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.

Table of Contents

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.

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 of Events:	Actor Action:	system Response:
	1. The user requests to login on the system.	2. The HydroTech system will prompt the user to enter their username and password.
	3. The user will provide their email address and password into the system and proceed	4. 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 *UserName* 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 *Password attribute* stored in the **User** entity related to the *UserID [PK]* record of the *UserName* 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 *Role_ID* [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]
Alternate: Courses:	[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”. [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”. [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 *UserName* 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: o
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.	
Typical Course of Events:	Actor Action:	system Response:
	1. The user requests to logout of the system.	2. The Hydrotech system will capture the user's logout request.
		3. The system will log the user out of the system. [ALT]
		4. 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:	• User must be registered on the system	
Trigger:	The user selects the forgot password option.	
Typical Course of Events:	Actor Action:	system Response:
	1. The user selects the forgot password option.	2. The HydroTech system will prompt the user to enter the following details. • Username • Phone number
	3. The user will enter their username and phone number. [ALT]	4. 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

		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
Alternate: Courses:	[ALT] **Step 3**: The user does not want to not change their password. Terminate use case. [ALT] **Step 4**: The username or the phone number was invalid, or username was not found in database. Return to step 3. [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 registered on the system	
Trigger:	The user requests to update password.	
Typical Course of Events:	Actor Action:	system Response:
	1. The user requests to update password.	2. The HydroTech system will prompt the user to enter the following details. ○ Current Password ○ New password ○ 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]
Alternate: Courses:	[ALT] Step 3: The user does not want to not change their password. Terminate use case. [ALT] Step 4: Password does not meet security requirements; password was not confirmed, or new password matches old password. Return to step 3.	
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 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 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.	
Typical Course of Events:	Actor Action:	system Response:
	1. The user requests to update email address.	**2.** The HydroTech system will prompt the user to enter their email address. **3.** [ALT]
	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]

<table>
<tr><td></td><td></td><td><b>6.</b>
The system updates the user’s email in the User table with the new email that they entered on the system.</td></tr>
<tr><td>Alternate:
Courses:</td><td colspan="2">[ALT] Step 3: The user does not want to not change their email address. End use case.
[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.</td></tr>
<tr><td>Conclusion:</td><td colspan="2">The case ends when the user’s email address has been updated successfully.</td></tr>
<tr><td>Post-condition:</td><td colspan="2">The user’s email address has been updated and they are able to log into the system with their newly created email address.</td></tr>
<tr><td>Business Rules:</td><td colspan="2">• Email needs to contain @ symbol.
• New email cannot be the same as the old email.</td></tr>
<tr><td>Implementation Constraints and Specifications:</td><td colspan="2">None</td></tr>
<tr><td>Assumptions:</td><td colspan="2">None</td></tr>
<tr><td>Open Issues:</td><td colspan="2">None</td></tr>
</table>

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.	
Typical Course of Events:	Actor Action:	system Response:
	1. The client requests to register an account on the system.	**2.** The system prompts the client to enter the following information from the **Client** entity: ○ ClientName ○ ClientSurname ○ 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: ○ Client_ID(PK) ○ User_ID(FK) ○ ClientName ○ ClientSurname ○ ClientEmail ○ ClientPhone The system stores the new client information into the **User** entity under the following attributes: ○ User_ID(PK) ○ Role_ID(FK) ○ Username ○ Password
Alternate: Courses:	[ALT] Step 4: The system provides an error message indicating that a field is empty or invalid. 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: 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 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.	
Trigger:	The client requests to update their details.	
Typical Course of Events:	Actor Action:	system Response:
	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.
	3. The client enters their updated details.	**4.** The system captures and validates the new client details entered by comparing the

		entered information with the information in the Client table and checks 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 updated client information into the Client entity under the following attributes: ○ ClientName ○ ClientSurname ○ ClientEmail ○ ClientPhone
Alternate: Courses:	[ALT] Step 4: The system notifies the client that a field is empty or invalid. Return to step 3.	
Conclusion:	The use case concludes when a client's personal information has been successfully updated.	
Post-condition:	• An clients account has been updated in the Client entity.	
Business Rules:	None	
Implementation Constraints and Specifications:	None	
Assumptions:	The client has internet connection	
Open Issues:	None	

Use Case Name:	Add Review	Use case type
Use Case ID:	2.5	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 add a review on the system. The system should allow the client to provide a review rating and a review description on the resort. The system will respond by saving the new review record 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 “View Client Homepage” screen.	
Trigger:	The client requests to add a new review to the system.	
Typical Course of Events:	Actor Action:	system Response:
	1. The client requests to add a new review to the system.	**2.** The system prompts the client to enter the new review details: ○ Number Rating ○ 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

		Review entity which has the following attributes: ○ Review_ID [PK] ○ Client_ID [FK] ○ RoomType_ID [FK] ○ Rating ○ Description [ALT]
Alternate: Courses:	[ALT] Step 4: Not all fields were filled in correctly, the system 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	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:	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.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:
	1. The client requests to view room booking type options available on the system by providing a check-in and check-out date.	**2.** The system retrieves the necessary room details from the following entities: The **Room** entity which contains the following attributes: ○ Room_ID [PK] ○ RoomType_ID [FK] ○ RoomStatus_ID [FK] ○ RoomNumber

		○ RoomFloor The **RoomType** entity which contains the following attributes: ○ RoomType_ID [PK] ○ TypeName ○ Description ○ 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:	[ALT] Step 3: The duration set by the client for their stay returns no room options. The client will be prompted to select an alternate date. [ALT] Step 4: The client decides to return to the homepage. Terminate use case.	
CONCLUSION:	The use case concludes when the client selects the room of their choice.	

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.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:
	1. The client requests to complete an accommodation booking on the system.	**2.** The system retrieves the following attributes from the selected **Room** entity: ○ Room_ID [PK] ○ RoomType_ID [FK] ○ RoomStatus_ID [FK] ○ RoomNumber ○ RoomFloor The system will then retrieve the

		following attributes from the **RoomType** entity using the selected *RoomType_ID [FK]* record from the **Room** entity: ○ TypeName ○ Description ○ 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 ○ 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 ○ 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.
	5. 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]
- ReferenceNum
- BookingDate
- NumberOfGuests
- EntryDate
- ExitDate

RoomPayment entity with the following attributes:

- RoomPayment_ID [PK]
- 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]
- 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]

ALTERNATE: COURSES:	[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. [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 occurrence in which the client wishes to cancel their room booking. The use case begins when the client calls the administrator of platinum island, and provides the administrator with the reference number, the administrator locates the room booking and cancels it as indicated. The use case concludes when the system changes 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 or equal to 48 hours before Check-In. An email is also sent to client informing him about the cancellation.	
PRE-CONDITION:	• The administrator needs to be logged onto the system. • The client’s accommodation booking status has to be “Booked”	
TRIGGER:	The client calls the administrator of Platinum Island to requests to cancel their room booking.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:
	1. The client calls the administrator of Platinum Island to requests to cancellation of their room booking.	**2.** The administrator requests the client to provide their reference number.
	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: ○ RoomBooking_ID [PK] ○ RoomBookingStatus_ID [FK] ○ Client_ID [FK] ○ Room_ID [FK] ○ BookingDate ○ NumberOfGuests ○ ReferenceCode ○ EntryDate ○ ExitDate The system will update the status of the client’s booking by reading the *Name* attribute from the **RoomBookingStatus** entity depending on when the cancellation of the booking occurs: – If the clients’ room cancellation occurs within 48 hours before their room check-in, which can be validated using the EntryDate attribute within the Room Booking entity and the current date of cancellation, the Room booking will have a status of “Cancelled” – If the clients’ Room cancellation occurs more than 48 hours before their Riim check-in, which can be validated using the EntryDate attribute within the Room Bookings entity and the current date of

		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.
ALTERNATE: COURSES:	[ALT] Step 4: The administrator could not find the reference code provided by the client. Return to Step 2. 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 the client before their booking check-in reminding them of their booking reservation. The use case concludes when the system sends an email to the client reminding them of the time of their check-in.		
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.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:	
	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: 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:	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 be logged onto the system.	
TRIGGER:	The administrator requests to add a new type of room.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:
	1. The administrator requests to add a new type of room.	**2.** The HydroTech system prompts the administrator to enter the following information from the **RoomType** 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
	3. The administrator enters the new type of room's details into the system.	**4.** The system captures and validates the new type of room's details and verifies that all specified fields have been inputted [ALT]
		5. The system autogenerates the **RoomType_ID** which is set to

		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: ○ TypeName ○ TypeDescription ○ RoomCapacity **RoomTypePrice** entity with the following attributes: ○ Price ○ Date
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:	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:	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 administrator requests to update the type of room.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:	
	1. The administrator requests to update the type of room.	**2.** The HydroTech system prompts the administrator to select a room type	
	3. The administrator selects the room type they wish to edit.	**4.** The system displays the room's type information that has been retrieved from the **RoomType** entity with the following attributes: o TypeName o TypeDescription o RoomCapacity The system will then retrieve the following attributes from the **RoomTypePrice** entity using the selected *RoomType_ID [FK]* record from the **RoomTypePrice** entity: o Price	

	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	

<table>
<tr><td>USE CASE NAME:</td><td>Delete Room Type</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>3.8.</td><td>Business Requirements: o</td></tr>
<tr><td>PRIORITY:</td><td>Low</td><td>System Analysis: þ</td></tr>
<tr><td>SOURCE:</td><td>Platinum Island Resort</td><td>System Design: o</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="2">Administrator</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="2">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="2">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="2">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="2">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="2">• The administrator needs to be logged onto the system.
• The room type must exist on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="2">The administrator requests to delete a room type.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td>ACTOR ACTION:</td><td>SYSTEM RESPONSE:</td></tr>
<tr><td><b>1.</b> The administrator requests to delete a room type.</td><td><b>2.</b> The HydroTech system prompts the administrator to select a room type that they would like to delete.</td></tr>
<tr><td><b>3.</b> The administrator selects the room type they wish to delete.</td><td><b>4.</b> The system prompts the administrator to confirm that they want to delete the selected room type.</td></tr>
<tr><td><b>5.</b> The administrator confirms that they want to delete the selected room type.
[ALT]</td><td><b>6.</b> The system removes the room type from the <b>RoomType</b> entity which has the following attributes:
– RoomType_ID(pk)
– TypeName
– TypeDescription
– RoomCapacity</td></tr>
<tr><td>ALTERNATE: COURSES:</td><td colspan="2">[ALT] Step 5: The administrator chooses to cancel the process. The Use Case is terminated.</td></tr>
<tr><td>CONCLUSION:</td><td colspan="2">The use case ends when a room type is successfully deleted from the room type entity.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="2">• A room type was removed from the RoomType entity</td></tr>
</table>

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: 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 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 logged onto the system.	
Trigger:	The client requests to make a Day Visit booking on the system.	
Typical Course of Events:	Actor Action:	system Response:
	1. The client requests to complete a Day Visit 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 ○ ClientPhone ○ ClientEmail 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 the required number of adults, children, and/or pensioners. [ALT]
	3. The client will provide the number of adults, children and/or pensioners.	**4.** The system validates that the maximum capacity has not been reached by referring to the capacity attribute in the **DayVisitDate** entity. [ALT] The system then prompts the user to view the day visit summary.
	5. The client selects the terms and conditions checkbox. [ALT]	**6.** The system allows the client to view the summary.
	7. The client selects to view the day visit summary. • • [ALT]	**8.** The system displays a summary of the booking made by fetching the most recent VAT record according to the Date attribute from the **VAT** entity with the following attributes: ○ 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] ○ Price ○ 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: ○ Ticket_ID [PK] ○ 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]
Alternate: Courses:	[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. [ALT] Step 4: The capacity has been reached and the client cannot book for that day. Terminate use case. [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 not wish to view the day visit summary. [ALT] Step 12: 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 **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: 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 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.	
Typical Course of Events:	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.
	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) ○ LowerAge ○ 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:	[ALT] Step 3a: The administrator selects the adult's category to update. [ALT] Step 3b: The administrator selects the pensioner category to update. [ALT] Step 5: The administrator chooses to cancel the changes	
Conclusion:	The use case concludes when the ticket price has been successfully updated and the system notifies the administrator that the ticket price has been successfully updated.	
Post-condition:	• The ticket price has been updated in the **DayVisitTicketPrice** entity.	
Business Rules:	Only the administrator can update a ticket price's details on the system	
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.		
Typical Course of Events:	Actor Action:	system Response:	
	1. 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. • [ALT]	**6.** The system allows the user to select the view summary.
	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 *Date* attribute from the **VAT** entity with the following attributes: ○ VATAmount ○ 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: ○ VenuePrice_ID [PK] ○ Venue_ID [FK] attribute of the **Venue** entity where the Venue_ID in the **VenuePrice**

		entity corresponds to the Venue_ID in the Venue entity. ○ VenuePriceAmount_ID ○ 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: ○ Event_ID [PK] ○ EventStatus_ID [FK] ○ Client_ID [FK] ○ Venue_ID [FK] ○ Date

		○ GuestList ○ EventTotal ○ ReferenceCode ○ Description • The **EventPayment** entity with the following attributes: ○ EventPayment_ID [PK] ○ Client_ID [FK] ○ Event_ID [FK] ○ PaymentType_ID [FK] ○ 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] ○ 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. [ALT]
Alternate: Courses:	[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. [ALT] Step 4: The date selected has been booked out. Return to Step 3.	

	[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: o
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 in which the client wishes to cancel their event booking. The use case begins when the client calls the administrator of platinum island, and provides the administrator with the reference number, the administrator locates the event and cancels it as indicated. The use case concludes when the system changes the status of the client’s event booking to “Cancelled” if cancelled less than 48 hours before Check-In and “Refund Pending” if the cancellation occurred more than 48 hours before Check-In. An email is also sent to client informing him about the cancellation.	
Pre-condition:	• The administrator needs to be logged onto the system. • The client already has an event booked at Platinum Island.	
Trigger:	The client calls the administrator of Platinum Island to requests to cancel their booking.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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

		record and selects the appropriate record. [ALT]
		5. The system will confirm whether the client would like to cancel their event booking reservation.
	6. The client confirms the cancellation of their event booking. [ALT]	**7.** The administrator confirms the 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, 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 “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 *ClientEmail* retrieved, the system will draft an email which will be sent to the client detailing their accommodation booking cancellation.
Alternate: Courses:	[ALT] Step 4: The administrator could not find the reference code provided by the client. Return to Step 2. [ALT] Step 6: The client refuses to confirm the cancellation of their event booking. Terminate Use Case.	
Post-condition:	• Dependant of the when the event booking cancellation occurred, the system will change the status of the client’s event booking to “Cancelled” 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 Reminder	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:	The use case begins when time prompts the system to send a reminder notification to the client notifying them of their event booking. The system should send an email to the client before their event booking check-in reminding them of their booking reservation. The use case concludes when the system sends an email to the client reminding them of the time 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.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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: Courses:	None	
Conclusion:	The use case concludes when the system generates an email which is sent to the client detailing their event reservation.	
Post-condition:	The client receives an email reminding them of their event 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:	Update Event Price	Use case type
Use Case ID:	5.6	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:	This use case describes an occurrence in which the administrator wishes to change the price of booking an event in the **VenuePrice** entity. The use case begins when the administrator wants to edit the venue price and enters modifications. The use case concludes when the venue price has been successfully updated.	
Pre-condition:	• The administrator needs to be logged onto the system.	
Trigger:	The administrator requests to update a venue price.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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** entity where the *Venue_ID* in the **VenuePrice** entity corresponds to the *Venue_ID* in the **Venue** entity.

		○ VenuePriceAmount ○ 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: ○ Venue_ID [PK] ○ 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. [ALT]	**6.** The modified venue's price is first validated before it is saved by the system on the **VenuePrice** entity under the following attributes: ○ VenuePrice_ID [PK] ○ Venue_ID [FK] ○ VenuePriceAmount ○ Date • [ALT]
		7. The system notifies the user that the venues price has been successfully updated.
Alternate: Courses:	[ALT] Step 5: The administrator chooses to not update the selected field and cancels the update of the venue price. 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. The use case concludes after the information is captured.	
Pre-condition:	• The administrator needs to be logged onto the system. • The employee must not already exist on the system.	
Trigger:	The administrator requests to add a new employee on the system.	
	Actor Action:	system Response:
Typical Course of Events:	**1.** 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: ○ ID_Number ○ Cell_Num ○ EmployeeName The system prompts the administrator to enter the employee details.

	3. 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: ○ Employee_ID(PK) ○ EmplloyeeType_ID(FK) ○ User_ID(FK) ○ ID_Num ○ Cell_Num ○ EmployeeName ○ HireDate
Alternate: Courses:	[ALT] Step 4a: The system provides an error message indicating that a field is empty. Return to Step 3. [ALT] Step 4b: The system provides an error message indicating that a field is invalid. Return to Step 3.	
Conclusion:	The use case concludes when a new employee has been successfully added onto the system.	
Post-condition:	• A new employee has been added to the **Employee** table.	
Business Rules:	• Only the administrator can register a new employee on the system	
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has internet 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:	This use case describes an occurrence in which the administrator wishes to change an employee's information in the **Employee** entity. The use case begins when the administrator wants to edit an employee's information and enters modifications, the employee's information has been successfully updated.	
Pre-condition:	• The administrator needs to be logged onto the system. • The employee must exist on the system.	
Trigger:	The administrator requests to update an employee’s details.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator requests to update an employee’s details.	**2.** The hydrotech system prompts the admin to select the employee they want to update.
	3. The administrator selects the employee whose details they want to update	**4.** 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. ○ ID_Number ○ Cell_Num ○ EmployeeName ○ HireDate And prompts the administrator to edit the field/s they wish to.
	5. 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: ○ Cell_Num ○ EmployeeName
Alternate: Courses:	None	
Conclusion:	The use case concludes when an employee has been successfully updated.	
Post-condition:	• The employee’s information has been updated in the employee entity.	
Business Rules:	Only the administrator can update an employee’s details on the system	
Implementation Constraints and Specifications:	None	
Assumptions:	The administrator has internet 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 incident in which an administrator wishes to remove an employee from the system. The administrator will look for the employee and proceed to delete them after receiving confirmation. The use case ends when the employee is deleted from the Employee object.	
Pre-condition:	• The administrator needs to be logged onto the system. • The employee must exist already on the system.	
Trigger:	The administrator requests to delete an employee.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator requests to delete an employee	**2.** The HydroTech system prompts the Administrator to select the employee they want to delete.
	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.
	5. The administrator confirms that they want to delete the selected employee. [ALT]	**6.** The system removes the employee from the Employee entity with the following attributes: o Employee_ID [PK] o EmployeeType_ID [FK] o User_ID [FK]

		o ID_Number o Cell_Num o EmployeeName o HireDate .
Alternate: Courses:	[ALT] Step 5- The administrator chooses to cancel the process. The Use 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 Specifications:	None	
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:	The use case describes the event where the administrator wants to add a new type of employee. The system will prompt the administrator to enter the TypeName and TypeDescription for the new type of employee. The system validates and saves the new employee type in the EmployeeType entity. The use case ends when the system displays a message to indicate that the employee type was successfully added.		
Pre-condition:	• The administrator needs to be logged onto the system.		
Trigger:	The administrator requests to add a new type of employee.		
Typical Course of Events:	Actor Action:	system Response:	
	1. The administrator requests to add a new type of employee.	**2.** The HydroTech system prompts the administrator to enter the following information from the **EmployeeType** entity: ○ 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 when the system successfully adds an employee type.	
Post-condition:	• A new type of employee was added to the EmployeeType entity.	
Business Rules:	• Only the administrator can add an employee type on the system.	
Implementation Constraints and Specifications:	None	
Assumptions:	• The administrator has internet 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:	The use case describes the event where the administrator wants to update an existing employee type. The system prompts the administrator to enter the attributes of the employee type. The system will save the updated details of employee type in the EmployeeType entity.	
Pre-condition:	• The administrator needs to be logged onto the system. • The employee type must exist on the system.	
Trigger:	The administrator requests to update the type of employee.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator requests to update the type of employee.	**2.** The HydroTech system prompts the administrator to select an employee type
	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: Courses:	None	
Conclusion:	The use case concludes when system successfully updates the employee type.	
Post-condition:	• An employee type was updated in the EmployeeType entity	
Business Rules:	• Only the administrator can update an employee type on the system.	
Implementation Constraints and Specifications:	None	
Assumptions:	• The administrator has internet 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 incident in which an administrator wishes to remove an employee type from the system. The administrator will look for the employee type and proceed to delete them after receiving confirmation. The use case ends when the employee type is deleted from the Employee type object.	
Pre-condition:	• The administrator needs to be logged onto the system. • The employee type must exist on the system.	
Trigger:	The administrator requests to delete an employee type.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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.
	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. [ALT]	**6.** The system removes the employee type from the **EmployeeType** entity which has the following attributes: – EmployeeType_ID [PK] – TypeName

<table>
<tr><td></td><td></td><td>– TypeDescription</td></tr>
<tr><td>Alternate:
Courses:</td><td colspan="2">[ALT] Step 5: The administrator chooses to cancel the process. The Use Case is terminated.</td></tr>
<tr><td>Conclusion:</td><td colspan="2">The use case ends when an employee type is successfully deleted from the Employee type entity.</td></tr>
<tr><td>Post-condition:</td><td colspan="2">• An employee type was removed from the EmployeeType entity</td></tr>
<tr><td>Business Rules:</td><td colspan="2">• Only the administrator can delete an employee type on the system.</td></tr>
<tr><td>Implementation Constraints and Specifications:</td><td colspan="2">None</td></tr>
<tr><td>Assumptions:</td><td colspan="2">• The administrator has internet connection.</td></tr>
<tr><td>Open Issues:</td><td colspan="2">None</td></tr>
</table>

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 client initiates the check-in process into their room booking. The system should allow the administrator to check in a client's room booking. Whereby the client will be asked to confirm their booking before their room booking record is retrieved from the RoomBooking entity. Once the administrator has found and confirmed the client's room booking, the status of their room booking will be changed to "Checked-in" and the admin will hand over the room keys to the client. 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"	
Pre-condition:	• The administrator needs to be logged onto the system. • The client's accommodation must have been made prior to check-in. • The screen displayed the "View Accommodations" screen.	
Trigger:	The client requests to check-in into their room booking reservation.	
Typical Course of Events:	Actor Action:	system Response:
	1. The client requests to check-in into their room booking reservation	**2.** 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 check-in. [ALT]
	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.
Alternate: Courses:	[ALT] Step 4: The system could not retrieve the Room Booking and Room Booking status information from the **RoomBooking** entity. Terminate use case. [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:	This use case describes the event where the Client request to check-out after using the room. The administrator responds by facilitating the check-out process. The administrator will search for the client specific event booking on the system. The system will retrieve the details of the event entity. Thereafter the administrator will confirm details from the client. The use case concludes when the administrator checks the client out of the system. The status in the **RoomBookingStatus** entity will update to "Checked Out"	
Pre-condition:	• The administrator is logged on to system. • The client has already checked-in the system.	
Trigger:	The client arrives at the resort and wishes to get checked-out by the administrator.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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.
	3. The client provides the booking reference number.	**4.** The system retrieves the information for the event booking from the **RoomBooking** entity that contains the following information: ○ Roombooking_ID [PK]

		○ 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]
		6. The administrator checks the client of the system.
		7. The system changes the Name attribute in the **RoomBookingStatus** to “Checked Out”
Alternate: Courses:	[ALT] Step 4: The system could not retrieve the Booking and Booking status information from the Booking entity. Use case ends. [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:	This use case describes the event where the administrator creates a new VAT 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 VAT record has successfully been created on the system.	
Pre-condition:	• The administrator needs to be on the *VAT View.* • The administrator needs to be logged onto the system.	
Trigger:	The administrator requests to create a new VAT record on the system.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator requests to create a new VAT record on the system.	**2.** The system prompts the administrator to enter the required information concerning the VAT record from the **VAT Table**: o VATAmountDate o Date
	3. The administrator enters 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: ○ VAT_ID (PK) ○ VATAmount ○ Date [ALT]
		6. The system will then display the new record within the VAT table
Alternate: Courses:	[ALT] Step 4: The administrator denies the information entered is correct. Return to Step 3. [ALT] Step 5: The system will throw an error due to incorrect information entered by the administrator. Return to Step 3	
Conclusion:	The use case concludes when the record is successfully added to the **VAT Table**.	
Post-condition:	• The new VAT record is successfully added to the **VAT Table** in system’s database.	
Business Rules:	1. 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 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:	This use case describes the event where the administrator updates a VAT record on the system. The use case begins when the administrator chooses to update a VAT record on the system. The system will retrieve the information of the selected record, where the system will prompt the user to enter the updated required information. The administrator enters the information required. The system verifies the information and then stores it. The use case concludes when the VAT record has been successfully updated on the system.	
Pre-condition:	• The administrator needs to be on the *VAT View.* • The administrator needs to be logged onto the system.	
Trigger:	The administrator requests to update a VAT record on the system.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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: ○ VAT_ID (PK) ○ VATAmount ○ Date [ALT]
		6. The system will then display the updated record within the VAT table.
Alternate: Courses:	[ALT] Step 4: The administrator denies the information entered is correct. Return to Step 3. [ALT] Step 5: The system will throw an error due to incorrect information entered by the administrator. Return to Step 3	
Conclusion:	The use case concludes when the VAT record has been successfully updated on the system.	

Post-condition:	- The updated VAT record is successfully added to the **VAT Table** in system's database.
Business Rules:	1. 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:	• This use case describes an occurrence in which the administrator wishes to change a reminders information in the **Reminder** entity. The use case begins when the administrator wants to edit a reminders information and enters modifications, and it concludes when the administrator is alerted that the reminders information has been successfully updated.	
Pre-condition:	• The administrator needs to be logged onto the system. • The reminder must exist on the system.	
Trigger:	• The administrator requests to update a reminders details.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator requests to update a reminders details.	**2.** The system prompts the user to select the reminder they want to update.
	3. The administrator selects the reminder they want to update	**4.** 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.

	5. The administrator edits the details of the reminder.	•
	•	**6.** The modified reminder information is saved by the system in the **Reminder** entity under the following attributes: ○ Reminder_ID (PK) ○ Reminder_Type ○ Reminder_Time •
	•	**7.** 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	

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:	This use case describes the event where the Client initiates the check-in process when he/she arrives at the reception. The Security Guard responds by facilitating the check-in process. The Security Guard will scan the client's QR code, and allow the client into resort. The Name attribute in the DayVisitStatus entity would change to "Checked-in".	
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.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:
	1. The Client arrives at Platinum Island Resort and requests to check in. [ALT]	**2.** The security guard responds by requesting the client to provide their QR code.
	3. The Client provides his/her 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".
ALTERNATE COURSES:	[ALT] Step 1: The client does not arrive that day, the Name Attribute of the DayVisitStatus entity updates to "Forfeited".	
	[ALT] Step 4: The system is not successful. Use case ends.	
CONCLUSION:	Client enters platinum island resort.	
POST-CONDITION:	• The Name attribute of the DayVisitStatus entity is updated to "Checked-In".	
BUSINESS RULES:	• Tickets are only valid for the day that is booked.	

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:	The use case describes an event when the client wishes to check in for their event. The use case starts when the client arrives at the resort, he then gives the administrator the booking reference number that was given to him. Once the administrator locates the booking that was made, he checks the user in. The use case ends when the status is updated in the **EventStatus** entity to “checked in”.	
Pre-condition:	• The administrator is logged on to system. • The client made an event booking on the system.	
Trigger:	The client arrives at the resort and wishes to get checked-in by the administrator.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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.
	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: ○ Event_ID [PK] ○ EventStatus_ID [FK] ○ Client_ID [FK] ○ Venue_ID [FK]

		○ Reminder_ID [FK] ○ GuestList ○ ReferenceCode ○ 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 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”
Alternate: Courses:	[ALT] Step 4: The system could not retrieve the Event and Event 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.	
Implementation Constraints and Specifications:	None	
Assumptions:	• The administrator has internet connection.	
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:	This use case describes the event where the Client request to check-out after using the venue. The administrator responds by facilitating the check-out process. The administrator will search for the client specific event booking on the system. The system will retrieve the details of the event entity. Thereafter the administrator will confirm details from the client. The use case concludes when the administrator checks the client out of the system. The status in the **EventStatus** entity will update to "Checked Out"	
Pre-condition:	• The administrator is logged on to system. • The client has already checked-in the system.	
Trigger:	The client arrives at the resort and wishes to get checked-out by the administrator.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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.
	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: ○ Event_ID [PK] ○ EventStatus_ID [FK]

		○ Client_ID [FK] ○ Venue_ID [FK] ○ Reminder_ID [FK] ○ GuestList ○ ReferenceCode ○ 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: Courses:	[ALT] Step 4: The system could not retrieve the Booking and Booking status information from the Booking entity. Use case ends. [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 the client initiates the refund process of their room booking reservation. The system should allow the administrator to refund a client's room booking. Whereby the client will be asked to confirm their booking before their room booking record is retrieved from the **RoomBooking** entity. Once the administrator has found the record, the administrator will request confirmation on the client's room booking refund and will then complete the client's refund payment off of the system. The system will then change the status of their room booking to "Refunded". The use case concludes 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.		
Pre-condition:	• The administrator needs to be logged onto the system. • The client's accommodation booking must have been cancelled first. • The client's room booking status must have a status of "Refund Pending"		
Trigger:	The client requests to refund their room booking reservation.		
Typical Course of Events:	Actor Action:	system Response:	
	1. 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. [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 **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: ○ RoomBooking_ID [PK] ○ Client_ID [FK] ○ RoomBookingStatus_ID [FK] ○ ReferenceNum ○ BookingDate

		○ NumberOfGuests ○ EntryDate ○ ExitDate The system will also retrieve the relevant RoomPayment record by using the *RoomBooking_ID [FK]* within the **RoomPayment** entity to retrieve the *Amount* attribute. The system retrieves the “Refunded” status by reading the *Name* 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: ○ Refund_ID [PK] ○ RoomBooking_ID [FK] ○ Amount [ALT]
Alternate: Courses:	[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 room booking was not valid for a room booking refund as it was cancelled less than 48 hours before the *EntryDate* attribute within the RoomBooking entity. Terminate Use Case. [ALT] Step 5: The client refuses to continue with their refund room 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 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 client initiates the refund process of 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 administrator completes the refund booking EFT, the transaction is logged on the system and the system updates the event booking status of the client's booking.	
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 status must have a status of "Refund Pending"	
Trigger:	The client requests to refund their event booking reservation.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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]

		○ ReferenceCode ○ Date ○ GuestList ○ EventTotal ○ Description The system will also retrieve the relevant EventPayment record by using the *Event_ID [FK]* within the **EventPayment** entity to retrieve the *Amount* attribute. The system retrieves the “Refunded” status by reading the *Name* 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: ○ Refund_ID [PK] ○ Event_ID [FK] ○ Amount [ALT]

Alternate: Courses:	[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 *Date* attribute within the Event entity. Terminate Use Case. [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

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 the administrator requests to add a new item to the system. The system should prompt the administrator to enter the items details after which the administrator will provide the item details. This use case concludes when the item has been successfully added to the **Item** entity.	
Pre-condition:	• The administrator is logged in has the necessary permissions to create a new item.	
Trigger:	The administrator requests to add an item.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator requests to add an item.	**2.** The system prompts the administrator to enter the required information regarding the Item from the **Item** entity: ○ Name ○ Description ○ QtyOnHand ○ ItemType_ID.Name [FK] which is connected to the **ItemType** entity by matching the

		ItemType_ID [FK] to the *ItemType_ID [PK]* in the **ItemType** entity.
	3. The administrator enters the required details for the new item.	**4.** The system will validate the information entered by the administrator and will prompt the administrator to confirm the item details once validated. [ALT]
	5. The administrator will confirm the creation of the new item record. [ALT]	**6.** The system will respond by creating a new record within the **Item** entity which has the following attributes: ○ Item_ID [PK] ○ ItemType_ID [FK] ○ Name ○ Description ○ QtyOnHand [ALT]
Alternate: Courses:	[ALT] Step 3: The administrator declines confirmation of adding the new item. Terminate Use Case. [ALT] Step 4: The information entered by the administrator returned a validation error. Return to Step 3. [ALT] Step 6: The system is unable to successfully create the Item record and will return an error. Terminate Use Case.	
Conclusion:	The use case concludes when the new item has been added to the system	
Post-condition:	A new item record has been 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 the administrator requests to update an existing item on the system. The system should prompt the administrator to provide the updated item record details after retrieving the appropriate record details. After which, the administrator will respond by providing the relevant updated item record details and will confirm the creation of the item record. This use case concludes when the item has been successfully updated.	
Pre-condition:	• The administrator is logged into the system	
Trigger:	The administrator requests to update an existing item record on the system.	
	Actor Action:	system Response:
Typical Course of Events:	**1.** 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: ○ 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: ○ Item_ID [PK] ○ ItemType_ID [FK] ○ Name ○ Description [ALT]
		7. The system notifies the user that the item has been successfully updated.
Alternate: Courses:	[ALT] Step 5: The administrator chooses to not update the selected field and cancels the update of the item. Terminate use case. [ALT] Step 6: The information entered by the administrator returns a validation error. Return to Step 3.	
Conclusion:	Use case concludes when item details have been successfully updated	
Post-condition:	The item record has been successfully 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 event where the administrator would like to delete a specific item record. The use case begins when the system will request confirmation on the item deletion before the administrator confirms the deletion of the record. The use case concludes when the item record has been deleted from the system.	
Pre-condition:	• The administrator is logged in onto the system. • The item exists on the system before it can be deleted.	
Trigger:	The administrator requests to delete an Item record from the system.	
	Actor Action:	system Response:
Typical Course of Events:	**1.** The administrator requests to delete an Item record from the system.	**2.** The system will retrieve the record from the **Item** entity using the Item_ID [PK] and retrieve the attributes with that associated record: ○ Item_ID [PK] ○ ItemType_ID [FK] ○ 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
		5. The system will display a notification informing the administrator of the successful deletion of the item record
Alternate: Courses:	[ALT] Step 3: The administrator will decide to cancel the deletion of the record. The use case terminates.	
Conclusion:	Use case concludes when item has been deleted from the **Item** entity.	
Post-condition:	The item record has been successfully removed from the **Item** entity.	
Business Rules:	None	
Implementation Constraints and Specifications:	None	
Assumptions:	None	
Open Issues:	None.	

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:	This use case begins when the administrator navigates through the side-navigation panel and clicks on the Item Type option. The system displays the Add Item Type option. The user clicks this option and is then prompted toad an item type. This use case concludes when the item type has been successfully added to the database.	
Pre-condition:	The Administrator is logged into the inventory system and has the necessary permissions to create a new item.	
Trigger:	The administrator clicks the Add item type button on the item type page.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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.
	3. The administrator enters the required details for the new item.	
	4. The administrator saves the new item to the **ItemType** entity with the following attributes: o ItemType_ID (PK) o ItemCategory_ID (FK) o Name o Description	**5.** The system saves the new record to the **ItemType** table. [ALT]

Alternate: Courses:	[ALT] Step 5: Not all fields were filled in correctly, the system 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 Specifications:	None
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:	This use case begins when the administrator navigates through the side-navigation panel and clicks on the Item type option. The system displays the update item type option. The user clicks this option and is then prompted to search for the item type to update. This use case concludes when the item type has been successfully updated.	
Pre-condition:	Admin is logged in	
Trigger:	The user navigates to the item details page and selects the "Update" button.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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: ○ ItemType_ID (PK) ○ ItemCategory_ID (FK) ○ Name ○ Description
	3. The user modifies the fields they want to update and submits the form.	**4.** The system validates the input and updates the item type with the new details in the **ItemType** entity.
		5. The system displays a success message indicating

		that the item has been updated.
Alternate: Courses:	[ALT] Step 4: If the user inputs invalid data in the form, the 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:	The Delete Item use case allows a user to delete an existing item type from the system. This use case is typically initiated by a user who needs to remove an item type from the system permanently. Use case concludes when item type has been removed from the database.	
Pre-condition:	Admin is logged in	
Trigger:	The admin requests to delete an item type from the system.	
Typical Course of Events:	Actor Action:	system Response:
	1. The admin requests to delete an item type from the system.	**2.** The system requests confirmation from the admin to delete the item type.
	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 item types from the system.	

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 *Items View.* • The administrator needs to be logged onto the system.	
Trigger:	The administrator requests to create a new Item Category record on the system.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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: ○ ItemCategory_ID (PK) ○ Name ○ Description [ALT]
		6. The system will then display the new record within the Item Category Table
Alternate: Courses:	[ALT] Step 4: The administrator denies the information entered is correct. Return to Step 3. [ALT] Step 5: The system will throw an error due to incorrect information entered by the administrator. Return to Step 3	
Conclusion:	The use case concludes when the record is successfully added to the **Item Category Table**.	
Post-condition:	• The new VAT record is successfully added to the **Item Category Table** in system’s database.	
Business Rules:	1. Only authorised users of the system will be permitted create an Item Category 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:	This use case describes the event where the administrator updates an Item Category record on the system. The use case begins when the administrator chooses to update an Item Category record on the system. The system will retrieve the information of the selected record, where the system will prompt the user to enter the updated 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 been successfully updated on 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.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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.
		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**: ○ ItemCategory_ID (PK)

		○ 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: Courses:	[ALT] Step 5: The administrator denies the information entered is correct. Return to Step 4. [ALT] Step 6: The system will throw an error due to incorrect information entered by the administrator. Return to Step 4.	
Conclusion:	The use case concludes when the Item Category record has been successfully updated on the system.	

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	
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 delete an Item Category record from the system.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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]
		3. The system will retrieve the record from the **ItemCategory**

		Table with the following attributes: ○ ItemCategory_ID (PK) ○ 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. [ALT]	**5.** The system will respond deleting the appropriate item category record from the **Item Category Table** with the following attributes: ○ 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:	[ALT] Step 3: The system will determine that there are item types associated with the item category. The system will then inform the administrator of the item types that are associated with the item category. The use case will terminate. [ALT] Step 5: The administrator will decide to cancel the deletion of the record. The use case terminates.	
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 of the system will be permitted to update an Item Category 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:	In this use case, the admin creates a new supplier that he may use to view the supplier's current details. The use case begins when the administrator creates a new supplier and establishes the relevant supplier information. Once the system stores the new supplier in the database, the use case is complete.	
Pre-condition:	• The admin must have access to internet. • The admin must be logged onto the system.	
Trigger:	The admin wishes to add a new supplier to the system.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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: ○ FullName ○ Email ○ ContactNum ○ Description They system then prompts the user to enter the supplier information.

<table>
<tr><td></td><td><b>3.</b> The admin enters the information from the previous step.</td><td><b>4.</b> 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]</td></tr>
<tr><td></td><td></td><td><b>5.</b> The system stores the new supplier information into the corresponding entities under the following attributes:
– Supplier_ID(PK)
– FullName
– Email
– ContactNum
– Description</td></tr>
<tr><td>Alternate:
Courses:</td><td colspan="2">[ALT] step 4a - The system provides an error message indicating that a field is empty. Go back to step 3.
[ALT]step 4b - The system provides an error message indicating that a field is invalid. Go back to step 3.</td></tr>
<tr><td>Conclusion:</td><td colspan="2">The use case concludes, and the new supplier is added onto the platinum island database.</td></tr>
<tr><td>Post-condition:</td><td colspan="2">The admins can now find the details of the new supplier on the system.</td></tr>
<tr><td>Business Rules:</td><td colspan="2">• Only admins can add new suppliers to the system.</td></tr>
<tr><td>Implementation Constraints and Specifications:</td><td colspan="2">None</td></tr>
<tr><td>Assumptions:</td><td colspan="2">The admin has internet access.</td></tr>
<tr><td>Open Issues:</td><td colspan="2">None</td></tr>
</table>

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 occurrence in which the administrator wishes to change a supplier's information in the **Supplier** entity. The use case begins when the administrator wants to edit a supplier's information and enters modifications, the supplier's information has been successfully updated.	
Pre-condition:	• The admin must have access to internet. • The admin must be logged onto the system.	
Trigger:	The administrator requests to update a supplier's details.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator requests to update a supplier's details.	**2.** The system prompts the admin to select a supplier that they want to update.
	3. The admin selects a supplier that they would like to update	**4.** 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 the supplier details is now updated and stored onto the database.	
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 access.	
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:	This use case covers an incident in which an administrator wishes to remove a supplier from the system. The administrator will look for the supplier and proceed to delete them after receiving confirmation. The use case ends when the supplier is deleted from the supplier object.	
Pre-condition:	• The admin must have access to internet. • The admin must be logged onto the system.	
Trigger:	The admin wants to delete an existing supplier.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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
	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: Courses:	[ALT]step 5: The admin declines the deletion, and the use case 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 internet access.	
Trigger:	The administrator requests to place a new order for stock	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator requests to place a new order for stock.	**2.** The HydroTech system prompts the administrator to enter the following information from the **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. ○ 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 decides not to place an order. Terminate use case.	
Conclusion:	The use case concludes when the new order is sent to the owner.	
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 internet 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 cancel an order of stock that was previously placed.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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.
	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 in 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.		
Typical Course of Events:	Actor Action:	system Response:	
	1. After the stock order has been placed and the supplier has dispatched the items, the administrator receives the shipment at the designated receiving area.		
	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	

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	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator wishes to generate a peak season day visit report	**2.** 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 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] ○ Client_ID [FK] ○ DayVisitDate_ID [FK] ○ RefCode ○ 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.
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 5: 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 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 Specifications:	None
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 logged onto the system.	
Trigger:	The administrator requests to generate an accommodation summary report on the system.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator requests to generate an accommodation summary report on the system.	**2.** 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 *Room_ID [FK].* The **Room** entity is then connected to the **RoomType** entity via the *RoomType_ID [FK]*. Once the system retrieves the number of instances belonging to each *RoomType_ID*. The following attributes are retrieved from the **RoomType** entity: ○ TypeName
		6. The system then filters the data retrieved by the date selected using the *EntryDate* and *ExitDate* 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 system prompts the administrator to print the depicted bar chart.
	8. The administrator selects to print the report. [ALT]	**9.** 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 8a: The administrator does not select to print report. Use Case Terminates. [ALT] Step 8b: The administrator selects a new start and end date to generate the report. Return to step 3.	
Conclusion:	The use case concludes when the administrator downloads the report.	
Post-condition:	The system prints the accommodation summary report.	
Business Rules:	None.	
Implementation Constraints and Specifications:	The administrator requires internet access.	
Assumptions:	None	
Open Issues:	None	

Use Case Name:	Generate Room Rating Report	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 event where the administrator would like to generate a Room Rating report. The use case begins when the administrator requests to generate a Room Rating Report. The system will then retrieve all the Room ratings for each room type and display the different proportions of ratings each type of room receives. The use case concludes when the administrator downloads the report.	
Pre-condition:	• The administrator must be logged into the system.	
Trigger:	The administrator requests to generate a Room Ratings report.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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. 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

		attributes of the Review Entity are retrieved: ○ 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.
	5. The administrator requests to download the report. [ALT]	**6.** The system downloads the report on the administrator's computer.
Alternate: Courses:	[ALT] Step 3: The system is unable to retrieve the necessary data. Use Case Terminates. [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 customer demographic report	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 event in which the administrator wishes to generate the customer demographic report. The use case begins when the administrator wishes to generate a customer demographic report on the system. The system displays a doughnut graph using the retrieved data on the different types of tickets bought over the period of the year. The use case ends when the administrator can view the customer demographic report and download it.	
Pre-condition:	• The administrator is logged onto the system.	
Trigger:	The administrator wishes to generate a customer demographic report.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator wishes to generate a customer demographic report.	**2.** The system responds by retrieving the necessary attributes from the following entities: **DayVisit, DayVisitTicket** and **DayVisitType** entities. The **DayVisit** entity has the following attributes: ○ DayVisit_ID [PK] ○ DayVisitStatus [FK]

		○ Client_ID [FK] ○ DayVisitDate_ID [FK] ○ NumberOfGuests ○ Total The **DayVisitTicket** entity has the following attributes: ○ Ticket_ID [PK] ○ DayVisit_ID[FK] ○ Client_ID [FK] (int) ○ DayVisitType_ID [FK] ○ NumberOfGuests The **DayVisitType** entity has the following attributes: ○ DayVisit_ID [PK] ○ Category ○ Price ○ 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:	[ALT] Step 1: The system could not retrieve any data regarding the number of ticket sales for 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: 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 event in which the administrator wishes to generate the event type report. The use case begins when the administrator wishes to generate the event type report. The system displays a pie chart using the retrieved data on the different types of events that the venue was used for. The use case ends when the administrator can view the event type report and download it.	
Pre-condition:	• The administrator is logged onto the system.	
Trigger:	The administrator wishes to generate an event type report.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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. The following attributes are retrieved from the **Event** entity to generate the report: ○ Description
		3. The system then filters the data according to the dates that were selected by the administrator. The system then displays the total instance that each event type was selected in a doughnut chart and table with the number

		of times the instance has occurred.
	4. The administrator selects to print the report. [ALT]	**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		
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.		
Typical Course of Events:	Actor Action:	system Response:	
	1. The administrator requests to generate a booking type remuneration report on the system.	**2.** 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_ID [FK]* in the **RoomBookingRefund** entity. ○ The **RoomBooking** 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 *RoomType_ID [FK].* ○ The event booking payment records will be retrieved via **EventPayment** entity using the Event*Payment_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 **EventRefund** entity. [ALT]
		3. The system will respond to the retrieval of record by performing the following processing: ○ Each record within the **RoomPayment** entity will

		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 day-visit purchase booking record through the use of the *DayVisit_ID [FK]* which is linked to the *DayVisit_ID [PK]* found

		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 Day-visits as well as each of their subcategories. Lastly, the system displays the generated information.
		4. The system prompts the administrator to print the depicted financial report.
	5. The administrator selects to print the report. [ALT]	**6.** The system responds by generating the PDF for download of the financial report.
Alternate: Courses:	[ALT] Step 2: The system is unable to retrieve the necessary data. Use Case Terminates. [ALT] Step 5: The administrator does not select to print report. Use Case Terminates.	
Conclusion:	The use case concludes when the administrator downloads the financial report.	
Post-condition:	The system prints the Booking Type Renumeration report.	
Business Rules:	None.	
Implementation Constraints and Specifications:	The administrator requires internet 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 event where the administrator would like to generate a Supplier Order List report. The use case begins when the administrator requests to generate a Supplier Order List report. The system will then retrieve all the Supplier Order records and display them in a list report. The use case concludes when the administrator downloads the report.	
Pre-condition:	• The administrator must be logged into the system.	
Trigger:	The administrator requests to generate a Supplier Order List report.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator requests to generate a Supplier Order List report.	2. 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]
		4. The system prompts the administrator to download the report.
	5. The administrator requests to download the report. [ALT]	6. The system downloads the report on the administrator’s computer.
Alternate: Courses:	[ALT] **Step 3**: The system is unable to retrieve the necessary data. **Use Case Terminates**.	

	[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:	This use case describes the event where the admin would like to generate a list of all employees. The system should allow the admin to generate a list of employees by retrieving all the employee's details and populates the list by displaying the employees full name, ID number, Employee type, contact number and Hire Date. The use case concludes when the admin downloads the report	
Pre-condition:	• The administrator is logged onto the system.	
Trigger:	The administrator wishes to generate an employee list report.	
Typical Course of Events:	Actor Action:	system Response:
	1. 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: The list is generated using the following attributes from the **Employee** entity: ○ ID_Num ○ 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:** ○ TypeName [ALT]
		3. The system then displays the list that was generated and prompts the administrator to print the report.
	4. The administrator chooses to print the report. [ALT]	**5.** The system displays the generated list in PDF Format.
Alternate: Courses:	[ALT] Step 1: An error occurred on the system and the employee list data could not be retrieved. [ALT] Step 3: The administrator does not select to print report. Use Case Terminates.	
Conclusion:	The use case concludes when the system generates the employee list report and displays it in PDF format for download.	
Post-condition:	The admin can now view a list of all the employee'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	

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 which the administrator wishes to generate a list displaying a list of available rooms in a given period of time. The use case begins when the administrator selects the dates in which he wants to generate the list. The system will then retrieve data using the given dates and create a list of available rooms. The use case ends when the administrator can download the available rooms list from the system.	
PRE-CONDITION:	• The administrator is logged onto the system.	
TYPICAL COURSE OF EVENTS:	Actor Actions	System Response
	1. 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.
		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: ○ 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. ○ RoomNumber_ID ○ 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.
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 8a: The administrator does not select to print report. Use Case Terminates. [ALT] Step 8b: The administrator selects a new start and end date to generate the report. Return to step 3	
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 system generates the available rooms list and displays it in PDF format for download.	

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:	This use case describes the event where the admin would like to generate a list of all suppliers. The system should allow the admin to generate a list of suppliers by retrieving all the supplier's details and populates the list by displaying the suppliers full name, Email, contact number and the description. The use case concludes when the admin downloads the report	
Pre-condition:	• The administrator is logged onto the system.	
Trigger:	The administrator wishes to generate an employee list report.	
Typical Course of Events:	Actor Action:	system Response:
	1. The administrator wishes to generate a supplier list.	**2.** The system responds by retrieving the following attributes from the **Supplier** entity: ○ FullName ○ Email ○ ContactNum ○ Description [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. [ALT]	**5.** The system responds by downloading the supplier report on the system.
Alternate: Courses:	[ALT] Step 1: The system could not retrieve any data regarding the suppliers. Terminate use case. [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	

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.

Rahaman

……………………..

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.

S Ghela

……………………..

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

6. Client sign-off

I Ya'qoob Tayob, 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