

OUR TEAM:

Ismaeel Rahaman u20427906 u20427906@tuks.co.za 065 814 5362

Nawailah Tarmohamed u21437972 u21437972@tuks.co.za 067 319 1637

Deshlan Pillay u21481084 u21481084@tuks.co.za 067 127 6684

Sameer Ghela u21445142 u21445142@tuks.co.za 066 239 5477

Sashin Gounden u20487062 u20487062@tuks.co.za 072 520 1968



From left to right: Sameer Ghela, Ismaeel Rahaman (Group leader), Nawailah Tarmohamed, Sashin Gounden, and Deshlan Pillay.

Team 7: Omnia

Project: Water Resort administration system

System Name: Hydrotech

Iteration 7 – Logical Use Case Narratives

This document will define our Logical Narratives for all the use cases of the HydroTech System. Including the following Subsystems: User, Client, Accommodation, Ticketing, Events, Admin, Inventory, and Reporting.



Group 7 – Omnia

Iteration 7 Logical Use Case Narratives



Table of Contents

1.	DOCUMENT INTRODUCTION		
2.	LOG	GICAL USE CASE NARRATIVES	
2	.1.	SUBSYSTEM 1 - USER	
2	.2.	SUBSYSTEM 2 – CLIENT	
2	.3.	SUBSYSTEM 3 - ACCOMMODATION	24
2	.4.	SUBSYSTEM 4 - TICKETING	4
2	.5.	SUBSYSTEM 5 – EVENTS	49
2	.6.	SUBSYSTEM 6 – ADMINISTRATION	
2	.7.	SUBSYSTEM 7 – INVENTORY	10
2	.8.	Subsystem 8 – Reports	132
4.	DOC	CUMENT CONCLUSION	16
5.	TEA	M SIGN-OFF	162
6.	CLIENT SIGN-OFF		



Group 7 – Omnia

Iteration 7 Logical Use Case Narratives



1. Document Introduction

This section contains all the logical use case diagrams which describe the standard procedures of actors alongside the system response, for each use case.



Narratives



2. Logical Use Case Narratives

2.1. Subsystem 1 - User

Use Case Name:	Login	Use case type
Use Case ID:	1.1	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	User	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the user logging into the system. This use case begins when the user selects the Login option from the Platinum Island Resort website. The user will enter their username and password in the respective fields. The system will validate their login credentials by comparing the details with the details stored in the system's database. The use case ends when the user gains the permissions to access the system.	
Pre-condition:	User must be registered on the system	
Trigger:	The user requests to login on the system.	
Typical Course	Actor Action:	system Response:
of Events:	The user requests to login on the system.	The HydroTech system will prompt the user to enter their username and password.
	The user will provide their email address and password into the system and proceed	The system validates the information entered in the email and password fields on the form to ensure that none





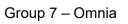
with their login. [ALT]	of the fields are empty and that a valid email address was entered.
[· ·=·]	[ALT]
•	5. The system validates the email address entered by the user matches a record found within the User entity by matching the email entered by the user to the <i>UserName</i> attribute found within the User entity.
	[ALT]
	6. The system hashes the password entered by the user. The system then validates that the password provided by user is correctly matched to the <i>Password attribute</i> stored in the User entity related to the <i>UserName</i> previously found.
	[ALT]
	7. The user's credentials are confirmed, and they are granted role specific access by referencing the permissions linked to the specific role on the system.
	The Role entity has the following attributes:
	o Role_ID [PK]
	o Name
	o Description
	The Permission entity has the following attributes:
	o Permission_ID [PK]
	o Name





	o Description	
	A role and its permission are linked in the RolePermission entity with the following composite key: o Role_ID [PK,FK]	
	o Permission_ID [PK,FK]	
	A user is linked to a specific role by the <i>Role_ID</i> [FK] in the User entity.	
	The system will then keep a record of the log in by creating an audit log record within the Audit_Log entity with the following attributes:	
	o Audit_Log [PK]	
	o Employee_ID [FK]	
	o Date	
	o Time	
	[ALT]	
	[ALT] Step 3a: The user forgot their password. They will select the forgot password button and the system will invoke Use Case 1.3 "Forgot Password".	
Alternate:	[ALT] Step 3b: The user does not have an active account and will be prompted to create an account on the system. Invoke Use Case 2.1 "Register Client".	
Courses:	[ALT] Step 4: The username and password field is invalid. Return to Step 3.	
	[ALT] Step 5: The username does not match an instance in the database. Return to step 3.	

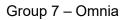






	[ALT] Step 6: The password entered by the user does not match the <i>UserName</i> attribute found within the User entity record does not match an. Return to step 3.
	[ALT] Step 7: The system failed to grant the user access to the system. Return to step 3
Conclusion:	The use case concludes when the user gains the permissions to access the system.
Post-condition:	The user is logged into the system. The user can access the system's functionalities associated with their user rights.
Business Rules:	Only registered users can login.
Implementation Constraints and Specifications:	Password in database must be hashed.
Assumptions:	None
Open Issues:	None

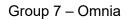






Use Case Name:	Log out	Use case type	
Use Case ID:	1.2	Business Requirements:o	
Priority:	High	System Analysis: 0	
Source:	Platinum Island resort	System Design: þ	
Primary Business Actor:	User		
Primary The system Actor:	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	None		
Description:	This use case describes the event where a user logs out of the system. The use case begins with the user requesting the log out of the system, the user clicks on the logout button on the screen. The system will respond by logging the user out. The use case concludes when the user is successfully logged out and redirected to the login page where they will be required to enter their details again.		
Pre-condition:	User must be logged into the system.		
Trigger:	The user requests to logout of the system.		
	Actor Action:	system Response:	
Typical Course	The user requests to logout of the system.	The Hydrotech system will capture the user's logout request.	
of Events:		The system will log the user out of the system.	
		[ALT]	
		The system will route the user to the login page.	

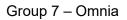






Alternate Courses:	[ALT] Step 3 : The system encountered an error while logging the user out and ask the user to attempt to log out again. Return to Step 1
Conclusion:	The use case concludes when the user is successfully logged out of the system
Post-condition:	
	The user will be redirected to the log in page. The user's accessibility has been removed from the system until they login again.
Business Rules:	None
Implementation Constraints and Specifications:	None
Assumptions:	None
Open Issues:	None







Use Case Name:	Forgot Password	Use case type
Use Case ID:	1.3	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	User	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event where a user has forgotten their password and has to reset it.	
	The use case begins with the user selecting the forgot password option. The system will request the user to enter their phone number. The system will validate that the phone number belongs to the user. After successful validation the system will SMS them an OTP. They will enter the OTP and thereafter enter their new password. The use case concludes when the user can successfully log in to the system using their newly created password.	
Pre-condition:	on: • User must be registered on the system	
Trigger:	gger: The user selects the forgot password option.	
	Actor Action:	system Response:
	The user selects the forgot password option.	The HydroTech system will prompt the user to enter the following details
Typical Course		Username
of Events:	0 TI '''	Phone number
	The user will enter their username and phone number. [ALT]	 The system validates the email to ensure that the phone number was entered and belongs to the user by searching for it in the User entity and validates that the phone

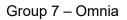


Group 7 – Omnia



	number was entered in the correct format. [ALT]
	5. When the phone number is located and verified the system will generate an OTP (One Time Password) and send it to the relevant phone number that was entered.
	6. The system will prompt the user to enter the OTP sent to their phone number via SMS.
7. The user enters the OTP they received	8. The system validates the entered OTP by matching it with the OTP generated. The system successfully validated the entered OTP. [ALT]
	9. The system prompts the user to enter a new password and to confirm the password.
10. The user enters a new password and confirms it.	11. The system will validate the entered password to ensure that it is secure, and that the password matches the confirmed password entered. The system will also ensure that the new password does not match the user's old password by hashing the entered password and comparing it to the Password attribute in the User table. The system successfully validated the entered password. [ALT]
	12. The system updates the user's newly created password in the User table's Password attribute







	after hashing the new	
	password.	
	13. The system will notify the	
	user that their password has been changed by	
	sending an SMS to them.	
	14. The system will route the	
	user back to the login page	
	[ALT] Step 3 : The user does not want to not change their	
	password. Terminate use case.	
Alternate:	[ALT] Step 4 : The username or the phone number was invalid, or username was not found in database. Return to step 3.	
Courses:	[ALT] Step 8: The OTP entered was invalid. Return to step 6.	
	[ALT] Step 11 : The entered password and confirmed password either do not match or do not comply with the security measures or the password is the same as the user's old password. Return to Step 10.	
Conclusion:	The case ends when the user's password has been updated	
	successfully and the system has returned to the login screen.	
Post-condition:	The user's password has been updated and they are able to log into the system with their newly created password.	
Business Rules:	Password needs to be at least 8 characters long.	
	Password must have a combination of uppercase and lowercase characters.	
	Password must include at least one number.	
	New password cannot be the same as the old password.	
Implementation Constraints and Specifications:	Password in database must be hashed.	
Assumptions:	None	
Open Issues:	None	
	-	





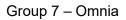
Use Case Name:	Update Password	Use case type
Use Case ID:	1.4	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	User	•
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	The situation where a user wants to change their password is described in this use case. The user initiates the use case by asking the system to update their password. Before the user can enter a new password, the system will ask them to enter their old one. They will then type their chosen password and confirm it by typing it again. When the system successfully updates the user's password, the use case is complete.	
Pre-condition:	User must be register	ered on the system
Trigger:	The user requests to upda	ate password.
	Actor Action:	system Response:
Typical Course of Events:	The user requests to update password.	 2. The HydroTech system will prompt the user to enter the following details. o Current Password o New password o Confirm new password
	3. The user will enter their current password, new password, and	4. The system checks to see if the new password, old password, and password fields are input correctly and adhere to security specifications. The Password attribute in the User table is used by the system to





	confirmation of the new password. [ALT]	compare the current password to the password that is already stored there. The system then checks to make sure the password entered by the user and the password currently stored for the user are different. The system updates the user's password in the User table with the new password that they entered on the system after hashing the new password. [ALT]	
		[ALI]	
Alternate:	[ALT] Step 3: The user do Terminate use case.	[ALT] Step 3: The user does not want to not change their password. Terminate use case.	
Courses: [ALT] Step 4: Password does not meet security required password was not confirmed, or new password match password. Return to step 3.		ned, or new password matches old	
Conclusion:	The case ends when the user's password has been updated successfully.		
Post-condition:	The user's password has been updated and they are able to log into the system with their newly created password.		
Business • Password needs to l		be at least 8 characters long.	
Rules:	e a combination of uppercase and s.		
	Password must include at least one number.		
New password cannot be the same as the old pass		ot be the same as the old password.	
Implementation Constraints and Specifications:	Password in database must be hashed.		
Assumptions:	None		
Open Issues:	None		

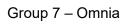






Use Case Name:	Update Email Address	Use case type	
Use Case ID:	1.5	Business Requirements:o	
Priority:	Medium	System Analysis:þ	
Source:	Platinum Island Resort	System Design:o	
Primary Business Actor:	User	•	
Primary The system Actor:	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	None		
Description:	The situation where a user wants to update their email address is described in this use case. The system should allow the user to update their email address, the user initiates the use case by inputting their "Current Email" and then entering the Email Address that they wish to use When the system successfully updates the user's email address, the use case is complete.		
Pre-condition:	User must be registered on the system		
Trigger:	The user requests to update their email address.		
	Actor Action:	system Response:	
	The user requests to update email address.	 The HydroTech system will prompt the user to enter their email address. [ALT] 	
Typical Course of Events:	4. The user will enter their current email address and new email address.[ALT]	5. The Email attribute in the User table is used by the system to compare the current email address to the email address that is already stored there. The system then checks to make sure the email address entered by the user contains @ symbol and that the email address currently stored for the user is different [ALT]	







	6.	
	The system updates the user's email in the User table with the new email that they entered on the system.	
	[ALT] Step 3: The user does not want to not change their email address. End use case.	
Alternate: Courses:	[ALT] Step 4a: New email matches old email, Return to step 3.	
	[ALT] Step 4b: The email address does not contain "@" symbol. Return to step 3.	
Conclusion:	The case ends when the user's email address has been updated successfully.	
Post-condition:	The user's email address has been updated and they are able to log into the system with their newly created email address.	
Business Rules:	Email needs to contain @ symbol.	
	New email cannot be the same as the old email.	
Implementation Constraints and Specifications:	None	
Assumptions:	None	
Open Issues:	None	



Narratives



2.2. Subsystem 2 – Client

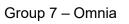
Use Case Name:	Register Client	Use case type	
Use Case ID:	2.1	Business Requirements: o	
Priority:	High	System Analysis: þ	
Source:	Platinum Island Resort	System Design: o	
Primary Business Actor:	Client		
Primary The system Actor:	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	None		
Description:	This use case begins when the client requests to register an account on the system. The system will prompt the client to enter their first name, last name, cell phone number, email address and a password. The system will validate the information and an account will be created for the client. The use case ends when the client has created an account.		
Pre-condition:	The client navigates to the register client page.		
Trigger:	The client requests to register an account.		
	Actor Action:	system Response:	
Turning Course	The client requests to register an account on the system.	The system prompts the client to enter the following information from the Client entity:ClientName	
Typical Course		 ClientSurname 	
of Events:		 ClientEmail 	
		 ClientPhone 	
	3. The client enters his/her information.	4. The system captures and validates the new client details entered by comparing the entered information with the	





		information in the Client table to ensure that the client does not already exist and checking the fields to ensure that the information is correct and that all fields specified as required have been inputted [ALT] 5. The system will respond by loading the OTP modal screen with an input field for the OTP.
		The system prompts the client to add the OTP.
	6. The client will enter the OTP that is sent to their number in the required field in the pop-up modal. [ALT]	7. The system captures and validates the OTP details entered and validates the fields using Angular to ensure that the information is correct and that all fields specified as required have been inputted.
		8. The system stores the new client information into the Client entity under the following attributes:
		o Client_ID(PK)
		o User_ID(FK)
		 ClientName
		 ClientSurname
		 ClientEmail
		 ClientPhone
		The system stores the new client information into the User entity under the following attributes:
		o User_ID(PK)
		o Role_ID(FK)
		 Username
		o Password
Alternate: Courses:	[ALT] Step 4: The system pro that a field is empty or invalid	ovides an error message indicating . Go back to step 3.

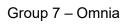






	[ALT] Step 6: The OTP is invalid. Re-enter the OTP.
Conclusion:	The use case concludes when the client has been registered on the system and their account has been created.
Post-condition:	 A new client has been added to the Client table. The new client will have access to certain parts of the system.
Business Rules:	None
Implementation Constraints and Specifications:	None
Assumptions:	The client has internet connection
Open Issues:	None

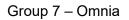






Use Case Name:	Update Client Details	Use case type	
Use Case ID:	2.3	Business Requirements: 0	
Priority:	High	System Analysis: þ	
Source:	Platinum Island Resort	System Design: o	
Primary Business Actor:	Client		
Primary The system Actor:	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	None		
Description:	This use case describes the event where the client wants to update their details. The use case starts when the client requests to update their details and will input changes to the details and ends when the client is notified that their profile details has been successfully updated.		
Pre-condition:	The client needs to be logged onto the system.		
	The client must exist on the system. The elient must exist to an electric details.		
Trigger:	The client requests to update their details.		
Typical Course of Events:	1. The client requests to update their details.	2. The system displays the client's details that has been retrieved from the Client entity with the following attributes: ClientName ClientSurname ClientEmail ClientPhone The system then prompts the client to enter their updated details.	
	The client enters their updated details.	4. The system captures and validates the new client details entered by comparing the	

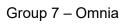






		ir a e c s	ntered information with the information in the Client table and checks the fields to insure that the information is orrect and that all fields pecified as required have een inputted. [ALT]
		u th	The system stores the pdated client information into ne Client entity under the ollowing attributes:
			ClientName
			ClientSurnameClientEmail
			 ClientPhone
Alternate: Courses:	[ALT] Step 4: The system noti or invalid. Return to step 3.	fies th	e client that a field is empty
Conclusion:	The use case concludes when		ent's personal information
	has been successfully update	d.	
Post-condition:	An clients account has been	upda	ted in the Client entity.
Business Rules:	None		
Implementation Constraints and Specifications:	None		
Assumptions:	The client has internet connec	tion	
Open Issues:	None		







Use Case Name:	Add Review	Use case type	
Use Case ID:	2.5	Business Requirements: 0	
Priority:	High	System Analysis: þ	
Source:	Platinum Island Resort	System Design: o	
Primary Business Actor:	Client	·	
Primary The system Actor:	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	None		
Description:	This use case begins when the client requests to add a review on the system.		
	· ·	client to provide a review rating and sort. The system will respond by to the Review entity.	
	The use case concludes when a new record is added to the Review entity.		
Pre-condition:	The client is logged in.		
	The client is on the "Vient	w Client Homepage" screen.	
Trigger:	The client requests to add a new review to the system.		
	Actor Action:	system Response:	
	The client requests to add a new review to the	2. The system prompts the client to enter the new review details:	
Typical Course of Events:	system.	 Number Rating 	
		o Review Description	
	3. The client enters the required details for the new review.	4. The system validates that all the required fields have been filled in correctly.	
		Once validated the system will create a new record within the	



Group 7 – Omnia



	Review entity which has the following attributes:
	o Review_ID [PK]
	○ Client_ID [FK]
	o RoomType_ID [FK]
	○ Rating
	 Description
	[ALT]
Alternate:	[ALT] Step 4: Not all fields were filled in correctly, the system
Courses:	displays an error message under the corresponding filed requesting the client to check the field.
Conclusion:	The use case concludes when the new review record has been added to the Review entity.
Post-condition:	A new review record is added in the Review table.
Business Rules:	None
Implementation Constraints and Specifications:	None
Assumptions:	Client is logged into the system
Open Issues:	None





2.3. Subsystem 3 - Accommodation

USE CASE NAME:	Filter Room Bookings		USE CASE TYPE	
USE CASE ID:	3.1	Bus	iness Requirements:	0
PRIORITY:	High		tem Analysis:	þ
SOURCE:	Platinum Island Resort		tem Design:	0
PRIMARY BUSINESS ACTOR:	Client			
PRIMARY THE SYSTEM ACTOR:	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	The use case begins when the client requests to filter available accommodation booking options on the system.			
	The system should allow the client to filter available room type bookings. The system will begin by prompting the client to select the check-in and check-out date where the system will filter the available rooms to the client.			
	The use case concludes when room are filtered according to the client's date selection following by the client will then select the room option of their choice.			
PRE-CONDITION:	 The client needs to be logged onto the system. The screen displays the types of accommodation options with their associated price. 			
TRIGGER:	The client requests to view available accommodation bookings on the system.			
	ACTOR ACTION:	SYS	STEM RESPONSE:	
TYPICAL COURSE OF EVENTS:	1. The client requests to view room booking type options available on the system by providing a check-in and check-out date.	neces the fo Th	ystem retrieves the sary room details from lowing entities: ne Room entity which ontains the following tributes: o Room_ID [PK] o RoomStatus_ID [FK] o RoomNumber	





		o RoomFloor	
		The RoomType entity which contains the following attributes: o RoomType_ID [PK] o TypeName o Description o RoomCapacity	
		The RoomTypePrice entity which contains the following attributes: RoomTypePrice_ID [PK] RoomType_ID [FK] Price Date	
		3. The system will then filter the client's selected duration of stay by reading the Name attribute from the RoomStatus entity, using the RoomStatus_ID [FK] attribute found within the Room entity, associated with each room where the room status is "Available" and will return all available room bookings for the client's requested booking duration.	
		The system prompts the client to select the filtered room booking of their choice. [ALT]	
	4. The client selects the room booking option of their choice. [ALT]	5. The system extends to "3.2 Make Room Booking"	
ALTERNATE: COURSES:	room options. The client will be date.	by the client for their stay returns no prompted to select an alternate	
	[ALT] Step 4: The client decides to return to the homepage. Terminate use case.		
CONCLUSION:	The use case concludes when choice.	the client selects the room of their	



Group 7 – Omnia



POST-CONDITION:	The system has filtered the available room accommodation options according to the dates selected by the client.
BUSINESS RULES:	None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	The client has access to an active internet connection.
ASSUMPTIONS:	None
OPEN ISSUES:	None





USE CASE NAME:	Make Room Booking	USE CASE TYPE
USE CASE ID:	3.2	Business Requirements: o
PRIORITY:	High	System Analysis: þ
SOURCE:	Platinum Island Resort	System Design: o
PRIMARY BUSINESS ACTOR:	Client	
PRIMARY THE SYSTEM ACTOR:	None	
OTHER PARTICIPATING ACTORS:	PayPal (ESA)	
OTHER INTERESTED STAKEHOLDERS:	Administrator	
DESCRIPTION:	The use case begins when the client requests to complete an accommodation booking on the system.	
	The system should allow the client to complete an accommodation booking. The system will prompt the client to provide the number of guests that will be occupying the room, the client will confirm their pre-populated information as well as indicate the number of guests that will be occupying the room. Once the client accepts the terms and conditions, they will proceed with the payment via the payment gateway, PayPal. Once the payment has been validated and is successful, the system will generate a unique reference number specific to the client's booking. The system will use the client email to generate and send an email detailing their booking summary information as well as will save the booking to the Booking entity. The use case concludes when the client receives an email summarising their booking.	
PRE-CONDITION:	 The client needs to be logged onto the system. The client has selected a room type option of their choice. 	
TRIGGER:	The client requests to make an accommodation booking on the system.	
	ACTOR ACTION:	SYSTEM RESPONSE:
TYPICAL COURSE OF EVENTS:	1. The client requests to complete an accommodation booking on the system.	2. The system retrieves the following attributes from the selected Room entity: OROOM_ID [PK] OROOMType_ID [FK] OROOMStatus_ID [FK] OROOMNumber OROOMFloor
		The system will then retrieve the





following attributes from the **RoomType** entity using the selected *RoomType_ID* [FK] record from the **Room** entity:

- o TypeName
- Description
- o RoomCapacity

The system will then retrieve the associated *RoomPrice_ID.Price* attribute from the

RoomTypePrice entity using the RoomType_ID [FK] attribute of the selected record as well as the date duration selected by the client.

The system then uses the Client_ID attribute within the Client entity to retrieve the record elements specific to the client:

- ClientName
- o ClientSurname
- ClientPhone
- ClientEmail

The system will then use the information retrieved from the client's record to pre-populate the input fields.

The system then fetches the most recent VAT record according to the *Date* attribute from the **VAT** entity with the following attributes:

- VATAmount
- o Date

The retrieved VAT amount will then be stored within a numeric variable, VATamount.

The system will then calculate the total amount payable by calculating the number of days selected by the client multiplied by the *Price* attribute found within the **RoomTypePrice** entity.





			The system then performs a further calculation by declaring a numeric variable, VATCost. Which is calculated by multiplying the totalCost variable by the VATamount variable (totalCost * VATamount) Using the RoomCapacity attribute found within RoomType entity, the system will restrict the number of guests chosen by the client to ensure the chosen capacity does not exceed the maximum capacity of the room type.
			The system prompts the client to provide the number of guests that will be occupying the room. [ALT]
;	3. The client will provide the number of guests that will be occupying the room.	4.	The system will prompt the client to accept the terms and conditions detailed.
	the client accepts the terms and conditions laid out by the business and proceeds with the payment of the booking. [ALT]	6.	The system validates the information entered by the client is valid and captures the information. Once successfully validated, the system will prompt the client to choose their method of payment. [ALT]
	7. The client will provide their preferred method of payment.	8.	The system will redirect the client to the PayPal system where the system will pass through the totalCost of the client's booking as well as the ClientName and ClientSurname to PayPal where they will be required provide the card holder's details to proceed with the payment for their booking.
		9.	Once the payment is successful, PayPal will send back a payment confirmation to the system. [ALT]





10. The system will then generate a unique reference number specific to the client's booking and will create a new record within the following entities:

RoomBooking entity with the following attributes:

- RoomBooking_ID [PK]
- Client_ID [FK]
- RoomBookingStatus_ID [FK]
- Reminder_ID [FK]
- o ReferenceNum
- BookingDate
- o NumberOfGuests
- EntryDate
- o ExitDate

RoomPayment entity with the following attributes:

- RoomPayment_ID [PK]
- o Client_ID [FK]
- RoomBooking_ID [FK]
- PaymentType_ID [FK]
- Amount

The system will assign the status of "Check-In" from the **RoomBookingStatus** entity to the client's booking record with the following attributes:

- RoomBookingStatus_ID [PK]
- o Name
- Description

The system will read the ClientEmail attribute from the Client entity to send an email detailing the booking summary information of the client's booking as well as their unique booking reference number.

[ALT]

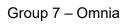


Group 7 – Omnia



	[ALT] Step 2: The system will return a 404-code error, as the client's details are unable to be found within the Client entity.
ALTERNATE: COURSES:	[ALT] Step 5a: The client refuses to accept the terms and conditions set out by the resort. Terminate use case.
	[ALT] Step 5b: The client chooses to cancel the booking by return to the booking homepage. Terminate use case.
	[ALT] Step 6: The information entered by the client was invalid and/or input fields were left blank. The system will prompt the client to re-enter their details. Return to Step 3.
	[ALT] Step 9: PayPal will return a failed payment from the client. Return to Step 7.
	[ALT] Step 10: The system failed to add the client's booking. The system alerts the admin of the error and that the booking wasn't made successfully. Terminate use case.
CONCLUSION:	The use concludes when the client's booking is successfully added to the Booking entity and the client receives an email detailing their booking information.
POST- CONDITION:	The booking has been stored in the Booking Entity and the room's booking status has been changed to "Check-In".
BUSINESS RULES:	The client must complete the total room reservation payment at checkout.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	The client requires internet access.
ASSUMPTIONS:	None
OPEN ISSUES:	None







USE CASE NAME:	Cancel Room Booking	USE CASE TYPE
USE CASE ID:	3.3	Business Requirements: o
PRIORITY:	High	System Analysis: þ
SOURCE:	Platinum Island Resort	System Design: o
PRIMARY BUSINESS ACTOR:	Client	
PRIMARY THE SYSTEM ACTOR:	Administrator	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes an occ to cancel their room booking.	urrence in which the client wishes
		client calls the administrator of e administrator with the reference es the room booking and cancels it
	the client's room booking to "Ca hours before Check-In and "Elig	he system changes the status of ncelled" if cancelled less than 48 ible for Refund" if the cancellation 48 hours before Check-In. An email im about the cancellation.
PRE-CONDITION:		to be logged onto the system. tion booking status has to be
TRIGGER:	The client calls the administrato cancel their room booking.	r of Platinum Island to requests to
	ACTOR ACTION:	SYSTEM RESPONSE:
TYPICAL COURSE	The client calls the administrator of Platinum Island to requests to cancellation of their room booking.	The administrator requests the client to provide their reference number.
OF EVENTS:	3. The client provides their reference number that is linked to their room booking.	4. The administrator searches for the client's event booking by matching the client's reference code to a room booking record and selects the appropriate record. [ALT]
		5. The system will confirm whether the client would like to





	cancel their room booking reservation.
6. The client confirms their accommodation booking cancellation. [ALT]	7. The administrator confirms the cancellation of the client's room booking by the matching the *ReferenceCode* attribute within the *RoomBooking* entity to the appropriate matching record which has the following attributes:

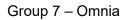


Group 7 – Omnia



	cancellation, the Room booking will have a status of "Eligible for Refund". Where the client will be eligible for a 25% return on their room booking payment.	
	8. The system will then retrieve the client's email address using the Client_ID [FK] attribute within the RoomBooking entity and match it to the Client_ID [PK] within the Client entity to retrieve the ClientEmail attribute.	
	Using the ClientEmail retrieved, the system will draft an email which will be sent to the client detailing their accommodation booking cancellation.	
	[ALT] Step 4: The administrator could not find the reference code provided by the client. Return to Step 2.	
ALTERNATE: COURSES:	ALT] Step 6: The client refuses to confirm the cancellation of their event booking. Terminate Use Case.	
CONCLUSION:	The client receives an email confirming their accommodation cancellation.	
POST-CONDITION:	Dependant of the when the room booking cancellation occurred, the system will change the status of the client's room booking to "Cancelled" if cancelled less than 48 hours before Check-In and "Eligible for Refund" if the cancellation occurred more than 48 hours before Check-In.	
	The client will receive an email detailing the cancellation of their room booking.	
BUSINESS RULES:	The client cancels more than or equal to 48 hours before check-in they get a 25% refund.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	The client has access to an active internet connection.	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

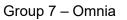






USE CASE NAME:	Send Accommodation Booking Reminder	USE CASE TYPE
USE CASE ID:	3.4	Business Requirements: o
PRIORITY:	Low	System Analysis: þ
SOURCE:	Platinum Island Resort	System Design: o
PRIMARY BUSINESS ACTOR:	Time	
PRIMARY THE SYSTEM ACTOR:	None	
OTHER PARTICIPATING ACTORS:	Client (ERA)	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	The use case begins when time prompts the system to send a reminder notification to the client.	
	The system should send an email to check-in reminding them of their bo	
	The use case concludes when the sclient reminding them of the time of	
PRE-CONDITION:	The system detects that the client needs to be notified of when their accommodation reservation.	
TRIGGER:	Time prompts the system to send a reminder notification to the client.	
	ACTOR ACTION:	SYSTEM RESPONSE:
TYPICAL COURSE OF EVENTS:	1. Time prompts the system to send a reminder notification to the client. 2.	The system retrieves the BookingDate, to periodically check for booking check-ins', by reading the Reminder_Time attribute from the Reminder entity to determine how many hours before the notification reminder is sent to the client, and Client_ID [FK] attributes from the RoomBooking entity and matches the Client_ID [FK] attribute to the Client_ID [PK] attribute within the Client entity to retrieve the ClientEmail attribute. [ALT]

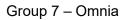






	3. The system uses the ClientEmail attribute retrieved to generate an email which is sent to the client's email to notify them of their accommodation before their reservation is due.	
ALTERNATE: COURSES:	None	
CONCLUSION:	The use case concludes when the system generates an email reminder which is sent to the client detailing their accommodation reservation.	
POST-CONDITION:	The client receives an email reminding them of their accommodation reservation.	
BUSINESS RULES:	The accommodation reservation notification is sent to the client 24 hours before their check-in.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

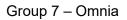






USE CASE NAME:	Add Room Type		USE CASE TYPE	
USE CASE ID:	3.6		Business Requirements:	0
PRIORITY:	Low		System Analysis:	þ
SOURCE:	Platinum Island Resort		System Design:	0
PRIMARY BUSINESS ACTOR:	Administrator	Administrator		
PRIMARY THE SYSTEM ACTOR:	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	The use case describes the event where the administrator wants to add a new type of room. The system will prompt the administrator to enter the TypeName and TypeDescription for the new type of room. The system validates and saves the new room type in the RoomType entity.			
PRE-CONDITION:	 The administrator needs to b 	e log	ged onto the system.	
TRIGGER:	The administrator requests to ad	dd a n	ew type of room.	
	ACTOR ACTION:		SYSTEM RESPONSE:	
TYPICAL COURSE OF EVENTS:	requests to add a new type of room.	p a fo R	The HydroTech system brompts the administrator to enter the collowing information from the comType entity: TypeName TypeDescription RoomCapacity The system will then retrieve the following attributes from the RoomTypePrice entity using the selected RoomType_ID [FK] record from the RoomTypePrice entity: Price	/e n /
	the new type of room's details into the system.	v: ro a	The system captures and alidates the new type of com's details and verifies the specified fields have been putted [ALT]	
			he system autogenerates t	

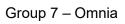






	the next consecutive number in the RoomType entity. The system stores the new type of room's details into the RoomType entity under the following attributes:
ALTERNATE: COURSES:	[ALT] Step 4b: The system displays a relevant error message that a field is empty/invalid. Return to step 3
CONCLUSION:	The use case concludes when the new room type is added to the room type entity.
POST-CONDITION:	A new type of room was added to the RoomType entity.
BUSINESS RULES:	Only the administrator can add a room type on the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	The administrator has internet connection.
OPEN ISSUES:	None







USE CASE NAME:	Update room Type		USE CASE TYPE	
USE CASE ID:	3.7		Business Requirements:	О
PRIORITY:	Low		System Analysis:	þ
SOURCE:	Platinum Island Resort		System Design:	0
PRIMARY BUSINESS ACTOR:	Administrator			
PRIMARY THE SYSTEM ACTOR:	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	The use case describes the event where the administrator wants to update an existing room type. The system prompts the administrator to enter the attributes of the room type. The system will save the updated details of room type in the RoomType entity.			
PRE-CONDITION:	 The administrator needs to be logged onto the system. The room type must exist on the system. 			
TRIGGER:	The room type must exist on The administrator requests to up		•	
THOOLIN.	ACTOR ACTION:	Juaic	SYSTEM RESPONSE:	
		р	The HydroTech system rompts the administrator to elect a room type	
TYPICAL COURSE OF EVENTS:	3. The administrator selects the room type they wish to edit.	rc ha R fc	the system displays the com's type information that as been retrieved from the comType entity with the collowing attributes: TypeName TypeDescription RoomCapacity The system will then retrieve the following attributes from the RoomTypePrice entity using the selected RoomType_ID [FK] record from the RoomTypePrice entity: Price	re n





	 5. The administrator edits the details of the room type. 6. The system captures and stores the updated details in RoomType entity under the following attributes: TypeName TypeDescription RoomCapacity And the RoomTypePrice entity under the following attributes: Price Date
ALTERNATE: COURSES:	None
CONCLUSION:	The system captures and stores the updated details in RoomType entity.
POST-CONDITION:	 A room type was updated in the RoomType entity
BUSINESS RULES:	Only the administrator can update a room type on the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	The administrator has internet connection.
OPEN ISSUES:	None





USE CASE NAME:	Delete Room Type	USE CASE TYPE	
USE CASE ID:	3.8.	Business Requirements: o	
PRIORITY:	Low	System Analysis: þ	
SOURCE:	Platinum Island Resort	System Design: o	
PRIMARY BUSINESS ACTOR:	Administrator		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case covers an incident in which an administrator wishes to delete a room type from the system. The administrator will look for the room type and proceed to delete it after receiving confirmation. The use case ends when the room type is deleted from the room type entity.		
PRE-CONDITION:	The administrator needs to be	e logged onto the system.	
	The room type must exist on	the system.	
TRIGGER:	The administrator requests to delete a room type.		
TRIOGER.	The administrator requests to de	lete a room type.	
HUGGER.	ACTOR ACTION:	lete a room type. SYSTEM RESPONSE:	
THOOLIL.	ACTOR ACTION:	••	
TYPICAL COURSE	ACTOR ACTION: 1. The administrator requests to delete a room type.	SYSTEM RESPONSE: 2. The HydroTech system prompts the administrator to select a room type that they	
	ACTOR ACTION: 1. The administrator requests to delete a room type. 3. The administrator selects the room type they wish to delete.	SYSTEM RESPONSE: 2. The HydroTech system prompts the administrator to select a room type that they would like to delete. 4. The system prompts the administrator to confirm that they want to delete the	
TYPICAL COURSE	ACTOR ACTION: 1. The administrator requests to delete a room type. 3. The administrator selects the room type they wish to delete. 5. The administrator confirms that they want to delete the selected room type.	2. The HydroTech system prompts the administrator to select a room type that they would like to delete. 4. The system prompts the administrator to confirm that they want to delete the selected room type. 6. The system removes the room type from the RoomType entity which has the following attributes: - RoomType_ID(pk) - TypeName - TypeDescription - RoomCapacity	
TYPICAL COURSE OF EVENTS:	1. The administrator requests to delete a room type. 3. The administrator selects the room type they wish to delete. 5. The administrator confirms that they want to delete the selected room type. [ALT] Step 5: The administrator of The Use Case is terminated.	2. The HydroTech system prompts the administrator to select a room type that they would like to delete. 4. The system prompts the administrator to confirm that they want to delete the selected room type. 6. The system removes the room type from the RoomType entity which has the following attributes: - RoomType_ID(pk) - TypeName - TypeDescription - RoomCapacity	





BUSINESS RULES:	None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	The client has internet connection.
OPEN ISSUES:	None





2.4. Subsystem 4 - Ticketing

Use Case Name:	Make ticket Booking	Use case type
Use Case ID:	4.2	Business Requirements: 0
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Client	
Primary The system Actor:	None	
Other Participating Actors:	PayPal (ESA)	
Other Interested Stakeholders:	Administrator	
Description:	The use case begins when the client requests to complete a Day Visit booking on the system.	
	The system should allow the client to complete a Day Visit booking. The system will prompt the client to provide the number of guests that will be visiting on the day, the client will confirm their pre-populated information as well as indicate the number of guests that will be visiting. Once the client accepts the terms and conditions, they will proceed with the payment via the payment gateway, PayPal. Once the payment has been validated and is successful, the system will generate a QR Code specific to the client's booking. The system will use the client email to generate and send an email detailing their booking summary information as well as will save the booking to the DayVisit entity.	
	The use case concludes when the client receives an email summarising their booking.	
Pre-condition:	The client needs to be log	gged onto the system.
Trigger:	The client requests to make a I	Day Visit booking on the system.
Typical Course	Actor Action:	system Response:
Typical Course of Events:	The client requests to complete a Day Visit booking on the system.	 The system uses the Client_ID attribute within the Client entity to retrieve the





		client:
		o ClientName
		 ClientSurname
		 ClientPhone
		 ClientEmail
	informa client's the inp client t numbe	ystem will then use the ation retrieved from the serecord to pre-populate out fields and prompts the to select the required er of adults, children, repensioners.
numb	per of adults, children maxion pensioners. been the control of the	system validates that the imum capacity has not neached by referring to capacity attribute in the VisitDate entity. [ALT]
	-	stem then prompts the view the day visit ary.
U 1 1110 0	s and conditions to view to vi	system allows the client ew the summary.
the da	ay visit summary. sumi by fe VAT Date entity	system displays a mary of the booking made etching the most recent record according to the attribute from the VAT y with the following outes: VATAmount decimal (6,2)





	 Date (DateTime) The retrieved
	The system then fetches the ticket prices from the prices from the DayVisitType entity with the following attributes:
	DayVisitType_ID [PK]
	o Price
	o Category
	The system will then calculate the total amount payable by calculating the guests that were selected by the client multiplied by the Price attribute found within the DayVisitType entity for each category.
	The system prompts the client to select a payment method.
9. The client chooses his preferred method of payment.	10. Once the client's details are validated, the system will redirect the client to PayPal where the system will pass through the total cost of the client's booking to PayPal where they will be required to pay for their booking.
11. The client makes the payment for the booking on PayPal.	12. The will then system will generate a QR Code specific to the client's booking and will create a new record within the following entity:
	DayVisitTicket entity with the following attributes:
	o Ticket_ID [PK]
	o DayVisit_ID [FK]





		○ DayVisitType_ID [FK]	
		GuestName	
		The system will read the 'ClientEmail' attribute from the Client entity to send an email detailing the booking summary information of the client's booking as well as their QR Code.	
		[ALT]	
		return a 404-code error, as the e found within the Client entity.	
	[ALT] Step 4: The capacity has been reached and the client cannot book for that day. Terminate use case.		
Alternate: Courses:	[ALT] Step 5: The client refuses to accept the terms and conditions set out by the resort. The View booking summary button will remain disabled.		
	ALT] Step 7: The client does r summary.	not wish to view the day visit	
		iled to add the client's booking. The e error and that the booking wasn't use case.	
Conclusion:	The use concludes when the client's booking is successfully added to the DayVisitTicket entity, and the client receives an email detailing their booking information as well as a QR Code.		
Post-condition:	The booking has been stored in the DayVisitTicket entity".		
Business Rules:	The client must complete the payment at checkout.		
Implementation Constraints and Specifications:	The client requires internet access.		
Assumptions:	None		
Open Issues:	None		





Use Case Name:	Update ticket price	Use case type
Use Case ID:	4.4	Business Requirements: 0
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes an occurrence in which the administrator wishes to change the price of one of three types of tickets in the DayVisitTicketPrice entity. The use case begins when the administrator wants to edit the ticket price and enters modifications, and it concludes when the administrator is alerted that the ticket price has been successfully updated.	
Pre-condition:	The administrator needs to be logged onto the system.	
Trigger:	The administrator requests to update a ticket price.	
	Actor Action:	system Response:
	1. The administrator requests to update a ticket price. 2.	The system displays the edit ticket price modal and prompts the user to select a category for which they want to update the price.
Typical Course of Events:	3. The administrator selects to update the children category. [ALT] 4.	The system displays the ticket types of information that has been retrieved from the DayVisitType table with the following attributes:
		DayVisitType_ID(PK)
		o LowerAge
		o UpperAge
		 TypeName





		•
		And prompts the administrator to edit the ticket price/s they wish to.
	5. The administrator edits the price/s of the ticket types and saves the changes [ALT]	6. The modified ticket type price/s is saved by the system in the DayVisitTicketPrice entity under the following attributes: • Price_ID (PK) • DayVisitType_ID (FK) • Price • Date
		7. The system notifies the user that the employee has been successfully updated.
Alternate: Courses:	update. [ALT] Step 3b: The administrate to update.	ator selects the adult's category to ator selects the pensioner category or chooses to cancel the changes
Conclusion:	The use case concludes when successfully updated and the that the ticket price has been	system notifies the administrator
Post-condition:	The ticket price has been updated in the DayVisitTicketPrice entity.	
Business Rules:	Only the administrator can up system	date a ticket price's details on the
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has internet connection	
Open Issues:	None	





2.5. Subsystem 5 – Events

Use Case Name:	Make Event Booking	Use case type
Use Case ID:	5.2	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Client	
Primary The system Actor:	None	
Other Participating Actors:	PayPal (ESA)	
Other Interested Stakeholders:	Administrator	
Description:	The use case begins when the client requests to complete an event booking on the system. The system should allow the client to complete an event booking. The system will prompt the client to provide their date of the event and upload a guest list on who will be attending. The client will also confirm their pre-populated information. Once the client accepts the terms and conditions, they will proceed with the payment via the payment gateway, PayPal. Once the payment has been validated and is successful, the system will use the client email to generate and send an email detailing their booking summary information as well as will save the booking to the Event entity. The use case concludes when the client receives an email summarising their booking.	
Pre-condition:	The client needs to be logged onto the system.	
Trigger:	The client requests to make an event booking on the system.	
	Actor Action:	system Response:
Typical Course of Events:	The client requests to make an event booking on the system.	2. The system uses the Client_ID attribute within the Client entity to retrieve the record elements specific to the client:





		 ClientName
		 ClientSurname
		○ ClientEmail
		 ClientCellphone
		The system will then use the information retrieved from the client's record to pre-populate the input fields and prompts the client to select a date and upload a guestlist. [ALT]
3.	The client will provide the required information and upload their guestlist.	4. The system checks that all the fields are correctly filled in. [ALT]
5.	The client selects the terms and conditions checkbox.	6. The system allows the user to select the view summary.
	• [ALT]	
7.	The client chooses to view the event summary. [ALT]	8. The system displays a summary of the booking made by fetching the most recent VAT record according to the <i>Date</i> attribute from the VAT entity with the following attributes:
		 VATAmount
		o Date
		The retrieved VAT amount will then be stored within a numeric variable, VATamount.
		The system then fetches the prices from the VenuePrice entity with the following attributes:
		o VenuePrice_ID [PK]
		 Venue_ID [FK] attribute of the Venue entity where the VenuePrice





	entity corresponds to the
	Venue_ID in the Venue entity.
	 VenuePriceAmount_ID
	o Date
	The system then performs a further calculation by declaring a numeric variable, totalCost. Which is calculated by multiplying the price attribute retrieved from the VenuePrice entity by the client duration (price * dateDurationt)
	9. The system prompts the client to select a preferred payment method.
10. The client chooses his preferred payment method.	11. Once the client's details are validated, the system will redirect the client to PayPal where the system will pass through the total cost of the client's booking to PayPal where they will be required provide the card holder's details to proceed with the payment for their booking.
	12. Once the payment is successful, PayPal will send back a payment confirmation to the system.
	[ALT]
	13. The system will then generate a booking reference number which will then be sent to the client and will create a new record within the following entities:
	The Event entity which has the following attributes:
	o Event_ID [PK]
	EventStatus_ID [FK]
	o Client_ID [FK]
	o Venue_ID [FK]
	o Date





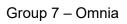
		 GuestList
		o ReferenceCode
		o Description
		The EventPayment entity with the following attributes:
		EventPayment_ID [PK]
		o Client_ID [FK]
		o Event_ID [FK]
		PaymentType_ID[FK]
		o Amount
		•
		The system will assign the status of "Check-In" from the EventStatus entity to the client's booking record with the following attributes:
		EventStatus_ID [PK]
		o Name
		Description
		·
		The system will read the <i>ClientEmail</i> attribute from the Client entity to send an email detailing the booking summary information of the client's booking.
		[ALT]
		will return a 404-code error, as the client's und within the Client entity.
Alternate:		
Courses:	[ALT] Step 4: The date sele Step 3.	ected has been booked out. Return to





	[ALT] Step 5: The client refuses to accept the terms and conditions set out by the resort. Disable view booking summary button.
	[ALT] Step 7: the client does view the summary. Return to Step 6.
	[ALT] Step 12: PayPal will return a failed payment from the client. Return to Step 7.
	[ALT] Step 13: The system failed to add the client's booking. The system alerts the admin of the error and that the booking wasn't made successfully. Terminate use case.
Conclusion:	The use concludes when the client's booking is successfully added to the Event entity, and the client receives an email detailing their booking information.
Post-condition:	The booking has been stored in the Event Entity.
Business Rules:	The client must complete the payment at checkout.
Implementation Constraints and Specifications:	The client requires internet access.
Assumptions:	None
Open Issues:	None







Use Case Name:	Cancel Event Booking	Use case type
Use Case ID:	5.3	Business Requirements: 0
Priority:	Medium	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Client	
Primary The system Actor:	Administrator	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes an occurrence wishes to cancel their event bo	
	The use case begins when the platinum island, and provides the reference number, the administ cancels it as indicated.	
	the client's event booking to "C hours before Check-In and "Re	the system changes the status of cancelled" if cancelled less than 48 fund Pending" if the cancellation before Check-In. An email is also but the cancellation.
Pre-condition:	The administrator needs to be	e logged onto the system.
	The client already has an even	ent booked at Platinum Island.
Trigger:	The client calls the administrator of Platinum Island to requests to cancel their booking.	
	Actor Action:	system Response:
Typical Course of Events:	The client calls the administrator of Platinum Island to requests to cancellation of their event booking.	2. The administrator requests the client to provide their reference number.
	3. The client provides their reference number that is linked to their booking.	4. The administrator searches for the client's event booking by matching the client's reference code to a booking





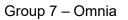
5. The system will confirm whether the client would like to cancel their event booking reservation. 7. The administrator confirms the cancellation of their event booking by the matching the *ReferenceCode* attribute within the *Event* entity to the appropriate matching record which has the following attributes: • Event_ID [PK] • EventStatus_ID [FK] • Client_ID [FK] • Venue_ID [FK] • Date • Guestlist • ReferenceCode • EventTotal • Description The system will update the status of the client's booking by reading the *Name* attribute from the *EventStatus* entity depending on when the cancellation of the booking occurs: • If the clients' event cancellation occurs within 48 hours before		record and selects the appropriate record. [ALT]
cancellation of their event booking. [ALT] cancellation of the client's event booking by the matching the ReferenceCode attribute within the Event entity to the appropriate matching record which has the following attributes: Event_ID [PK] EventStatus_ID [FK] Client_ID [FK] Venue_ID [FK] Date Guestlist ReferenceCode EventTotal Description The system will update the status of the client's booking by reading the Name attribute from the EventStatus entity depending on when the cancellation of the booking occurs: If the clients' event cancellation occurs within 48 hours before		whether the client would like to cancel their event booking
their event check-in, which can be validated	cancellation of their event booking.	cancellation of the client's event booking by the matching the ReferenceCode attribute within the Event entity to the appropriate matching record which has the following attributes: Event_ID [PK] EventStatus_ID [FK] Client_ID [FK] Venue_ID [FK] Date Guestlist ReferenceCode EventTotal Description The system will update the status of the client's booking by reading the Name attribute from the EventStatus entity depending on when the cancellation of the booking occurs: If the clients' event cancellation occurs within 48 hours before their event check-in,





		within the Event entity and the current date of cancellation, the event booking will have a status of "Cancelled" - If the clients' event cancellation occurs more than 48 hours before their event check-in, which can be validated using the Date attribute within the Event entity and the current date of cancellation, the event booking will have a status of "Refund Pending". Where the client will be eligible for a 25% return on their event payment.
		8. The system will then retrieve the client's email address using the Client_ID [FK] attribute within the RoomBooking entity and match it to the Client_ID [PK] within the Client entity to retrieve the ClientEmail attribute.
		Using the <i>ClientEmail</i> retrieved, the system will draft an email which will be sent to the client detailing their accommodation booking cancellation.
Alternate:	[ALT] Step 4: The administrato provided by the client. Return to	r could not find the reference code to Step 2.
Courses:	[ALT] Step 6: The client refuses their event booking. Terminate	
Post-condition:	occurred, the system will	he event booking cancellation change the status of the client's led" if cancelled less than 48

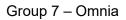






	hours before Check-In and "Refund Pending" if the cancellation occurred more than 48 hours before Check-In. The client will receive an email detailing the cancellation of their event booking.
Business Rules:	 Client's will be offered a refund on their reservation bookings should they cancel their booking 48 hour or more before their reservation check-in. A refund of 25% is offered should they client be eligible for a refund.
Implementation Constraints and Specifications:	The client requires internet access.
Assumptions:	None
Open Issues:	None

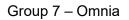






Use Case Name:	Send Event Booking Reminde	r Use case type
Use Case ID:	5.4	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Time	,
Primary The system Actor:	None	
Other Participating Actors:	Client (ERA)	
Other Interested Stakeholders:	None	
Description:	_	ne prompts the system to send a ent notifying them of their event
		mail to the client before their event em of their booking reservation.
	The use case concludes when client reminding them of the tire	the system sends an email to the me of their check-in.
Pre-condition:	The system detects that the client needs to be notified of when their event reservation.	
Trigger:	Time prompts the system to send a reminder notification to the client.	
	Actor Action:	system Response:
Typical Course of Events:	Time prompts the system to send a reminder notification to the client.	2. The system retrieves the Date, to periodically check for booking check-ins', by reading the Reminder_Time attribute from the Reminder entity to determine how many hours before the notification reminder is sent to the client, and Client_ID [FK] attributes from the Event entity and matches the Client_ID [FK] attribute to the Client_ID [PK] attribute within the Client

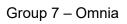






		entity to retrieve the ClientEmail attribute.
		3. The system uses the ClientEmail attribute retrieved to generate an email which is sent to the client's email to notify them of their event booking before their reservation is due.
Alternate:	None	
Courses:	140110	
Conclusion:	The use case concludes when which is sent to the client deta	n the system generates an email illing their event reservation.
Post-condition:	The client receives an email r reservation.	eminding them of their event
Business Rules:	 The accommodation res client 24 hours before th 	ervation notification is sent to the eir check-in.
Implementation Constraints and Specifications:	None	
Assumptions:	None	
Open Issues:	None	







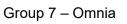
Use Case Name:	Update Event Price	Use case type
Use Case ID:	5.6	Business Requirements: 0
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes an ocadministrator wishes to change the VenuePrice entity.	e the price of booking an event in
	The use case begins when the venue price and enters modifice	e administrator wants to edit the cations.
	The use case concludes when successfully updated.	the venue price has been
Pre-condition:	The administrator needs to be logged onto the system.	
Trigger:	The administrator requests to	update a venue price.
	Actor Action: system Response:	
Typical Course of Events:	The administrator requests to update the venue price.	 2. The system retrieves the appropriate record from the system by matching the VenuePrice_ID attribute chosen by the administrator to the VenuePrice_ID [PK] record within the VenuePrice entity which holds the following attributes: VenuePrice_ID [PK] Venue_ID [FK] attribute of the Venue_ID in the VenuePrice entity corresponds to the Venue_ID in the Venue entity.





		o VenuePriceAmount
		o Date
		The system then retrieves the corresponding Venue record using the Venue_ID [FK] within the VenuePrice entity to the Venue_ID [PK] in the Venue entity which has the following attributes:
		o Venue_ID [PK]
		o VenueName
		 VenueDescription
		 Capacity
		•
		The system then prompts the administrator to enter the updated venue price details.
	3. The administrator provides the updated price of the venue.	4. The system requests the administrator to confirm the updated venue price details.
	5. The administrator confirms the updated record entry for the venue price.	6. The modified venue's price is first validated before it is saved by the system on the VenuePrice entity under the following attributes:
	[ALT]	o VenuePrice_ID [PK]
		o Venue_ID [FK]
		 VenuePriceAmount
		o Date
		• [ALT]
		7. The system notifies the user that the venues price has been successfully updated.
Alternate: Courses:	[ALT] Step 5: The administrat selected field and cancels the Terminate use case.	-







	[ALT] Step 6: The information entered by the administrator returns a validation error. Return to Step 3.
Conclusion:	The use case concludes when the venue price has been successfully updated and the system notifies the administrator that the venue price has been successfully updated.
Post-condition:	The venue price has been updated in the VenuePrice entity.
Business Rules:	Only the administrator can update a venues price detail on the system.
Implementation Constraints and Specifications:	None
Assumptions:	The administrator has internet connection
Open Issues:	None





2.6. Subsystem 6 – Administration

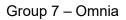
Use Case Name:	Add employee	Use case type
Use Case ID:	6.2.	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	·
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	The use case begins when the administrator wishes to add a newly hired employee to the system. The administrator will enter all the necessary	
	information onto the system. T information is captured.	he use case concludes after the
Pre-condition:	The administrator needs to be logged onto the system.	
	The employee must not alre	ady exist on the system.
Trigger:	The administrator requests to system.	add a new employee on the
	Actor Action:	system Response:
	The administrator requests to add a new employee on the system.	2. The HydroTech system will prompt the administrator to enter the following information from the Employee table:
Typical Course		o ID_Number
of Events:		o Cell_Num
		 o EmployeeName
		The system prompts the administrator to enter the employee details.





	The admin will enter the necessary information of the new employee	4. The system captures and validates the new employee attributes entered by checking the fields to ensure that the information is correct and that all fields specified as required have been inputted [ALT]
		5. The system stores the new employee information into the corresponding entities under the following attributes:
		o Employee_ID(PK)
		EmplloyeeType_ID(FK)
		○ User_ID(FK)
		o ID_Num
		o Cell_Num
		 EmployeeName
		o HireDate
Alternate:	[ALT] Step 4a: The system pr that a field is empty. Return to	ovides an error message indicating o Step 3.
Courses:		
Courses.	[ALT] Step 4b: The system pr that a field is invalid. Return to	ovides an error message indicating o Step 3.
Conclusion:	The use case concludes whe successfully added onto the s	The state of the s
Post-condition:	A new employee has been	added to the Employee table.
Business Rules:	Only the administrator can system	register a new employee on the
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has interne	t connection.
Open Issues:	None	







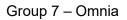
Use Case Name:	Update employee	Use case type
Use Case ID:	6.3	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	Employee entity. The use case wants to edit an employee's info	an employee's information in the begins when the administrator
Pre-condition:	The administrator needs to be	e logged onto the system.
	The employee must exist on the trial of trial of the trial of the trial of the trial of trial of the trial of the trial of	he system.
Trigger:	The administrator requests to up	pdate an employee's details.
	Actor Action:	system Response:
	The administrator requests to update an employee's details. 2	 The hydrotech system prompts the admin to select the employee they want to update.
Typical Course of Events:	3. The administrator selects the employee whose details they want to update 4. The administrator selects the employee whose details they want to update.	I. The system displays the employee information that has been retrieved from the Employee table with the following attributes:
		Employee_ID(PK)
		 EmployeeType_ID (FK) attribute of the EmployeeType entity where the EmployeeType_ID in the EmployeeType entity





		corresponds to the EmployeeType_ID in the Employee entity.
		 User_ID (FK) attribute of the User entity where the User_ID in the User entity corresponds to the User_ID in the Employee entity.
		o ID_Number
		o Cell_Num
		 EmployeeName
		o HireDate
		And prompts the administrator to edit the field/s they wish to.
	The administrator edits the details of the employee.	
		6. The modified employee information is saved by the system in the Employee entity under the following attributes:
		o Cell_Num
		○ EmployeeName
Alternate: Courses:	None	
Conclusion:	The use case concludes whe successfully updated.	n an employee has been
Post-condition:	The employee's information has been updated in the employee entity.	
Business Rules:	Only the administrator can up system	date an employee's details on the
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has interne	t connection







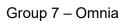
Use Case Name:	Delete employee	Use case type
Use Case ID:	6.4.	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case covers an incider wishes to remove an employee administrator will look for the er them after receiving confirmation employee is deleted from the E	from the system. The mployee and proceed to delete on. The use case ends when the
Pre-condition:	The administrator needs to be	e logged onto the system.
	The employee must exist alre	eady on the system.
Trigger:	The administrator requests to d	delete an employee.
	Actor Action:	system Response:
	The administrator requests to delete an employee	2. The HydroTech system prompts the Administrator to select the employee they want to delete.
Typical Course of Events:	3. The administrator selects the employee whom they want to delete.	4. The system prompts the administrator to confirm that they want to delete the selected employee.
or Evento.	5. The administrator confirms that they want to delete the selected employee.	6. The system removes the employee from the Employee entity with the following attributes:
	[ALT]	o Employee_ID [PK]
		o EmployeeType_ID [FK]
		o User_ID [FK]





	o ID_Number
	o Cell_Num
	o EmployeeName
	o HireDate
Alternate:	
	[ALT] Step 5- The administrator chooses to cancel the process. The Use Case is terminated.
Courses:	The Ose Case is terminated.
Conclusion:	The use case ends when an employee is successfully deleted
	from the Employee entity.
Post-condition:	The employee's information has been deleted from the employee entity.
Business Rules:	Only the administrator can update an employee's details on the
	system
Implementation	
Constraints and	None
Specifications:	
Assumptions:	The administrator has internet connection
Open Issues:	None







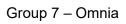
Use Case Name:	Add Employee Type	Use case type
Use Case ID:	6.6	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	,
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	to add a new type of employe administrator to enter the Typ new type of employee. The sy	eName and TypeDescription for the ystem validates and saves the new eeType entity. The use case ends nessage to indicate that the
Pre-condition:	The administrator needs to	be logged onto the system.
Trigger:	The administrator requests to	add a new type of employee.
	Actor Action:	system Response:
	The administrator requests to add a new	The HydroTech system prompts the
	type of employee.	administrator to enter the
		following information from the
Typical Course		EmployeeType entity:
of Events:		o TypeName
		 TypeDescription
	3. The administrator enters the new type of employee's details into the system.	4. The system captures and validates the new type of employee's details to ensure that all specified fields have been inputted.
		[ALT]





		5. The system autogenerates the EmployeeType_ID which is set to the next consecutive number in the EmployeeType entity. The system stores the new type of employee's details into the EmployeeType entity under the following attributes: TypeName TypeDescription
Alternate: Courses:	[ALT] Step 4b: The system displays a relevant error message that a field is empty/invalid. Return to step 3	
Conclusion:	The use case concludes wher employee type.	the system successfully adds an
Post-condition:	A new type of employee was entity.	s added to the EmployeeType
Business Rules:	Only the administrator can a system.	ndd an employee type on the
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has intern	et connection.
Open Issues:	None	

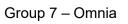






Use Case Name:	Update Employee Type	Use case type
Use Case ID:	6.7	Business Requirements: o
Priority:	Medium	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	to update an existing employee	vent where the administrator wants e type. The system prompts the putes of the employee type. The details of employee type in the
Pre-condition:	The administrator needs to b	pe logged onto the system.
	The employee type must exist	st on the system.
Trigger:	The administrator requests to	update the type of employee.
	Actor Action: system Response:	
	The administrator requests to update the type of employee.	2. The HydroTech system prompts the administrator to select an employee type
Typical Course of Events:	3. The administrator selects the employee type they wish to edit.	 4. The system displays the employee's type information that has been retrieved from the EmployeeType entity with the following attributes: TypeName TypeDescription
	5. The administrator edits the details of the employee type.	6. The system captures and stores the updated details in EmployeeType entity under either of the following attributes:

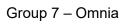






		 TypeName
		 TypeDescription
Alternate:	None	
Courses:	None	
Conclusion:	The use case concludes when employee type.	n system successfully updates the
Post-condition:	An employee type was upda	ated in the EmployeeType entity
Business Rules:	Only the administrator can usystem.	update an employee type on the
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has interr	net connection.
Open Issues:	None	

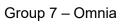






Use Case Name:	Delete Employee Type	Use case type
Use Case ID:	6.8	Business Requirements: o
Priority:	Low	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case covers an incide wishes to remove an employed administrator will look for the edelete them after receiving corwhen the employee type is delobject.	e type from the system. The employee type and proceed to nfirmation. The use case ends
Pre-condition:	The administrator needs to b	pe logged onto the system.
	The employee type must exist	st on the system.
Trigger:	The administrator requests to	delete an employee type.
	Actor Action:	system Response:
	The administrator requests to delete an employee type.	2. The HydroTech system prompts the administrator to select an employee type that they would like to delete.
Typical Course of Events:	3. The administrator selects the employee type they wish to delete.	4. The system prompts the administrator to confirm that they want to delete the selected employee type.
	5. The administrator confirms that they want to delete the selected employee type.	6. The system removes the employee type from the EmployeeType entity which has the following attributes:
	[ALT]	EmployeeType_ID [PK]
		- TypeName

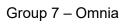






	- TypeDescription
Alternate:	[ALT] Step 5: The administrator chooses to cancel the process.
Courses:	The Use Case is terminated.
Conclusion:	The use case ends when an employee type is successfully deleted from the Employee type entity.
Post-condition:	An employee type was removed from the EmployeeType entity
Business Rules:	Only the administrator can delete an employee type on the system.
Implementation	
Constraints and	None
Specifications:	
Assumptions:	The administrator has internet connection.
Open Issues:	None







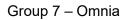
Use Case Name:	Room Check-In	Use case type
Use Case ID:	6.10	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Client	
Primary The system Actor:	Administrator	
Other Participating Actors:	None	
Other Interested Stakeholders:	Owner	
Description:	The use case begins when the c into their room booking.	lient initiates the check-in process
	The system should allow the admonom booking. Whereby the clier booking before their room booking RoomBooking entity. Once the acconfirmed the client's room book booking will be changed to "Checover the room keys to the client."	nt will be asked to confirm their ng record is retrieved from the dministrator has found and ing, the status of their room
	The use case concludes when the room keys to the client and their "Checked-In"	ne administrator hands over the room booking status is changed to
Pre-condition:	The administrator needs to	be logged onto the system.
	The client's accommodation check-in.	n must have been made prior to
	The screen displayed the "	View Accommodations" screen.
Trigger:	The client requests to check-in ir	nto their room booking reservation.
Typical Course	Actor Action: sy	stem Response:
Typical Course of Events:	check-in into their room	The administrator requests the client to provide their room booking details.





3. The client provides their room booking details.	4. The administrator searches and selects the relevant room booking reservation. The system responds by retrieving the client's accommodation booking by using the RoomBooking_ID attribute to locate the select room booking from the RoomBooking entity with the following attributes: RoomBooking_ID [PK] Client_ID [FK] RoomBookingStatus_ID [FK] Reminder_ID [FK] ReferenceNum BookingDate NumberOfGuests EntryDate ExitDate The administrator requests the client to provide their booking reference number to confirm their room checkin.
5. The client will provide their booking reference number.	6. The administrator will match the reference number given by the client to the reference number
	pulled up on the screen. [ALT] 7. The administrator confirms that the
	client's room is ready to be checked in. The employee hands over the key to the client. [ALT]

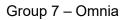






	8. The system retrieves the "Checked-In" status by reading the Name attribute within the RoomBookingStatus entity and updates the status of the client's room bookings.
	[ALT] Step 4: The system could not retrieve the Room Booking and Room Booking status information from the RoomBooking entity. Terminate use case.
Alternate: Courses:	[ALT] Step 6: The reference number given by the client does not match the reference number displayed on the system. Return to step 5.
	[ALT] Step 7: The room is not yet ready to be checked in. Inform Client to come back later. Return to step 1.
Conclusion:	The use case concludes when the administrator hands over the room keys to the client and their room booking status is changed to "Checked-In".
Post-condition:	The room booking status of the of the client's room booking is changed to "Checked-In"
Business Rules:	None
Implementation Constraints and Specifications:	The client has access to an active internet connection.
Assumptions:	None
Open Issues:	None







Use Case Name:	Room check-out	Use case type
Use Case ID:	6.11	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Client	
Primary The system Actor:	Administrator	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	check-out after using the room facilitating the check-out proce for the client specific event bod will retrieve the details of the e administrator will confirm detail concludes when the administrator	vent where the Client request to a. The administrator responds by ess. The administrator will search oking on the system. The system event entity. Thereafter the ells from the client. The use case eator checks the client out of the mBookingStatus entity will update
Pre-condition:	The administrator is logg	ed on to system.
	The client has already ch	necked-in the system.
Trigger:	The client arrives at the resort the administrator.	and wishes to get checked-out by
	Actor Action:	system Response:
Typical Course	The client arrives at the resort and wishes to get checked-out by the administrator.	2. The administrator asks the client for the booking reference number.
Typical Course of Events:	3. The client provides the booking reference number.	 The system retrieves the information for the event booking from the RoomBooking entity that contains the following information: Roombooking_ID [PK]



Group 7 – Omnia



	RoomBookingStatus_ID [FK]
	○ Client_ID [FK]
	○ Reminder_ID [FK]
	○ ReferenceNum
	○ BookingDate
	 NoOfGuests
	[ALT]
	5. By using the referenceNum attribute in the RoomBooking entity, the administrator searches for the booking that was made.
	[ALT]
	The administrator checks the client of the system.
	7. The system changes the Name attribute in the RoomBookingStatus to "Checked Out"
Alternate:	[ALT] Step 4: The system could not retrieve the Booking and Booking status information from the Booking entity. Use case ends.
Courses:	[ALT] Step 5: The booking could not be found. Go back to step 3.
Conclusion:	The client is checked-out and can no longer use the booked room for that day.
Post-condition:	The Name in the RoomBookingStatus entity has changed to "Checked Out"
Business Rules:	Only the administrator can check bookings in.
	The room check-out time is 10:00
Implementation Constraints and Specifications:	None
Assumptions:	The administrator has internet connection.
Open Issues:	None





Use Case Name:	Create VAT	Use case type
Use Case ID:	6.13	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	a new VAT record on the syste system prompts the administra The administrator enters the in verifies the information and th	event where the administrator creates em. The use case begins when the ator to enter the required information. Information required. The system en stores it. The use case concludes cessfully been created on the system.
Pre-condition:	The administrator needs to be on the VAT View.	
	The administrator needs	s to be logged onto the system.
Trigger:	The administrator requests to system.	create a new VAT record on the
	Actor Action:	system Response:
Typical Course of Events:	The administrator requests to create a new VAT record on the system. 3. The administrator enters the required information.	2. The system prompts the administrator to enter the required information concerning the VAT record from the VAT Table: o VATAmountDate o Date
	the required information into the input fields.	





	4. The administrator confirms the information entered when they select the '+' option. [ALT]	5. The system will validate the information entered is correct. Once the information is validated, the system will create a new record within the VAT Table which has the following attributes:
		o VAT_ID (PK)
		 VATAmount
		o Date
		[ALT]
		6. The system will then display the new record within the VAT table
Alternate:	[ALT] Step 4: The administrat correct. Return to Step 3.	tor denies the information entered is
Courses:	[ALT] Step 5: The system will information entered by the ac	throw an error due to incorrect Iministrator. Return to Step 3
Conclusion:	The use case concludes whe the VAT Table .	n the record is successfully added to
Post-condition:	 The new VAT record is in system's database. 	successfully added to the VAT Table
Business Rules:	Only authorised users of a VAT record on the system	of the system will be permitted create stem.
Implementation Constraints and Specifications:	None	
Assumptions:	None	
Open Issues:	None	





Use Case Name:	Update VAT	Use case type
Use Case ID:	6.14	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	the administrator chooses to uestimate the system will retrieve the inwhere the system will prompt required information. The adm	system. The use case begins when update a VAT record on the system. formation of the selected record, the user to enter the updated ninistrator enters the information the information and then stores it. In the VAT record has been
Pre-condition:	The administrator needsThe administrator needs	s to be on the VAT View. s to be logged onto the system.
Trigger:	The administrator requests to system.	update a VAT record on the
	Actor Action:	system Response:
Typical Course of Events:	The administrator requests to update a VAT record on the system.	2. The system retrieves the appropriate record from the system by matching the 'VAT_ID' attribute chosen by the administrator to the 'VAT_ID' record within the VAT Table: o VAT_ID (PK)
		o VATAmount
		o Date The system will then display the following attributed retrieved





		from the VAT record from the VAT Table: o VATAmount
		o Date
		The system will then prompt the administrator to enter the updated information within the appropriate input fields.
	3. The administrator enters the updated required information into the input fields.	
	4. The administrator confirms the updated information entered when they select the 'Save' option. [ALT]	5. The system will validate the information entered is correct. Once the information is validated, the system will update the chosen VAT record within the VAT Table which has the following attributes:
		o VAT_ID (PK)
		VATAmountDate
		[ALT]
		6. The system will then display the updated record within the VAT table.
Alternate:	[ALT] Step 4: The administrat correct. Return to Step 3.	or denies the information entered is
Courses:	[ALT] Step 5: The system will information entered by the ad	throw an error due to incorrect ministrator. Return to Step 3
Conclusion:	The use case concludes whe successfully updated on the s	

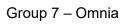


Group 7 – Omnia



Post-condition:	 The updated VAT record is successfully added to the VAT Table in system's database.
Business Rules:	 Only authorised users of the system will be permitted create a VAT record on the system.
Implementation Constraints and Specifications:	None
Assumptions:	None
Open Issues:	None

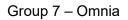






Use Case Name:	Update reminder timer	Use case type
Use Case ID:	6.17	Business Requirements:
Priority:	High	System Analysis: ☑
Source:	Platinum Island Resort	System Design: □
Primary Business Actor:	Administrator	
Primary The system Actor:	• None	
Other Participating Actors:	• None	
Other Interested Stakeholders:	• None	
Description:	Reminder entity. The use case	ge a reminders information in the se begins when the administrator ormation and enters modifications, lministrator is alerted that the
Pre-condition:	The administrator needs to be	e logged onto the system.
	The reminder must exist on the second contract of the second co	he system.
Trigger:	The administrator requests to	update a reminders details.
	Actor Action:	system Response:
	The administrator requests to update a reminders details.	The system prompts the user to select the reminder they want to update.
Typical Course of Events:	the reminder they want to update	 The system displays the reminder information that has been retrieved from the Reminder table with the following attributes: Reminder_ID (PK) Reminder_Type Reminder_Time And prompts the administrator to edit the field/s they wish to.







	The administrator edits the details of the reminder.
	The modified reminder information is saved by the system in the Reminder entity under the following attributes: Reminder_ID (PK) Reminder_Type Reminder_Time
	•
	The system notifies the user that the reminder has been successfully updated.
Alternate: Courses:	• None
Conclusion:	The use case concludes when a reminder has been successfully updated and the system notifies the administrator that the reminder has been successfully updated.
Post-condition:	The reminders information has been updated in the reminder entity.
Business Rules:	Only the administrator has access.
Implementation Constraints and Specifications:	• None
Assumptions:	The administrator has internet connection
Open Issues:	None



Group 7 – Omnia



USE CASE NAME:	Check-in ticket	USE CASE TYPE
USE CASE ID:	6.19.	Business Requirements:
PRIORITY:	High	System Analysis: ☑
SOURCE:	Platinum Island Resort	System Design:
PRIMARY BUSINESS ACTOR:	Client	
PRIMARY THE SYSTEM ACTOR:	Security Guard	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	Guard responds by facilitating the	rrives at the reception. The Security e check-in process. The Security ode, and allow the client into resort.
PRE-CONDITION:	The Client must have a valid booking.	
	The Client has arrived on the correct check-in date	
TRIGGER:	The Client arrives at Platinum Island Resort and requests to check in.	
	111.	and reservand requests to sheek
	ACTOR ACTION:	SYSTEM RESPONSE:
TVDICAL COLIDER		·
TYPICAL COURSE OF EVENTS:	ACTOR ACTION: 1. The Client arrives at Platinum Island Resort and	SYSTEM RESPONSE: 2. The security guard responds by requesting the client to provide
	ACTOR ACTION: 1. The Client arrives at Platinum Island Resort and requests to check in. [ALT] 3. The Client provides his/her	SYSTEM RESPONSE: 2. The security guard responds by requesting the client to provide their QR code. 4. The security guard scans the
	ACTOR ACTION: 1. The Client arrives at Platinum Island Resort and requests to check in. [ALT] 3. The Client provides his/her	 SYSTEM RESPONSE: 2. The security guard responds by requesting the client to provide their QR code. 4. The security guard scans the QR code. 5. The scan is successful, Client is allowed to enter platinum
OF EVENTS:	ACTOR ACTION: 1. The Client arrives at Platinum Island Resort and requests to check in. [ALT] 3. The Client provides his/her QR code.	SYSTEM RESPONSE: 2. The security guard responds by requesting the client to provide their QR code. 4. The security guard scans the QR code 5. The scan is successful, Client is allowed to enter platinum island resort. [Alt] 6. The Name attribute of the DayVisitStatus entity is updated to "Checked-In".
OF EVENTS:	ACTOR ACTION: 1. The Client arrives at Platinum Island Resort and requests to check in. [ALT] 3. The Client provides his/her QR code. [ALT] Step 1: The client does not	SYSTEM RESPONSE: 2. The security guard responds by requesting the client to provide their QR code. 4. The security guard scans the QR code 5. The scan is successful, Client is allowed to enter platinum island resort. [Alt] 6. The Name attribute of the DayVisitStatus entity is updated to "Checked-In". a arrive that day, the Name Attribute tes to "Forfeited".
OF EVENTS:	ACTOR ACTION: 1. The Client arrives at Platinum Island Resort and requests to check in. [ALT] 3. The Client provides his/her QR code. [ALT] Step 1: The client does not of the DayVisitStatus entity updates	SYSTEM RESPONSE: 2. The security guard responds by requesting the client to provide their QR code. 4. The security guard scans the QR code 5. The scan is successful, Client is allowed to enter platinum island resort. [Alt] 6. The Name attribute of the DayVisitStatus entity is updated to "Checked-In". It arrive that day, the Name Attribute tes to "Forfeited".
OF EVENTS: ALTERNATE COURSES:	ACTOR ACTION: 1. The Client arrives at Platinum Island Resort and requests to check in. [ALT] 3. The Client provides his/her QR code. [ALT] Step 1: The client does not of the DayVisitStatus entity updated [ALT] Step 4: The system is not stated that the client enters platinum island resorted.	SYSTEM RESPONSE: 2. The security guard responds by requesting the client to provide their QR code. 4. The security guard scans the QR code 5. The scan is successful, Client is allowed to enter platinum island resort. [Alt] 6. The Name attribute of the DayVisitStatus entity is updated to "Checked-In". It arrive that day, the Name Attribute tes to "Forfeited".



Group 7 – Omnia



IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	 The client must have made the booking in advance.
ASSUMPTIONS:	None
OPEN ISSUES:	None





Use Case Name:	Event check-in	Use case type
Use Case ID:	6.21	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Client	
Primary The system Actor:	Administrator	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	arrives at the resort, he then on reference number that was give locates the booking that was r	vent when the client wishes to se case starts when the client gives the administrator the booking ven to him. Once the administrator made, he checks the user in. The is is updated in the EventStatus
Pre-condition:	The administrator is logo The client made an ever	ged on to system. nt booking on the system.
Trigger:	The client arrives at the resort the administrator.	and wishes to get checked-in by
	Actor Action:	system Response:
	The client arrives at the resort and wishes to get checked-in by the administrator.	2. The administrator asks the client for the booking reference number.
Typical Course of Events:	3. The client provides the booking reference number.	 The system retrieves the information for the event booking from the Event entity that contains the following information: Event_ID [PK]
		EventStatus_ID [FK]
		○ Client_ID [FK]
		○ Venue_ID [FK]

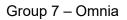


Group 7 – Omnia



the booking that was made. [ALT] 6. The system notifies the administrator that the event is ready to be checked in. [ALT] 7. The system changes the Name attribute in the EventStatus to "Checked In" [ALT] Step 4: The system could not retrieve the Event and Even status information from the Event entity. Use case ends. [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" Business Rules: Only the administrator can check bookings in. The event check-in time can begin from 7am.		○ Reminder_ID [FK]
o Date o EventTotal [ALT] 5. By using the referenceCode attribute in the Event entity, the administrator searches fo the booking that was made. [ALT] 6. The system notifies the administrator that the event is ready to be checked in. [ALT] 7. The system changes the Name attribute in the EventStatus to "Checked In" [ALT] Step 4: The system could not retrieve the Event and Even status information from the Event entity. Use case ends. [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" Business Rules: • Only the administrator can check bookings in. • The event check-in time can begin from 7am.		o GuestList
S. By using the referenceCode attribute in the Event entity, the administrator searches fo the booking that was made. [ALT] 6. The system notifies the administrator that the event is ready to be checked in. [ALT] 7. The system changes the Name attribute in the EventStatus to "Checked In" [ALT] Step 4: The system could not retrieve the Event and Even status information from the Event entity. Use case ends. [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" • Only the administrator can check bookings in. • The event check-in time can begin from 7am.		o ReferenceCode
[ALT] 5. By using the referenceCode attribute in the Event entity, the administrator searches fo the booking that was made. [ALT] 6. The system notifies the administrator that the event is ready to be checked in. [ALT] 7. The system changes the Name attribute in the EventStatus to "Checked In" [ALT] Step 4: The system could not retrieve the Event and Even status information from the Event entity. Use case ends. [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" • Only the administrator can check bookings in. • The event check-in time can begin from 7am.		o Date
5. By using the referenceCode attribute in the Event entity, the administrator searches fo the booking that was made. [ALT] 6. The system notifies the administrator that the event is ready to be checked in. [ALT] 7. The system changes the Name attribute in the EventStatus to "Checked In" [ALT] Step 4: The system could not retrieve the Event and Even status information from the Event entity. Use case ends. [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" Business Rules: Only the administrator can check bookings in. The event check-in time can begin from 7am.		○ EventTotal
attribute in the Event entity, the administrator searches fo the booking that was made. [ALT] 6. The system notifies the administrator that the event is ready to be checked in. [ALT] 7. The system changes the Name attribute in the EventStatus to "Checked In" [ALT] Step 4: The system could not retrieve the Event and Even status information from the Event entity. Use case ends. [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" Business Rules: • Only the administrator can check bookings in. • The event check-in time can begin from 7am.		[ALT]
6. The system notifies the administrator that the event is ready to be checked in. [ALT] 7. The system changes the Name attribute in the EventStatus to "Checked In" [ALT] Step 4: The system could not retrieve the Event and Even status information from the Event entity. Use case ends. [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" Business Rules: Only the administrator can check bookings in. The event check-in time can begin from 7am.		attribute in the Event entity, the administrator searches for
administrator that the event is ready to be checked in. [ALT] 7. The system changes the Name attribute in the EventStatus to "Checked In" [ALT] Step 4: The system could not retrieve the Event and Even status information from the Event entity. Use case ends. [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" • Only the administrator can check bookings in. • The event check-in time can begin from 7am.		[ALT]
Alternate: Courses: [ALT] Step 4: The system could not retrieve the Event and Even status information from the Event entity. Use case ends. [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" • Only the administrator can check bookings in. • The event check-in time can begin from 7am.		administrator that the event is ready to be checked in.
Alternate: Courses: [ALT] Step 4: The system could not retrieve the Event and Even status information from the Event entity. Use case ends. [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" • Only the administrator can check bookings in. • The event check-in time can begin from 7am.		7. The system changes the
Alternate: Courses: [ALT] Step 5: The event booking could not be found. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" Only the administrator can check bookings in. The event check-in time can begin from 7am.		
Courses: [ALT] Step 5: The event booking could not be lound. Terminate use case. [ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" Only the administrator can check bookings in. The event check-in time can begin from 7am.		[ALT] Step 4: The system could not retrieve the Event and Event status information from the Event entity. Use case ends.
[ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later. Conclusion: The client is checked-in and can use the booked venue for that day. Post-condition: The StatusName in the EventStatus entity has changed to "Checked-In" Business Rules: Only the administrator can check bookings in. The event check-in time can begin from 7am.		
Dost-condition: The StatusName in the EventStatus entity has changed to "Checked-In" Business Rules: Only the administrator can check bookings in. The event check-in time can begin from 7am.		[ALT] Step 6: The event is not yet ready to be checked in. Inform Client to come back later.
"Checked-In" Only the administrator can check bookings in. The event check-in time can begin from 7am.	Conclusion:	
The event check-in time can begin from 7am.	Post-condition:	,
	Business Rules:	Only the administrator can check bookings in.
Implementation		The event check-in time can begin from 7am.
Constraints and None Specifications:		None
Assumptions: • The administrator has internet connection.	Assumptions:	The administrator has internet connection.
Open Issues: None	Open Issues:	None







Use Case Name:	Event check-out	Use case type
Use Case ID:	6.22	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Client	•
Primary The system Actor:	Administrator	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	check-out after using the vent facilitating the check-out proof for the client specific event be will retrieve the details of the administrator will confirm deta concludes when the administ	event where the Client request to ue. The administrator responds by ess. The administrator will search poking on the system. The system event entity. Thereafter the ails from the client. The use case rator checks the client out of the entStatus entity will update to
Pre-condition:	The administrator is logThe client has already of	
Trigger:	The client arrives at the resort and wishes to get checked-out by the administrator.	
	Actor Action:	system Response:
	The client arrives at the resort and wishes to get checked-out by the administrator.	The administrator asks the client for the booking reference number.
Typical Course of Events:	3. The client provides the booking reference number.	4. The system retrieves the information for the event booking from the Event entity that contains the following information:
		o Event_ID [PK]
		o EventStatus_ID [FK]

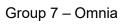


Group 7 – Omnia



	○ Client_ID [FK]
	○ Venue_ID [FK]
	Reminder_ID [FK]
	○ GuestList
	o ReferenceCode
	o Date
	○ EventTotal
	[ALT]
	5. By using the referenceCode attribute in the Event entity, the administrator searches for the booking that was made.
	[ALT]
	6. The administrator checks the client of the system.
	7. The system changes the Name attribute in the EventStatus to "Checked Out"
Alternate:	[ALT] Step 4: The system could not retrieve the Booking and Booking status information from the Booking entity. Use case ends.
Courses:	
	[ALT] Step 5: The booking could not be found. Go back to step 3.
Conclusion:	The client is checked-out and can no longer use the booked venue for that day.
Post-condition:	The StatusName in the EventStatus entity has changed to "Checked Out"
Business Rules:	Only the administrator can check bookings in.
	The event check-out time is 8pm.
Implementation Constraints and Specifications:	None
Assumptions:	The administrator has internet connection.
Open Issues:	None







Use Case Name:	Refund Room Booking	Use case type
Use Case ID:	6.26	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Client	
Primary The system Actor:	Administrator	
Other Participating Actors:	None	
Other Interested Stakeholders:	Owner	
Description:	The use case begins when t their room booking reservation	he client initiates the refund process of on.
	booking. Whereby the client before their room booking re RoomBooking entity. Once record, the administrator will room booking refund and will	the administrator has found the request confirmation on the client's I then complete the client's refund The system will then change the status
	refund booking EFT, the tran	en the administrator completes the isaction is logged on the system and boking status of the client's booking.
Pre-condition:	The administrator need	ds to be logged onto the system.
	 The client's accommod cancelled first. 	dation booking must have been
	The client's room book "Refund Pending" " " " " " " " " " " " " " " " " " "	ing status must have a status of
Trigger:	The client requests to refund	I their room booking reservation.
Typical Course	Actor Action:	system Response:
Typical Course of Events:	The client requests to refund their room booking reservation.	2. The administrator requests the client to provide their room booking reference number.





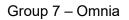
3. The client provides their room booking reference number.	4. The administrator searches and selects the relevant room booking reservation. The administrator requests confirmation on the refund process by communicating the amount due to the client.
	[ALT]
5. The client will confirm the refund process as well as the amount due.	6. The administrator requests the client to provide their banking account details.
[ALT] 7. The client will provide their bank account details.	8. The administrator completes an online EFT payment by making use of the client's bank account details provided and the <i>Amount</i> attribute value from the RoomPayment entity
	9. Once the transaction has been approved and completed. The administrator will confirm the refund of the client's room booking reservation.
	The system then responds by retrieving the client's accommodation booking by using the RoomBooking_ID attribute to locate the selected room booking from the RoomBooking entity with the following attributes:
	o RoomBooking_ID [PK]
	○ Client_ID [FK]
	RoomBookingStatus_ID [FK]
	o ReferenceNum
	○ BookingDate





		 NumberOfGuests
		o EntryDate
		o ExitDate
		The system will also retrieve the relevant RoomPayment record by using the <i>RoomBooking_ID [FK]</i> within the RoomPayment entity to retrieve the <i>Amount</i> attribute.
		The system retrieves the "Refunded" status by reading the <i>Name</i> attribute within the RoomBookingStatus entity and updates the status of the client's room bookings.
		The system will create a new record within the RefundRoomBooking entity with the following attributes:
		o Refund_ID [PK]
		o RoomBooking_ID [FK]
		o Amount
		[ALT]
		e number given by the client does not numbers displayed on the system
Alternate: Courses:	booking refund as it was ca	room booking was not valid for a room ncelled less than 48 hours before the ne RoomBooking entity. Terminate Use
	[ALT] Step 5: The client refu booking process. Terminate	uses to continue with their refund room Use Case.







	[ALT] Step 9a: The transaction was not approved and was unsuccessful. Return to Step 8.
	[ALT] Step 9b: The system was unable to create a new record within the RefundRoomBooking entity and will return an error to the system. Terminate Use Case.
Conclusion:	The use case concludes when the administrator when the administrator completes the refund booking EFT, the transaction is logged on the system and the system updates room booking status of the client's booking.
Post-condition:	The room booking status of the of the client's room booking is changed to "Refunded" and a new record is added to the RoomBookingRefund entity.
Business Rules:	 The system will offer a 25% refund amount on the total payment of the client's room booking should the cancellation have occurred more than 48 hours before the client's check- in time.
	 The system will not offer a refund option to client's who cancel their room booking reservation less than 48 hours before the client's check-in time
Implementation Constraints and Specifications:	None
Assumptions:	The client provides the correct banking details.
Open Issues:	None





Use Case Name:	Refund Event Booking	Use case type
Use Case ID:	6.27	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Client	
Primary The system Actor:	Administrator	
Other Participating Actors:	None	
Other Interested Stakeholders:	Owner	
Description:	The use case begins when the coof their event booking reservation	•
	The system should allow the administrator to refund a client's event booking. Whereby the client will be asked to confirm their booking before their event booking record is retrieved from the Event entity. Once the administrator has found the record, the administrator will request confirmation on the client's event booking refund and will then complete the client's refund payment off of the system. The system will then change the status of their event booking to "Refunded".	
	The use case concludes when the refund booking EFT, the transact the system updates the event booking.	tion is logged on the system and
Pre-condition:	The administrator needs to	be logged onto the system.
	The client's event booking	must have been cancelled first.
	 The client's event booking "Refund Pending" 	status must have a status of
Trigger:	The client requests to refund the	ir event booking reservation.
Typical Course	Actor Action:	system Response:
of Events:	The client requests to refund their event booking reservation. 2.	The administrator requests the client to provide their event booking reference code.





3. The client provides their event booking reference code.	4. The administrator searches and selects the relevant event booking reservation. The administrator requests confirmation on the refund process by communicating the amount due to the client. [ALT]
5. The client will confirm the refund process as well as the amount due. [ALT]	6. The administrator requests the client to provide their banking account details.
7. The client will provide their bank account details.	8. The administrator completes an online EFT payment by making use of the client's bank account details provided and the Amount attribute value from the EventPayment entity 9. Once the transaction has been approved and completed. The administrator will confirm the refund of the client's event booking reservation. The system then responds by retrieving the client's event booking by using the Event_ID attribute to locate the selected event booking from the Event entity with the following attributes: © Event_ID [PK] © Client_ID [FK] © Venue_ID [FK] © EventStatus_ID [FK]





	o ReferenceCode
	o Date
	o GuestList
	 EventTotal
	 Description
	The system will also retrieve the relevant EventPayment record by using the <i>Event_ID [FK]</i> within the EventPayment entity to retrieve the <i>Amount</i> attribute.
	The system retrieves the "Refunded" status by reading the <i>Name</i> attribute within the EventStatus entity and updates the status of the client's event bookings.
	The system will create a new record within the EventRefund entity with the following attributes:
	o Refund_ID [PK]
	o Event_ID [FK]
	o Amount
	[ALT]





	[ALT] Step 4a: The reference number given by the client does not match any of the reference numbers displayed on the system Return to Step 3.
	[ALT] Step 4b: The client's event booking was not valid for an event booking refund as it was cancelled less than 48 hours before the <i>Date</i> attribute within the Event entity. Terminate Use Case.
Alternate:	
Courses:	[ALT] Step 5: The client refuses to continue with their refund event booking process. Terminate Use Case.
	[ALT] Step 9a: The transaction was not approved and was unsuccessful. Return to Step 8.
	[ALT] Step 9b: The system was unable to create a new record within the EventRefund entity and will return an error to the system. T erminate Use Case.
Conclusion:	The use case concludes when the administrator when the administrator completes the refund booking EFT, the transaction is logged on the system and the system updates event booking status of the client's booking.
Post-condition:	The event booking status of the of the client's event booking is changed to "Refunded" and a new record is added to the EventRefund entity.
Business Rules:	The system will offer a 25% refund amount on the total payment of the client's event booking should the cancellation have occurred more than 48 hours before the client's check-in time.
	 The system will not offer a refund option to client's who cancel their event booking reservation less than 48 hours before the client's check-in time
Implementation Constraints and Specifications:	None
Assumptions:	The client provides the correct banking details.
Open Issues:	None



Narratives



2.7. Subsystem 7 – Inventory

Use Case Name:	Add Item	Use case type
Use Case ID:	7.2	Business Requirements: o
Priority:	Low	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	Owner/Supplier	
Description:	This use case begins when th new item to the system.	e administrator requests to add a
		e administrator to enter the items strator will provide the item details.
	This use case concludes when added to the Item entity.	n the item has been successfully
Pre-condition:	The administrator is logo permissions to create a	ged in has the necessary new item.
Trigger:	The administrator requests to	add an item.
	Actor Action:	system Response:
		2. The system prompts the administrator to enter the required information regarding the Item from the Item entity:
Typical Course of Events:	1 The administrator	o Name
	The administrator requests to add an item.	 Description
		o QtyOnHand
		 ItemType_ID.Name [FK] which is connected to the ItemType entity by matching the





	3. The administrator enters the required details for the new item.	ItemType_ID [FK] to the ItemType_ID [PK] in the ItemType entity. 4. The system will validate the information entered by the administrator and will prompt the administrator to confirm the item details once validated.
	5. The administrator will	[ALT] 6. The system will respond by
	confirm the creation of the new item record. [ALT]	creating a new record within the Item entity which has the following attributes: o Item_ID [PK]
		ltemType_ID [FK]Name
		Description
		○ QtyOnHand
		[ALT]
	[ALT] Step 3: The administrat the new item. Terminate Use	or declines confirmation of adding Case.
Alternate: Courses:	[ALT] Step 4: The information returned a validation error. Re	
	[ALT] Step 6: The system is u Item record and will return an	nable to successfully create the error. Terminate Use Case.
Conclusion:	The use case concludes whe the system	n the new item has been added to
Post-condition:	A new item record has been o	created within the Item entity
Business Rules:	None	
Implementation Constraints and Specifications:	None	
Assumptions:	None	
Open Issues:	None	





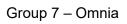
Use Case Name:	Update Item	Use case type
Use Case ID:	7.3	Business Requirements: o
Priority:	Low	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case begins when th an existing item on the system	e administrator requests to update า.
	The system should prompt the updated item record details af record details. After which, the providing the relevant updated confirm the creation of the iter	ter retrieving the appropriate administrator will respond by titem record details and will
	This use case concludes when updated.	n the item has been successfully
Pre-condition:	The administrator is logg	ged into the system
Trigger:	The administrator requests to the system.	update an existing item record on
	Actor Action:	system Response:
Typical Course of Events:	The administrator requests to update an existing item record on the system.	2. The system retrieves the appropriate record from the system by matching the Item_ID attribute chosen by the administrator to a specific record which matches the Item_ID record within the Item entity which has the following attributes:
		o Item_ID [PK]
		 ItemType_ID [FK] attribute of the Item entity which





		corresponds to the ItemType_ID [PK] within the ItemType entity. Name Description The system will then respond by requesting the administrator to provide the updated item details.
	3. The administrator provides the items updated details.	4. The system requests the administrator to confirm the updated item details.
	5. The administrator confirms the updated record entry for the item. [ALT]	6. The modified item's details are first validated before it is saved by the system in the Item entity under the following attributes:
		o Item_ID [PK]
		o ItemType_ID [FK]
		o Name
		 Description
		[ALT]
		7. The system notifies the user that the item has been successfully updated.
Alternate:		tor chooses to not update the e update of the item. Terminate use
Courses:	[ALT] Step 6: The information returns a validation error. Ret	n entered by the administrator turn to Step 3.
Conclusion:	Use case concludes when ite updated	em details have been successfully
Post-condition:	The item record has been su	ccessfully updated on the system.
Business Rules:	None	
Assumptions:	None	
Open Issues:	None.	

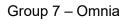






Use Case Name:	Delete Item	Use case type
Use Case ID:	7.4	Business Requirements: o
Priority:	Low	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	Owner	
Description:	This use case describes the ellike to delete a specific item re	vent where the administrator would ecord.
	The use case begins when the on the item deletion before the deletion of the record.	e system will request confirmation administrator confirms the
	The use case concludes when from the system.	the item record has been deleted
Pre-condition:	The administrator is logg	jed in onto the system.
	The item exists on the sy	ystem before it can be deleted.
Trigger:	The administrator requests to system.	delete an Item record from the
	Actor Action:	system Response:
		2. The system will retrieve the record from the Item entity using the Item_ID [PK] and retrieve the attributes with that associated record:
Typical Course of Events:	1. The administrator	o Item_ID [PK]
	requests to delete an Item record from the system.	ItemType_ID [FK]
	,	o Name
		 Description
		The system will confirm the deletion of the record by







		prompting the administrator the confirm the record's deletion.
	3. The administrator confirms the deletion of the item record. [ALT]	4. The system will respond deleting the appropriate item record from the Item entity with the following attributes: Item_ID [PK] ItemType [FK] Name Description
		The system will display a notification informing the administrator of the successful deletion of the item record
Alternate: Courses:	[ALT] Step 3: The administrate of the record. The use case te	or will decide to cancel the deletion erminates.
Conclusion:	Use case concludes when iter entity.	m has been deleted from the Item
Post-condition:	The item record has been suc entity.	cessfully removed from the Item
Business Rules:	None	
Implementation Constraints and Specifications:	None	
Assumptions:	None	
Open Issues:	None.	

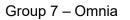


Group 7 – Omnia



Use Case Name:	Add Item Type	Use case type
Use Case ID:	7.6	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	Supplier	
Description:	the side-navigation panel and system displays the Add Item option and is then prompted to	e administrator navigates through clicks on the Item Type option. The Type option. The user clicks this pad an item type. This use case has been successfully added to
Pre-condition:	The Administrator is logged in the necessary permissions to	to the inventory system and has create a new item.
Trigger:	The administrator clicks the Adtype page.	dd item type button on the item
	Actor Action:	system Response:
	The administrator clicks the Add item type button on the item type page.	2. The system displays a modal with the required details for the new item.
Turning Course	3. The administrator enters the required details for the new item.	
Typical Course of Events:	4. The administrator saves the new item to the ItemType entity with the following attributes:	The system saves the new record to the ItemType table.[ALT]
	ItemType_ID (PK)ItemCategory_ID (FK)NameDescription	

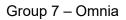






Alternate:	[ALT] Step 5: Not all fields were filled in correctly, the system displays a pop up requesting the user to check all fields
Courses:	displays a pop up requesting the user to check all fields
Conclusion:	The use case concludes when the new item type has been added to the system
Post-condition:	A new item is present in the Item Type table
Business Rules:	Only administrators can add new items onto the system.
Implementation	
Constraints and	None
Specifications:	
Assumptions:	Administrator is logged into the system
Open Issues:	None

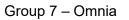






Use Case Name:	Update Item Type	Use case type
Use Case ID:	7.7	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	Owner	
Description:	the side-navigation panel and system displays the update ite option and is then prompted to	e administrator navigates through clicks on the Item type option. The m type option. The user clicks this o search for the item type to les when the item type has been
Pre-condition:	Admin is logged in	
Trigger:	The user navigates to the item "Update" button.	details page and selects the
	Actor Action:	system Response:
Typical Course of Events:	The administrator wishes to update the item type details on the system.	2. The system displays with the current details of the item type from the ItemType entity with the following attributes: o ItemType_ID (PK) o ItemCategory_ID (FK) o Name o Description 4. The system validates the input
	fields they want to update and submits the form.	and updates the item type with the new details in the ItemType entity.The system displays a
		success message indicating

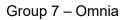






	that the item has been updated.
Alternate:	[ALT] Step 4: If the user inputs invalid data in the form, the
Courses:	system displays an error message and prompts the user to correct the errors before resubmitting.
Conclusion:	Use case concludes when item type details have been updated
Post-condition:	Item type details has been changed in the ItemType entity.
Business Rules:	Only administrators can update item type details.
Implementation Constraints and Specifications:	None
Assumptions:	Administrator has internet access.
Open Issues:	None.







Use Case Name:	Delete Item Type	Use case type
Use Case ID:	7.8	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	Owner	
Description:	item type from the system. Th a user who needs to remove a	ows a user to delete an existing is use case is typically initiated by an item type from the system udes when item type has been
Pre-condition:	Admin is logged in	
Trigger:	The admin requests to delete	an item type from the system.
	Actor Action:	system Response:
	The admin requests to delete an item type from the system.	2. The system requests confirmation from the admin to delete the item type.
Typical Course of Events:	3. The admin confirms the deletion. [ALT]	 4. The system removes the item type from the Item table permanently where the ItemType_ID matches the ID of the item type requested to be deleted. 5. The system notifies admin
		that item type was deleted.
Alternate: Courses:	[ALT] Step 3: User selects does not confirm, end use case.	
Conclusion:	Use case concludes when item type has been deleted	
Post-condition:	Item type removed from database.	
Business Rules:	Only administrators can delete	e item types from the system.

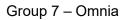


Group 7 – Omnia



Assumptions:	Administrator has internet access.
Open Issues:	None.







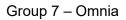
Use Case Name:	Add Item Category	Use case type
Use Case ID:	7.10	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Administrator	•
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event where the administrator creates a new Item Category record on the system. The use case begins when the system prompts the administrator to enter the required information. The administrator enters the information required. The system verifies the information and then stores it. The use case concludes when the Item Category record has successfully been created on the system.	
Pre-condition:	The administrator needs navigate from the <i>Items View</i> . The administrator needs to be leaved onto the acceptance.	
Trigger:	The administrator needs to be logged onto the system. The administrator requests to create a new Item Category record on the system.	
	Actor Action:	system Response:
Typical Course of Events:	The administrator requests to create a new Item Category record on the system.	 2. The system prompts the administrator to enter the required information regarding the Item Category from the Item Category Table: o Name o Description
	3. The administrator enters the required information into the input fields.	, ···





	4. The administrator confirms the information entered when they select the 'Create' option. [ALT]	5. The system will validate the information entered is correct. Once the information is validated, the system will create a new record within the Item Category Table which has the following attributes: o ItemCategory_ID (PK) Name o Description
		[ALT]
		6. The system will then display the new record within the Item Category Table
Alternate:	[ALT] Step 4: The administrate correct. Return to Step 3.	or denies the information entered is
Courses:	[ALT] Step 5: The system will information entered by the ad	throw an error due to incorrect ministrator. Return to Step 3
Conclusion:	The use case concludes when to the Item Category Table.	n the record is successfully added
Post-condition:	The new VAT record is s Category Table in system	successfully added to the Item em's database.
Business Rules:	Only authorised users of create an Item Category	f the system will be permitted record on the system.
Implementation Constraints and Specifications:	None	
Assumptions:	None	
Open Issues:	None	







Use Case Name:	Update Item Category	Use case type
Use Case ID:	7.11	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	begins when the administrator Category record on the syster information of the selected rec the user to enter the updated administrator enters the inform verifies the information and th	cord on the system. The use case of chooses to update an Item of the system will retrieve the cord, where the system will prompt required information. The mation required. The system
Pre-condition:	The administrator needs to be logged onto the system.	
Trigger:	The administrator requests to update an Item Category record on the system.	
	Actor Action:	system Response:
	The administrator requests to update an Item Category record on the system.	2. The system prompts the administrator to enter the updated required information regarding the item category.
Typical Course of Events:		3. The system retrieves the appropriate record from the system by matching the 'ItemCategory_ID' attribute chosen by the administrator to the 'ItemCategory_ID' record within the Item Category_Table: o ItemCategory_ID (PK)



Group 7 – Omnia



		o Name
		 Description
		The system will then display the 'name' and 'description' attributes retrieved from the Item Category record within the appropriate input fields, where the record can be updated by the administrator.
	4. The administrator enters the updated required information into the input fields.	
	5. The administrator confirms the updated information entered when they select the 'Save option. [ALT]	6. The system will validate the information entered is correct. Once the information is validated, the system will update the chosen Item Category record within the Item Category Table which has the following attributes: ItemCategory_ID (PK) Name Description
		[ALT]
		7. The system will then display the updated record within the Item Category table.
Alternate:	[ALT] Step 5: The administrator denies the information entered is correct. Return to Step 4.	
Courses:	[ALT] Step 6: The system will information entered by the ad	throw an error due to incorrect ministrator. Return to Step 4.
Conclusion:	The use case concludes whe been successfully updated or	n the Item Category record has n the system.

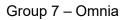


Group 7 – Omnia



Post-condition:	The updated Item Category record is successfully added to the Item Category Table in system's database.
Business Rules:	 Only authorised users of the system will be permitted to update an Item Category record on the system.
Implementation Constraints and Specifications:	None
Assumptions:	None
Open Issues:	None







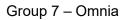
Use Case Name:	Delete Item Category	Use case type
Use Case ID:	7.12	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Administrator	1
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event where the administrator would like to delete a specific item category record. The use case begins with the system verifying there are no sale items associated with the item category. The system will confirm the item category deletion before the administrator confirms the deletion of the record. The use case concludes when the item category record has been deleted from the system.	
Pre-condition:	The administrator needs to be logged onto the system.	
	The item category must exist on the system before it can be deleted	
Trigger:	The administrator requests to from the system.	delete an Item Category record
	Actor Action:	system Response:
Typical Course of Events:	The administrator requests to delete an Item Category record from the system.	2. The system matches the ItemCategory_ID (PK) for the item category selected by the administrator to the item category from the ItemCategory Table, with the ItemCategory_ID (FK) retrieved from the ItemType Table, to ensure no inventory item types are associated with the selected item category. [ALT]
		The system will retrieve the record from the ItemCategory





		Table with the following attributes:
		○ ItemCategory_ID (PK)
		o Name
		 Description
		The system will confirm the deletion of the record by prompting the administrator the confirm the record's deletion.
	4. The administrator confirms the deletion of the item category record by selecting the 'Delete' option.	5. The system will respond deleting the appropriate item category record from the Item Category Table with the following attributes:
	[ALT]	o ItemCategory_ID (PK)
		○ Name
		 Description
		6. The system will display a notification informing the administrator of the successful deletion of the item category record
Alternate: Courses:	associated with the item cate	determine that there are item types gory. The system will then inform ypes that are associated with the will terminate.
Courses.	[ALT] Step 5: The administrat of the record. The use case to	or will decide to cancel the deletion erminates.
Conclusion:	The use case concludes when the system notifies the administrator of the deletion of the item category record.	
Post-condition:	The Item Category record is successfully deleted from the Item Category Table in system's database.	
Business Rules:	 Only authorised users o update an Item Categor 	of the system will be permitted to y record on the system.
Implementation Constraints and Specifications:	None	
Assumptions:	None	







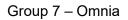
Use Case Name:	Add supplier	Use case type
Use Case ID:	7.14	Business Requirements: o
Priority:	Low	System Analysis: þ
Source:	Platinum Island resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	use to view the supplier's curre	
Pre-condition:	The admin must have access to internet.	
	The admin must be logged o	nto the system.
Trigger:	The admin wishes to add a new	w supplier to the system.
	Actor Action:	system Response:
	The admin wishes to add a new supplier to the system.	2. The Hydrotech system will prompt the admin to enter the following information from the Supplier table with the following attributes:
Typical Course		o FullName
of Events:		o Email
		o ContactNum
		 Description
		They system then prompts the user to enter the supplier information.





	3. The admin enters the information from the previous step.	4. The system captures and validates the new supplier attributes entered by checking the fields to ensure that the information is correct and that all fields specified as required have been inputted. [ALT]
		The system stores the new supplier information into the corresponding entities under
		the following attributes:
		Supplier_ID(PK)FullName
		– FullName – Email
		- ContactNum
		Description
	[ALT] step 4a - The system p that a field is empty. Go back	rovides an error message indicating to step 3.
Alternate: Courses:	[ALT]step 4b - The system pr that a field is invalid. Go back	ovides an error message indicating to step 3.
Conclusion:	The use case concludes, and platinum island database.	I the new supplier is added onto the
Post-condition:	The admins can now find the system.	details of the new supplier on the
Business Rules:	Only admins can ad	ld new suppliers to the system.
Implementation Constraints and Specifications:	None	
Assumptions:	The admin has internet acces	SS.
Open Issues:	None	

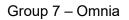






Use Case Name:	Update supplier	Use case type
Use Case ID:	7.15	Business Requirements: o
Priority:	Low	System Analysis: þ
Source:	Platinum Island resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes an occ administrator wishes to change Supplier entity. The use case b wants to edit a supplier's inform the supplier's information has b	a supplier's information in the begins when the administrator nation and enters modifications,
Pre-condition:	The admin must have accessThe admin must be logged or	
Trigger:	The administrator requests to u	pdate a supplier's details.
	Actor Action:	system Response:
	The administrator requests to update a supplier's details.	2. The system prompts the admin to select a supplier that they want to update.
Typical Course of Events:	3. The admin selects a supplier that they would like to update	 The system displays the Supplier information that has been retrieved from the Supplier entity with the following attributes: FullName Email ContactNum Description

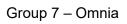






		And prompts the admin to enter the updated details for the supplier.
	5. The administrator edits the details of the supplier.	6. The modified supplier information is saved by the system in the Supplier entity under the following attributes:
		FullName
		– Email
		ContactNum
		Description
Alternate: Courses:	None	
Conclusion:	The use case concludes, and and stored onto the database	the supplier details is now updated
Post-condition:	The details regarding the supplier have been updated and stored in the Supplier table.	
Business Rules:	Only admins can update details for suppliers on the system.	
Implementation Constraints and Specifications:	None	
Assumptions:	The admin has internet acces	SS.
Open Issues:	None	

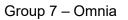






Use Case Name:	Delete supplier	Use case type
Use Case ID:	7.16	Business Requirements: o
Priority:	Low	System Analysis: þ
Source:	Platinum Island resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	will look for the supplier and pro	om the system. The administrator occeed to delete them after case ends when the supplier is
Pre-condition:	The admin must have access	to internet.
	The admin must be logged or	nto the system.
Trigger:	The admin wants to delete an e	existing supplier.
	Actor Action: system Response:	
	The admin wants to delete an existing supplier.	2. The system prompts the admin to select a supplier
	3. The admin selects a supplier that they would like to delete	4. The system prompts the admin to confirm that they would like to delete the selected supplier
Typical Course of Events:	5. The admin then confirms that they want to delete the selected supplier [ALT]	6. The system removes the supplier from the Supplier entity with the following attributes:
		Supplier_ID(PK)
		FullName
		– Email
		ContactNum

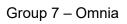






	Description	
Alternate:	[ALT]step 5: The admin declines the deletion, and the use case	
Courses:	is terminated.	
Conclusion:	The use case concludes, and the supplier is now deleted from the database. The admin is notified about this deletion.	
Post-condition:	The supplier is removed from the Supplier entity.	
Business Rules:	Only the admin can delete supplier's details on the system.	
Implementation Constraints and Specifications:	None	
Assumptions:	The admin has internet access.	
Open Issues:	None	

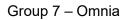






Use Case Name:	Place supplier order	Use case type
Use Case ID:	7.18	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	Supplier	
Other Interested Stakeholders:	None	
Description:	This use case outlines the event in which an administrator initiates a stock order for a business. The process commences when the administrator carefully selects the desired quantity, supplier, and specific stock items required for the business operations. Once all selections are made, the administrator proceeds to place the order, triggering an automated email notification to the chosen supplier. This comprehensive process ensures a seamless and efficient stock replenishment procedure.	
Pre-condition:	The administrator needs to be logged onto the system.	
	The administrator must have	e internet access.
Trigger:	The administrator requests to	place a new order for stock
	Actor Action:	system Response:
	The administrator requests to place a new order for stock.	2. The HydroTech system prompts the
	order for stock.	administrator to enter the
Typical Course		following information from the
of Events:		SupplierOrder entity:
		 FullName using the Supplier_ID [PK] in the Supplier entity that corresponds with the Supplier_ID [FK] in the SupplierOrder entity.

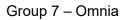






		 Name using the Item_ID [PK] in the Item entity that corresponds with the Item_ID [FK] in the SupplierOrder entity.
		o Quantity
		The system prompts the admin to enter the above information.
	3. The administrator enters the order information.	4. The HydroTech system captures the entered order information using the Supplier_Order
	5. The admin chooses to place the order [ALT]	6. The order is sent via email to the chosen supplier.
Alternate: Courses:	[ALT] Step 5: The admin decident Terminate use case.	des not to place an order.
Conclusion:	The use case concludes when owner.	n the new order is sent to the
Post-condition:	The order is placed, and an email is sent to supplier.	
Business Rules:	Only the administrator can place orders	
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has interr	net connection.
Open Issues:	None	

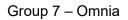






Use Case Name:	Cancel supplier order	Use case type
Use Case ID:	7.19	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Administrator	·
Primary The system Actor:	None	
Other Participating Actors:	Supplier	
Other Interested Stakeholders:	None	
Description:	This use case outlines a situation in which the administrator wishes to cancel the order that was placed to the supplier. The use case starts when the administrator clicks the cancel button, an automated email is then sent to the supplier indicating to him that the previous order that was placed needs to be cancelled as it is no longer needed.	
Pre-condition:	The administrator needs to be logged onto the system.	
	The administrator must have internet access.	
	An order must previously be placed.	
Trigger:	The administrator requests to previously placed.	cancel an order of stock that was
	Actor Action:	system Response:
Typical Course	The administrator requests to cancel an order of stock that was previously placed.	2. The HydroTech system displays a pop-up confirmation to make sure that the admin wants to cancel the order.
of Events:	3. The administrator confirms that he wishes to cancel the order that was placed. [ALT]	4. The HydroTech system then sends an automated email to the appropriate supplier informing them that the order should be cancelled. The order is then deleted from the Supplier_Order table.

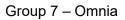






	The SupplierOrderStatus entity is linked to the SupplierOrder entity via the SupplierOrderStatus_ID [PK, FK]. The system will retrieve the name attribute from the SupplierOrderStatus table. The name attribute will change to "cancelled".
	5. The order is sent via email to the chosen supplier and the order will be cancelled.
Alternate: Courses:	[ALT] Step 3: The admin decides not to cancel an order. Terminate use case.
Conclusion:	The use case concludes when the order that was placed is now cancelled.
Post-condition:	An email is sent to the supplier informing him about the cancellation.
Business Rules:	 Only the administrator can cancel orders. The order can only be cancelled within 24 hours.
Implementation Constraints and Specifications:	None
Assumptions:	The administrator has internet connection.
Open Issues:	None







Use Case Name:	Receive supplier order	Use case type
Use Case ID:	7.20	Business Requirements:o
Priority:	Medium	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Administrator	•
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	Supplier	
Description:	This use case outlines an event i	n which the admin wants to
Pre-condition:	The administrator needs to be logged onto the system.	
	The administrator must have internet access.	
	An order must previously be placed.	
Trigger:	This use case describes the process of receiving and tracking stock after placing an order. Once the administrator receives the stock, they mark it as checked or received in the system. Additionally, the system automatically removes information about received orders from the table once the order is checked or cancelled and an email will be sent to the supplier informing him that the stock was received.	
	Actor Action:	system Response:
Typical Course of Events:	After the stock order has been placed and the supplier has dispatched the items, the administrator receives the shipment at the designated receiving area.	
of Evolito.	2. The administrator cross-checks the received items against the order details to ensure accuracy, verifying the quantity, item names, and any additional specifications.	





	 3. The administrator then navigates to the view supplier orders screen where a list of placed orders is displayed and checks the received order. 4. The system shows that the order was checked and confirmed that the supplier sent the stock. [ALT] 	
	 5. The administrator chooses to confirm that the order has been received. 6. The SupplierOrderStatus entity is linked to the SupplierOrder entity via the SupplierOrderStatus_ID [PK, FK]. The system will retrieve the name attribute from the SupplierOrderStatus table. The name attribute will change to "received". 	
	7. An email is sent to the supplier informing him the stock was received by the business.	
Alternate: Courses:	[ALT] Step 4: The administrator decides not to check the checkbox indicating that the order was not received. Terminate use case.	
Conclusion:	The use case concludes when the order that was placed, is now received.	
Post-condition:	The checkbox is checked indicating the stock was received and confirmed by the administrator.	
Business Rules:	Only the administrator can mark the received orders as checked.	
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has internet connection.	
Open Issues:	None	



Narratives



2.8. Subsystem 8 – Reports

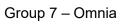
Use Case Name:	Generate peak season day visit report	Use case type
Use Case ID:	8.2	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island resort	System Design: o
Primary Business Actor:	Administrator (PBA)	•
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	In this case, the administrator asks for the creation of a detailed report that shows day visits at the busiest times of the year. The system effectively collects information on reservations for day trips and groups them according to each month. The algorithm then dynamically creates a line graph that visually illustrates the variances in ticket sales throughout various months. The use case concludes when the administrator downloads the report.	
Pre-condition:	The administrator is logged onto the system.	
Trigger:	The administrator wishes to generate a peak season day visit report	
	Actor Action:	system Response:
Typical Course	The administrator wishes to generate a peak season day visit report	The system requests the administrator to enter the date period for the accommodation summary report
of Events:	The administrator provides their date period of choice	The system validates that the date period entered by the administrator is valid. [ALT]
	5.	. The system retrieves the number of instances that a





		DayVisit booking was made on the system and retrieves the attributes from the DayVisit entity.
		The following attributes will be retrieved from the DayVisit entity:
		○ DayVisit_ID [PK]
		○ DayVisitStatus [FK]
		o Client_ID [FK]
		○ DayVisitDate_ID [FK]
		o RefCode
		o Total
		[ALT]
		6. The system then displays the total number of ticket sales filtered by month.
		The system prompts the admin to print the report.
	7. The administrator chooses to print the report. [ALT]	8. The system then responds by generating the PDF for download of the report with the currently selected date aggregation.
	ALT] Step 4: The system does information provided by the acReturn to step 3.	-
Alternate:	[ALT] Step 5: The system is u data. Use Case Terminates.	nable to retrieve the necessary
Courses:	[ALT] Step 7a: The administra Use Case Terminates.	itor does not select to print report.
	[ALT] Step 7b: The administra date to generate the report. R	eturn to step 3.







Conclusion:	The use case concludes when the system generates the peak season day visit report and displays it in PDF format for download.
Post-condition:	The administrator can now view the data on which month of the year had the most sales.
Business Rules:	A time period must be specified.
Implementation	
Constraints and	None
Specifications:	
Assumptions:	The administrator has internet connection.
Open Issues:	None





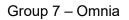
Use Case Name:	Generate Accommodation Summary	Use case type
Use Case ID:	8.3	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island Resort	System Design: o
Primary Business Actor:	Administrator	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	The use case begins when the administrator requests to generate an accommodation summary report on the system. The system should allow the administrator to generate an accommodation summary report. The system will request the time period they would like to generate the report for. The system will then retrieve all the room bookings with their associated room booking types for the booked room bookings and then display the total count that the room type was booked per month in a triple bar graph. The use case concludes when the admin downloads the report.	
Pre-condition:	The admin needs to be logger	ed onto the system.
Trigger:	The administrator requests to gene summary report on the system.	erate an accommodation
		stem Response:
Typical Course of Events:	The administrator requests to generate an	The system requests the administrator to enter the date period for the accommodation summary report.





3.	The administrator provides their date period of choice.	4.	The system validates that the date period entered by the administrator is valid.
			[ALT]
		5.	The system retrieves the number of instances related to a specific Room Type as well as the total number of instances per Room Type according to the selected date period. This is done by counting the number of instances of the RoomBooking entity linked to a specific RoomType entity instance.
			The RoomBooking entity is connected to the RoomType entity via the Room entity which is joined to it via the <i>Room_ID [FK]</i> . The Room entity is then connected to the RoomType entity via the <i>RoomType_ID [FK]</i> .
			Once the system retrieves the number of instances belonging to each RoomType_ID. The following attributes are retrieved from the RoomType entity:
			o TypeName
		6.	The system then filters the data retrieved by the date selected using the <i>EntryDate</i> and <i>ExitDate</i> attributes within the Booking entity. The system then displays the
			total instance of each room booking per room type which is filtered by month in a bar chart.
			[ALT]

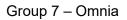






		7 The eveters prepare the
		7. The system prompts the administrator to print the
		depicted bar chart.
		depicted bai chart.
	8. The administrator selects	9. The system responds by
	to print the report.	generating the PDF for
	541 -	download of the report with
	[ALT]	the currently selected date
		aggregation.
	[ALT] Step 4: The system doe	s not verify or validate the
	information provided by the ac	dministrator, there is an error.
	Return to step 3.	
	[ALT] Chara Co The assertance is so	
	data. Use Case Terminates.	nable to retrieve the necessary
Alternate:	data. Ose Case Terminates.	
Courses:		
	[ALT] Step 8a: The administra	tor does not select to print report.
	Use Case Terminates.	досе постолно римиторони
	[ALT] Step 8b: The administra	tor selects a new start and end
	date to generate the report. R	eturn to step 3.
Conclusion:	The use case concludes when	the administrator downloads the
301101010111	report.	and the state of t
Dook on ditions	·	- d-k
Post-condition:	The system prints the accomr	nodation summary report.
Business Rules:	None.	
Implementation		
Constraints and	The administrator requires into	ernet access.
Specifications:	,	
Accumptions	None	
Assumptions:	INUTIE	
Open Issues:	None	







Use Case Name:	Generate Room Rating Report	t Use case type
Use Case ID:	8.4	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Administrator	•
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the e would like to generate a Roon	
		ort. The system will then retrieve all om type and display the different
	The use case concludes when report.	n the administrator downloads the
Pre-condition:	The administrator must	be logged into the system.
Trigger:	The administrator requests to	generate a Room Ratings report.
	Actor Action:	system Response:
Typical Course	The administrator requests to generate a Room Ratings report.	2. The system retrieves the total number of instances available for each Rating. This is done by counting the number of instances of the Review Entity.
of Events:		The RoomType entity is connected to the Review entity via the RoomType_ID (FK).
		Once the system retrieves the number of intances belonging to each Review_ID. The following

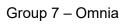


Group 7 – Omnia



		attributes of the Review Entity are retrieved:
		o Rating
		3. The system then using this data filters it by the Rating using the Rating attribute in the Review Entity.
		The system will display the data which is filtered with the average ratings per room type from when the data was first recorded.
		[ALT]
		4. The system requests the administrator to download the report.
	The administrator requests to download the report.	6. The system downloads the report on the administrator's computer.
	[ALT]	
	[ALT] Step 3: The system is u data. Use Case Terminates.	nable to retrieve the necessary
Alternate:		
Courses:	[ALT] Step 5: The administrat the report. Use Case Termina	or does not choose to download tes.
Conclusion:	The use case concludes when report.	n the administrator downloads the
Post-condition:	The system downloads	the report.
Business Rules:	None	
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has interne	t connection
Open Issues:	None	







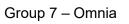
Use Case Name:	Generate customer demograp	hic Use case type
Use Case ID:	8.5	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island resort	System Design: o
Primary Business Actor:	Administrator (PBA)	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	The use case describes an even wishes to generate the custom	
	a customer demographic report displays a doughnut graph using different types of tickets bough	ng the retrieved data on the at over the period of the year.
	The use case ends when the a customer demographic report	
Pre-condition:	The administrator is logged of	onto the system.
Trigger:	The administrator wishes to ge report.	enerate a customer demographic
	Actor Action:	system Response:
Typical Course of Events:	The administrator wishes to generate a customer demographic report.	 The system responds by retrieving the necessary attributes from the following entities: DayVisit, DayVisitTicket and DayVisitType entities.
		The DayVisit entity has the following attributes:
		o DayVisit_ID [PK]
		o DayVisitStatus [FK]





		o Client_ID [FK]
		DayVisitDate_ID [FK]
		 NumberOfGuests
		o Total
		The DayVisitTicket entity has the following attributes:
		o Ticket_ID [PK]
		o DayVisit_ID[FK]
		o Client_ID [FK] (int)
		DayVisitType_ID [FK]
		 NumberOfGuests
		The DayVisitType entity has the following attributes:
		o DayVisit_ID [PK]
		o Category
		o Price
		o Date
		[ALT]
		3. The system will then do a calculation on the total amount of ticket sales done in the past year and then group them by each category using the category attribute In the DayVisitType entity.
		4. The system then prompts the administrator to print the report.
	5. The administrator clicks the "Print" button. [ALT]	6. The system finally generates the report in PDF.
Alternate: Courses:		ld not retrieve any data regarding the year. Terminate use case.

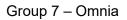






	[ALT] Step 4: The administrator does not select to print report. Terminate use case.
Conclusion:	The use case concludes when the system generates the customer demographic report and displays it in PDF format for download.
Post-condition:	The administrator can now view they report that was generated by the system.
Business Rules:	Only the administrator can generate the report.
Implementation Constraints and Specifications:	None
Assumptions:	The administrator has internet connection.
Open Issues:	None

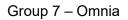






Use Case Name:	Generate Event type report.	Use case type
Use Case ID:	8.6	Business Requirements:
200 Gast 12.	0.0	0
Priority:	High	System Analysis: þ
Source:	Platinum Island resort	System Design: o
Primary Business	Administrator (PBA)	
Actor:	7.10.1	
Primary The system Actor:	None	
Other Participating		
Actors:	None	
Other Interested	None	
Stakeholders:		
Description:	wishes to generate the event	
	the event type report. The sys	e administrator wishes to generate tem displays a pie chart using the types of events that the venue was
	The use case ends when the a type report and download it.	administrator can view the event
Pre-condition:	The administrator is logged	onto the system.
Trigger:	The administrator wishes to g	enerate an event type report.
	Actor Action:	system Response:
	The administrator requests to generate an event type summary report on the system	2. The system retrieves the number of instances related to a specific event type. This is done by counting the number
		of instances a specific event was booked in the Event entity.
Typical Course of Events:		was booked in the Event
		was booked in the Event entity. The following attributes are retrieved from the Event entity to generate the report:

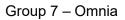






	of times the instance has occurred. 4. The administrator selects to print the report. [ALT] of times the instance has occurred. 5. The system responds by generating the PDF for download of the report with the currently selected date aggregation
Alternate: Courses:	 [ALT] Step 4: The system does not verify or validate the information provided by the administrator, there is an error. Return to step 3. [ALT] Step 6: The system is unable to retrieve the necessary data. Use Case Terminates. [ALT] Step 7a: The administrator does not select to print report. Use Case Terminates. [ALT] Step 7b: The administrator selects a new start and end date to generate the report. Return to step 3
Conclusion:	The use case concludes when the system generates the event type report and displays it in PDF format for download.
Post-condition:	The administrator can now view the data on which event type has had the most frequent selection.
Business Rules:	A time period must be specified.Only the administrator can generate the report.
Implementation Constraints and Specifications:	None
Assumptions:	The administrator has internet connection.
Open Issues:	None







Use Case Name:	Generate Booking Type Renumeration Report	Use case type	
Use Case ID:	8.7	Business Requirements: o	
Priority:	High	System Analysis: þ	
Source:	Platinum Island Resort	System Design: o	
Primary Business Actor:	Administrator	1	
Primary The system Actor:	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	None		
Description:	This use case describes the event where the administrator would like to generate a booking type renumeration report. The system should allow the administrator to generate this financial report.		
	The system retrieves all the payments related to each type of booking made on the system. The system will then populate a table that will display the overall amount paid per each subcategory of a booking type, the overall total for each booking type and the overall total of all the booking payments made on the system.		
	The use case concludes when the administrator chooses to print the report		
Pre-condition:	The admin needs to be	logged onto the system.	
Trigger:	The administrator requests to generate a booking type remuneration report on the system.		
	Actor Action:	system Response:	
Typical Course of Events:	1. The administrator requests to generate a booking type remuneration report on the system.	 The system responds by retrieving the information from the following entities: DayVisitPayment, DayVisit, DayVisitTicket, DayVisitType, RoomPayment, RoomBooking, Room, RoomType, RoomBookingRefund, 	





EventPayment, **EventRefund**, and **Event**

All the above are linked in the database as follows:

- The day visit payment records will be retrieved via DayVisitPayment entity using the DayVisitPayment_ID [PK] which is connected to the DayVisit entity via the DayVisit_ID [FK] in the DayVisitPayment entity.
- The day visit ticket records will be retrieved via DayVisitTicket entity using the DayVisitTicket_ID [PK] which is connected to the DayVisit entity via the DayVisit_ID [FK] in the DayVisitTicket entity.
- The day visit type records will be retrieved via DayVisitType entity using the DayVisitType_ID [PK] which is connected to the DayVisitTicket entity via the DayVisitType_ID [FK] in the DayVisitTicket entity.
- The room booking payment records will be retrieved via
 RoomPayment entity using the RoomPayment_ID [PK] which is connected to the RoomBooking entity via the RoomBooking_ID [FK] in the RoomPayment entity.
- The room booking refund records will be retrieved





via RoomBookingRefund entity using the Refund_ID [PK] which is connected to the RoomBooking entity via the RoomBooking entity via the RoomBookingRefund entity. The RoomBookingRefund entity. The RoomBooking entity via the Room entity which is joined to it via the Room entity is then connected to the RoomType entity via the Room entity is then connected to the RoomType_ID [FK]. The event booking payment records will be retrieved via EventPayment entity using the Event_aument_ID [PK] which is connected to the Event entity via the Event_D [FK] in the Event_D [FK] in the Event_Payment entity. The event booking refunds records will be retrieved via EventRefund entity. The event booking refunds records will be retrieved via EventRefund entity. The event booking refunds records will be retrieved via EventRefund entity using the Refund_ID [PK] which is connected to the Event entity via the Event_ID [FK] in the Event_ID [
entity is connected to the RoomType entity via the Room entity which is joined to it via the Room entity which is joined to it via the Room_ID [FK]. The Room entity is then connected to the RoomType_ID [FK]. The event booking payment records will be retrieved via EventPayment entity using the EventPayment_ID [PK] which is connected to the Event entity via the Event_ID [FK] in the EventPayment entity. The event booking refunds records will be retrieved via EventPayment entity. The event booking refunds records will be retrieved via EventRefund entity using the Refund_ID [PK] which is connected to the Event entity via the Event en				RoomBookingRefund entity using the Refund_ID [PK] which is connected to the RoomBooking entity via the RoomBooking_ID [FK] in the RoomBookingRefund
payment records will be retrieved via EventPayment entity using the EventPayment_ID [PK] which is connected to the Event entity via the Event_ID [FK] in the EventPayment entity. The event booking refunds records will be retrieved via EventRefund entity using the Refund_ID [PK] which is connected to the Event entity via the Event_ID [FK] in the Event_ID [FK] in the EventRefund entity. [ALT] The system will respond to the retrieval of record by performing the following processing: Each record within the			0	entity is connected to the RoomType entity via the Room entity which is joined to it via the Room_ID [FK]. The Room entity is then connected to the RoomType entity via the
refunds records will be retrieved via EventRefund entity using the Refund_ID [PK] which is connected to the Event entity via the Event_ID [FK] in the EventRefund entity. [ALT] 3. The system will respond to the retrieval of record by performing the following processing: • Each record within the			0	payment records will be retrieved via EventPayment entity using the EventPayment_ID [PK] which is connected to the Event entity via the Event_ID [FK] in the
retrieval of record by performing the following processing: o Each record within the		[^		refunds records will be retrieved via EventRefund entity using the Refund_ID [PK] which is connected to the Event entity via the Event_ID [FK] in the
 Each record within the 			retriev	al of record by performing
				Each record within the



Iteration 7 Logical Use Case Narratives



be linked a room booking record through the use of the RoomBooking ID [FK]. Within the each of those records, the system will find the corresponding room which was booked within the room booking by linking the Room_ID [FK] within the RoomBooking entity to the Room ID [PK] in the Room entity. Within the **Room** entity, the system will read the *TypeName* attribute of the room types using the RoomType_ID [FK]. The system will then read the Amount attribute from each record found within the **RoomPayment** entity and group each record found into its associated room type category. The system will then total each Amount record found within the RoomPayment entity according to which room type the payment is associated to.

Each record within the RoomBookingRefund entity will be linked a room booking record through the use of the RoomBooking_ID [FK] which is linked to the RoomBooking_ID [PK] found within the RoomBooking entity. The system will then total each Amount record found within the RoomBookingRefund entity.





- Each record within the **EventPayment** entity will be linked an event booking record through the use of the Event ID [FK] which is linked to the Event ID [PK] found within the **Event** entity. Within the **Event** entity, the system will read the Description attribute which details the type of event chosen by the client. The system will then read the Amount attribute from each record found within the **EventPayment** entity and group each record found into its associated event type category. The system will then total each Amount record found within the **EventPayment** entity according to which event type the payment is associated to.
- Each record within the
 EventRefund entity will
 be linked an event
 booking record through
 the use of the Event_ID
 [FK] which is linked to the
 Event_ID [PK] found
 within the Event entity.
 The system will then total
 each Amount record
 found within the
 EventRefund entity.
- Each record within the DayVisitPayment entity will be linked to a dayvisit purchase booking record through the use of the DayVisit_ID [FK] which is linked to the DayVisit_ID [PK] found



Iteration 7 Logical Use Case Narratives

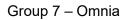


within the **DayVisit** entity. Each day visit booking is comprised of multiple tickets which is identified by the *Ticket_ID [PK]* within the **DayVisitTicket** entity. Each ticket is categorised into a type, the system will use the DayVisitType_ID [FK] attribute to determine what each type of ticket per day visit purchase is categorised into by reading the Category attribute within the **DayVisitType** entity. The system will then read the Amount attribute from each record found within the **DayVisitPayment** entity and group each record found into its associated day visit ticket type category. The system will then total each Amount record found within the DayVisitPayment entity according to which day visit ticket types the payment is associated to.

Once the system has successfully generated the subtotals for each booking type. The system will aggregate each booking type subcategory into an overall total per each booking type.

The system will then aggregate each refund table according to their booking type: Event and Room Booking where each total will be deducted from the overall sales made at the resort.

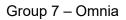






		Once the system has calculated each total per booking type, it will then aggregate each of totals to generate the overall total of every booking made on the system for the three booking types: Accommodation, Events, and Dayvisits as well as each of their subcategories. Lastly, the system displays the generated information.
		The system prompts the administrator to print the depicted financial report.
	5. The administrator selects to print the report.	6. The system responds by generating the PDF for download of the financial report.
	[ALT]	
Alternate: Courses:	Use Case Terminates.	unable to retrieve the necessary data. ator does not select to print report. Use
Conclusion:	The use case concludes wh financial report.	en the administrator downloads the
Post-condition:	The system prints the Booki	ng Type Renumeration report.
Business Rules:	None.	
Implementation Constraints and Specifications:	The administrator requires in	nternet access.
Assumptions:	None	
Open Issues:	None	







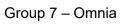
Use Case Name:	Generate Supplier Order List Report	Use case type
Use Case ID:	8.8	Business Requirements:o
Priority:	High	System Analysis:þ
Source:	Platinum Island Resort	System Design:o
Primary Business Actor:	Administrator	•
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the even would like to generate a Suppli	
	The use case begins when the generate a Supplier Order List retrieve all the Supplier Order report.	•
	The use case concludes when report.	the administrator downloads the
Pre-condition:	The administrator must b	e logged into the system.
Trigger:	The administrator requests to generate a Supplier Order List report.	
	Actor Action:	system Response:
Typical Course of Events:	The administrator requests to generate a Supplier Order List report.	 The system retrieves the total number of instances available for each Supplier Order. This is done by counting the number of instances of the Supplier Order Entity linked to a specific Supplier Entity instance. The Supplier_Order entity is
		connected to the Supplier entity via the Supplier_ID (PK,FK). The Supplier_Order entity is then connected to the





		SupplierOrderStatus entity via the OrderStatus_ID (PK,FK)
		Once the system retrieves the number of intances belonging to each Order_ID. The following attributes of the Supplier_Order entity are retrieved:
		o Date
		o Quantity
		The Supplier_ID [PK,FK] is also retrieved from the SupplierOrder entity. The Supplier name attribute is retrieved from the Supplier entity.
		The Supplier_Order entity is connected to the Item entity via the Item_ID(PK, FK). The Item_ID[PK, FK] is retrieved from the Item entity. The item name attribute is retrieved from the Item entity.
		3. The system will then display a list of all supplier orders by showing all instances of a supplier order record. [ALT]
		[ALT]
		The system prompts the administrator to download the report.
	The administrator requests to download the report.	The system downloads the report on the administrator's computer.
	[ALT]	
Alternate:	[ALT] Step 3 : The system is u data. Use Case Terminates .	nable to retrieve the necessary
Courses:	data. 000 0000 ferminates.	







	[ALT] Step 5: The administrator does not choose to download the report. Use Case Terminates .
Conclusion:	The use case concludes when the administrator downloads the report.
Post-condition:	The system downloads the report.
Business Rules:	None
Implementation Constraints and Specifications:	None
Assumptions:	The administrator has internet connection
Open Issues:	None





Use Case Name:	Generate Employee list.	Use case type
Use Case ID:	8.9	Business Requirements: o
Priority:	High	System Analysis: þ
Source:	Platinum Island resort	System Design: o
Primary Business Actor:	Administrator (PBA)	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	generate a list of all employee admin to generate a list of em employee's details and popula employees full name, ID num	ates the list by displaying the
Pre-condition:	The administrator is logged	onto the system.
Trigger:	The administrator wishes to g	enerate an employee list report.
	Actor Action: system Response:	
	The administrator wishes to generate an employee list on the system.	2. The system responds by retrieving the attributes from the Employee and EmployeeType entities with the following attributes:
Typical Course of Events:		The list is generated using the following attributes from the Employee entity:
		o ID_Num
		o Cell_Num
		 EmployeeName
		○ HireDate





		The Type_name attribute will be retrieved from the Employee_Type table by referencing the EmployeeType_ID attribute in
		the Employee table where the following attributes will be retrieved:
		Employee_Type:
		o TypeName
		[ALT]
		3. The system then displays the list that was generated and prompts the administrator to print the report.
	The administrator chooses to print the report.	The system displays the generated list in PDF Format.
	[ALT]	
Alternate:	[ALT] Step 1: An error occurre list data could not be retrieved	ed on the system and the employee d.
Courses:	[ALT] Step 3: The administrate Use Case Terminates.	or does not select to print report.
Conclusion:	The use case concludes when employee list report and displ	n the system generates the ays it in PDF format for download.
Post-condition:	The admin can now view a lis the system with their details.	t of all the employee's stored on
Business Rules:	Only the administrator contact.	an generate the report.
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has interr	net connection.
Open Issues:	None	





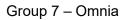
USE CASE NAME:	Available rooms list	USE CASE TYPE
USE CASE ID:	8.10	Abstract: "
PRIORITY:	High	Extension: "
SOURCE:	Platinum Island resort	
PARTICIPATING ACTORS:	Administrator (PBA)	
DESCRIPTION:	The use case describes an event in generate a list displaying a list of a time.	n which the administrator wishes to vailable rooms in a given period of
	The use case begins when the adn he wants to generate the list. The s the given dates and create a list of	•
	The use case ends when the admir rooms list from the system.	nistrator can download the available
PRE-CONDITION:	The administrator is logged o	nto the system.
	Actor Actions	System Response
	The administrator requests to generate available rooms list on the system.	2. The system retrieves the current date in which the available rooms would need to be generated.
TYPICAL COURSE OF EVENTS:		3. The system then retrieves the necessary information related to a specific room from the Room and RoomType entities where the RoomStatus is Available.
		The system retrieves the relevant data from the Room entity with the following attributes:
		o Room_ID [PK] (int)
		 RoomType_ID [FK] (int) where the RoomType_ID in the Room entity corresponds with the





		RoomType_ID in the RoomType entity.
		o RoomNumber_ID
		o Reminder_ID [FK]
		 RoomFloor
		Using the RoomType_ID in the Room entity, the system will retrieve the following attributes in the RoomType entity: TypeName Description
		·
		[ALT]
		4. The system prompts the administrator to print the depicted list.
	5. The administrator selects to print the report.[ALT]	6. The system responds by generating the PDF for download of the report.
	[ALT] Step 4: The system does not provided by the administrator, there	•
ALTERNATE COURSES:	[ALT] Step 8a: The administrator do Case Terminates.	oes not select to print report. Use
	[ALT] Step 8b: The administrator se generate the report. Return to step	
POST-CONDITION:	The administrator can now view the data on the different available rooms in a given time period.	
CONCLUSION:	The use case concludes when the rooms list and displays it in PDF fo	•

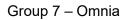






Use Case Name:	Generate Supplier list	Use case type
Use Case ID:	8.11	Business Requirements: o
Priority:	Medium	System Analysis: þ
Source:	Platinum Island resort	System Design: o
Primary Business Actor:	Administrator (PBA)	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	generate a list of all suppliers. admin to generate a list of sup supplier's details and populate suppliers full name, Email, co	opliers by retrieving all the
Pre-condition:	The administrator is logged	onto the system.
Trigger:	The administrator wishes to g	enerate an employee list report.
	Actor Action:	system Response:
	The administrator wishes to generate a supplier list.	2. The system responds by retrieving the following attributes from the Supplier entity:FullName
		o FullName o Email
Typical Course		ContactNum
of Events:		Description
		3 2333р.ш
		[ALT]
		3. The system then loads the supplier list and prompts the administrator to download the report.







	 4. The administrator chooses to download the report. 5. The system responds by downloading the supplier report on the system. 	
	[ALT]	
	[ALT] Step 1: The system could not retrieve any data regarding the suppliers. Terminate use case.	
Alternate:		
Courses:	[ALT] Step 4: The administrator does not select to print report. Use Case Terminates.	
Conclusion:	The admin can view the downloaded supplier list report on the system.	
Post-condition:	The admin can now view a list of all the supplier's stored on the system with their details.	
Business Rules:	Only the administrator can generate the report.	
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has internet connection.	
Open Issues:	None	



Iteration 7 Logical Use Case Narratives



4. Document Conclusion

The above document contained the logical narratives for the User, Client, Accommodation, Ticketing, Events, Admin, Inventory, and Reporting Subsystem. This concludes the detailed descriptions given for each logical use case.





5. Team sign-off

I, Ismaeel Rahaman, confirm that I have read the requirement changes of the Platinum Island Resort and agree to all of the information contained in this document. I confirm that the information stated above is factual and that I am aware of any and all changes made to this document between drafts. Furthermore, I confirm that I have contributed to this document equally as much as my teammates.



Ismaeel Rahaman

I, Nawailah Tarmohamed, confirm that I have read the requirement changes of the Platinum Island Resort and agree to all of the information contained in this document. I confirm that the information stated above is factual and that I am aware of any and all changes made to this document between drafts. Furthermore, I confirm that I have contributed to this document equally as much as my teammates.



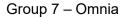
Nawailah Tarmohamed

I, Deshlan Pillay, confirm that I have read the requirement changes of the Platinum Island Resort and agree to all of the information contained in this document. I confirm that the information stated above is factual and that I am aware of any and all changes made to this document between drafts. Furthermore, I confirm that I have contributed to this document equally as much as my teammates.



Deshlan Pillay







I, Sameer Ghela, confirm that I have read the requirement changes of the Platinum Island Resort and agree to all of the information contained in this document. I confirm that the information stated above is factual and that I am aware of any and all changes made to this document between drafts. Furthermore, I confirm that I have contributed to this document equally as much as my teammates.

Similar

Sameer Ghela

I, Sashin Gounden, confirm that I have read the requirement changes of the Platinum Island Resort and agree to all of the information contained in this document. I confirm that the information stated above is factual and that I am aware of any and all changes made to this document between drafts. Furthermore, I confirm that I have contributed to this document equally as much as my teammates.

Sashin Gounden



Iteration 7 Logical Use Case Narratives



6. Client sign-off

I <u>Ya'qoob Tayob</u>, on behalf of the Platinum Island resort, acknowledge that I have received and reviewed the work and confirm that it is up to quality standards.

Signature

