Adwait Purao - 2021300101

Viraj Bhalerao -2022301002

TE COMPS - B

SE LAB

Experiment 2: UML Diagrams

Use Case Diagram:

Table 1: List of Actors

Actor	**Description**
Admin	The admin is responsible for managing the entire ferry ticketing system. • They have access to the system's backend and can make changes as needed. • Admins can create and update routes and ticket prices. • They handle user accounts, including staff and customers, and can resolve any issues or disputes.
Customer	Customers are end-users who want to purchase ferry tickets for their travel. • They visit the system's website or app to search for ferry routes, check schedules, and buy tickets. • Customers can view ticket availability, select seats or cabins, and make payments securely. • They receive confirmation and e-tickets after successful bookings.
Staff	Staff members are employees who work for the ferry company and use the ticketing system. • They include ferry operators, ticketing agents, and onboard crew. • Staff use the system to check-in passengers, validate tickets, and manage boarding procedures. • They may also assist customers with inquiries and provide on-the-ground support.

Table 2: List of Use-Cases

#	Use Case	Description
UC1	Search Ticket	Allows the users to search for a ticket.
UC2	View Current Bookings	Allows the user to view the current booking.
UC3	Manage Revenue	Allows the admin to view and allocate the finances to various parts of the system.
UC4	Book Ticket	Helps the users to book tickets
UC5	Make Payment	Allows users to make payments to the allocated ferries.

UC6	Cancel Ticket	Allows users to cancel the allocated ticket.
UC7	Schedule ferries	Allows the admin to schedule the timing of the ferries.
UC8	View Report	Allows the admin to view a detailed report of all aspects of the system.
UC9	Manage Staff	Allows the admin to add , update and delete staff.
UC10	Set fares	Allows the Admin to change the rates for ferries.

Use Case Scenarios:

Table 3: UC1 – Search Ticket

Use Case:	UC1. Make reservation	
Goal:	Search Ticket	
Actors:	- User	
Pre-condition:	The desired ticket should exist	
Post-condition:	No changes should be made, just the user should be given to perform the read operation.	
Mainline Scenario:	Actor Actions	System Actions
	1. Enter details of the ticket	
	3. The user can now view the ticket.	2. Check whether such a ticket exists
Alternate Flows:	2. The system will display "Incorrect Details" and the user will have to enter his details again. 5. No rooms available for the route given by the user. 7. System crashes while searching for the ticket, the user is redirected to the page where they need to fill the details of the ticket again.	

Table 4: UC2 – View Current Bookings

Use Case:	UC2. View Current Bookings	
Goal:	To view the tickets booked by the user	
Actors:	- User	
Pre-condition:	The User should have logged into the system	
Post-condition:	No Changes	
Mainline Scenario:	Actor Actions	System Actions
	1. Log in to the system.	2. Authenticate the user's credentials.
	3. Navigate to the "View Current Bookings" section.	4. Retrieve the user's booked tickets.
	6. User reviews the booked tickets.	5. Display the list of booked tickets to the user.
Alternate Flows:	2a. If the user's credentials are invalid: - System displays an error message. - User corrects the credentials and retries. 4a. If there are no booked tickets for the user: - System displays a message indicating no bookings found. This use case allows a logged-in user to view their booked tickets, and the system handles authentication and retrieval of the booking information.	

Table 5: UC3 – Manage Revenue

Use Case:	UC3. Manage Revenue	
Goal:	Manage Revenue	
Actors:	• Admin	
Pre-condition:	Log in to the system	
Post-condition:	Changes in Database	
Mainline Scenario:	Actor Actions	System Actions
	1. Log in to the system as an admin.	6. Authenticate the admin's credentials.
	2. Access the "Manage Revenue" section.	7. Provide access to the "Manage Revenue" section.
	3. View and analyze revenue data.	8. Retrieve and present revenue data for analysis.
	4. Make adjustments or changes as needed.	9. Allow the admin to make adjustments.
	5. Save the changes made.	10. Update the database with any changes made by the admin.
		11. Notify the admin of successful changes.
Alternate Flows:	• If the admin's credentials are invalid: • System displays an error message. • Admin corrects the credentials and retries. • If there is an issue with updating the revenue data in the database: • System displays an error message. • Admin can choose to retry or cancel the operation. These tables outline the mainline and alternate flows for the "Manage Revenue" use case, including actions taken by both the actor (admin) and the system.	

Table 6: UC4 – Book Ticket

Use Case:	UC4. Book Ticket	
Goal:	Book Ticket	
Actors:	• User, Ferry Ticketing System	
Pre-condition:	Log in to the system	
Post-condition:	Payments should be collected, and database of the ship must be updated	
Mainline Scenario:	Actor Actions	System Actions
	1. Log in to the system.	4. Authenticate the user's credentials.
	2. Provide booking details (e.g., route, date, number of tickets).	5. Validate the provided booking details.

<table>
<tr><td></td><td>
<table>
<tr><td>3. Select payment method and confirm the booking.</td><td>6. Calculate the total cost of the tickets.</td></tr>
<tr><td></td><td>7. Collect payment from the user.</td></tr>
<tr><td></td><td>8. Update the ship's database with the booking information.</td></tr>
<tr><td></td><td>9. Notify the user of the successful booking.</td></tr>
</table>
</td></tr>
<tr><td>Alternate Flows:</td><td>
<ul>
<li>If the user's credentials are invalid:
<ul>
<li>System displays an error message.</li>
<li>User corrects the credentials and retries.</li>
</ul></li>
<li>If the provided booking details are incomplete or invalid:
<ul>
<li>System displays an error message.</li>
<li>User corrects the information and retries.</li>
</ul></li>
<li>If there is an issue with payment processing:
<ul>
<li>System displays an error message.</li>
<li>User can choose to retry the payment or use a different payment method.</li>
</ul></li>
</ul>
These tables outline the mainline and alternate flows for the "Book Ticket" use case, including actions taken by both the actor (user) and the system.
</td></tr>
</table>

Table 7: UC5 – Make Payment

<table>
<tr><td>Use Case:</td><td>UC5. Make Payment</td></tr>
<tr><td>Goal:</td><td>Completing successful payment after the purchase of ticket</td></tr>
<tr><td>Actors:</td><td>• User, Ferry Ticketing System</td></tr>
<tr><td>Pre-condition:</td><td>Log in to the system</td></tr>
<tr><td>Post-condition:</td><td>Changes in Database of Revenue</td></tr>
<tr><td>Mainline Scenario:</td><td>
<table>
<tr><th>Actor Actions</th><th>System Actions</th></tr>
<tr><td>1. Log in to the system.</td><td>6. Authenticate the user's credentials.</td></tr>
<tr><td>2. Select the ticket(s) for payment.</td><td>7. Retrieve the selected ticket(s) for payment.</td></tr>
<tr><td>3. Choose a payment method (e.g., credit card, debit card).</td><td>8. Present the total payment amount to the user.</td></tr>
<tr><td>4. Provide payment details.</td><td>9. Collect payment information securely.</td></tr>
<tr><td>5. Confirm the payment.</td><td>10. Process the payment transaction.</td></tr>
<tr><td></td><td>11. Update the Database of Revenue with the payment information.</td></tr>
<tr><td></td><td>12. Notify the user of the successful payment.</td></tr>
</table>
</td></tr>
<tr><td>Alternate Flows:</td><td>• If the user's credentials are invalid:
  ◦ System displays an error message.
  ◦ User corrects the credentials and retries.
• If the selected ticket(s) are no longer available:
  ◦ System displays an error message.
  ◦ User can choose different ticket(s) or cancel the payment.
• If there is an issue with payment processing:
  ◦ System displays an error message.
  ◦ User can choose to retry the payment or use a different payment method.
These tables outline the mainline and alternate flows for the "Make Payment" use case, including actions taken by both the actor (user) and the system.</td></tr>
</table>

Table 8: UC6 – Cancel Ticket

<table>
<tr><td>Use Case:</td><td>UC6. Cancel Ticket</td></tr>
<tr><td>Goal:</td><td>Cancel the book ticket</td></tr>
<tr><td>Actors:</td><td>• Staff
• Admin
• Customer</td></tr>
<tr><td>Pre-condition:</td><td>Log in to the system</td></tr>
<tr><td>Post-condition:</td><td>No Changes</td></tr>
<tr><td>Mainline Scenario:</td><td>
<table>
<tr><th>Actor Actions</th><th>System Actions</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td>1. Clicks on the
“Cancel Ticket”
button.</td><td>2. Displays the customer cancel ticket details.</td></tr>
<tr><td>4. Checks for the ticket details andcancel them.</td><td>3. Validates the details of the customer.</td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td>Alternate Flows:</td><td>• Actor Selects "Cancel Ticket" Option
• Actor (User) selects the "Cancel Ticket" option in the application or website.
• System Verifies Ticket Information
• System prompts the user to enter or verify ticket information, such as booking/reference number or username.
• Actor Provides Ticket Information
• Actor provides the necessary ticket information.
• System Validates Ticket
• System validates the provided information to ensure it matches an existing reservation.</td></tr>
</table>

Table 9: UC7 – Schedule ferries

<table>
<tr><td>Use Case:</td><td>UC7. Schedule ferries</td></tr>
<tr><td>Goal:</td><td>Schedules ferries for customer</td></tr>
<tr><td>Actors:</td><td>• Admin</td></tr>
<tr><td>Pre-condition:</td><td>Log in to the system</td></tr>
<tr><td>Post-condition:</td><td>Changes in Database</td></tr>
<tr><td>Mainline Scenario:</td><td>
<table>
<tr><th>Actor Actions</th><th>System Actions</th></tr>
<tr><td>1. Clicks on the “Schedules ferries”button.</td><td>2. Page appears to fill in the ferriesdetails.</td></tr>
<tr><td>3. Enter the new ferries details.</td><td>4. Validates the ferries details</td></tr>
<tr><td></td><td>5. Creates a new schedule for ferries andupdates the database.</td></tr>
</table>
</td></tr>
<tr><td>Alternate Flows:</td><td>• If no ferry is available for the selected route or time, an alternate flow is triggered:
• System informs the user that no ferry is available.
• Actor may choose to:
• Select an alternate date or time.
• Choose a different departure or destination.
• Cancel the reservation process.</td></tr>
</table>

Table 10: UC8 – View Report

<table>
<tr><td>Use Case:</td><td>UC8. View Report</td></tr>
<tr><td>Goal:</td><td>Allows the admin to view a detailed report of all aspects of the system.</td></tr>
<tr><td>Actors:</td><td>• Admin</td></tr>
<tr><td>Pre-condition:</td><td>Log in to the system</td></tr>
<tr><td>Post-condition:</td><td>Changes in Database</td></tr>
<tr><td>Mainline Scenario:</td><td>
<table>
<tr><th>Actor Actions</th><th>System Actions</th></tr>
<tr><td>1. Clicks on the “view report” button.</td><td>2. Page appears of report.</td></tr>
<tr><td>3. Select view report.</td><td>4. Validates the report details</td></tr>
<tr><td></td><td>5. Asks for confirmation before view the report</td></tr>
<tr><td></td><td>6,generate the full report</td></tr>
</table>
</td></tr>
<tr><td>Alternate Flows:</td><td>If there is an issue with generating the report, an alternate flow is triggered:
• System displays an error message, indicating the reason for the failure.
• Actor may choose to:
• Retry generating the report.
• Select a different report.</td></tr>
</table>

	• Contact support for assistance.

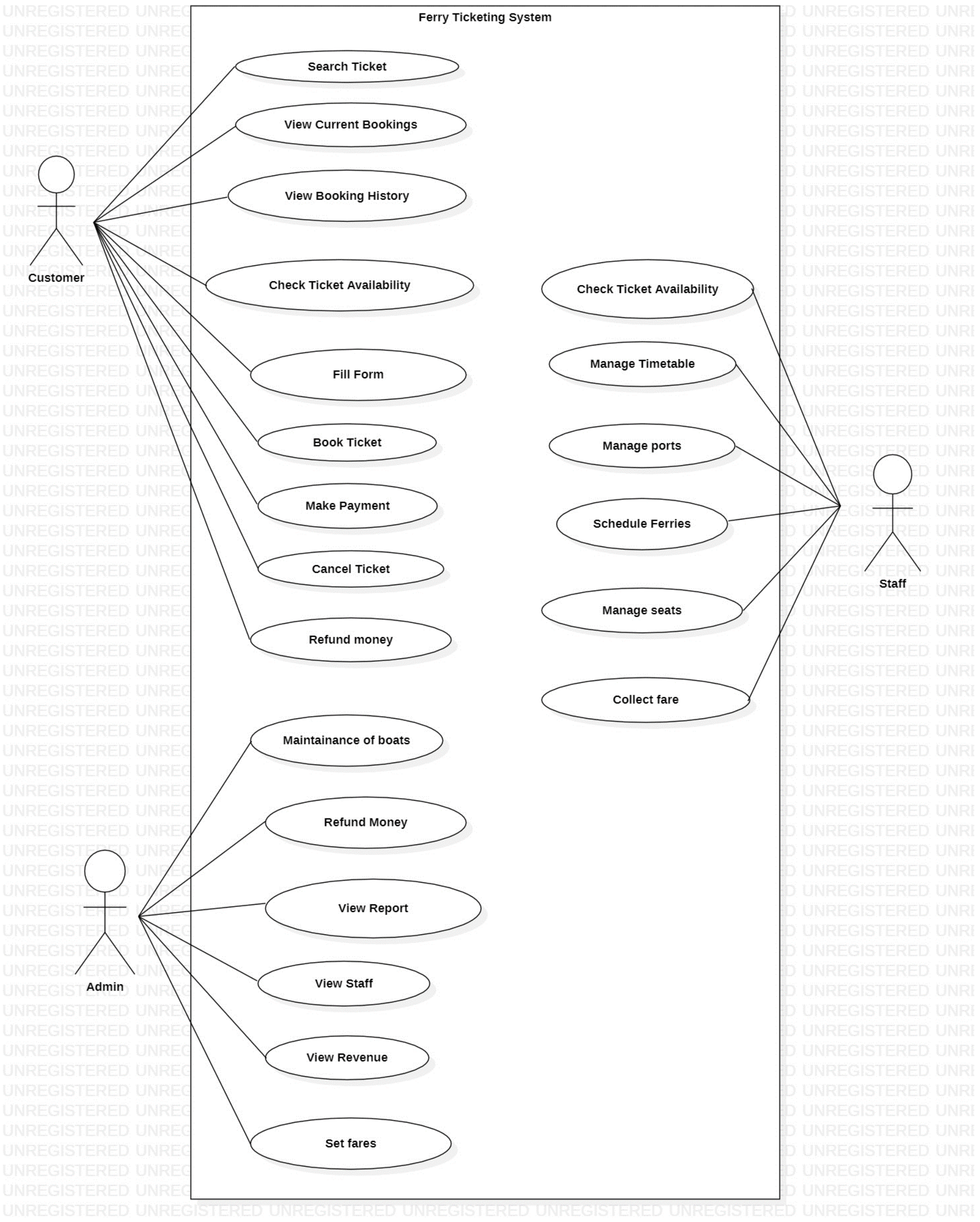
Table 11: UC9 – Manage Staff

<table>
<tr><td>Use Case:</td><td colspan="2">UC9. Manage Staff</td></tr>
<tr><td>Goal:</td><td colspan="2">Allows the admin to add , update and delete staff.</td></tr>
<tr><td>Actors:</td><td colspan="2">• Admin</td></tr>
<tr><td>Pre-condition:</td><td colspan="2">Log in to the system</td></tr>
<tr><td>Post-condition:</td><td colspan="2">Changes in Database</td></tr>
<tr><td>Mainline Scenario:</td><td>Actor Actions</td><td>System Actions</td></tr>
<tr><td></td><td>1. Clicks on the “update staff ”detail button.</td><td>2. Page appears to select the staff detailed</td></tr>
<tr><td></td><td>3. add staff detailed</td><td>4. Displays the room details</td></tr>
<tr><td></td><td>5. Edits the staff details.</td><td>6. Validates the details.</td></tr>
<tr><td></td><td></td><td>7. Updates staff details and the database.</td></tr>
<tr><td>Alternate Flows:</td><td colspan="2">• If there is an issue executing the selected staff management action, an alternate flow is triggered:
• System displays an error message, indicating the reason for the failure.
• Actor may choose to:
• Retry the action with corrected information.
• Choose a different staff management action.
• Contact support or a higher-level administrator for assistance.</td></tr>
</table>

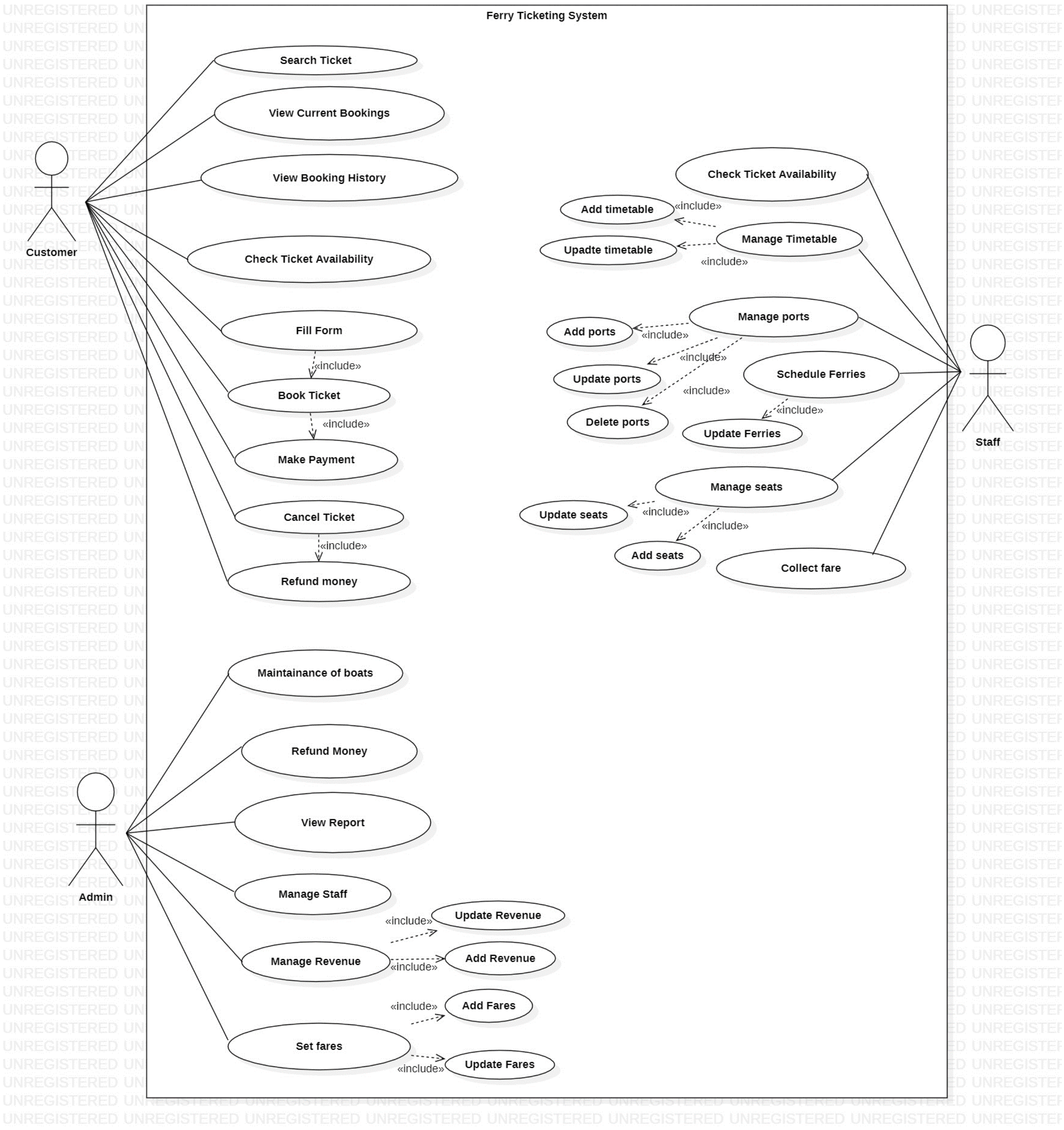
Table 12: UC10 – Set fares

<table>
<tr><td>Use Case:</td><td colspan="2">UC10. Set fares</td></tr>
<tr><td>Goal:</td><td colspan="2">Change the Rate for a fares</td></tr>
<tr><td>Actors:</td><td colspan="2">• Admin</td></tr>
<tr><td>Pre-condition:</td><td colspan="2">Log in to the system</td></tr>
<tr><td>Post-condition:</td><td colspan="2">Changes in Database</td></tr>
<tr><td>Mainline Scenario:</td><td>Actor Actions</td><td>System Actions</td></tr>
<tr><td></td><td>1. Clicks on the “set fare rate” button.</td><td>2. Page appears to select the fares.</td></tr>
<tr><td></td><td>3. Select the fares.</td><td>4. Display the fare details.</td></tr>
<tr><td></td><td>5. Edits the fare rates.</td><td>6. Validates the fare details.</td></tr>
<tr><td></td><td></td><td>7. Updates fare rates and the database.</td></tr>
<tr><td>Alternate Flows:</td><td colspan="2">If there is an issue executing the selected fare management action, an alternate flow is triggered:
• System displays an error message, indicating the reason for the failure.
• Actor may choose to:
• Retry the action with corrected information.
• Choose a different fare management action.
• Contact support or a higher-level administrator for assistance.</td></tr>
</table>

Level 1:


Ferry Ticketing System
Customer
Search Ticket
View Current Bookings
View Booking History
Check Ticket Availability
Fill Form
Book Ticket
Make Payment
Cancel Ticket
Refund money
Staff
Check Ticket Availability
Manage Timetable
Manage ports
Schedule Ferries
Manage seats
Collect fare
Admin
Maintainance of boats
Refund Money
View Report
View Staff
View Revenue
Set fares

Level 2:


Ferry Ticketing System
Customer
Search Ticket
View Current Bookings
View Booking History
Check Ticket Availability
Fill Form
«include»
Book Ticket
«include»
Make Payment
Cancel Ticket
«include»
Refund money
Check Ticket Availability
Add timetable
«include»
Manage Timetable
Upadte timetable
«include»
Manage ports
Add ports
«include»
Update ports
«include»
Delete ports
«include»
Schedule Ferries
«include»
Update Ferries
Staff
Manage seats
Update seats
«include»
«include»
Add seats
Collect fare
Admin
Maintainance of boats
Refund Money
View Report
Manage Staff
«include»
Update Revenue
Manage Revenue
«include»
Add Revenue
«include»
Add Fares
Set fares
«include»
Update Fares

Conclusion:

From the above experiment, we have learned the following:

How to implement a Use case diagram based on Use case scenarios.