UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

SOFTWARE REQUIREMENTS SPECIFICATION

Mobile Car Wash Booking & Management System

Group no : CS-01

Academic year 2019/2020

Table of Contents

Table of Figures **3**
Group Details **5**
1. Introduction **6**
1.1 Purpose 6
1.2 Document Conventions 6
1.3 Intended audience 6
1.4 Domain description 6
1.5 Current system and its limitations 7
1.6 Project Scope 8
1.6.1 In-Scope 8
1.6.2 Out-Scope 8
1.7 Project Goal 8
1.8 Constraints and assumptions 9
1.8.1 Project constraints and assumptions 9
1.8.2 System constraints and assumptions 9
2. Overall Description **10**
2.1 Product Perspective 10
2.2 Stakeholders 10
2.3 Operating Environment 10
3. Feasibility Study **11**
3.1 Technical Feasibility 11
3.2 Economic Feasibility 11
3.3 Legal and Ethical Feasibility 11
3.4 Operational Feasibility 12
3.5 Scheduling Feasibility 12
4. External Interface Requirements **13**
4.1 User Interfaces 13
4.1.1 User UI 13
4.1.2 Customer UI 14
4.1.3 Manager UI 15
4.1.4 System Admin UI 16

4.1.5 Service Team Leader UI 17
4.2 Hardware Interfaces 17
4.3 Software Interfaces 18
5. System Features 19
5.1 Use Case Diagram 19
5.2 Use Case Narratives 20
5.3 Functional Requirements 40
5.3.1 User 40
5.3.2 Customer 40
5.3.3 Manager 40
5.3.4 System Admin 41
5.3.5 Service Team Leader 41
6. Quality Attributes 42
6.1 Quality Attribute Requirements 42
6.1.1 General Scenarios 42
6.1.2 Concrete Scenarios 45
6.2 Tactics to achieve Quality Attribute requirements 50
6.2.1 Tactics for Availability 50
6.2.2 Tactics for Security 50
6.2.3 Tactics for Usability 51
6.2.4 Tactics for Modifiability 51
6.2.5 Tactics for Performance 51
7. System Architecture 52
7.1 Derived Component Diagram 52
7.2 Components and their responsibilities 53
8. System Design 56
8.1 Class Diagram 56
8.2 EER Diagram 57
8.3 EER Mapping 58
8.4 Sequence Diagrams 59
8.5 Activity Diagrams 81
8.6 State Transition Diagrams 96

References **100**

Table of Figures

Figure 1: Current System 7
Figure 2: User UI 13
Figure 3: Customer UI 14
Figure 4: Manager UI 15
Figure 5: System Admin UI 16
Figure 6: Service Team Leader UI 17
Figure 7: Use Case Diagram 19
Figure 8: System Architecture - Derived Component Diagram 52
Figure 9: Class Diagram 56
Figure 10: EER Diagram 57
Figure 11: Sequence Diagram - Login 59
Figure 12: Sequence Diagram - Password Reset 60
Figure 13: Sequence Diagram - View Reviews 61
Figure 14: Sequence Diagram - Registration 62
Figure 15: Sequence Diagram - Make Reservation 63
Figure 16: Sequence Diagram - View and Modify Upcoming Reservations 64
Figure 17: Sequence Diagram - View and Rate Past Reservations 65
Figure 18: Sequence Diagram - Give Reviews 66
Figure 19: Sequence Diagram - Edit Account Details 66
Figure 20: Sequence Diagram - Delete Account 67
Figure 21: Sequence Diagram - View and Assign STL for Reservations 68
Figure 22: Sequence Diagram - View Completed Reservations 69
Figure 23: Sequence Diagram - View Dashboard 70
Figure 24: Sequence Diagram - Add/Update/Delete Employee 71
Figure 25: Sequence Diagram - Add/Update/Delete Equipment 72
Figure 26: Sequence Diagram - Add/Update/Delete Service Details 73
Figure 27: Sequence Diagram - Add/Update/Delete Service Team Leader 74
Figure 28: Sequence Diagram - View/Search Employee 75
Figure 29: Sequence Diagram - View/Search Equipment 76
Figure 30: Sequence Diagram - View/Search Service Details 77
Figure 31: Sequence Diagram - View/Search Service Team Leader 78
Figure 32: Sequence Diagram - View Service Dashboard 79
Figure 33: Sequence Diagram - View Calendar 80
Figure 34: Activity Diagram - Login 81
Figure 35: Activity Diagram - View Reviews 81
Figure 36: Activity Diagram - Register 82
Figure 37: Activity Diagram - Make Reservation 83
Figure 38: Activity Diagram - View Upcoming Reservations 84
Figure 39: Activity Diagram - View Past Reservations 85

Figure 40: Activity Diagram - Give Reviews 86
Figure 41: Activity Diagram - Edit Account Details 86
Figure 42: Activity Diagram - Delete Account 87
Figure 43: Activity Diagram - View Reservations 88
Figure 44: Activity Diagram - View Completed Reservations 89
Figure 45: Activity Diagram - View Dashboard 90
Figure 46: Activity Diagram - View Service Dashboard 90
Figure 47: Activity Diagram - Add/Update/Delete Employee 91
Figure 48: Activity Diagram - Add/Update/Delete Service Team Leader 91
Figure 49: Activity Diagram - View/search Employee 92
Figure 50: Activity Diagram - View/search Service Team Leader 92
Figure 51: Activity Diagram - Add/Update/Delete Service Details 93
Figure 52: Activity Diagram - Add/Update/Delete Equipment 93
Figure 53: Activity Diagram - View/Search Service Details 94
Figure 54: Activity Diagram - View/Search Equipment 94
Figure 55: Activity Diagram - View Calendar 95
Figure 56: State Transition Diagram - User Account 96
Figure 57: State Transition Diagram - Reservation 97
Figure 58: State Transition Diagram - Service Team 98
Figure 59: State Transition Diagram - Equipment 99

Group Details

Name	Registration Number	Index Number	Email Address
ILMA B.F.	2019/CS/061	19000618	2019cs061@stu.ucsc.cmb.ac.lk
ABDULLA M.N.	2019/CS/001	19000014	2019cs001@stu.ucsc.cmb.ac.lk
ALUTHWATHTA A.G.N.P.	2019/CS/005	19000057	2019cs005@stu.ucsc.cmb.ac.lk
RODRIGO P.C.A.	2019/CS/147	19001479	2019cs147@stu.ucsc.cmb.ac.lk

Project Supervisor: Ms. M.W.A.C.R Wijesinghe

Project Co - Supervisor: Ms. G.K.K Perera

Client of the Project

Name: W.M.Anura Bandara Wijesinghe

Address: Malwathugoda Auto Service, Kaudella, Galagedara.

Contact Number: +94 77 329 3347

1. Introduction

1.1 Purpose

This software requirements specification document covers the functional requirements, quality attribute requirements, design documents and other critical information of the entire system that is being developed by the team.

1.2 Document Conventions

The following abbreviations have been used in this document.

ST – Service Team

STL – Service Team Leader

DB – Database

UI – User Interface

1.3 Intended audience

This document is intended to provide information about the system being developed to the Project Examination Board (PEB) of UCSC, the supervisors and co-supervisors, client of the project and to the development team (CS-01).

1.4 Domain description

The current car wash process of Malwathugoda Auto Service is that the vehicles that visit the wash centre take the service on **first come first serve** basis.

The wash services currently provided by the service centre are **exterior wash, interior cleaning and sanitization.** The wash service is provided on separate vehicle types. The price depends on both wash package and vehicle type.

The service, equipment, employee and reservation records are managed manually with records on spreadsheets and paper.

1.5 Current system and its limitations

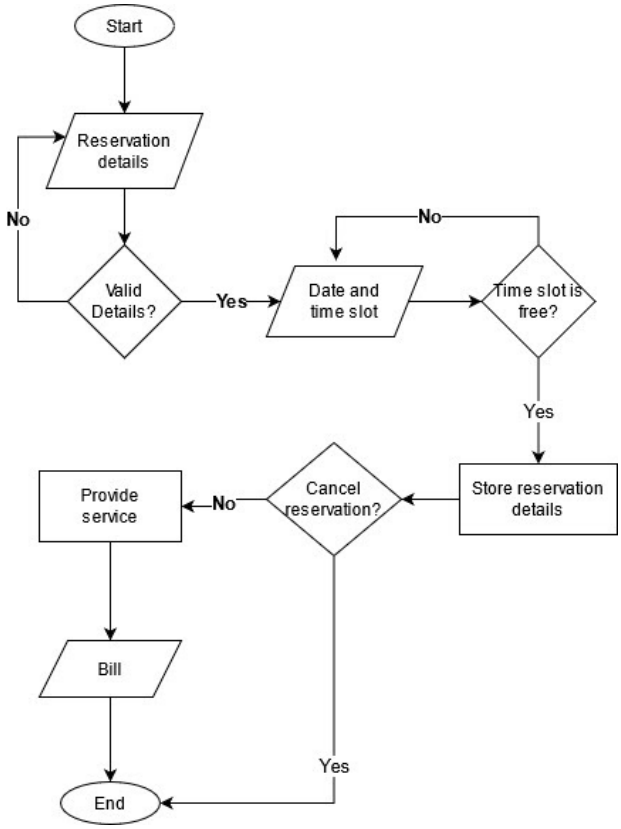


Figure 1: Current System

When implementing a mobile car wash service, the customers would be making reservations through phone calls providing details. Valid reservation details in the above diagram refer to regional addresses, valid mobile numbers and vehicle details of a customer.

After obtaining the details, the staff must check if the requested time slot is available. If not, the customer is informed and both parties agree on a suitable time slot.

The reservation details are then recorded and the service is provided if the customer does not cancel the reservation meantime.

The current system amounts to several weaknesses/limitations.

1. On the reserved time itself, the services provided to the customers might get delayed due to various reasons.
2. Time wasted on a phone call around 3-5 minutes is an inconvenience to the customer and the manager.

3. All records of services done, reservations, employee data are maintained on paper or manually through spreadsheets by a staff member. The client finds it difficult to keep them updated.
4. The client needs to manually check for available time slots for a customer reservation.

A proper online reservation and management system would overcome the above-mentioned weaknesses and limitations.

1.6 Project Scope

1.6.1 In-Scope

The following is the outline of all aspects of functionalities delivered by our project.

- Web application
- Creating and managing user profiles.
- Managing employees and service details.
- Smart calendar to make and view reservations.
- Automated report generation (Dashboard).
- Tracking customer location through Google maps.
- Customer review system.
- Online reservation system.
- Generating bills for the service provided.
- User authentication

1.6.2 Out-Scope

The following will not be part of the project or its outcomes.

- Mobile application
- Online payment
- Rating individual employees

1.7 Project Goal

The goal of this project is to create a fully-functional user-friendly online reservation system for Malwathugoda Auto Service to manage online bookings and other business aspects of the mobile car wash services.

1.8 Constraints and assumptions

We have identified two major aspects of constraints and assumptions as below.

1.8.1 Project constraints and assumptions

- No frameworks allowed.
- The project shall be finished within one year considering the academic calendar.
- All four members of the team have sufficient knowledge and skills in terms of software development.

1.8.2 System constraints and assumptions

- All users of the system must have a proper internet connection to operate the system.
- All users of the system have basic IT and Language skills to operate the system.
- Cancellation and rescheduling of a reservation can only take place until 24 hours before the scheduled time.
- The Manager assigns service teams to all the customer reservations.
- The Service Team Leader can only see the customer reservations for the current day in his calendar.
- The System Admin handles the technical/database-related operations of the business through the website.

2. Overall Description

2.1 Product Perspective

The product being developed is a stand-alone, new, self-contained product.

2.2 Stakeholders

PEB - Project Examination Board of UCSC

Actors of the system

Manager – Manager of the business

System Admin – Staff member who handles the technical aspect of business management

Service Team Leader – Person-in-charge of the service team who provides car wash services

Customer – Customers of the business

User – Generalized actor of the above four actors.

2.3 Operating Environment

The components of the system operate within any modern web-browser environment which supports HTML and JavaScript.

3. Feasibility Study

An analysis was conducted on the following five aspects to evaluate the feasibility of the proposed system.

3.1 Technical Feasibility

The technologies and resources below are available for the development of the system.

- For frontend development HTML5, CSS3 and JavaScript will be used.
- For backend development PHP and MySQL will be used.
- As IDEs, Visual Studio Code and Atom will be used.
- XAMPP, draw.io (UMLs), Github, Microsoft Word, Microsoft Excel and
- Trello will be used as tools and utilities for this project.
- All four members are equipped with laptops and have the required technical skills for the development of this project.

3.2 Economic Feasibility

The product would be financially beneficial for the client after considering the cost of development mentioned below.

- Open-source software and tools will be used for development.
- The client has agreed to purchase a domain on Hostinger with an annual subscription (9$/yr).
- The cost of using Google map APIs is manageable to the client.
- A free hosting server is already available.
- No additional hardware cost.
- Cost of communication among the team members and the client (via Zoom and MS Teams) is manageable.

3.3 Legal and Ethical Feasibility

The proposed system conforms to the legal and ethical requirements below.

- Customer feedback will be kept confidential.
- User details will be stored securely in the system.
- The client has no legal constraints regarding the operation of the system.

3.4 Operational Feasibility

The product will be easily operatable and usable according to the results below.

- All users of the system have basic IT and Language skills to operate the system.
- Service team leaders have mobile phones with internet connection to access the system at any time.
- Manager and System Admin have separate laptops with internet connection to manage the system.

3.5 Scheduling Feasibility

The system will be developed smoothly by dividing the work equally between each member of the development team and will be deployed according to the proposed timeline.

According to the study conducted in above-mentioned areas, the software product will be technically, economically, legally, ethically, operationally and in terms of scheduling, feasible.

4. External Interface Requirements

4.1 User Interfaces

URL (for quality images): https://drive.google.com/file/d/1H-8DHAun4DzvPTxtZwNKBdRMZGZvzmKN/view?usp=sharing

4.1.1 User UI

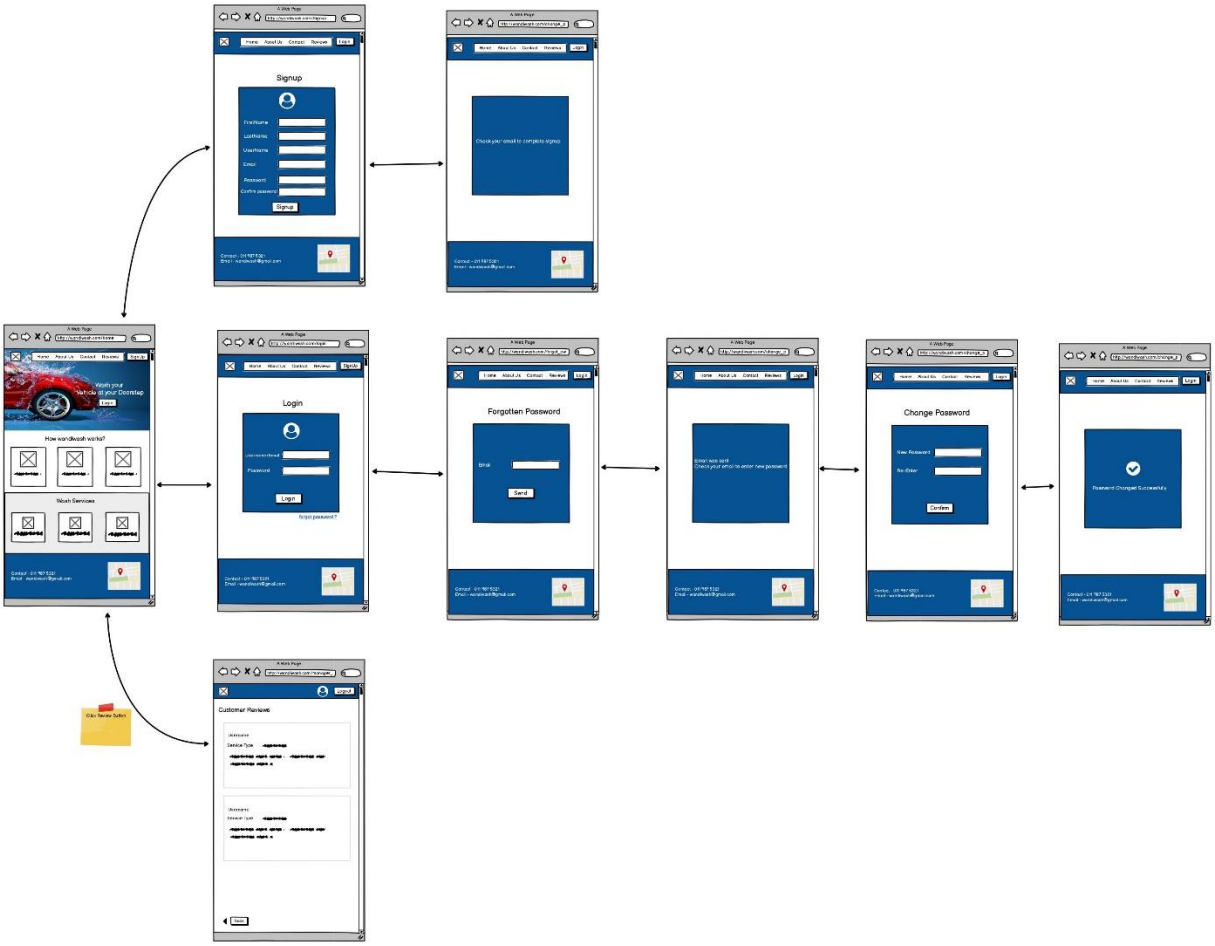


Figure 2: User UI

4.1.2 Customer UI

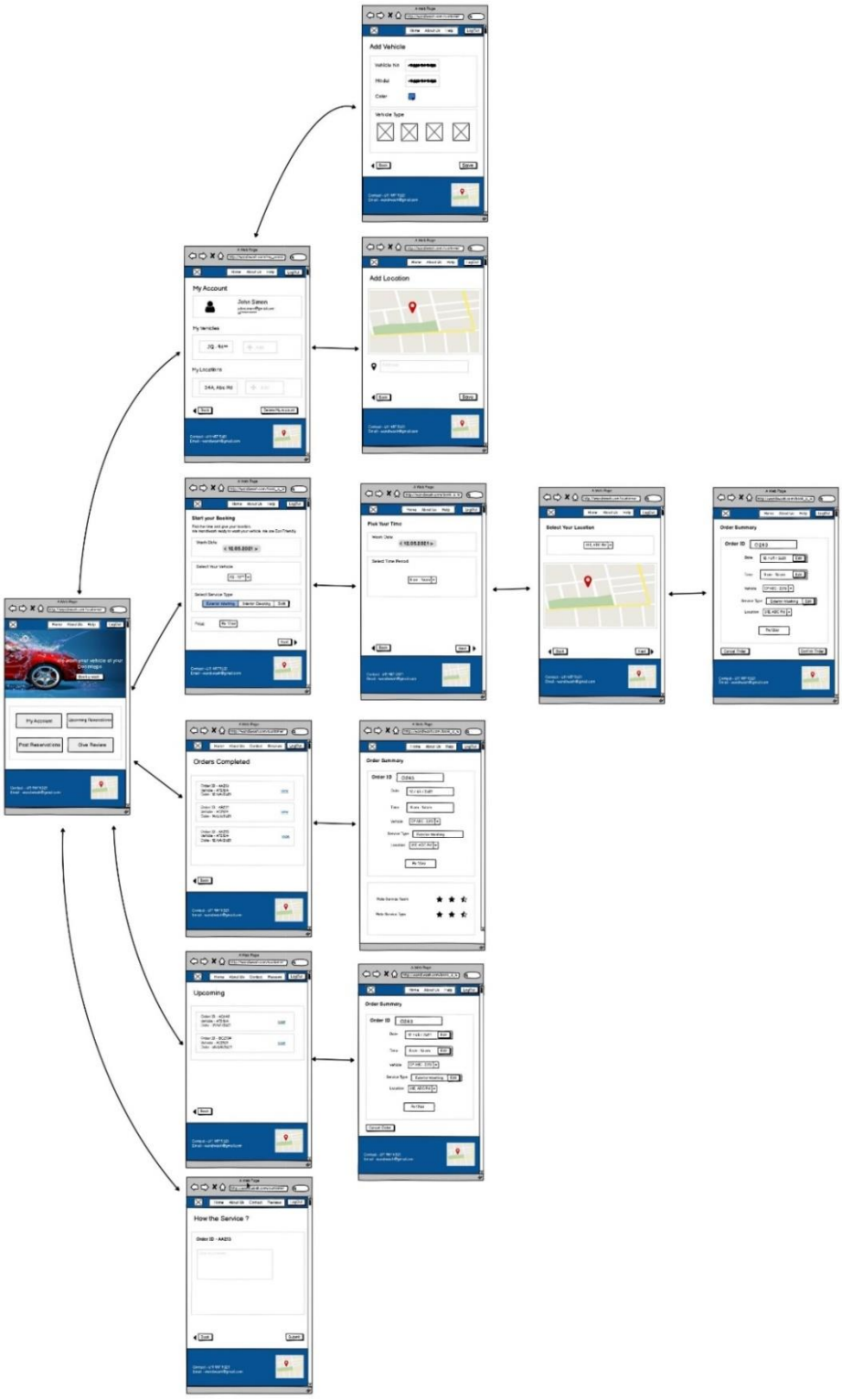


Figure 3: Customer UI

4.1.3 Manager UI

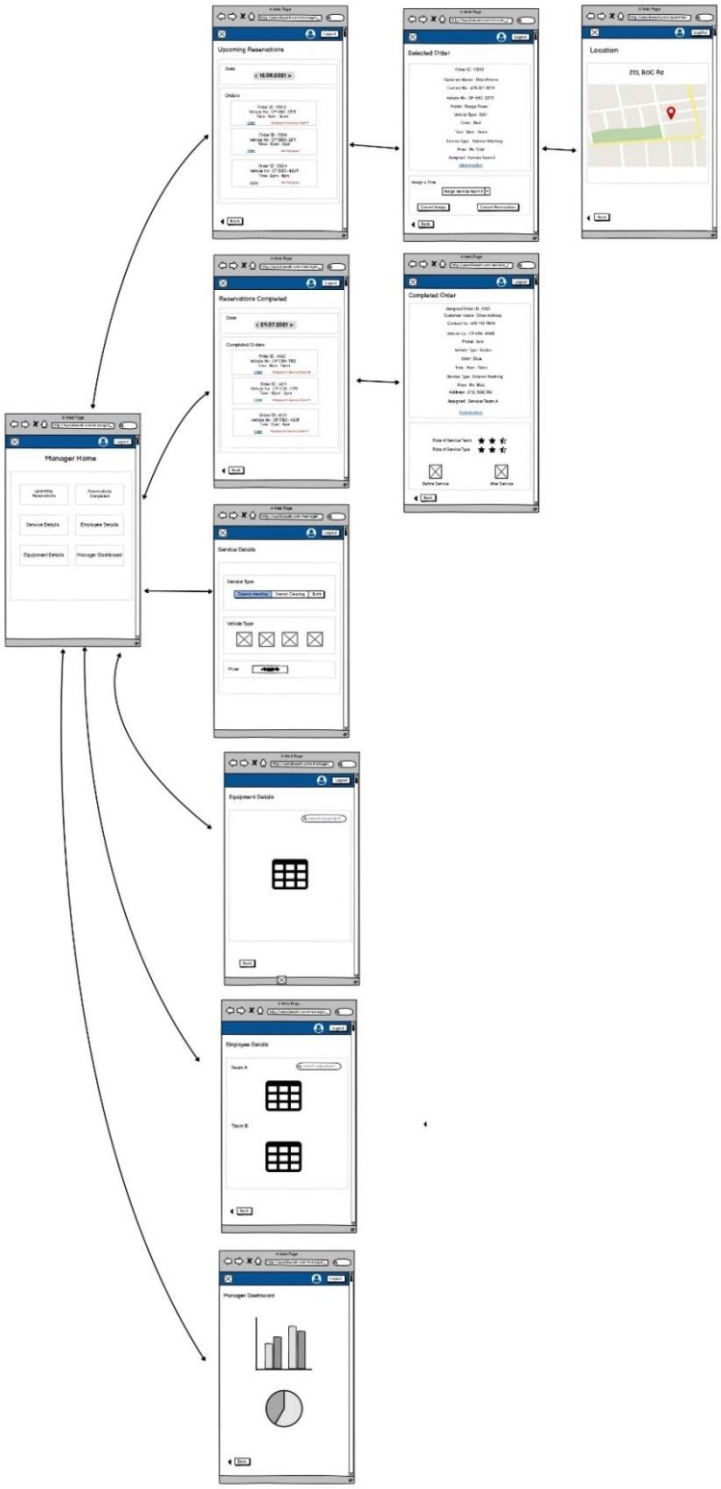


Figure 4: Manager UI

4.1.4 System Admin UI

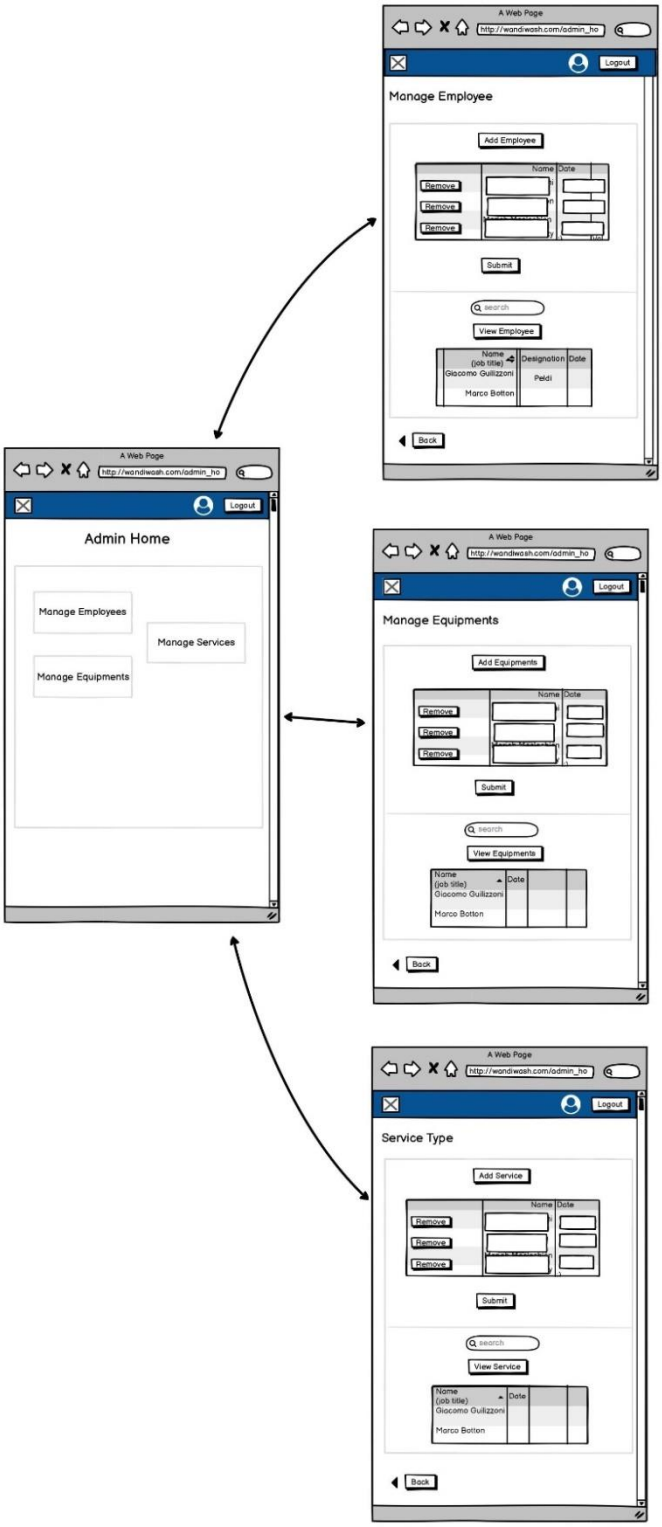


Figure 5: System Admin UI

4.1.5 Service Team Leader UI

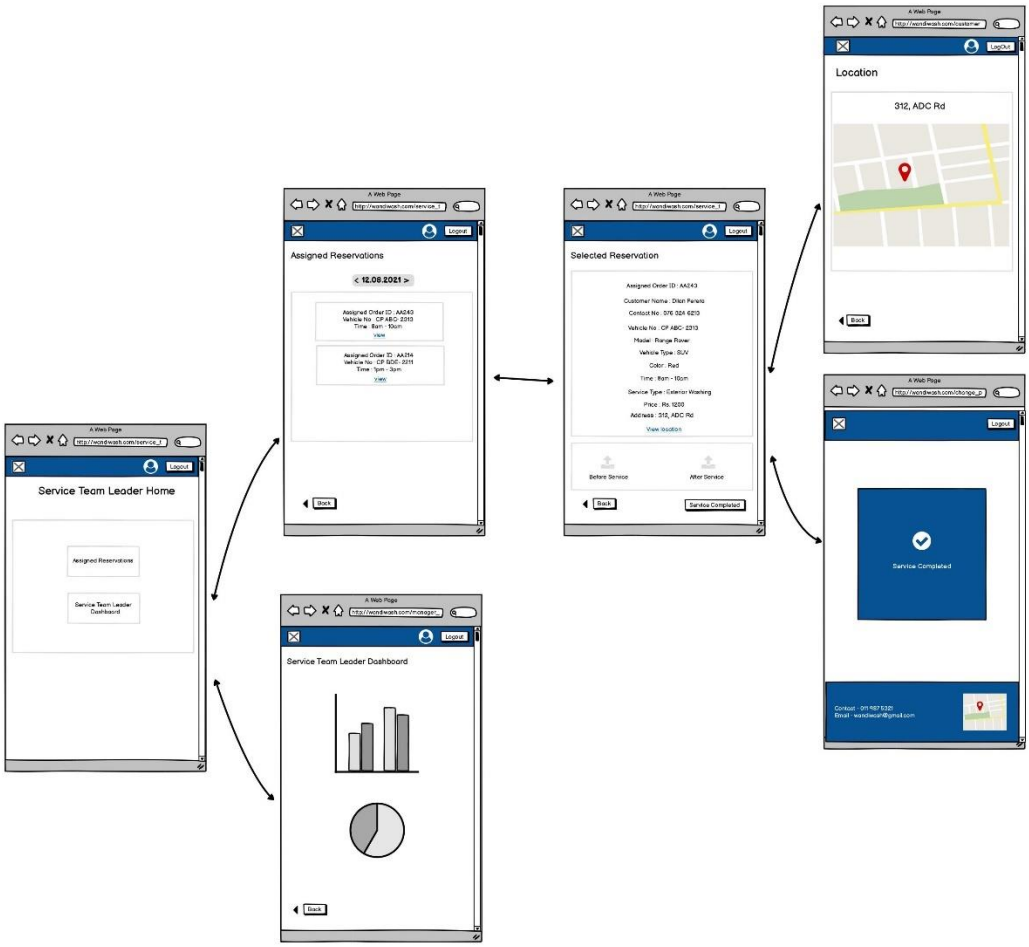


Figure 6: Service Team Leader UI

4.2 Hardware Interfaces

Since the application must run over the internet, all the hardware required to connect to the internet will be a hardware interface for the system. For example, Modem, WAN – LAN, Ethernet Cross-Cable etc.

The system is designed to be compatible and responsive with desktop computers, laptops and mobile phones.

4.3 Software Interfaces

The system will be using Apache HTTP server and hence is compatible with any operating system.

The system will be using MySQL database to store and retrieve data.

The system will be built using PHP considering development team size and the type of web application being built (middle web application). PHP would provide the system better performance at all stages and easy modification at any time. The system also uses built-in PHP functions for security purposes.

The system will be using charts.js for data visualization on dashboards.

The system will be using PHPMailer to send email to the customers of the system.

The system will be using Google map API to get the customer locations.

5. System Features

5.1 Use Case Diagram

URL (for quality image): https://files.fm/f/s8cgdckxe

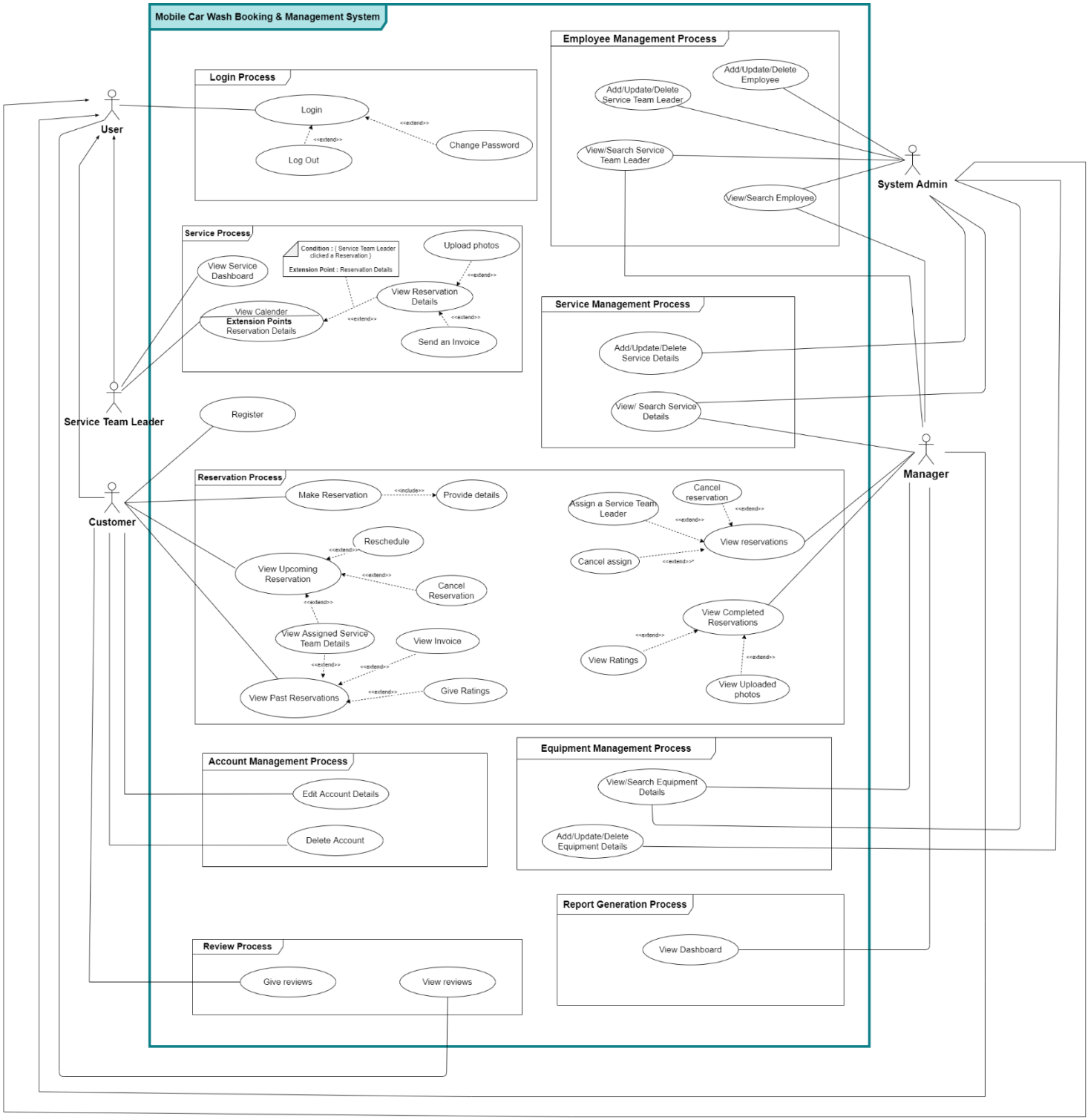


Figure 7: Use Case Diagram

5.2 Use Case Narratives

Author(s): Aluthwaththa A.G.N.P. Date: 2021/06/10 Version: 01

<table>
<tr><td>Use Case Name:</td><td>Login</td><td rowspan="4">Use Case Type
System Requirements:</td></tr>
<tr><td>Use Case Id:</td><td>001</td></tr>
<tr><td>Priority:</td><td>High</td></tr>
<tr><td>Source:</td><td>Requirements - R1</td></tr>
<tr><td>Primary Business Actor:</td><td colspan="2">User</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 describes the event of a user logging into the system. The user accesses the system and provides the username and password. Once the details are accepted, the user is directed to the relevant homepage.</td></tr>
<tr><td>Precondition:</td><td colspan="2">None</td></tr>
<tr><td>Trigger:</td><td colspan="2">This use case is initiated when a user initiates the user login process.</td></tr>
<tr><td rowspan="2">Typical course of events:</td><td>Actor Action</td><td>System Response</td></tr>
<tr><td>Step 1: The user clicks ‘login’.

Step 2: The user provides the username and the password.</td><td>Step 3: The system verifies that the details provided by the user are valid.

Step 4: If the user is a customer, the system directs the user to the customers’ homepage.
If the user is the manager, the system directs the user to the manager’s homepage.
If the user is the system admin, the system directs the user to the system admin’s homepage.
If the user is a service team leader, the system directs the user to the service team leaders’ homepage.</td></tr>
</table>

Alternate Courses:	**Alt-Step 2:** The user requests to change the password. The system sends an email to reset the password. Once the user sets a new password using the email, the system updates the database and redirects the user to the homepage. **Alt-Step 3:** Details provided by the user are not valid. The user is notified the information provided is invalid.
Conclusion:	This use case concludes when the user is successfully redirected to a homepage or in the case of an unregistered user when the user is redirected to the registration page or when an email is sent to change the password.
Postcondition:	None
Assumptions:	All the users of the system must be registered within the system. Reviews can be viewed by all users before logging in to the system. Reservation/rating services are available to users only after they log into the system.

Author(s): Abdulla M.N. Date: 2021/06/10 Version: 01

Use Case Name:	View Reviews	**Use Case Type** **Business Requirements:**
Use Case Id:	002	
Priority:	Low	
Source:	Requirement – R1	
Primary Business Actor:	User	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of a user viewing the customers' reviews.	
Precondition:	-	
Trigger:	This use case is initiated when the user chooses 'View Reviews'.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The user chooses 'View Reviews'.	**Step 2:** The system displays an interface that shows the customer reviews.
Alternate Courses:	None	
Conclusion:	This use case concludes when the system displays the reviews of the customers.	

Post condition:	None
Assumptions:	None

Author(s): Aluthwaththa A.G.N.P. Date: 2021/06/10 Version: 01

Use Case Name:	Register	**Use Case Type** **System Requirements:**
Use Case Id:	003	
Priority:	High	
Source:	Requirements - R1	
Primary Business Actor:	Customer	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of a customer registering into the system. The customer accesses the system and provides the details, username and a password. Once the details are accepted, an email is sent for verification.	
Precondition:	None	
Trigger:	This use case is initiated when a customer initiates the user registration process.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The customer clicks "Register". **Step 3:** The customer provides details, a username and a password. **Step 7:** The customer verifies the email..	**Step 2:** The system redirects the customer to the registration page. **Step 4:** The system checks the validity and uniqueness of the details provided. **Step 5:** The system updates the database with the user details. **Step 6:** The system sends an email to the customer for verification. **Step 8:** The system updates the database.
Alternate Courses:	**Alt-Step 5:** The details provided by the customer are invalid or not unique. The system informs this to the customer.	
Conclusion:	This use case concludes when the customer is successfully registered and when an error message is displayed to the customer.	
Postcondition:	None	
Assumptions:	None.	

Author(s): Abdulla M.N. Date: 2021/06/10 Version: 01

<table>
<tr><td>Use Case Name:</td><td>Make Reservation</td><td rowspan="4">Use Case Type
Business Requirements:</td></tr>
<tr><td>Use Case Id:</td><td>004</td></tr>
<tr><td>Priority:</td><td>High</td></tr>
<tr><td>Source:</td><td>Requirement – R1</td></tr>
<tr><td>Primary Business Actor:</td><td colspan="2">Customer</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 describes the event of a Customer reserving an available and convenient date and time slot by selecting his/her vehicle category, wash package with prices and giving the location.</td></tr>
<tr><td>Precondition:</td><td colspan="2">The user must be registered within the system.
The user must be logged in.</td></tr>
<tr><td>Trigger:</td><td colspan="2">This use case is initiated when the ‘Make a Reservation’ button is clicked.</td></tr>
<tr><td rowspan="2">Typical course of events:</td><td>Actor Action</td><td>System Response</td></tr>
<tr><td>Step 1: The customer clicks the ‘Make a Reservation’.

Step 4: The customer provides the details.

Step 6: The customer confirms the reservation.</td><td>Step 2: The system validates the user.
Step 3: The system displays an interface with a smart calendar to select available and convenient date and time and to select vehicle types, wash packages with prices and to get the customer location.
Step 5: The system accepts the details and displays the reservation details with the price.
Step 7: The system updates the database.</td></tr>
<tr><td>Alternate Courses:</td><td colspan="2">Alt-Step 5: User has not provided necessary and accurate information to make a reservation. User is notified that the reservation is unavailable due to insufficient or incorrect information.
Alt-Step 6: If the reservation is not confirmed by the customer, the system will redirect the customer back to the Home page.</td></tr>
<tr><td>Conclusion:</td><td colspan="2">This use case concludes when the system notifies the user of the successful/unsuccessful reservation.</td></tr>
<tr><td>Post condition:</td><td colspan="2">None</td></tr>
<tr><td>Assumptions:</td><td colspan="2">None</td></tr>
</table>

Author(s): Abdulla M.N. Date: 2021/06/10 Version: 01

Use Case Name:	View Upcoming Reservation	**Use Case Type** **Business Requirements:**
Use Case Id:	005	
Priority:	High	
Source:	Requirement – R1	
Primary Business Actor:	Customer	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of a customer viewing his/her upcoming reservations.. The customer can choose to see the details of each reservation such as assigned service team leaders and also choose to cancel/reschedule them.	
Precondition:	The user must be registered within the system. The user must be logged in.	
Trigger:	This use case is initiated when the 'View Upcoming Reservations' button is clicked.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The user clicks the 'View upcoming Reservations' button.	**Step 2:** The system displays the user's upcoming reservations with details.
Alternate Courses:	**Alt-Step 3:** The customer clicks the reschedule button and chooses a relevant date and time. The system removes the reservation from the current time slot and assigns it to the new time slot. **Alt-Step 4:** Customer clicks the cancel button of the relevant reservation and confirms it. The current reservation is cancelled by the system. **Alt-Step 5:** Customer clicks 'View Service Team Details' of relevant reservations. The system shows the details of the service team assigned to the reservation.	
Conclusion:	This use case concludes when the system displays the reservations or the service team details or when the system updates the database after the user reschedules/cancels the reservation.	
Post condition:	None	
Assumptions:	Customers can only reschedule/cancel the reservation within the time up to 24 hours before the reserved service time.	

Author(s): Abdulla M.N. Date: 2021/06/10 Version: 01

Use Case Name:	View Past Reservations	**Use Case Type** **Business Requirements:**
Use Case Id:	005	
Priority:	Medium	
Source:	Requirement – R1	
Primary Business Actor:	Customer	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of a customer viewing his/her past reservations. The customer can choose to see the details of each past reservation such as invoice / assigned service team details and give ratings on the service.	
Precondition:	The user must be registered within the system. The user must be logged in.	
Trigger:	This use case is initiated when the 'View Past Reservations' button is clicked.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The user clicks the 'View Past Reservations' button.	**Step 2:** The system displays the user's past reservations with details.
Alternate Courses:	.**Alt-Step 3:** Customer clicks 'View Service Team Details' of relevant reservations. The system shows the details of the service team assigned to the reservation. **Alt-Step 4:** Customer clicks 'View Invoice' of relevant reservation. The system shows the invoice of the reservation. **Alt-Step 5**: Customer clicks give ratings of relevant reservations and gives the rating. The system updates the database.	
Conclusion:	This use case concludes when the system displays the reservations or the invoice or the service team details or when the system updates the database after the user gives the ratings.	
Post condition:	None	
Assumptions:	None	

Author(s): Abdulla M.N. Date: 2021/06/10 Version: 01

Use Case Name:	Give Reviews	**Use Case Type** **Business Requirements:**
Use Case Id:	006	
Priority:	Low	

Source:	Requirement – R1	
Primary Business Actor:	Customer	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of a Customer giving reviews about the service.	
Precondition:	The user must be registered within the system. The user must be logged in.	
Trigger:	This use case is initiated when the 'Provide a Review' button is clicked.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The user clicks the 'Provide a Review' button. **Step 3:** The user types the review and submits.	**Step 2:** The system displays an interface to type the review. **Step 4:** The system stores the review and displays a 'thank you' message.
Alternate Courses:	None	
Conclusion:	This use case concludes when the system displays the 'thank you' message.	
Post condition:	None	
Assumptions:	Customer provides reviews only after a service is provided to him/her.	

Author(s): Abdulla M.N. Date: 2021/06/10 Version: 01

Use Case Name:	Edit Account Details	**Use Case Type** **Business Requirements:**
Use Case Id:	007	
Priority:	Medium	
Source:	Requirement – R1	
Primary Business Actor:	Customer	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of a customer editing account details.	

Precondition:	The user must be registered within the system. The user must be logged in.	
Trigger:	This use case is initiated when ‘My Account’ is clicked.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The user clicks ‘My Account’. **Step 3:** Customer clicks the edit button and replaces the details with new details.	**Step 2:** The system displays his/her account details. **Step 4:** The system validates the user. **Step 5:** The system updates the database with the new details.
Alternate Courses:	**None**	
Conclusion:	This use case concludes when the system displays the account details or system updates the database after the user edits his/her account details.	
Post condition:	None	
Assumptions:	None	

Author(s): Abdulla M.N. Date: 2021/06/10

Version: 01

Use Case Name:	Delete Account	**Use Case Type** **Business Requirements:**
Use Case Id:	008	
Priority:	Medium	
Source:	Requirement – R1	
Primary Business Actor:	Customer	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of a customer deleting the account.	
Precondition:	The user must be registered within the system. The user must be logged in.	
Trigger:	This use case is initiated when ‘My Account’ is clicked.	

Typical course of events:	Actor Action	System Response
	Step 1: The user clicks 'My Account'. **Step 3:** Customer clicks the delete button.	**Step 2:** The system displays his/her account details. **Step 4:** The system validates the user. **Step 5:** The system updates the database.
Alternate Courses:	**None**	
Conclusion:	This use case concludes when the system updates the database after the user deletes his/her account.	
Post condition:	None	
Assumptions:	None	

Author(s): Abdulla M.N. Date: 2021/06/10

Version: 01

Use Case Name:	View Reservations	**Use Case Type** **Business Requirements:**
Use Case Id:	009	
Priority:	High	
Source:	Requirement – R1	
Primary Business Actor:	Manager	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the manager viewing the upcoming reservations. The Manager can choose to assign a service team leader to any reservations of a particular day or cancel the assignment or cancel a reservation.	
Precondition:	The user must be registered within the system. The user must be logged in.	
Trigger:	This use case is initiated when the manager clicks 'View Reservations'.	
Typical course of	**Actor Action**	**System Response**

events:	**Step 1:** The manager clicks the 'View Reservations' button. **Step 4:** The manager chooses a particular reservation.	**Step 2:** The system validates the user. **Step 3:** The system displays an interface that shows the customer reservations with details. **Step 5:** The system displays the details of the current reservation.
Alternate Courses:	**Alt-Step 6:** The manager assigns a service team leader to the relevant reservation. The system updates the calendar and the database **Alt-Step 7:** The manager cancels an assignment of a service team leader to the relevant reservation. The system updates the calendar and the database. **Alt-Step 8:** The manager cancels a reservation. The system updates the calendar and the database.	
Conclusion:	This use case concludes when the system displays the reservation details or a service team leader is assigned or cancelled from an assignment to a reservation or when a reservation is cancelled..	
Post condition:	None	
Assumptions:	The manager assigns a service team leader to all the customer reservations.	

Author(s): Ilma B.F. Date: 2021/06/10 Version: 01

Use Case Name:	View Completed Reservations	**Use Case Type** **Business Requirements:**
Use Case Id:	010	
Priority:	Medium	
Source:	Requirement – R1	
Primary Business Actor:	Manager	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the manager viewing the completed reservations. The Manager can view ratings of the relevant reservation or view uploaded photos by the service team leader.	
Precondition:	The user must be registered within the system.	

	The user must be logged in.	
Trigger:	This use case is initiated when the manager clicks 'View Completed Reservations'.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The manager clicks the 'View Completed Reservations' button.	**Step 2:** The system validates the user. **Step 3:** The system displays an interface that shows the past reservations.
Alternate Courses:	**Alt-Step 4:** The manager chooses a particular reservation. The system displays the details of the current reservation. **Alt-Step 5:** The manager clicks 'view ratings'. The system displays the ratings of the particular reservation. **Alt-Step 6:** The manager clicks 'view photos. The system displays the photos of the particular reservation.	
Conclusion:	This use case concludes when the system displays the reservation details or the details of a particular reservation or the photos of a reservation.	
Post condition:	None	
Assumptions:	None	

Author(s): Abdulla M.N. Date: 2021/06/10 Version: 01

Use Case Name:	View Dashboard	**Use Case Type** **Business Requirements:**
Use Case Id:	011	
Priority:	Medium	
Source:	Requirement – R1	
Primary Business Actor:	Manager	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the manager viewing the monthly revenue and number of bookings as charts and graphs.	
Precondition:	The user must be registered within the system. The user must be logged in.	
Trigger:	This use case is initiated when the manager chooses 'View Dashboard'.	

Typical course of events:	Actor Action	System Response
	Step 1: The Manager chooses the 'View Dashboard'.	**Step 2:** The system validates the user. **Step 3:** The system displays the Dashboard.
Alternate Courses:	None.	
Conclusion:	This use case concludes when the system displays the Dashboard to the manager.	
Post condition:	None	
Assumptions:	None	

Author(s): Ilma B.F.

Date: 2021/06/10
Version: 01

Use Case Name:	Add/ Update/ Delete Employee	**Use Case Type Business Requirements:**
Use Case Id:	012	
Priority:	Medium	
Source:	Requirements - R1	
Primary Business Actor:	System Admin	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the system admin managing employee details within the system. The system admin can choose to add/update/delete employee details within the system. Once the change has been made, relevant changes are updated within the database.	
Precondition:	The user must be registered within the system. The user must be logged in.	
Trigger:	This use case is initiated when the system admin clicks 'Manage Employee Details'.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The system admin chooses 'Manage Employee Details'. **Step 3:** The system admin chooses either to add/update/delete Employee details	**Step 2:** The system validates the user. .**Step 4 :** System update the database.
Alternate	None	

Courses:	
Conclusion:	This use case concludes when the system admin has successfully added / updated/ deleted relevant employee records.
Post condition:	None
Assumptions:	The system admin makes changes that do not violate the domain constraints.

Author(s): Ilma B.F.

Date: 2021/06/10
Version: 01

Use Case Name:	Add/ Update/ Delete Service Team Leader	**Use Case Type Business Requirements:**
Use Case Id:	013	
Priority:	Medium	
Source:	Requirements - R1	
Primary Business Actor:	System Admin	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the system admin managing service team leaders' details within the system. The system admin can choose to add/update/delete service team leaders details within the system. Once the change has been made, relevant changes are updated within the database.	
Precondition:	The user must be logged in.	
Trigger:	This use case is initiated when the system admin clicks 'Manage Service Team Leaders Details'.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The system admin chooses 'Manage Service Team Leaders Details'. **Step 3:** The system admin chooses either to add/update/delete Service Team Leader details.	**Step 2:** The system validates the user. **Step 4:** The System updates the Database.
Alternate Courses:	None.	
Conclusion:	This use case concludes when the system admin has successfully added/updated/deleted relevant service team leaders' records.	

Post condition:	None
Assumptions:	The system admin makes changes that do not violate the domain constraints.

Author(s): Ilma B.F. Date: 2021/06/10 Version: 01

Use Case Name:	View / Search Employee	**Use Case Type** **Business Requirements:**
Use Case Id:	014	
Priority:	Medium	
Source:	Requirements - R1	
Primary Business Actor:	System Admin/ Manager	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the system admin or manager viewing/searching employee details within the system.	
Precondition:	The user must be logged in.	
Trigger:	This use case is initiated when the system admin/manager clicks 'View Employee Details'.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The system admin/manager chooses 'View Employee Details'.	**Step 2:** The system validates the user. **Step 3:** The system displays details with options to search and filter.
Alternate Courses:	None	
Conclusion:	This use case concludes when the system displays the employee details.	
Post condition:	None	
Assumptions:	None	

Author(s): Ilma B.F. Date: 2021/06/10
Version: 01

Use Case Name:	View / Search Service Team Leader	**Use Case Type**

Use Case Id:	015	**Business Requirements:**
Priority:	Medium	
Source:	Requirements - R1	
Primary Business Actor:	System Admin / Manager	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the system admin/manager viewing/searching service team leader details within the system.	
Precondition:	The user must be logged in.	
Trigger:	This use case is initiated when the system admin/manager clicks 'View Service Team Leader Details'.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The system admin/manager chooses 'View Service Team Leader Details'.	**Step 2:** The system validates the user. **Step 3:** The system displays details with options to search and filter.
Alternate Courses:	None	
Conclusion:	This use case concludes when the system displays the service team leader details.	
Post condition:	None	
Assumptions:	None	

Author(s): Ilma B.F. Date: 2021/06/10 Version: 01

Use Case Name:	Add/ Update/ Delete Service Details	**Use Case Type** **Business Requirements:**
Use Case Id:	016	
Priority:	Medium	
Source:	Requirements - R1	
Primary Business Actor:	System Admin	
Other Participating Actors:	None	

Other Interested Stakeholders:	None	
Description:	This use case describes the event of the system admin managing service details within the system. The system admin can choose to add/update/delete service details within the system. Once the change has been made, relevant changes are updated within the database.	
Precondition:	The user must be logged in.	
Trigger:	This use case is initiated when the system admin clicks 'Manage Service Details'.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The system admin chooses 'Manage Service Details'. **Step 3:** The system admin chooses to add/update/delete Service details.	**Step 2:** The system validates the user. **Step 4:** The System updates the Database.
Alternate Courses:	None.	
Conclusion:	This use case concludes when the system admin has successfully added/updated/deleted relevant service records.	
Post condition:	None	
Assumptions:	The system admin makes changes that do not violate the domain constraints.	

Author(s): Ilma B.F. Date: 2021/06/10 Version: 01

Use Case Name:	View / Search Service Details.	**Use Case Type** **Business Requirements:**
Use Case Id:	017	
Priority:	Medium	
Source:	Requirements - R1	
Primary Business Actor:	System Admin / Manager	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the system admin/manager viewing/searching service details within the system.	
Precondition:	The user must be logged in.	

Trigger:	This use case is initiated when the system admin/manager clicks ‘View Service Details’.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The system admin/manager chooses ‘View Service Details’.	**Step 2:** The system validates the user. **Step 3:** The system displays details with options to search and filter.
Alternate Courses:	None	
Conclusion:	This use case concludes when the system displays the service details.	
Post condition:	None	
Assumptions:	None	

Author(s): Ilma B.F. Date: 2021/06/10 Version: 01

Use Case Name:	Add/ Update/ Delete Equipment Details	**Use Case Type** **Business Requirements:**
Use Case Id:	018	
Priority:	Medium	
Source:	Requirements - R1	
Primary Business Actor:	System Admin	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the system admin managing equipment details within the system. The system admin can choose to add/update/delete equipment details within the system. Once the changes are made, they are updated within the database.	
Precondition:	The user must be logged in.	
Trigger:	This use case is initiated when the system admin clicks ‘Manage Equipment Details’.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The system admin chooses ‘Manage Equipment Details’. **Step 3:** The system admin chooses either to add/update/delete equipment details.	**Step 2:** The system validates the user. **Step 4:** The System updates the Database.

Alternate Courses:	None.
Conclusion:	This use case concludes when the system admin has successfully added/updated/deleted relevant equipment records.
Post condition:	None
Assumptions:	The system admin makes changes that do not violate the domain constraints.

Author(s): Abdulla M.N. Date: 2021/06/10 Version: 01

Use Case Name:	View / Search Equipment Details.	**Use Case Type Business Requirements:**
Use Case Id:	019	
Priority:	Medium	
Source:	Requirements - R1	
Primary Business Actor:	System Admin/ Manager	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of the system admin/ manager viewing/searching equipment details within the system.	
Precondition:	The user must be logged in.	
Trigger:	This use case is initiated when the system admin/manager clicks View Equipment Details'.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The system admin/manager chooses 'View Equipments Details'.	**Step 2:** The system validates the user. **Step 3:** The system displays details with options to search and filter.
Alternate Courses:	None	
Conclusion:	This use case concludes when the system displays the equipment details.	
Post condition:	None	
Assumptions:	None	

Author(s): Abdulla M.N. Date: 2021/06/10 Version: 01

Use Case Name:	View Service Dashboard	**Use Case Type Business Requirements:**
Use Case Id:	020	
Priority:	Medium	
Source:	Requirement – R1	
Primary Business Actor:	Service Team Leader	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of a service team leader viewing the details of services completed by his/her team from a chart/graph.	
Precondition:	The user must be registered within the system. The user must be logged in.	
Trigger:	This use case is initiated when the service team leader chooses 'View Service Dashboard'.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The Service Team Leader chooses the 'View Service Dashboard'.	**Step 2:** The system validates the user. **Step 3:** The system displays the Service Dashboard.
Alternate Courses:	None.	
Conclusion:	This use case concludes when the system displays the Dashboard to the service team leader.	
Post condition:	None	
Assumptions:	None	

Author(s): Abdulla M.N. Date: 2021/06/10 Version: 01

Use Case Name:	View Calendar	**Use Case Type Business Requirements:**
Use Case Id:	021	
Priority:	High	
Source:	Requirement – R1	
Primary Business Actor:	Service Team Leader	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event of a service team leader viewing the customer reservations for the particular day. The service team leader can choose a certain reservation he/she has within the day and see the details of the reservations, upload photos and send an invoice to the customer.	
Precondition:	The user must be registered within the system. The user must be logged in.	
Trigger:	This use case is initiated when a service team leader clicks 'View Calendar'.	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: The service team leader chooses the 'View Calendar' option.	**Step 2:** The system validates the user. **Step 3:** The system displays an interface that shows the customer reservations of the particular day.
Alternate Courses:	**Alt-Step 4:** The service team leader clicks 'View Reservation Details'. He/she is shown the details of each reservation. **Alt-Step 5:** The service team leader clicks 'Send Invoice'. An invoice is sent to the relevant customer through the system. **Alt-Step 6:** The service team leader clicks 'Upload picture' and uploads -before and after- service pictures. The system updates the database.	
Conclusion:	This use case concludes when the system displays the calendar or displays each reservation details or sends an invoice to the customer or stores the uploaded photos of the reservation.	
Post condition:	None	
Assumptions:	Service Team Leader sends an invoice to the customer only after the service has been provided.	

5.3 Functional Requirements

5.3.1 User

- Users should be able to login/logout to and from their accounts.
- Users should be able to change their account passwords.
- Users should be able to recover a forgotten password.
- Users should be able to view customers' reviews.

5.3.2 Customer

- Customers should be able to register into the system by creating an account and providing their details.
- Customers should be able to update their account details and delete their account if needed.
- Customers should be able to add one or more locations to their account using Google maps.
- Customers should be able to request a wash package for his/her vehicle at any given location on an available time slot at a given day by making a reservation through the web app.
- Customers should be able to reschedule their reservations up until 24 hours before the reserved time.
- Customers should be able to cancel their reservations up until 24 hours before the reserved time.
- Customers should be able to view upcoming and past bookings through their accounts.
- Customers should be able to give reviews about the service.
- Customers should be able to rate the service provided after a reservation.

5.3.3 Manager

- The manager should be able to view upcoming reservations.
- The manager should be able to view reservation details such as details of the customer, vehicle, address and the package chosen.
- The manager should be able to cancel reservations.
- The manager should be able to select and assign a service team leader to each reservation.
- The manager should be able to cancel the assignment of a service team leader to a reservation.
- The manager should be able to view past reservations and the relevant details.
- The manager should be able to access the dashboard and view monthly reports of the business as graphs and charts.
- The manager should be able to view the details of all the employees.
- The manager should be able to view the details of all the service packages.
- The manager should be able to view the details of all the equipment.
- The manager should be able to view ratings of a particular reservation.
- The manager should be able to see the uploaded photos of a reservation uploaded by the service team leader.

5.3.4 System Admin

- The system admin should be able to view/add/update/delete service records in the system.
- The system admin should be able to view/add/update/delete employee records in the system.
- The system admin should be able to view/add/update/delete equipment records in the system.
- The system admin should be able to view/add/update/delete service team leader records in the system.

5.3.5 Service Team Leader

- Service Team Leaders should be able to view the daily reservations assigned by the manager on their calendar.
- Service Team Leaders should be able to view details of a reservation assigned to them.
- Service Team Leaders should be able to view the customer location of a reservation in Google maps.
- Service Team Leaders should be able to upload photos of a reservation after providing the service.
- Service Team Leaders should be able to send an invoice through the system to a particular customer after they are done providing the service.
- Service Team Leaders should be able to see reports of their past reservations in the dashboard.

6. Quality Attributes

6.1 Quality Attribute Requirements

6.1.1 General Scenarios

Availability

Portion of scenario	**Possible Values**
Source	End User
Stimulus	Incorrect Response
Artifact	Process
Environment	Normal Operation
Response	Operate in a degraded mode while repair is being affected
Response Measure	Maximum 2 hours

Portion of scenario	**Possible Values**
Source	End User
Stimulus	Incorrect Timing
Artifact	Process
Environment	Normal Operation
Response	Operate in a degraded mode
Response Measure	Maximum 2 hours

Security

Portion of scenario	**Possible Values**
Source	Correctly Identified Individual
Stimulus	Tries to modify information
Artifact	Data within the system
Environment	Normal Operation
Response	Transactions are carried out in a fashion that, Data is protected from unauthorized access; Data is not being manipulated without authorization.
Response Measure	System not compromised when a particular data value is compromised

Usability

Portion of scenario	**Possible Values**
Source	End User
Stimulus	Learn system features
Artifact	System
Environment	Runtime
Response	Interface is familiar to user; Interface is usable in an unfamiliar context
Response Measure	User satisfaction

Modifiability

Portion of scenario	**Possible Values**
Source	Developer
Stimulus	Add/Modify functionality
Artifact	Code
Environment	Design time
Response	Make modification
Response Measure	No new defects introduced

Performance

Portion of Scenario	**Possible values**
Source	End User
Stimulus	Arrival of a stochastic event
Artifact	System services
Environment	Normal mode
Response	Process event
Response Measure	Less than 3 seconds

Portion of Scenario	**Possible values**
source	End User
stimulus	Arrival of a stochastic event
artifact	Data within the system
environment	Normal Mode
response	Process event
Response measure	Less than 3 seconds

6.1.2 Concrete Scenarios

Availability

1. Incorrect Response Issues

Portion of scenario	**Possible Values**
Source	Internal
Stimulus	Incorrect bill
Artifact	Process
Environment	Normal Operation
Response	Operate in a degraded mode while repair is being affected
Response Measure	Maximum 2 hours

Portion of scenario	**Possible Values**
Source	Internal
Stimulus	System shows incorrect free time slots to customer
Artifact	Process
Environment	Normal Operation
Response	Operate in a degraded mode while repair is being affected
Response Measure	Maximum 2 hours

Portion of scenario	**Possible Values**
Source	Internal
Stimulus	Reports on Dashboard contains incorrect values
Artifact	Process
Environment	Normal Operation
Response	Operate in a degraded mode while repair is being affected
Response Measure	Maximum 2 hours

2. Incorrect Timing Issues

Portion of scenario	**Possible Values**
Source	Internal
Stimulus	Invoice takes more than 5 seconds to send
Artifact	Process
Environment	Normal Operation
Response	Operate in a degraded mode while repair is being affected
Response Measure	Maximum 2 hours

3. Omission

Portion of scenario	**Possible Values**
Source	Internal
Stimulus	Reservation functionalities unresponsive.
Artifact	Process
Environment	Normal Operation
Response	Operate in a degraded mode while repair is being affected
Response Measure	Maximum 2 hours

Portion of scenario	**Possible Values**
Source	Internal
Stimulus	Calendar unresponsive
Artifact	Process
Environment	Normal Operation
Response	Operate in a degraded mode while repair is being affected
Response Measure	Maximum 2 hours

Security

Portion of scenario	**Possible Values**
Source	Correctly Identified Individual
Stimulus	Tries to modify information
Artifact	Data within the system
Environment	Normal Operation
Response	Transactions are carried out in a fashion that, Data is protected from unauthorized access; Data is not being manipulated without authorization.
Response Measure	System not compromised when a particular data value is compromised

Usability

Portion of scenario	**Possible Values**
Source	End User
Stimulus	Learn system features
Artifact	System
Environment	Runtime
Response	Interface is familiar to user; Interface is usable in an unfamiliar context
Response Measure	User satisfaction

Modifiability

Portion of scenario	Possible Values
Source	Developer
Stimulus	Add/Modify functionality
Artifact	Code
Environment	Design time
Response	Make modification
Response Measure	No new defects introduced

Portion of scenario	Possible Values
Source	Developer
Stimulus	Wishes to modify UI
Artifact	Code
Environment	Design time
Response	Make modification
Response Measure	No new defects introduced

Performance

Portion of Scenario	Possible values
Source	End User
Stimulus	Register/Login (stochastic)
Artifact	System services
Environment	Normal mode
Response	Process event
Response Measure	Average latency of 5 seconds

Portion of Scenario	**Possible values**
Source	Customer
Stimulus	Make reservation (stochastic)
Artifact	System services
Environment	Normal mode
Response	Process event
Response Measure	Average latency of 5 seconds

Portion of Scenario	**Possible values**
source	End User
stimulus	Initiate database transactions (stochastic)
artifact	Database
environment	Normal Mode
response	Process event
Response measure	Average latency of 5 seconds

6.2 Tactics to achieve Quality Attribute requirements

6.2.1 Tactics for Availability

Repair tactics:

The degradation tactic [1] – Individual component failures would gracefully reduce system functionality rather than causing total system failure. Other system components with critical functionalities are maintained in the presence of such component failures.

Fixing any incorrect response/incorrect timing issues while the system is in degraded mode.

Optimizing database (using indexes etc.)

Reintroduction tactics:

The shadow tactic – Operating the previously failed and upgraded component in ‘a shadow mode’ for a predefined amount of time prior to reverting it back to an active role. During this time, its behavior is monitored for correctness.

6.2.2 Tactics for Security

Resist attacks:

Authenticate actors - Using password-based authentication mechanisms to provide secure login into the system.

Authorize actors – Ensuring that an authenticated actor has the rights to access/modify either data or services by providing access control mechanisms within the system for each actor.

React to attacks:

Prevent user from logging in for a predefined amount of time after multiple failed login attempts.

Taking frequent backups to prevent data loss.

6.2.3 Tactics for Usability

Separating user interface (MVC) – Separating the user interface so that critical user interface choices can be made without having to recode.

Support user initiative – Using MVC to support user initiative tasks (e.g., cancelling a reservation) by providing multiple views of the data.

Providing a 'Help' feature to the users in using other system features.

6.2.4 Tactics for Modifiability

Reduce the size of a Module:

Split Module - Using Component-Based-Software-Engineering (CBSE) to split the whole system into components that are smaller modules with minimum functionalities to reduce the average cost of future changes.

Reduce coupling:

Encapsulation – Introducing explicit interfaces to each component through which other components can communicate with. External components can only interact with the component through the exposed interface.

Restrict dependencies – Using layered architecture (MVC) to restrict the dependencies or interactions among the modules of the system.

6.2.5 Tactics for Performance

Using Component-Based-Software-Engineering (CBSE) to limit the dependencies among the components and therefore to increase performance [2].

Choosing a good web host.

Optimizing Database (Using indexes).

7. System Architecture

7.1 Derived Component Diagram

URL (for quality image): https://files.fm/f/bey9a3r2x

The derived component diagram [3] below shows the architecture of the system.

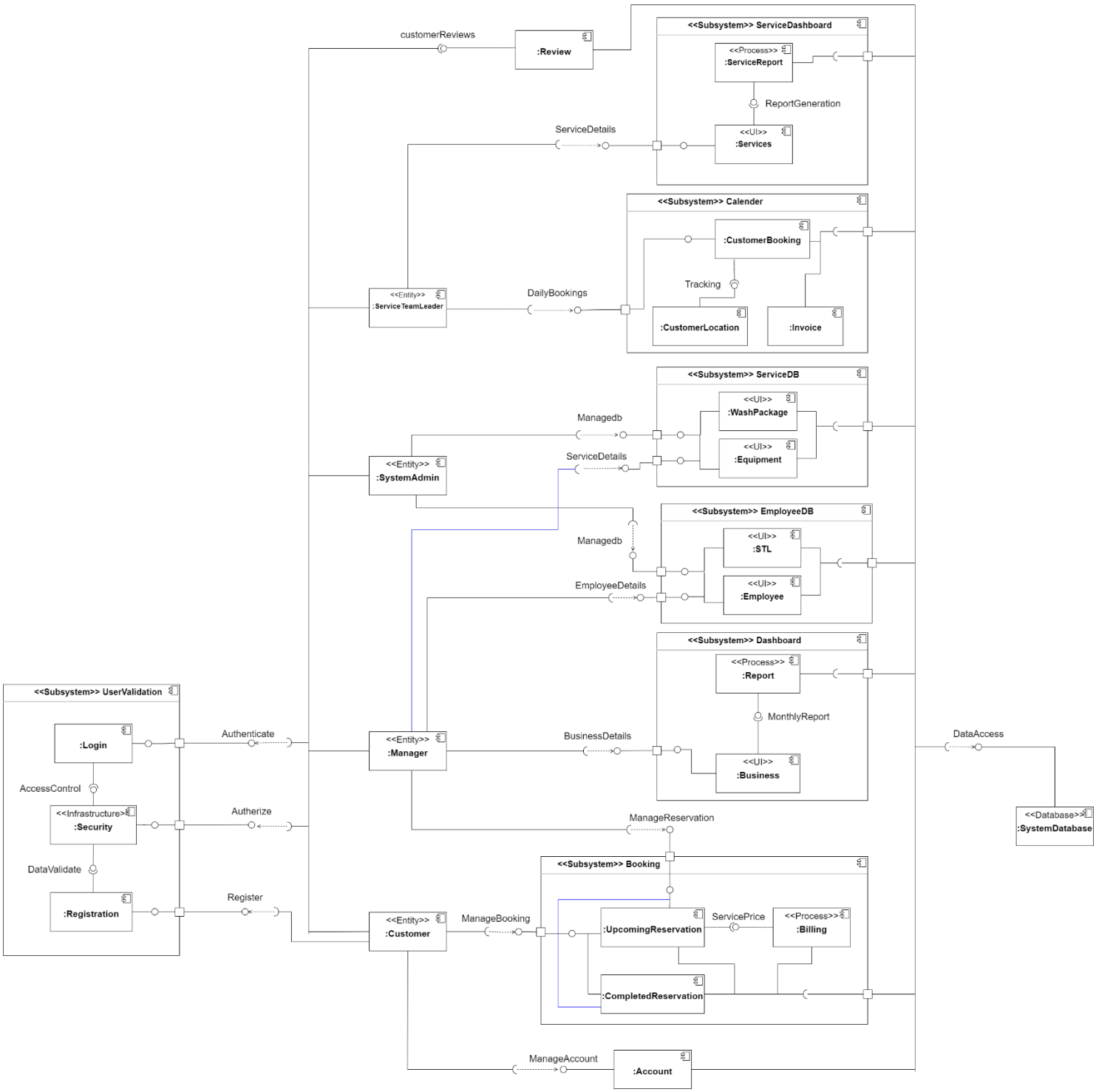
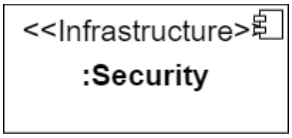
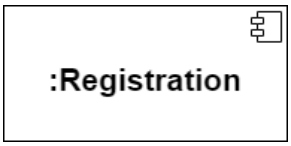
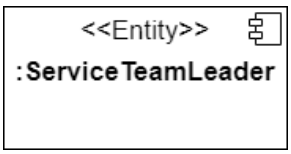
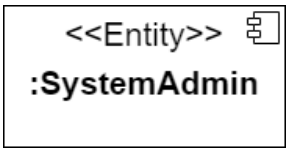
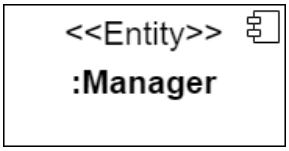
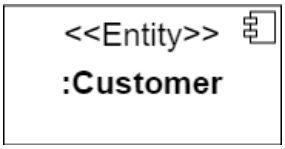
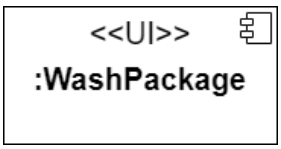
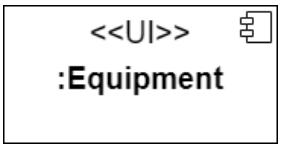
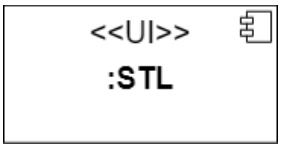
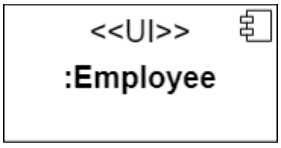
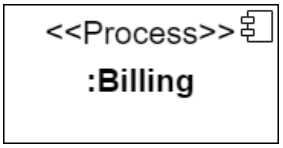


Figure 8: System Architecture - Derived Component Diagram

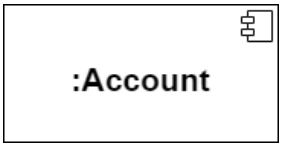
7.2 Components and their responsibilities

	Has the functionalities needed for user login
	Provides security functionalities such as user validation, email verification, user authorization etc.
	Has the functionalities needed for customer registration
	Entity component which provides the functionalities of a service team leader
	Entity component which provides the functionalities of the system admin
	Entity component which provides the functionalities of the manager
	Entity component which provides the functionalities of a customer
	Provides storing and retrieving customer reviews functionalities
	Analyzes past reservations of a service team leader and generates reports
	UI component which presents the service reports of a service team leader graphically

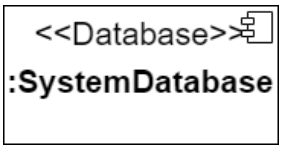
	Provides assigned reservation details of the current day for a service team leader
	Tracks customer location with the use of Google maps API
	Generates and sends an invoice to the customer for a particular service
	UI component to manage wash package details in the system
	UI component to manage details of servicing equipment and cleaning consumables
	UI component to manage service team leader details in the system
	UI component to manage employee details in the system
	Analyzes revenue of the reservations within the past month and generates business reports
	UI component which presents the business reports graphically
	Contains functionalities for the customer and manager to manage the upcoming reservations such as rescheduling, cancelling, assigning service team leaders etc.
	Contains functionalities for the customer and manager to view details of past reservations



Calculates the service price of a reservation



Provides functionalities for a customer to manage their account



Transfers data to and from the database for the relevant queries

8. System Design

8.1 Class Diagram

Below are the classes [3] of the system, their attributes and methods and their interactions.

URL (for quality image): https://files.fm/f/f5awvkmj8

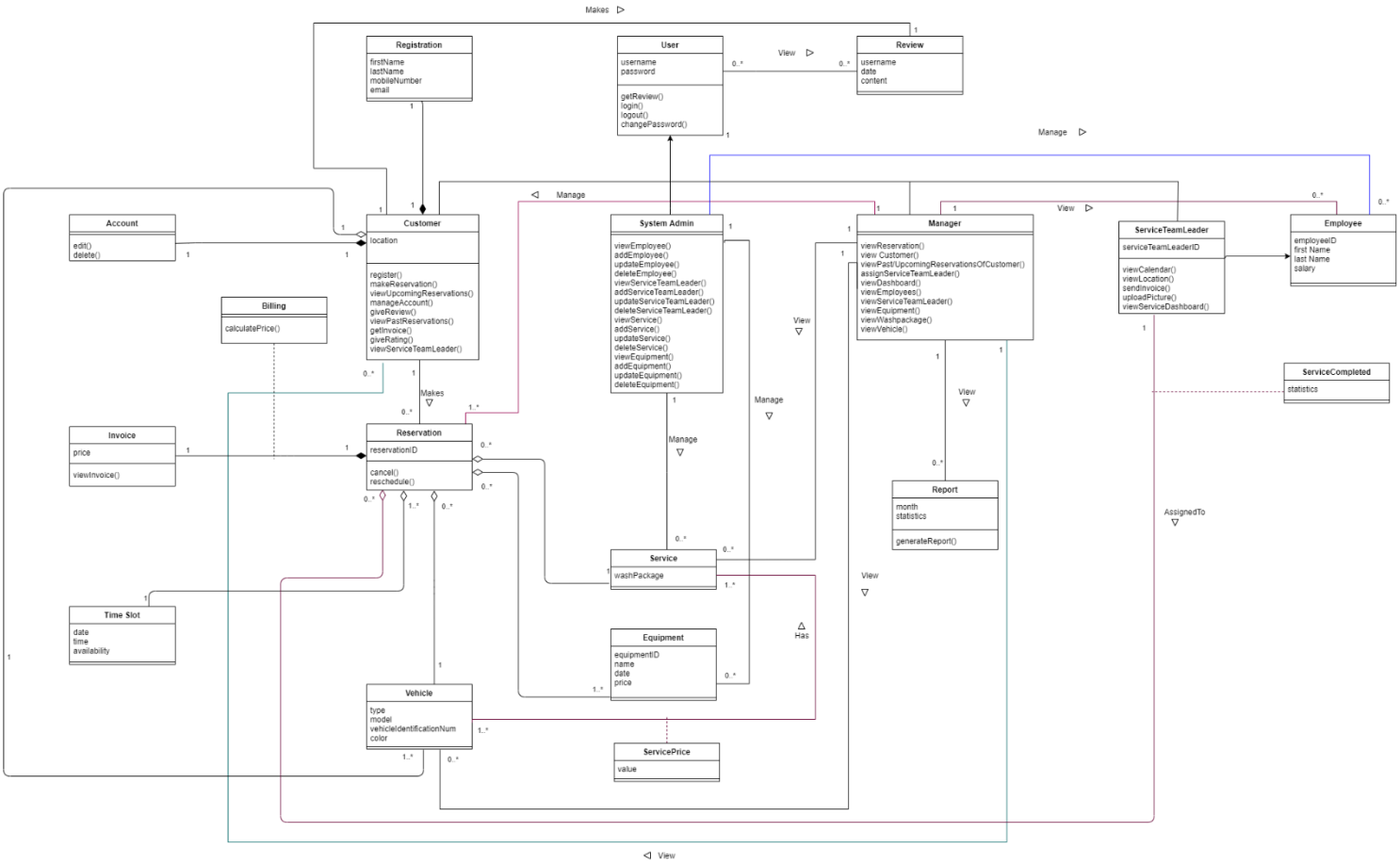


Figure 9: Class Diagram

8.2 EER Diagram

URL (for quality image): https://files.fm/f/eerpp2n7y

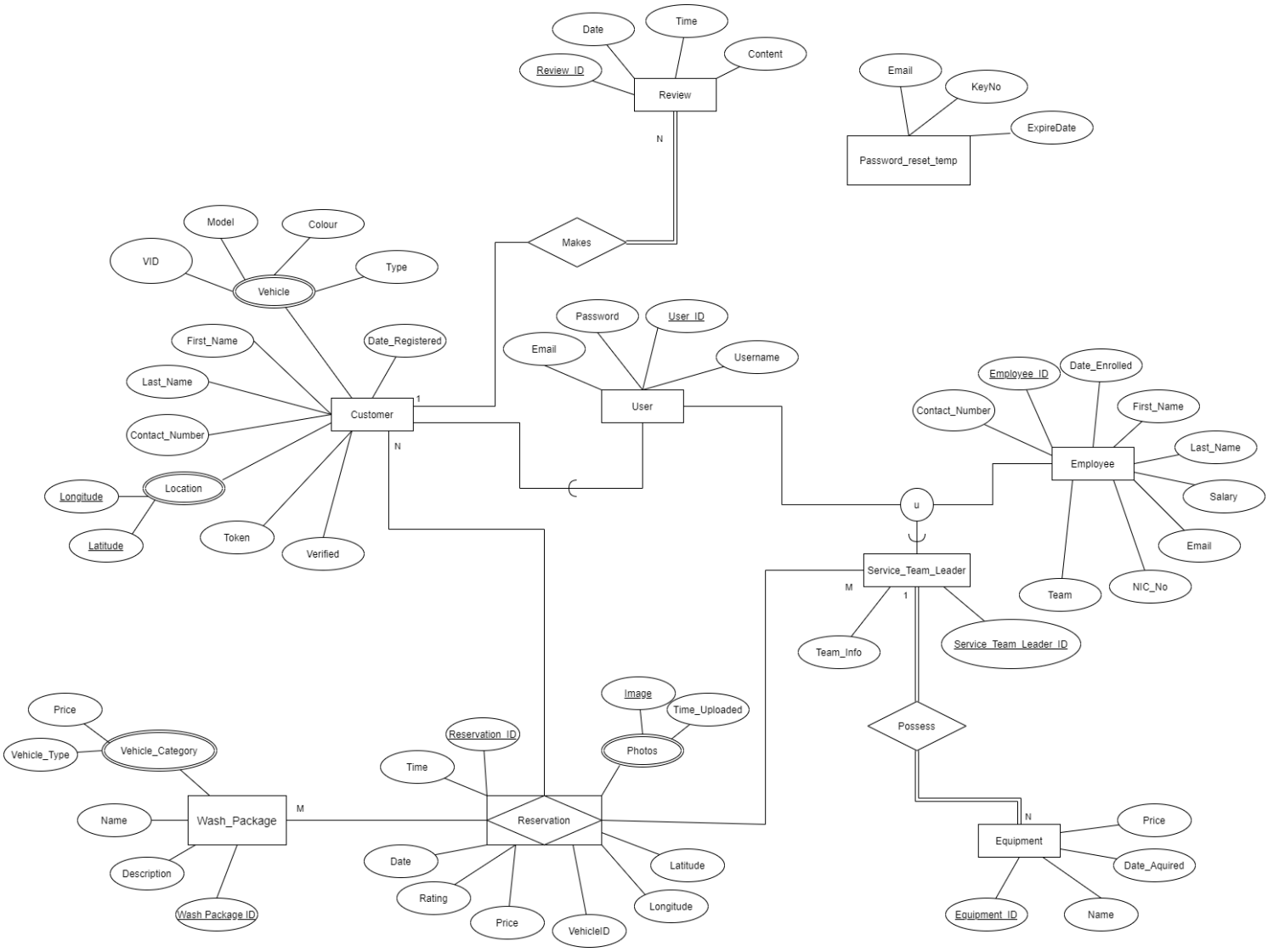
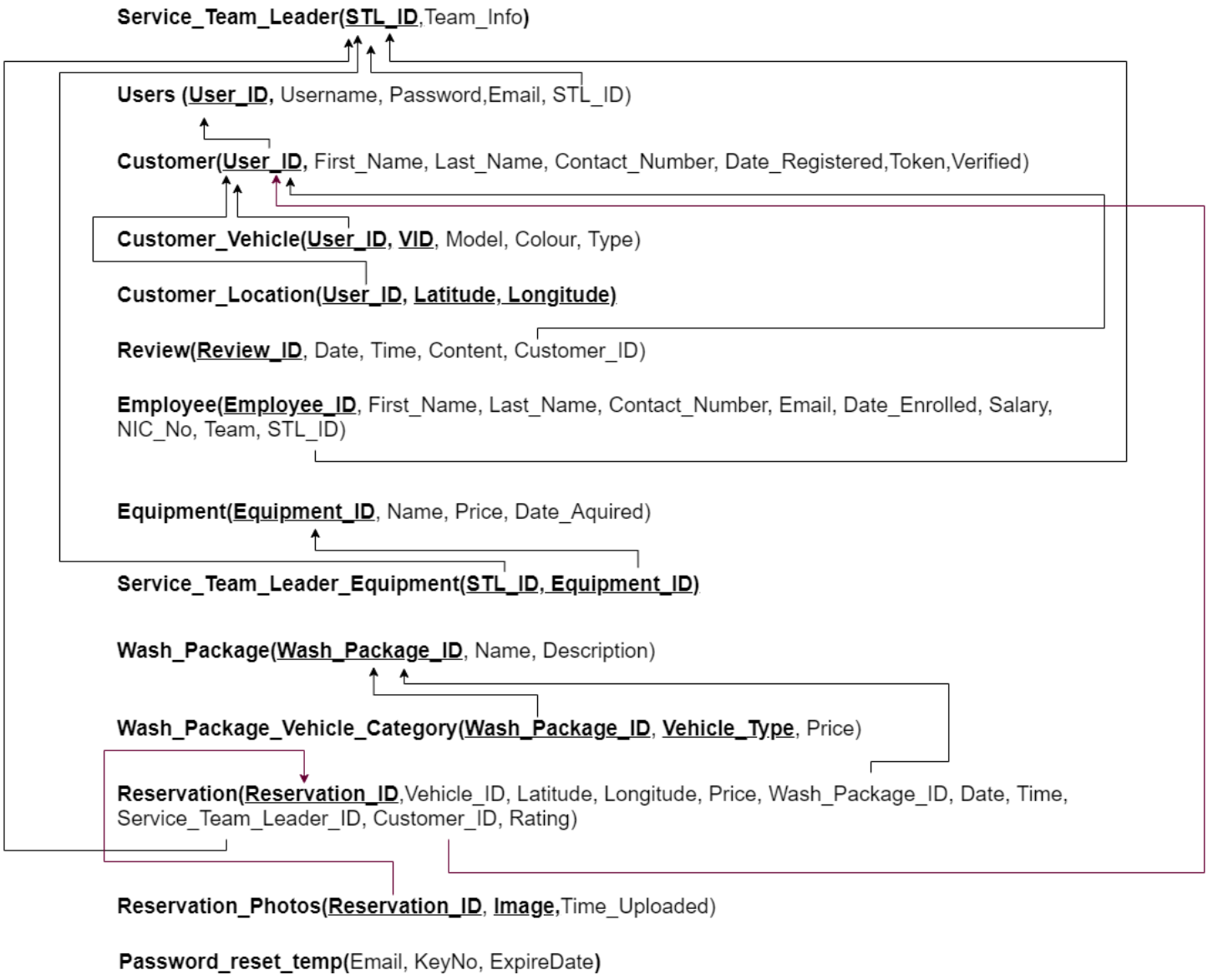


Figure 10: EER Diagram

8.3 EER Mapping



Primary keys are indicated by underlining.

Foreign keys are indicated using arrows.

8.4 Sequence Diagrams

The following sequence diagrams are drawn to show the interactions between classes and objects and their sequences for the use cases.

Login

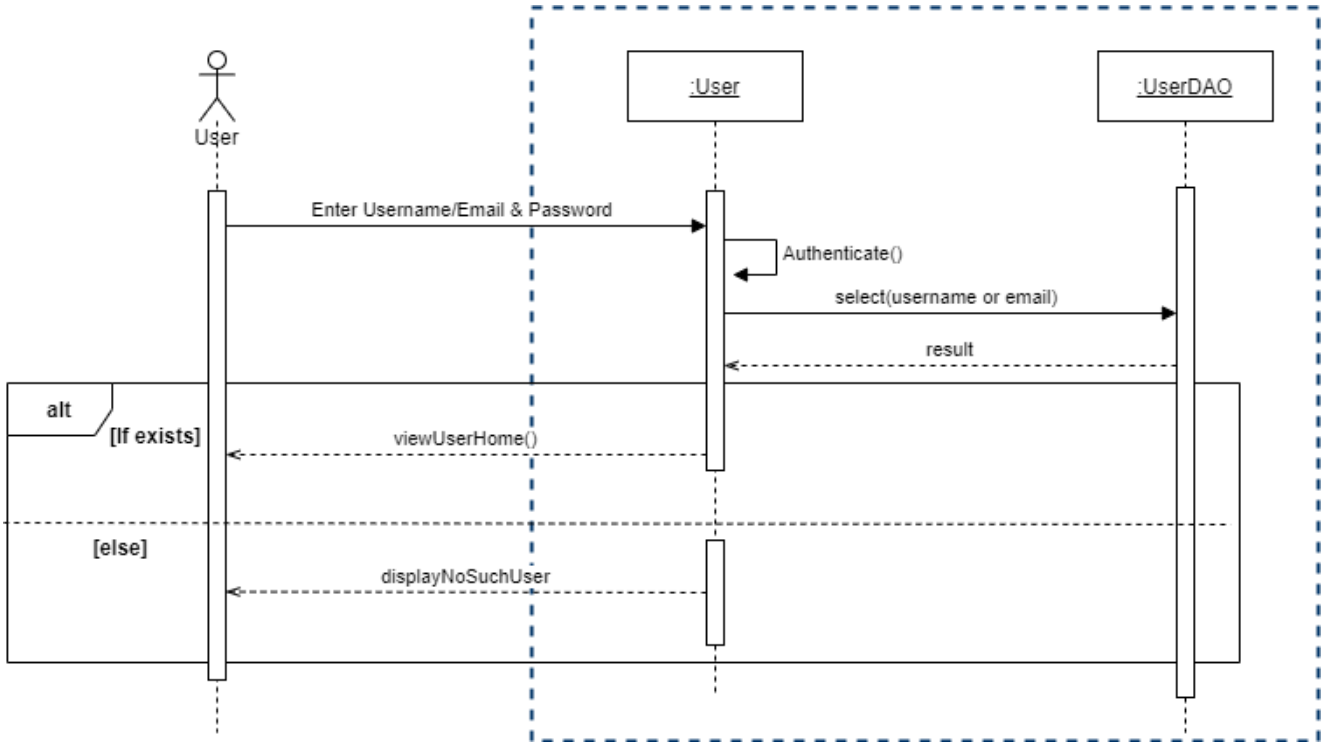


Figure 11: Sequence Diagram - Login

Password Reset

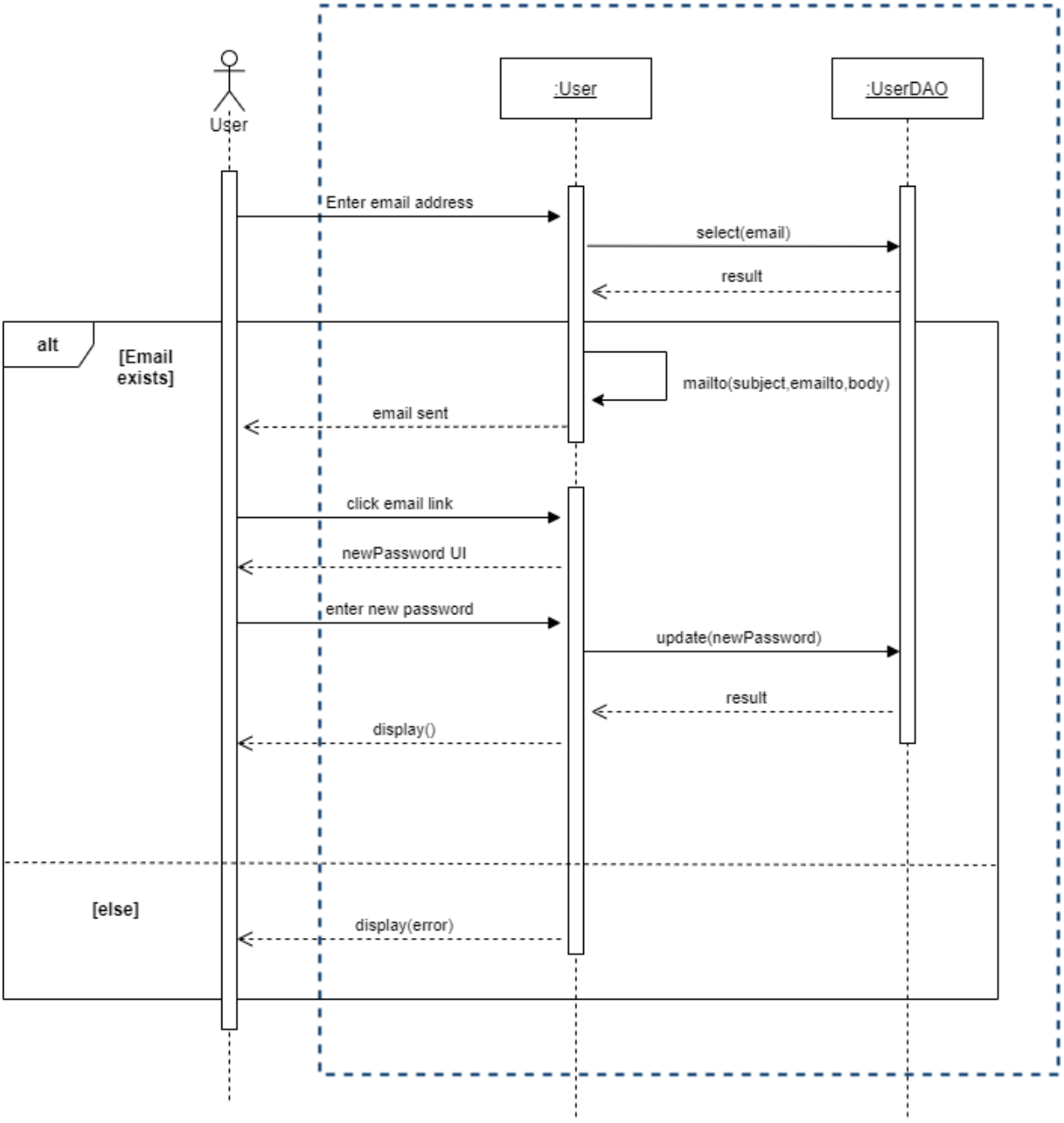


Figure 12: Sequence Diagram - Password Reset

View Reviews

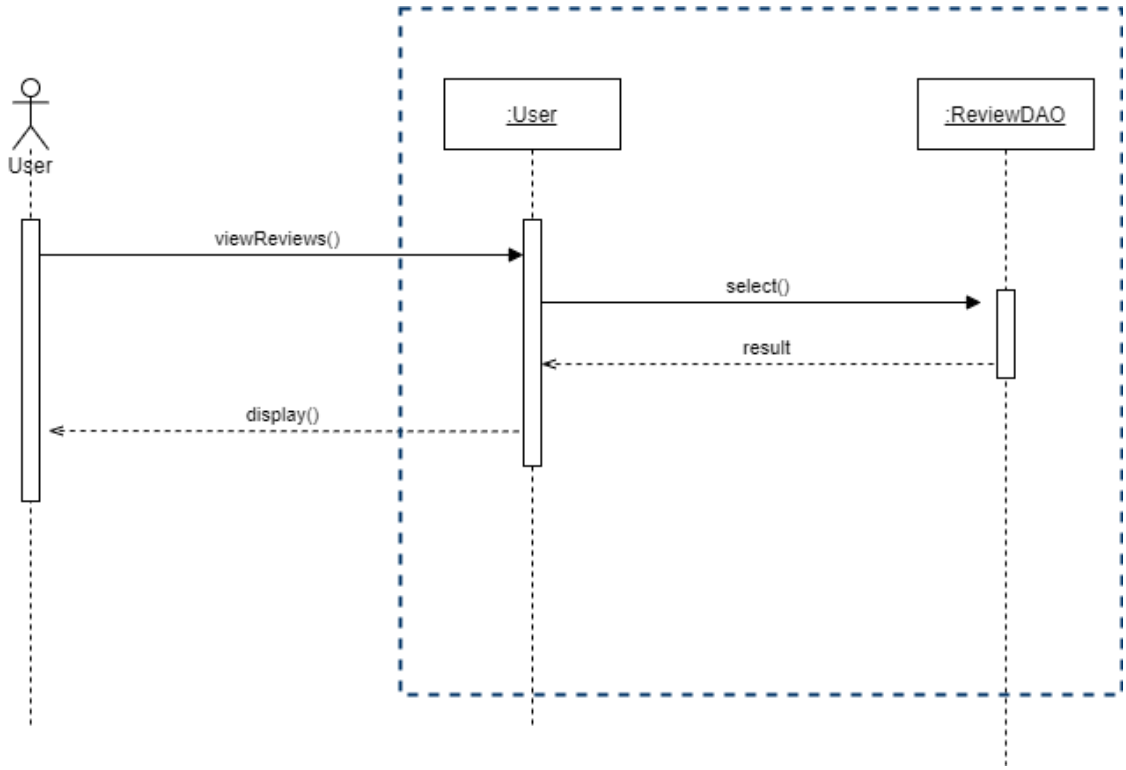


Figure 13: Sequence Diagram - View Reviews

Registration

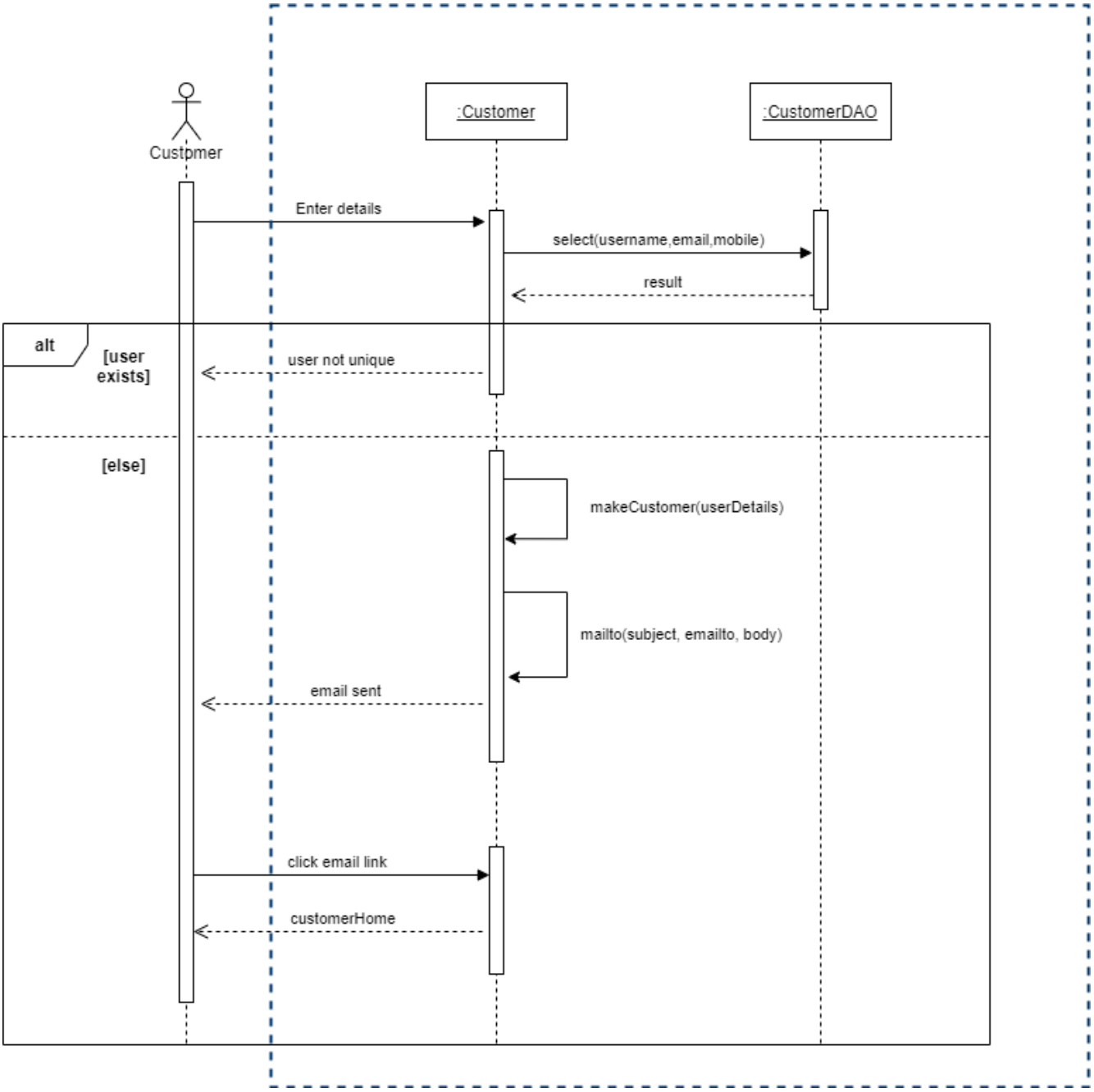


Figure 14: Sequence Diagram - Registration

Make Reservation

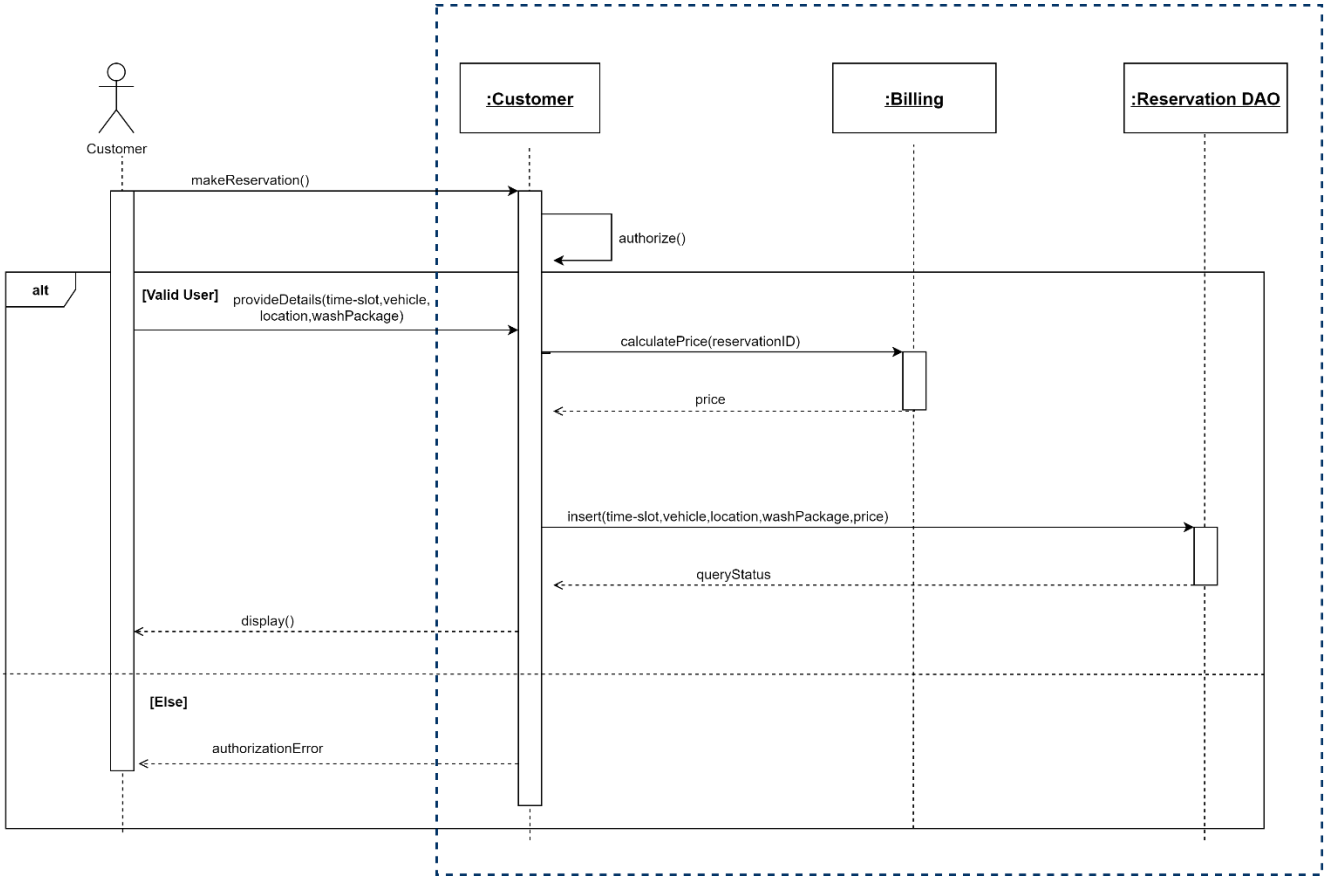


Figure 15: Sequence Diagram - Make Reservation

View and Modify Upcoming Reservations

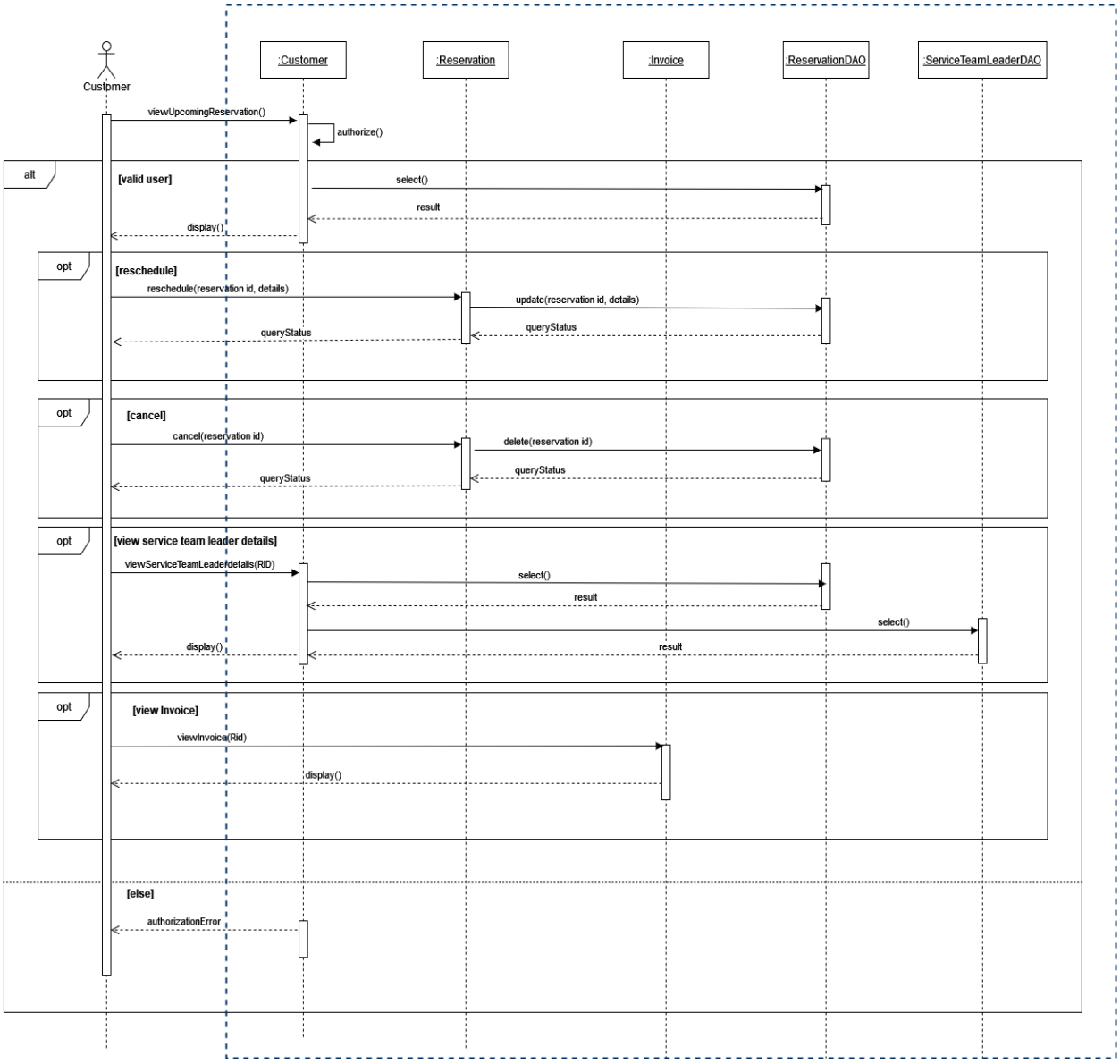


Figure 16: Sequence Diagram - View and Modify Upcoming Reservations

View and Rate Past Reservations

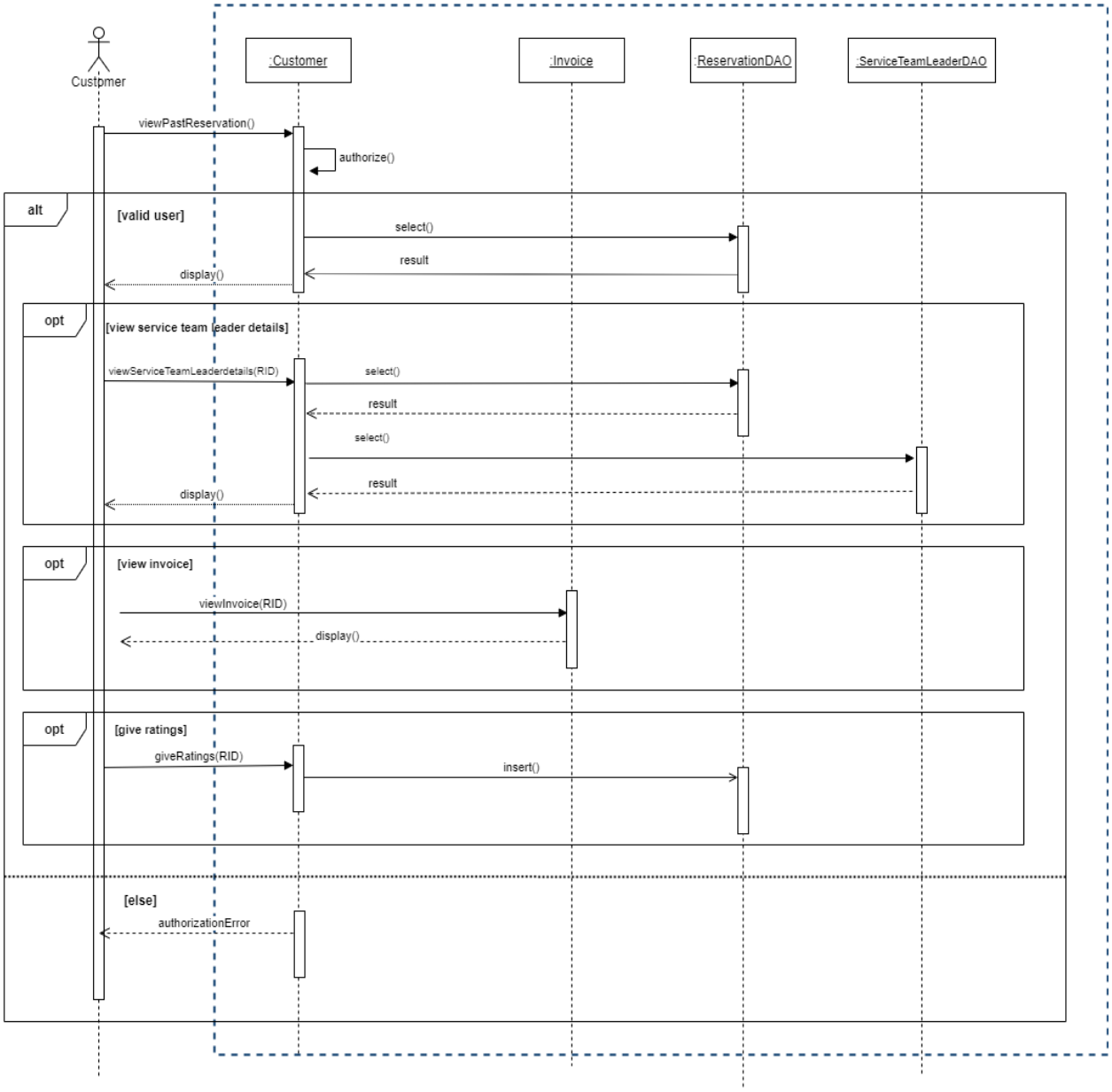


Figure 17: Sequence Diagram - View and Rate Past Reservations

Give Reviews

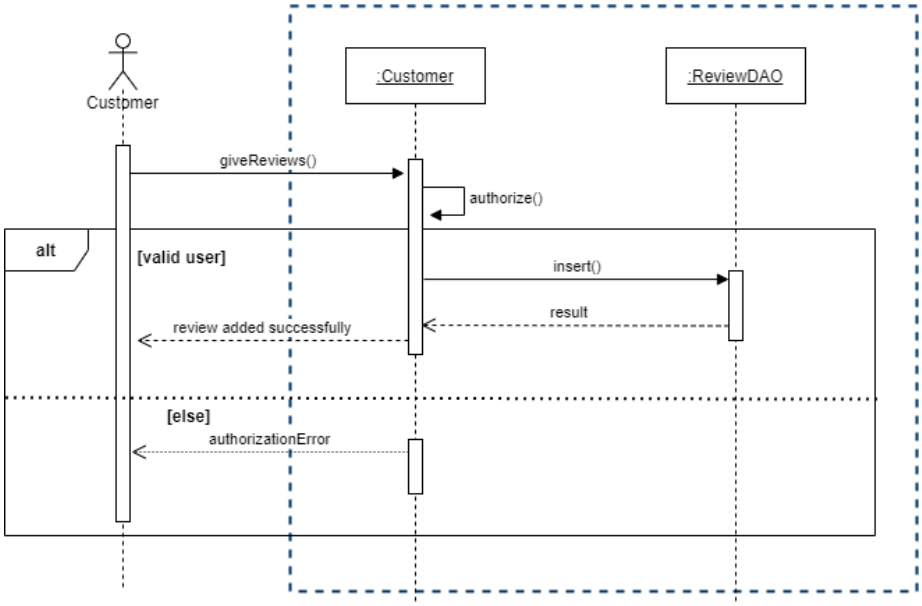


Figure 18: Sequence Diagram - Give Reviews

Edit Account Details

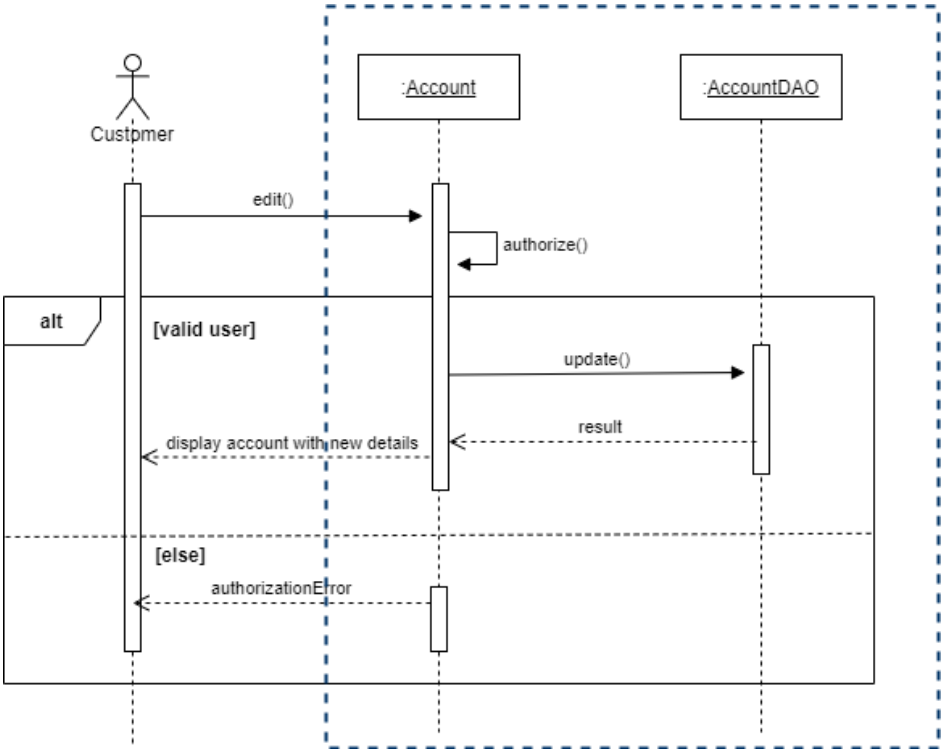


Figure 19: Sequence Diagram - Edit Account Details

Delete Account

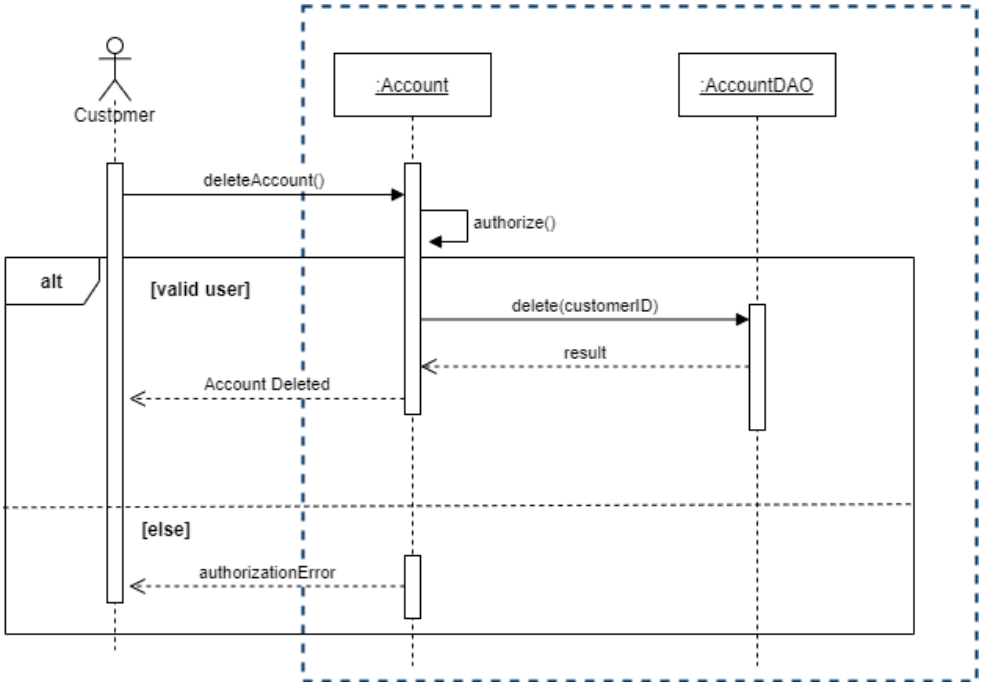


Figure 20: Sequence Diagram - Delete Account

View and Assign STL for Reservations

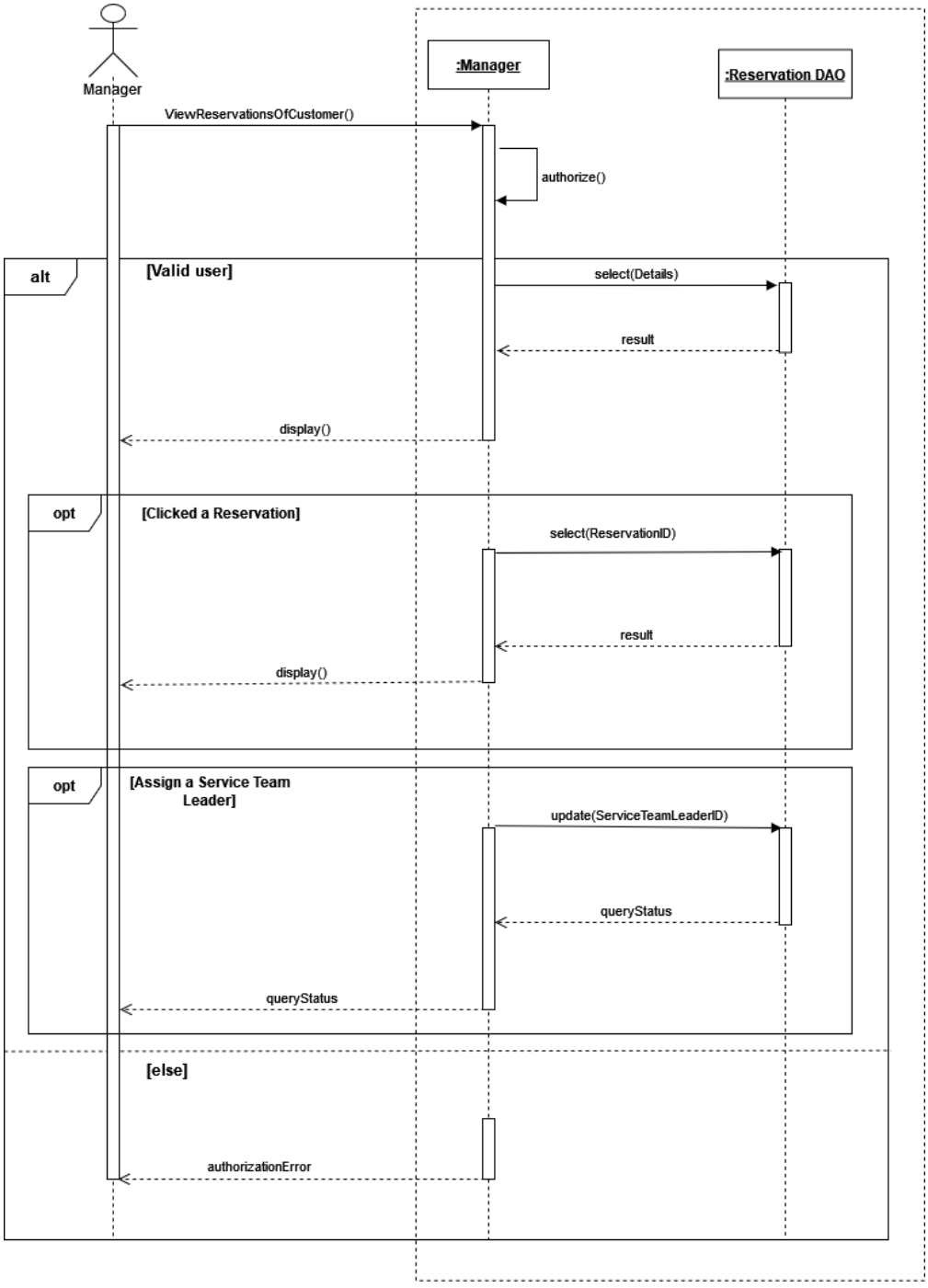


Figure 21: Sequence Diagram - View and Assign STL for Reservations

View Completed Reservations

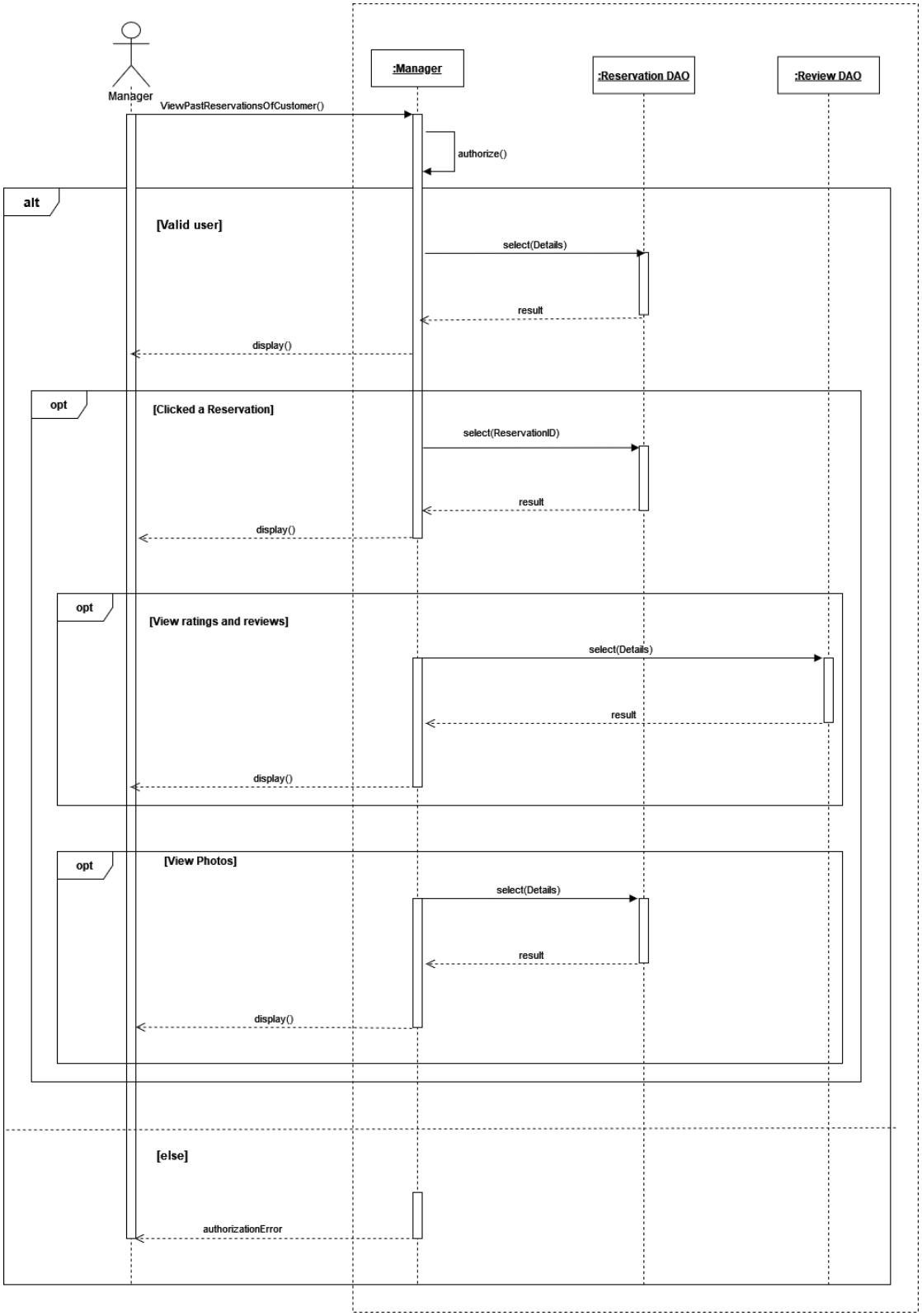


Figure 22: Sequence Diagram - View Completed Reservations

View Dashboard

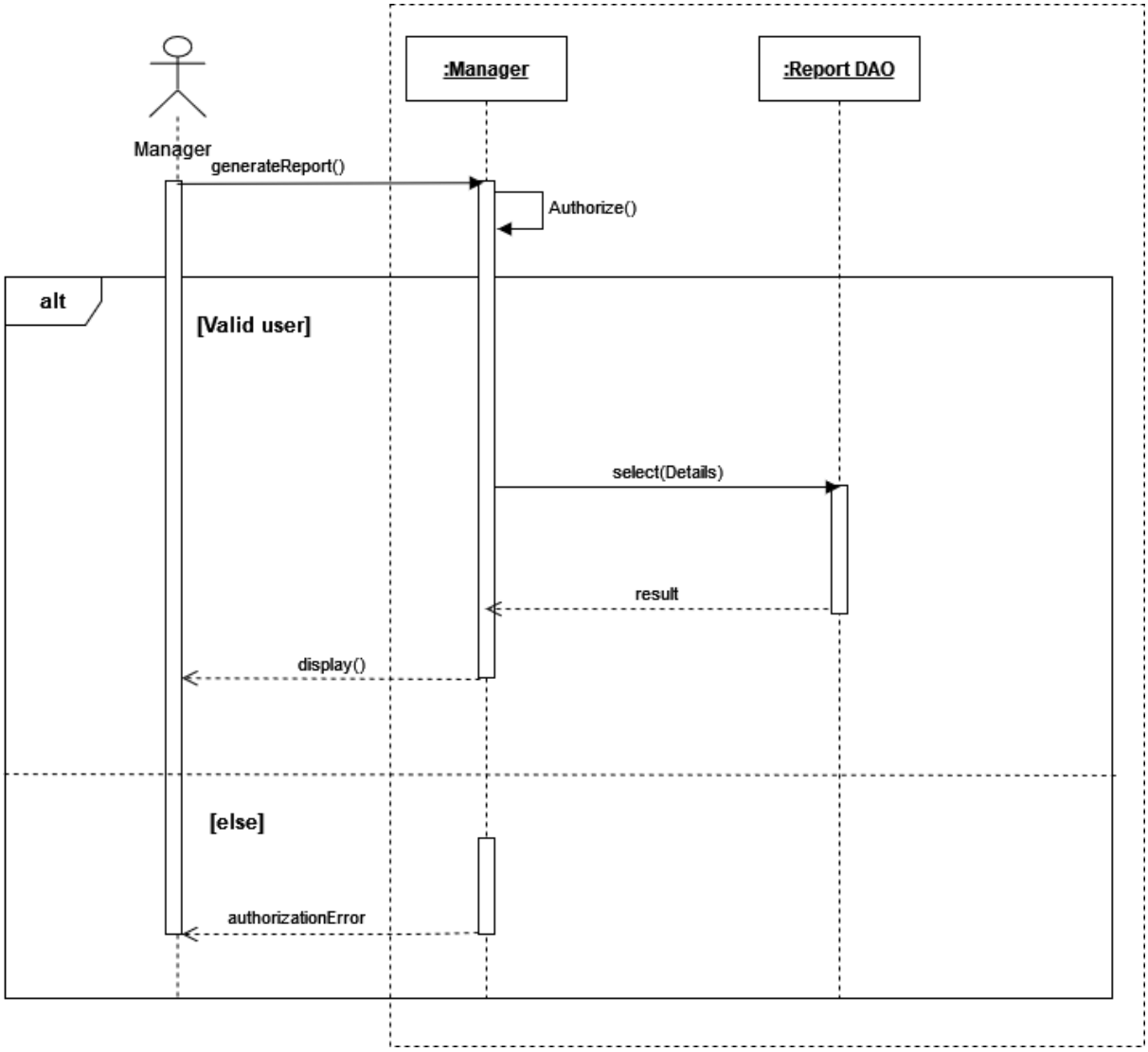


Figure 23: Sequence Diagram - View Dashboard

Add/Update/Delete Employee

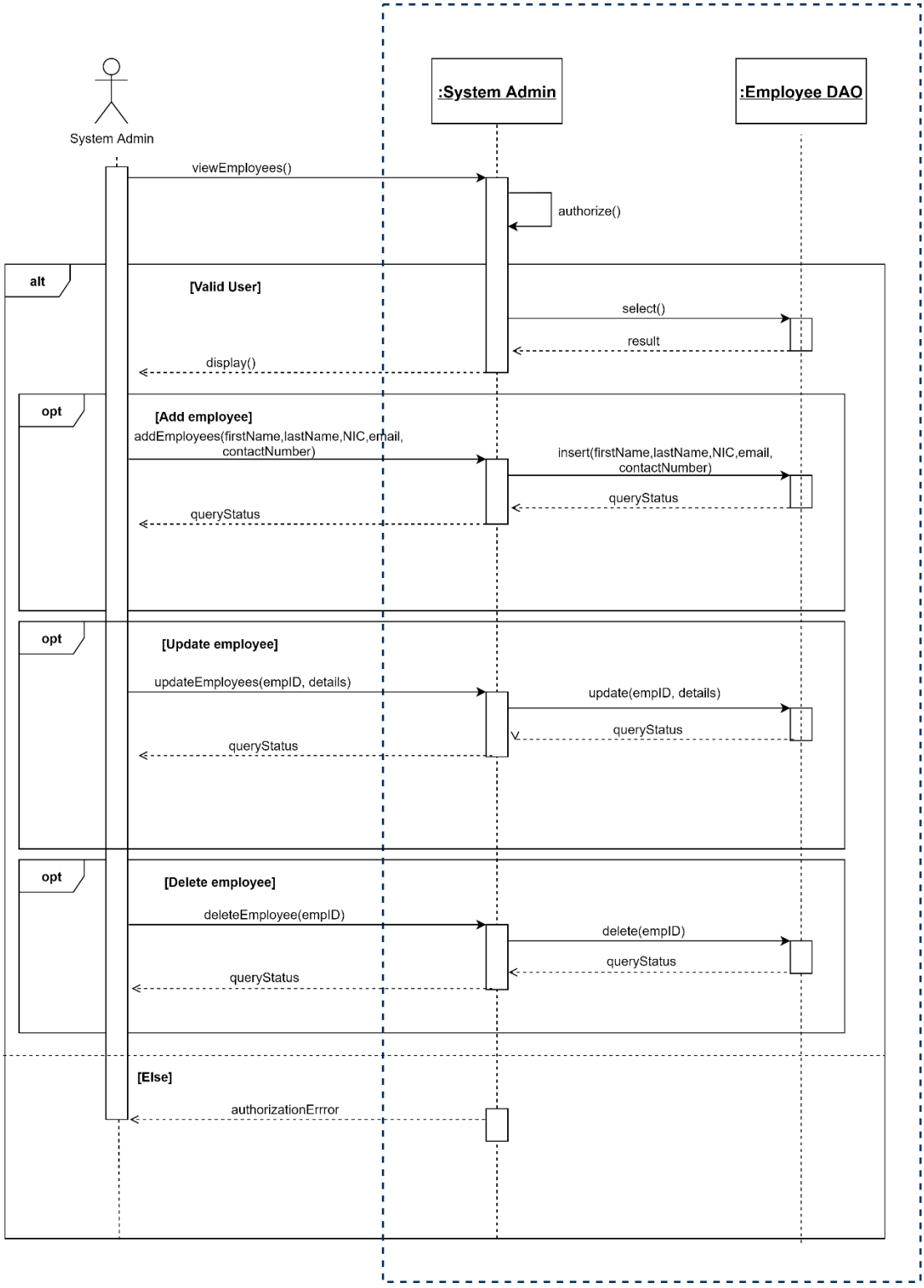


Figure 24: Sequence Diagram - Add/Update/Delete Employee

Add/Update/Delete Equipment

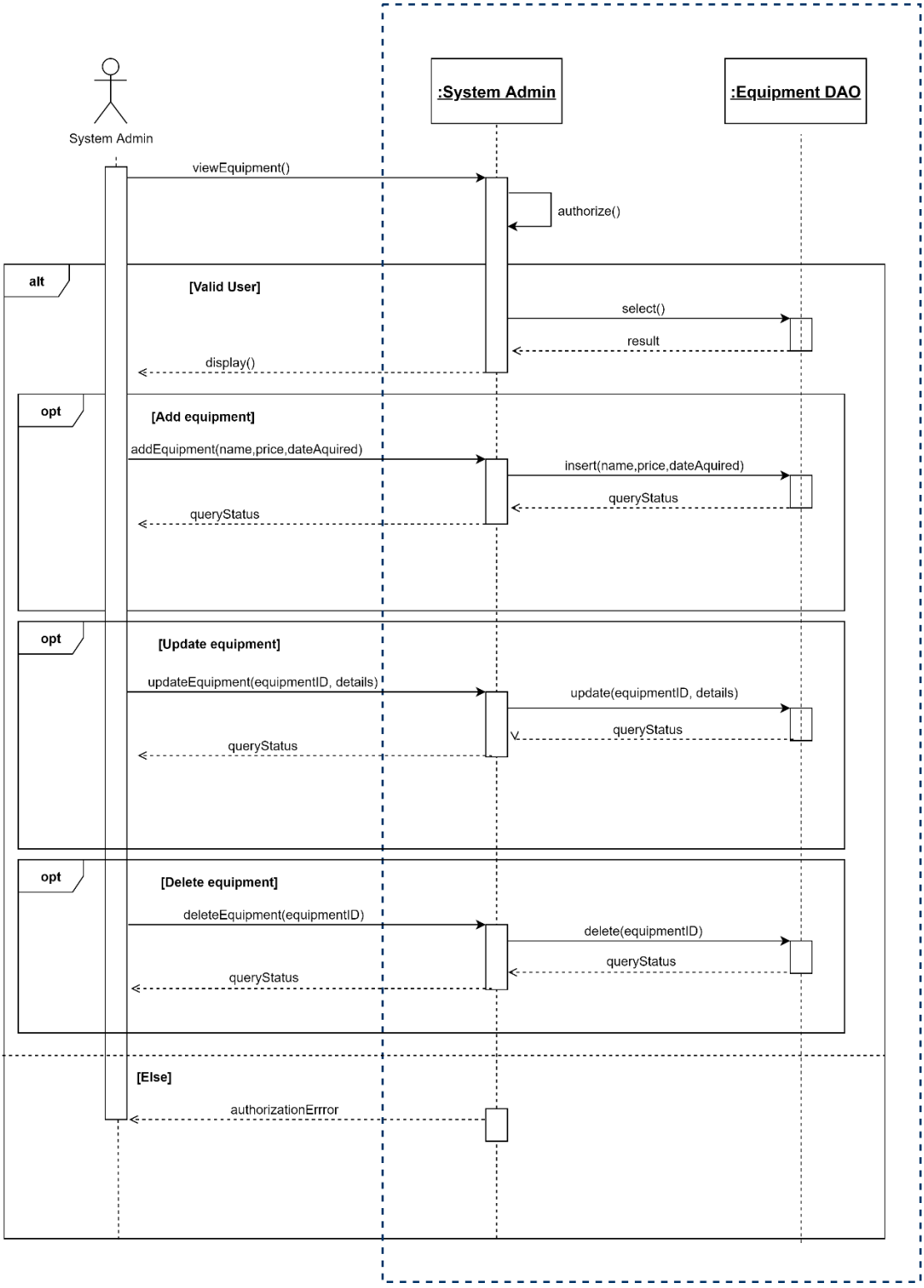


Figure 25: Sequence Diagram - Add/Update/Delete Equipment

Add/Update/Delete Service Details

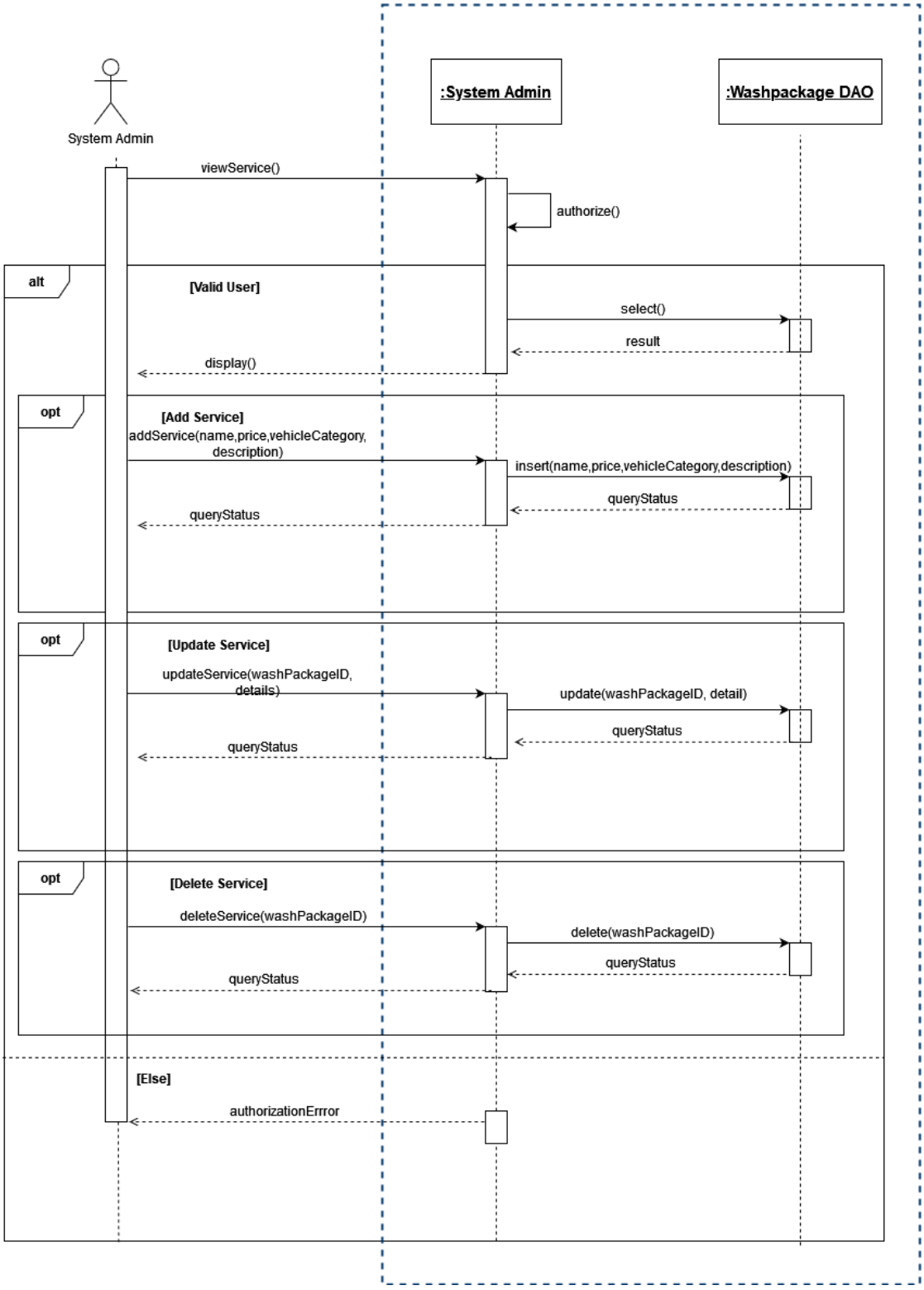


Figure 26: Sequence Diagram - Add/Update/Delete Service Details

Add/Update/Delete Service Team Leader

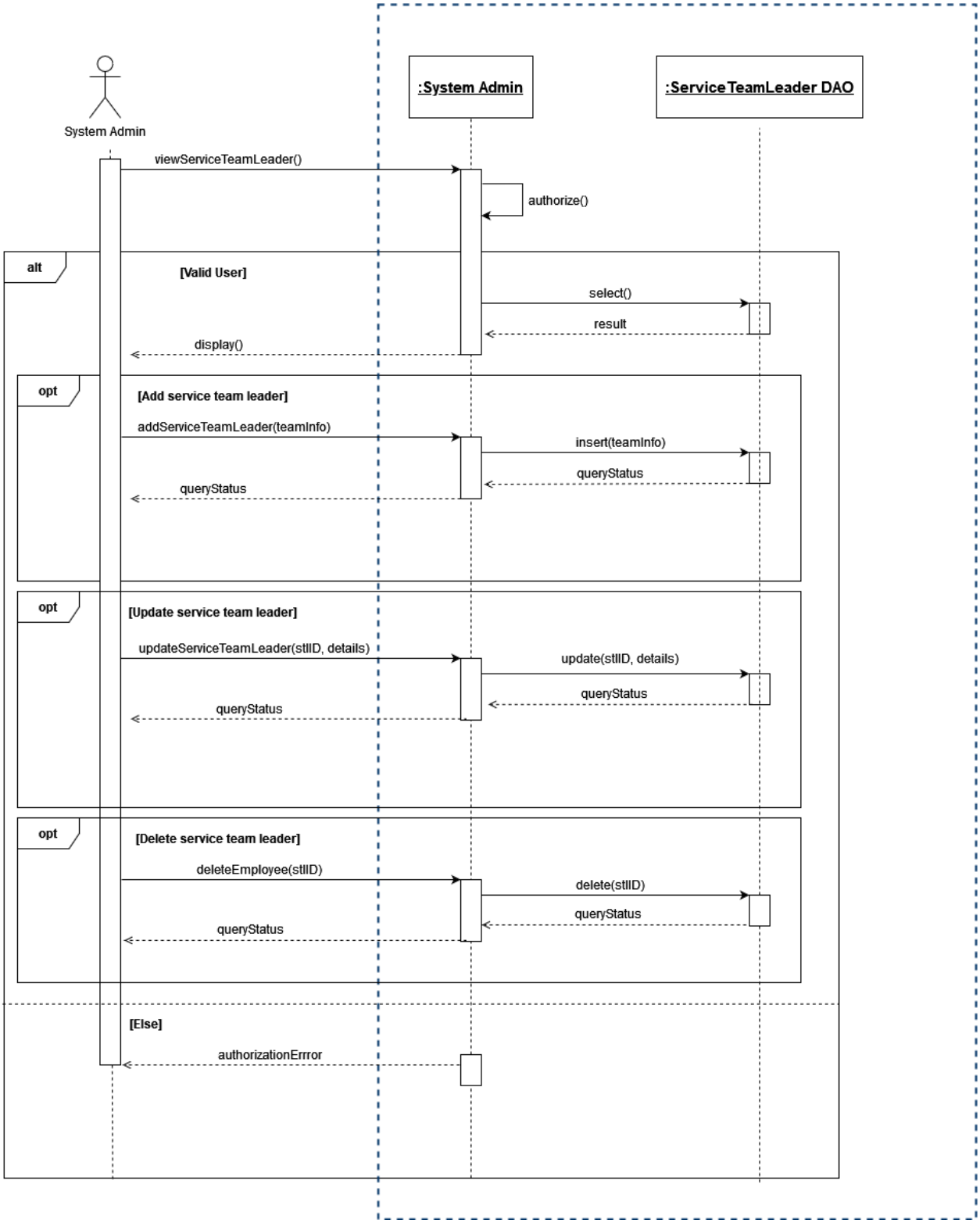


Figure 27: Sequence Diagram - Add/Update/Delete Service Team Leader

View/Search Employee

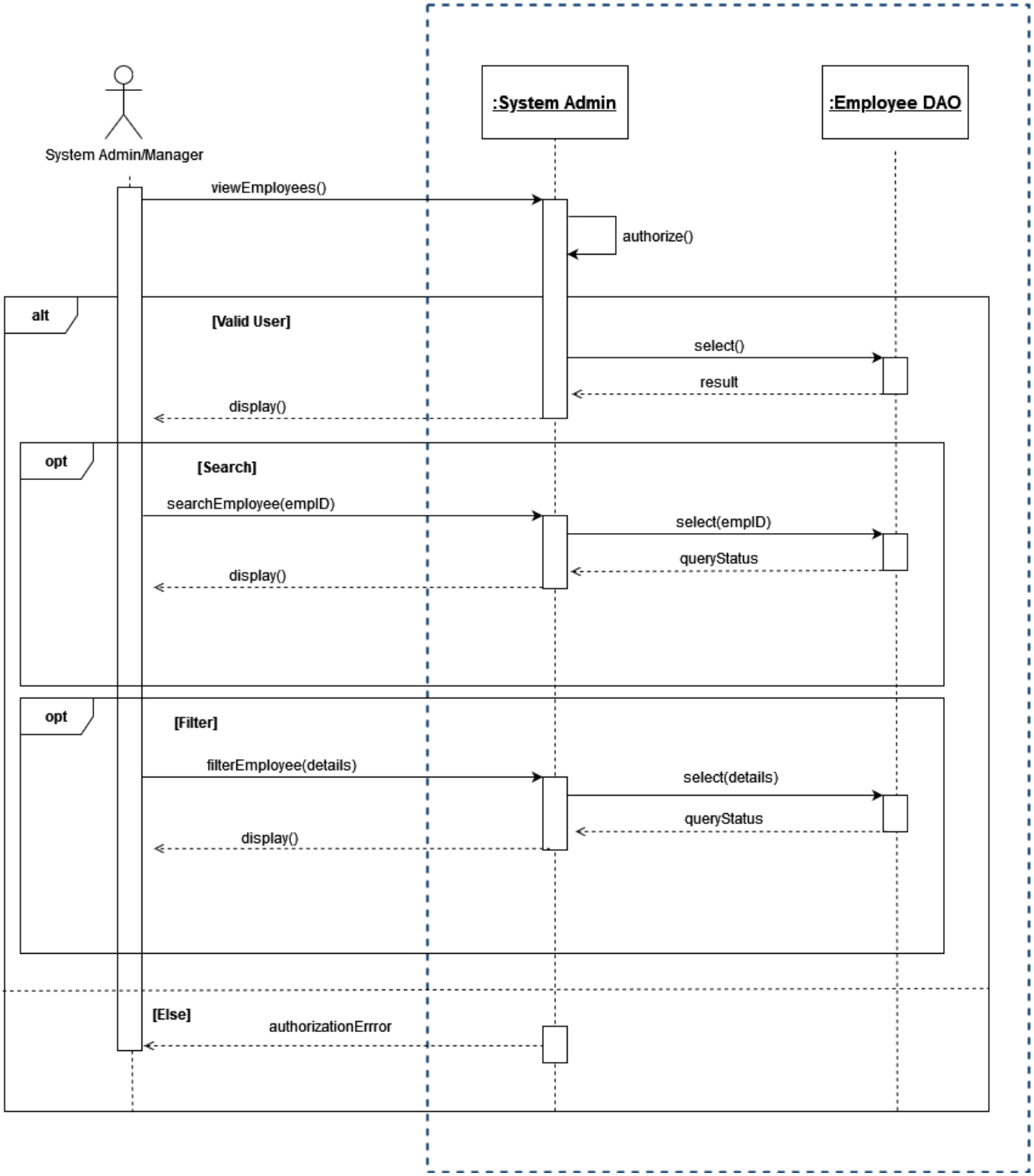


Figure 28: Sequence Diagram - View/Search Employee

View/Search Equipment

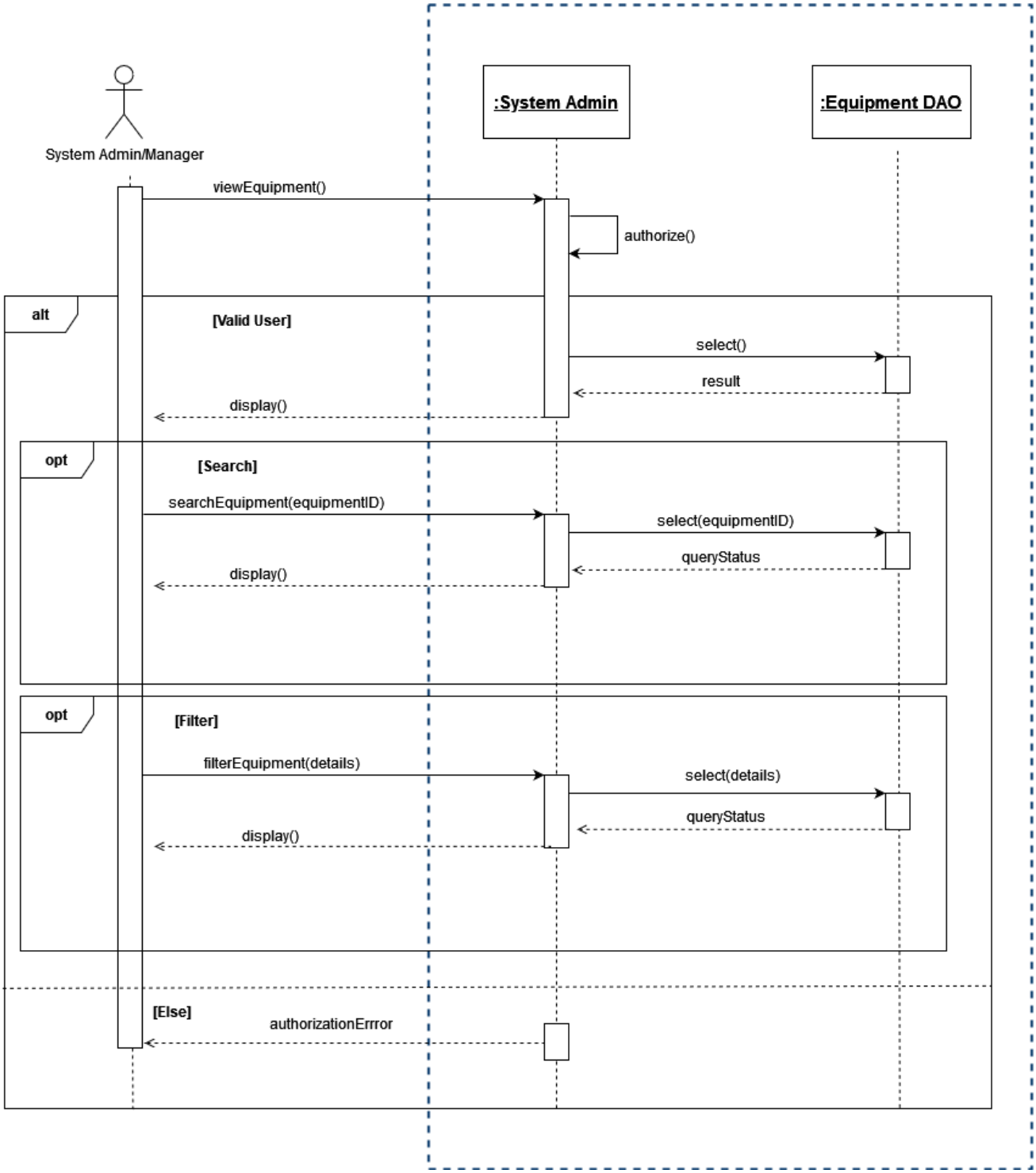


Figure 29: Sequence Diagram - View/Search Equipment

View/Search Service Details

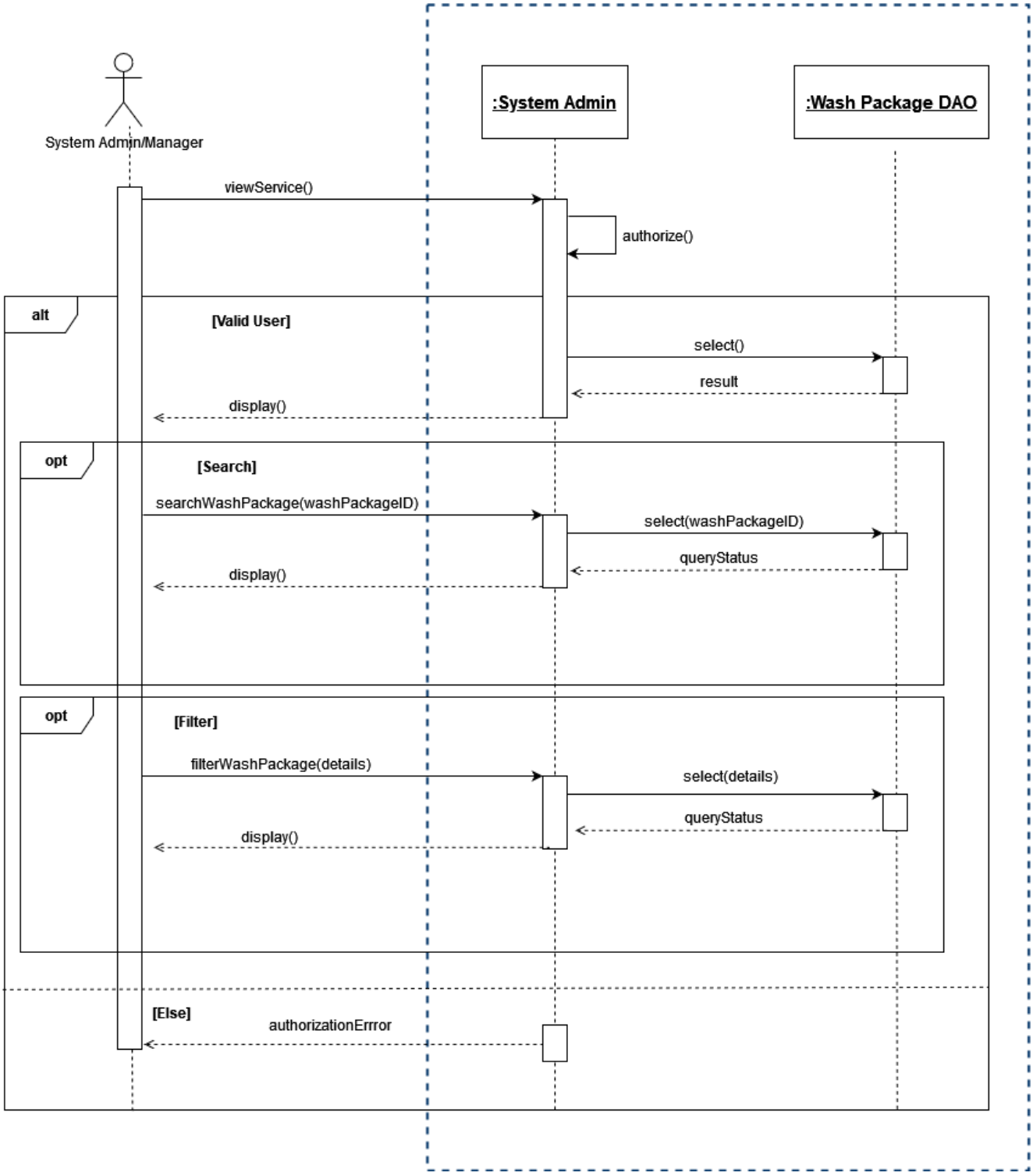


Figure 30: Sequence Diagram - View/Search Service Details

View/Search Service Team Leader

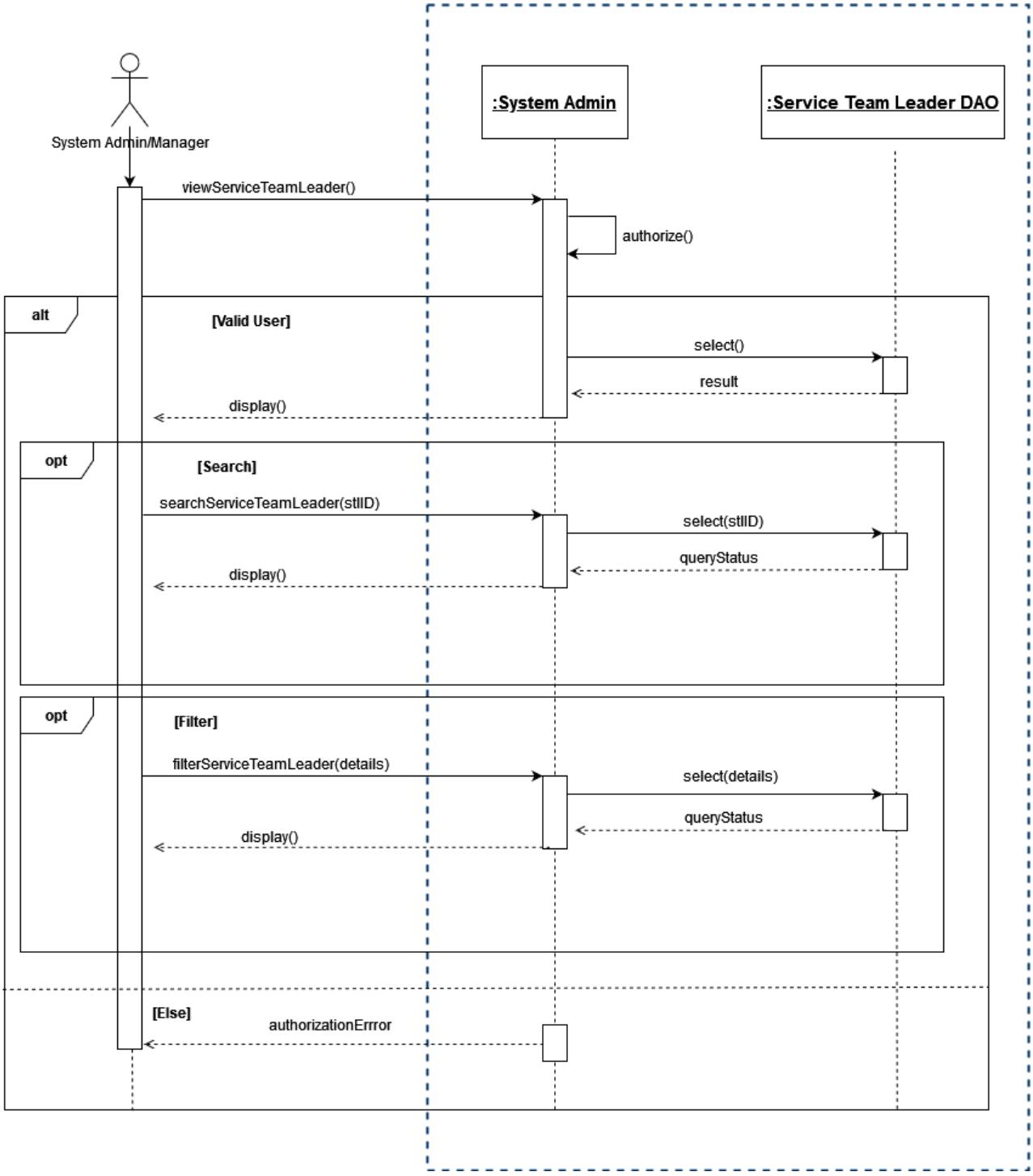


Figure 31: Sequence Diagram - View/Search Service Team Leader

View Service Dashboard

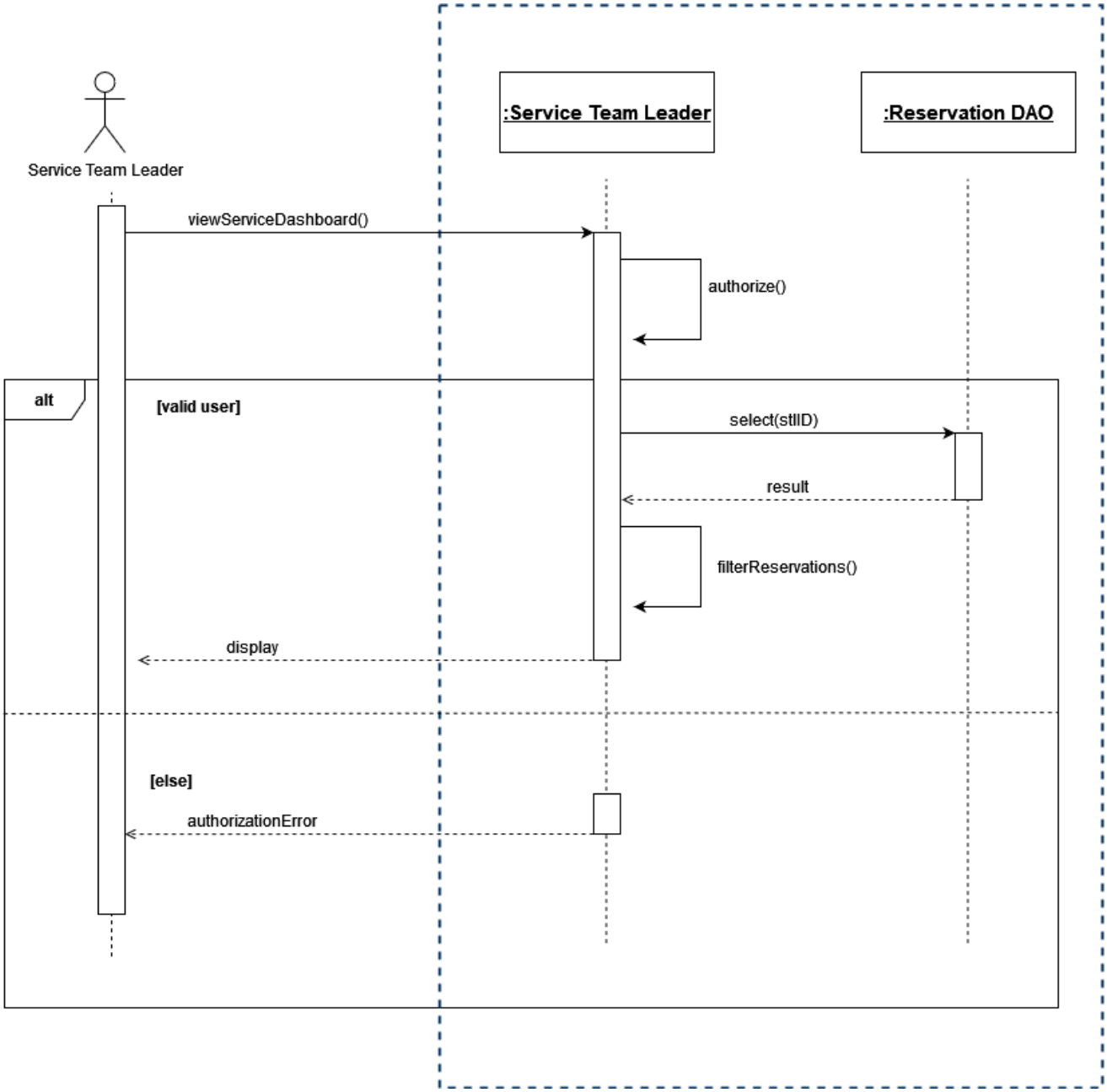


Figure 32: Sequence Diagram - View Service Dashboard

View Calendar

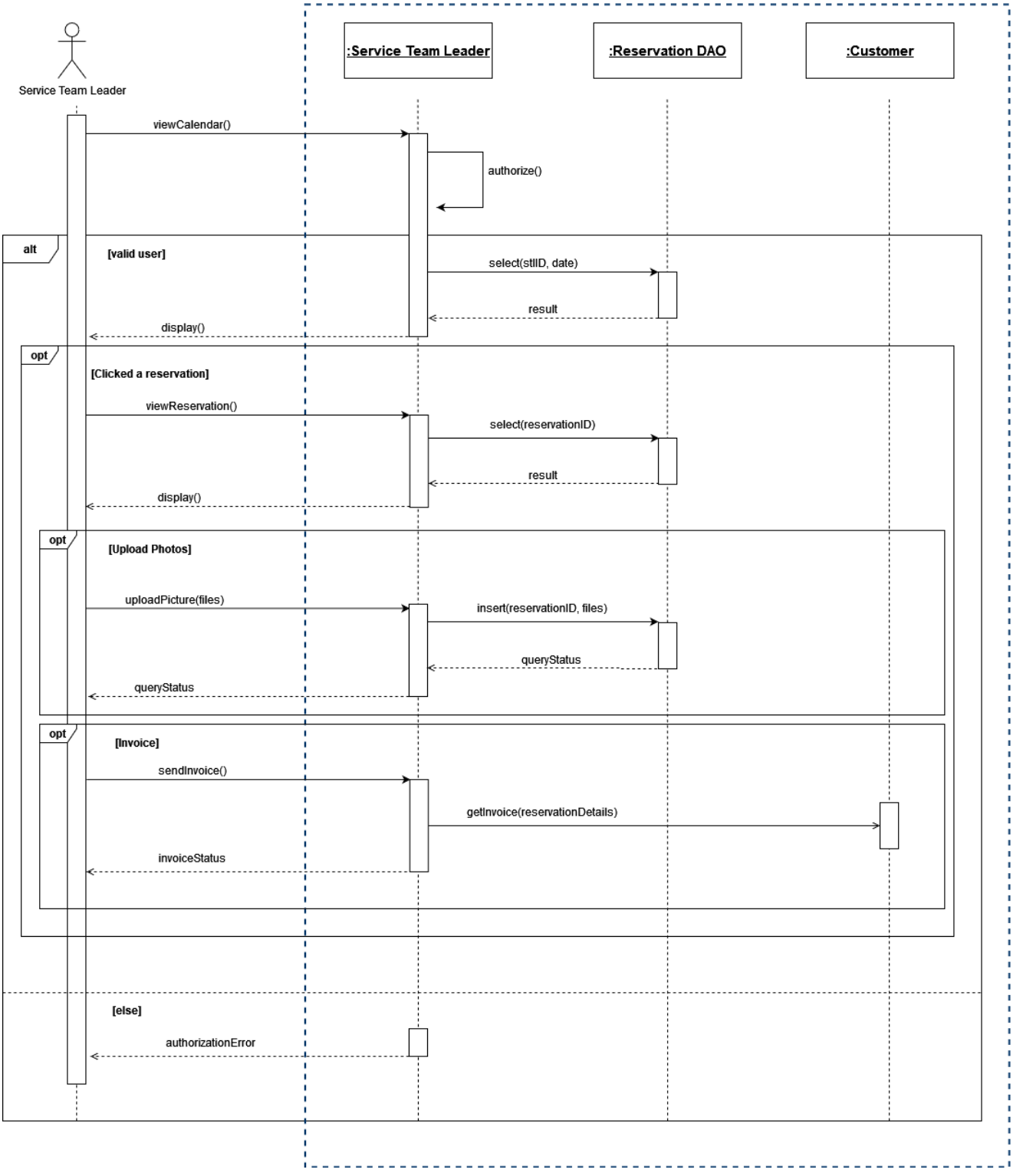


Figure 33: Sequence Diagram - View Calendar

8.5 Activity Diagrams

The following activity diagrams are drawn to show the flow of activities of the use cases.

User Login

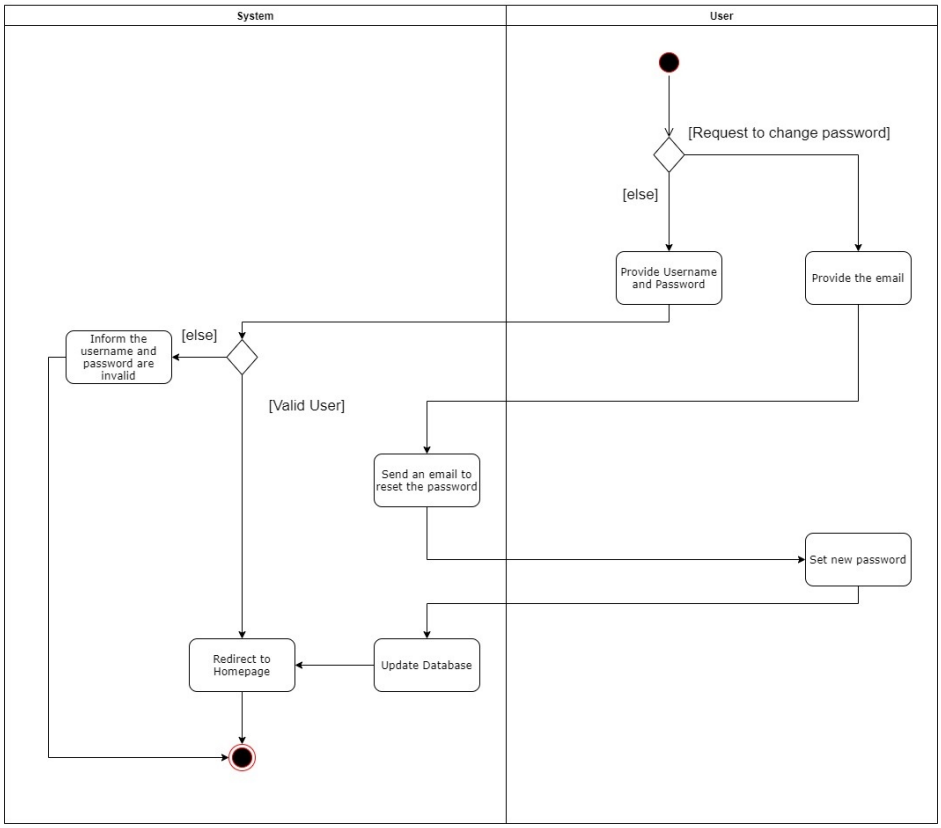


Figure 34: Activity Diagram - Login

View Reviews

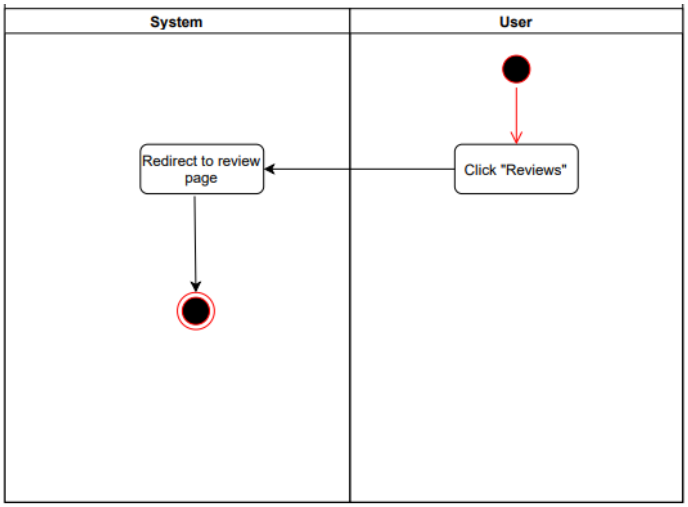


Figure 35: Activity Diagram - View Reviews

Registration

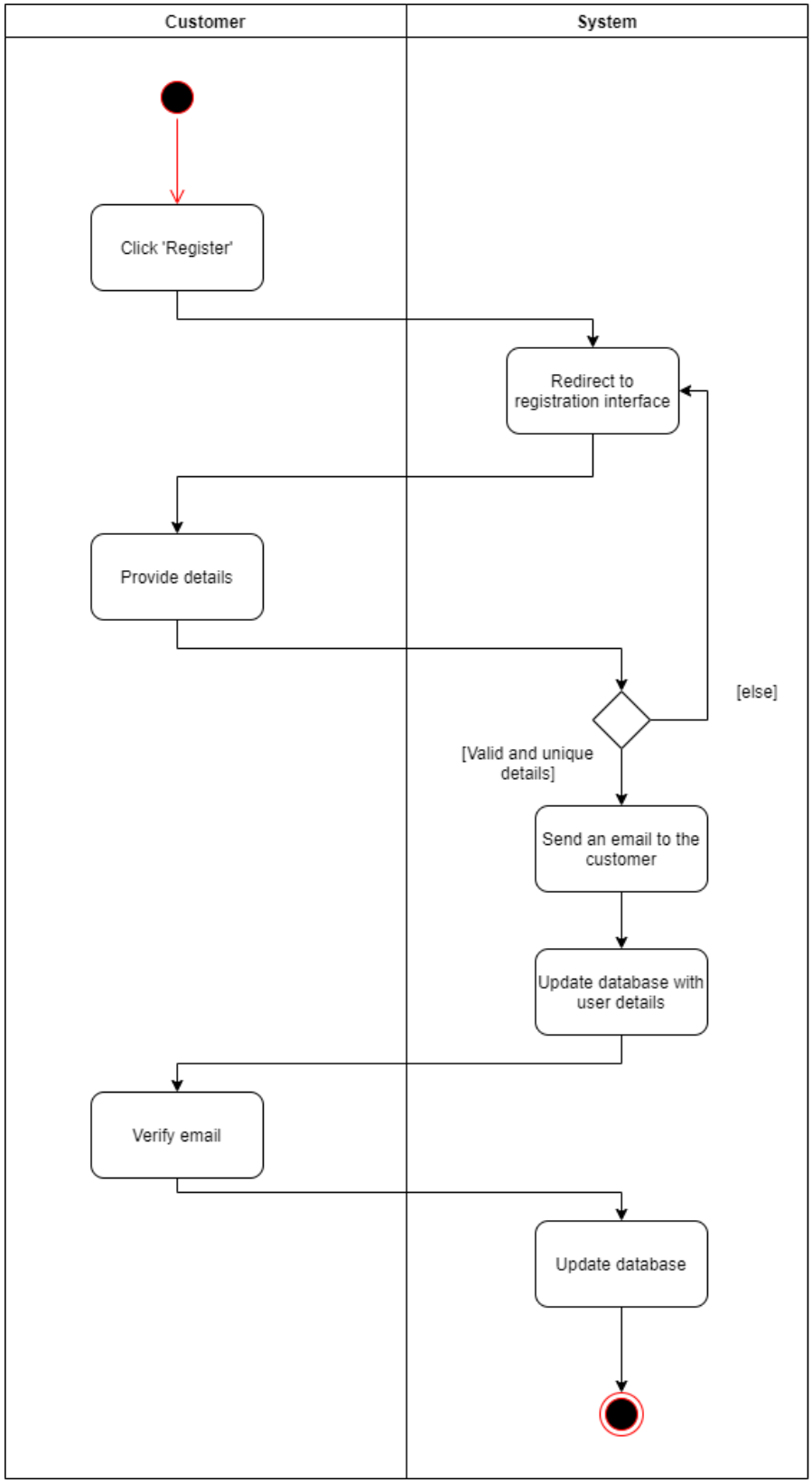


Figure 36: Activity Diagram - Register

Make Reservation

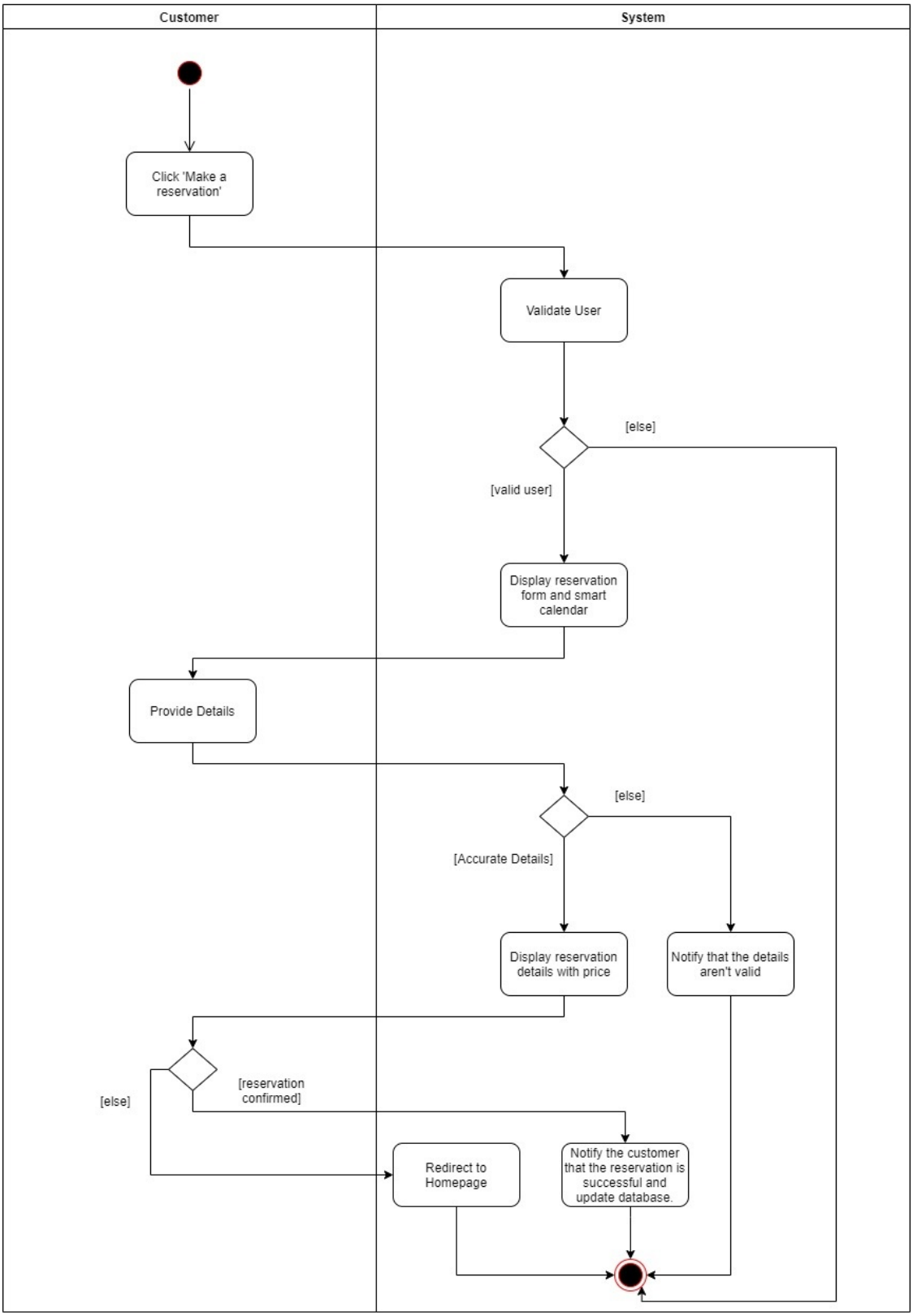


Figure 37: Activity Diagram - Make Reservation

View Upcoming Reservations

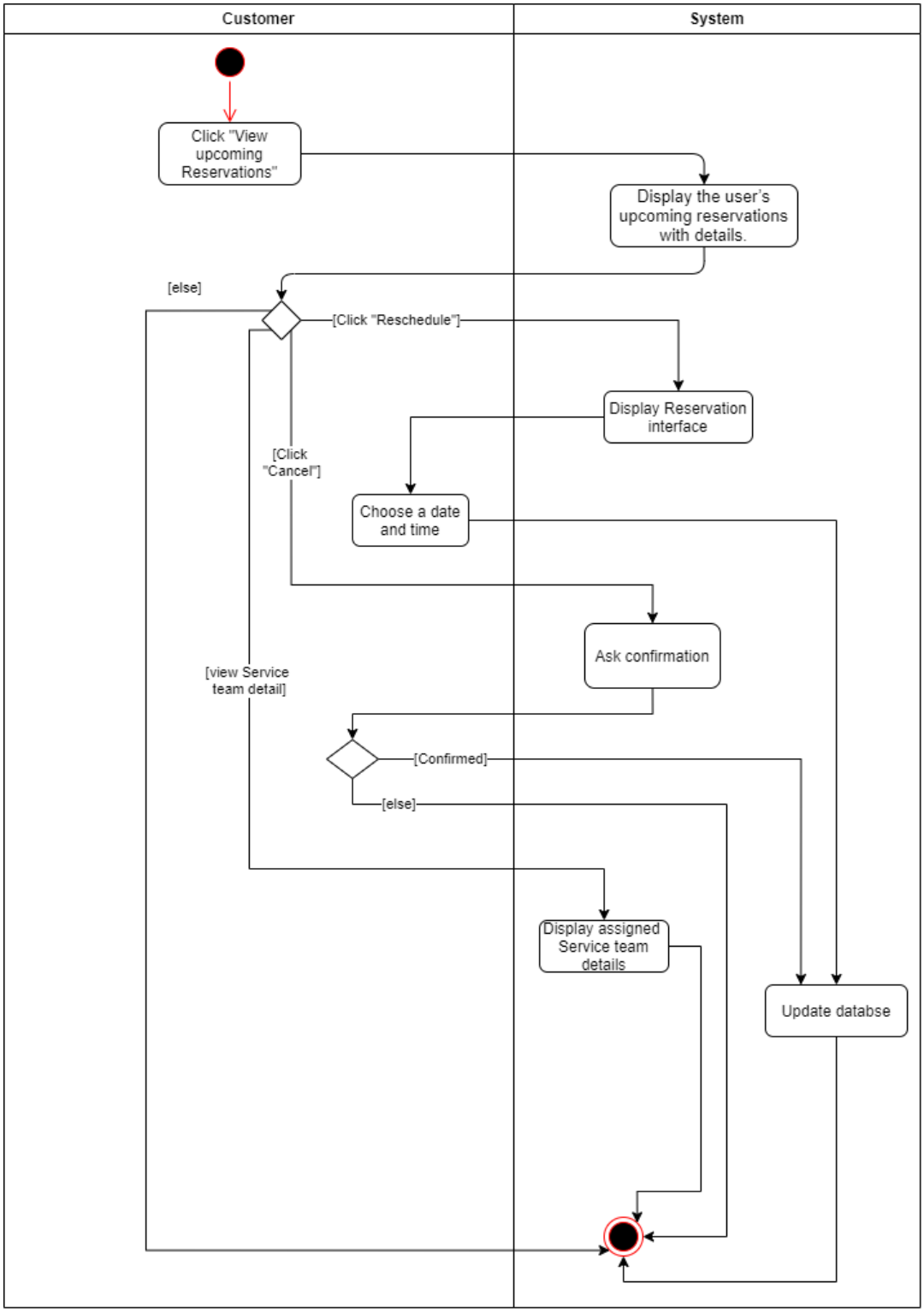


Figure 38: Activity Diagram - View Upcoming Reservations

View Past Reservations

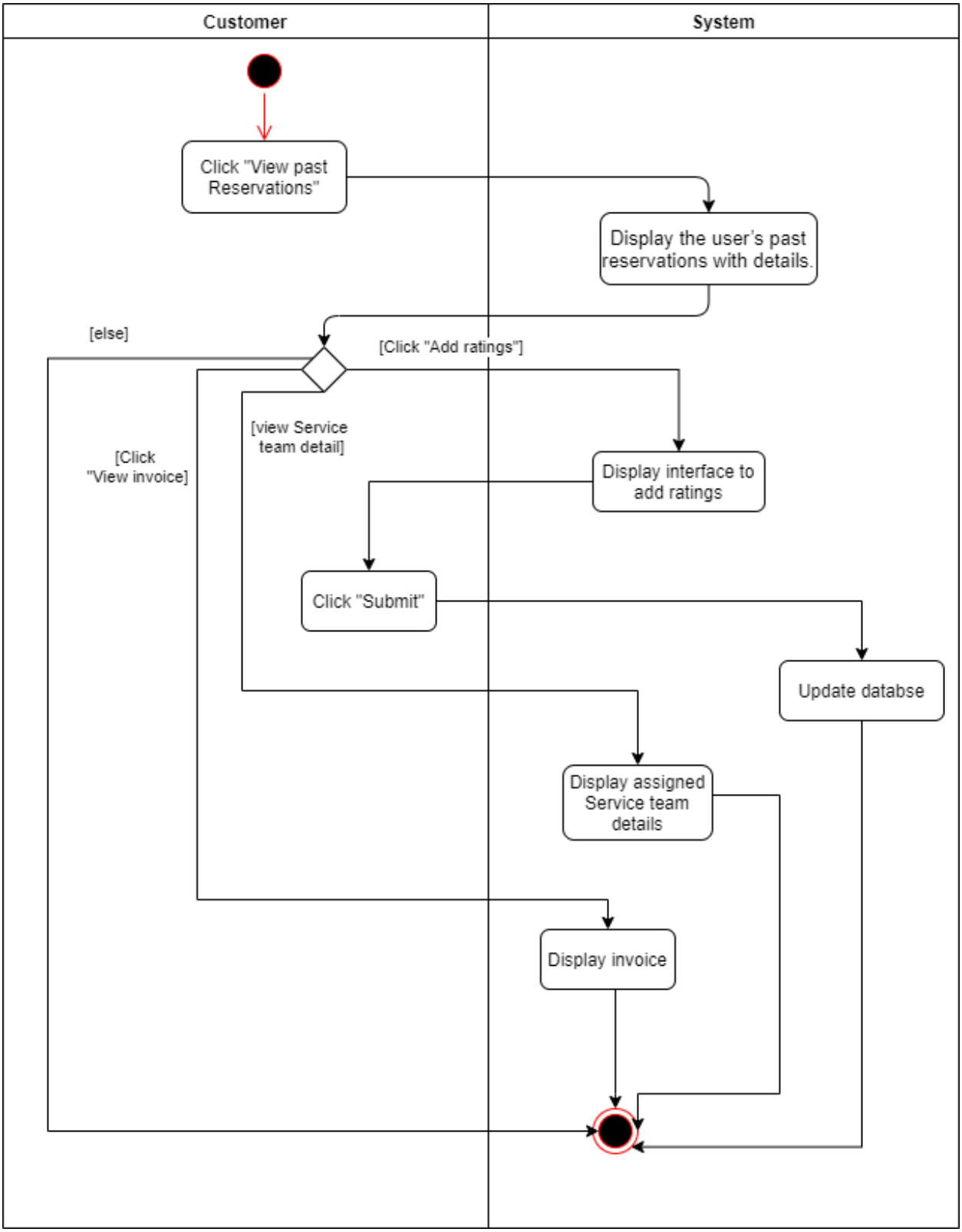


Figure 39: Activity Diagram - View Past Reservations

Give Reviews

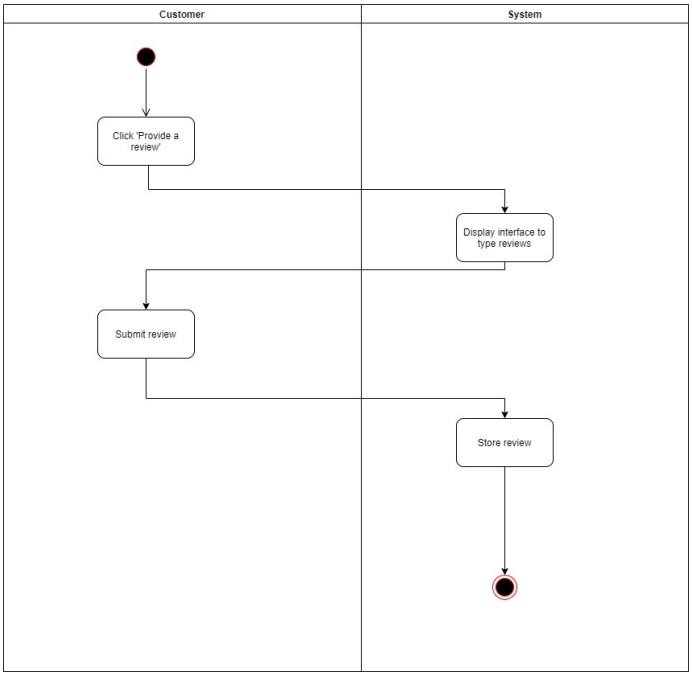


Figure 40: Activity Diagram - Give Reviews

Edit Account Details

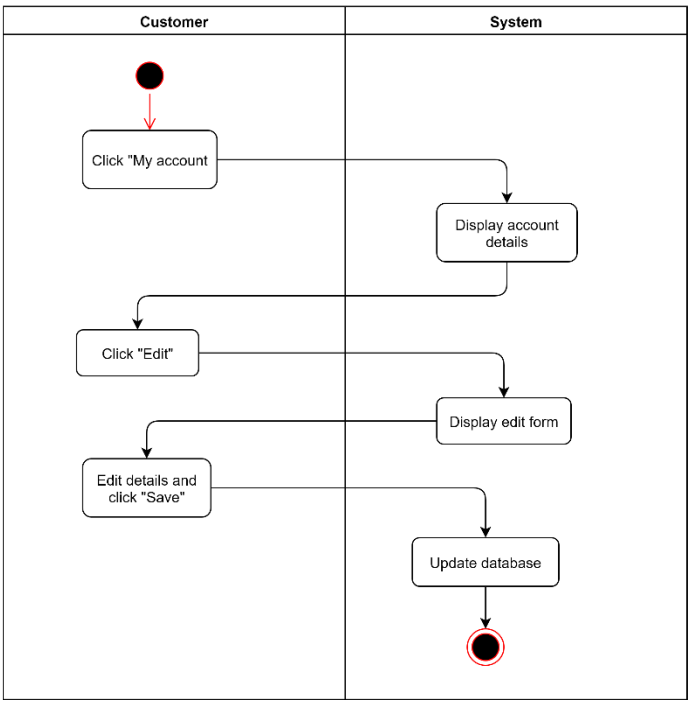


Figure 41: Activity Diagram - Edit Account Details

Delete Account

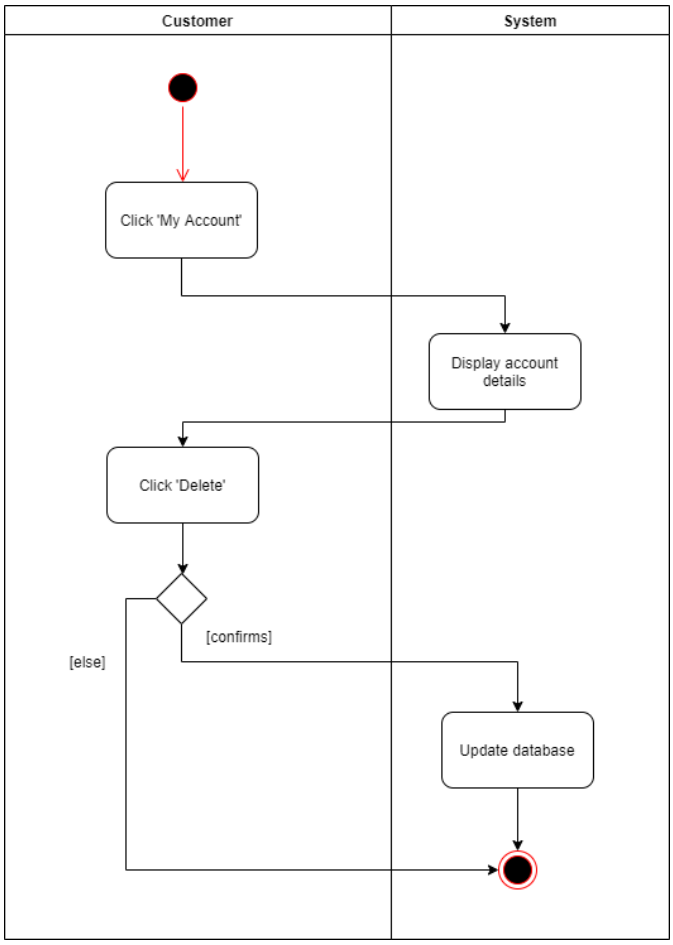


Figure 42: Activity Diagram - Delete Account

View Reservations

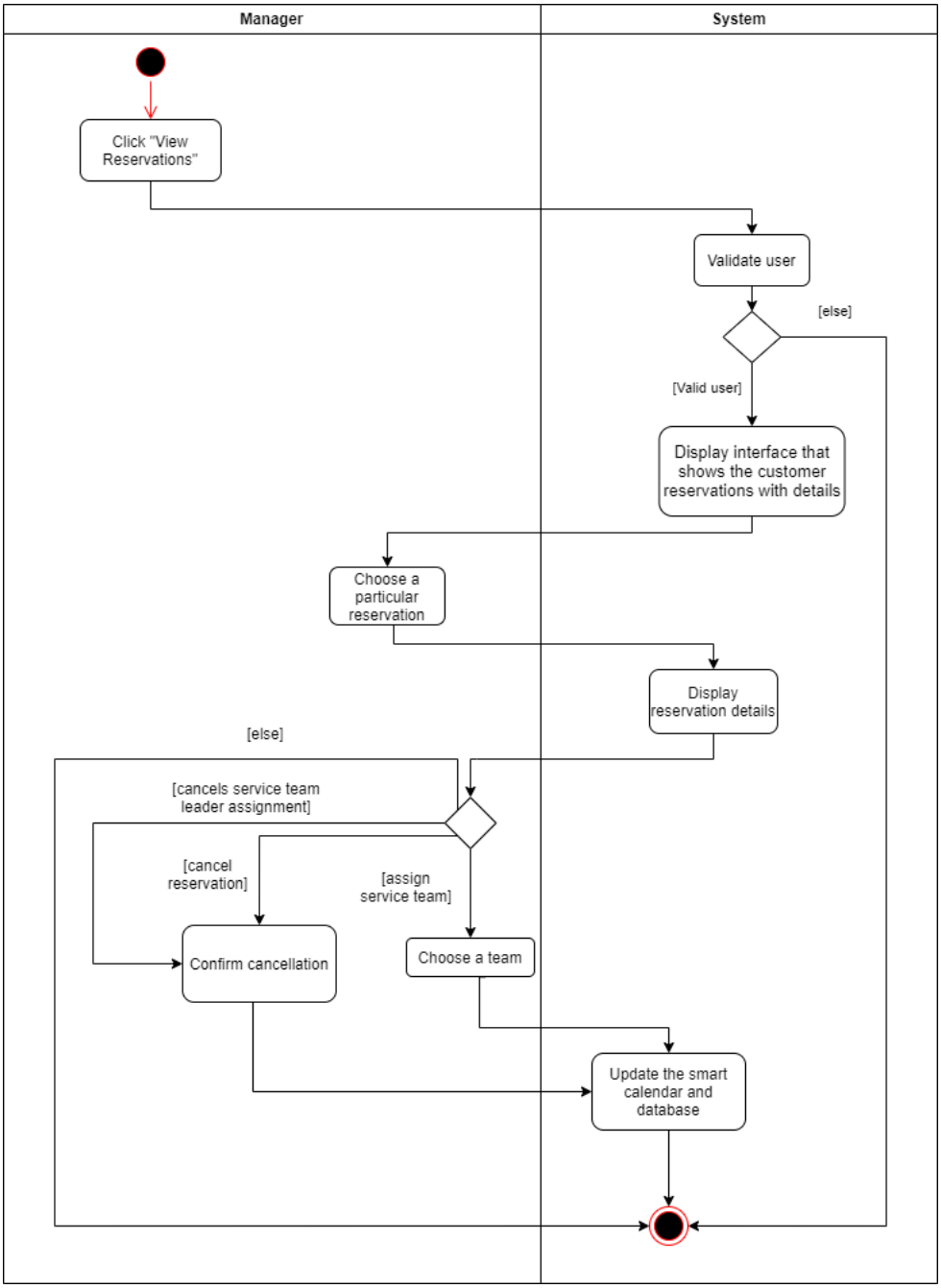


Figure 43: Activity Diagram - View Reservations

View Completed Reservations

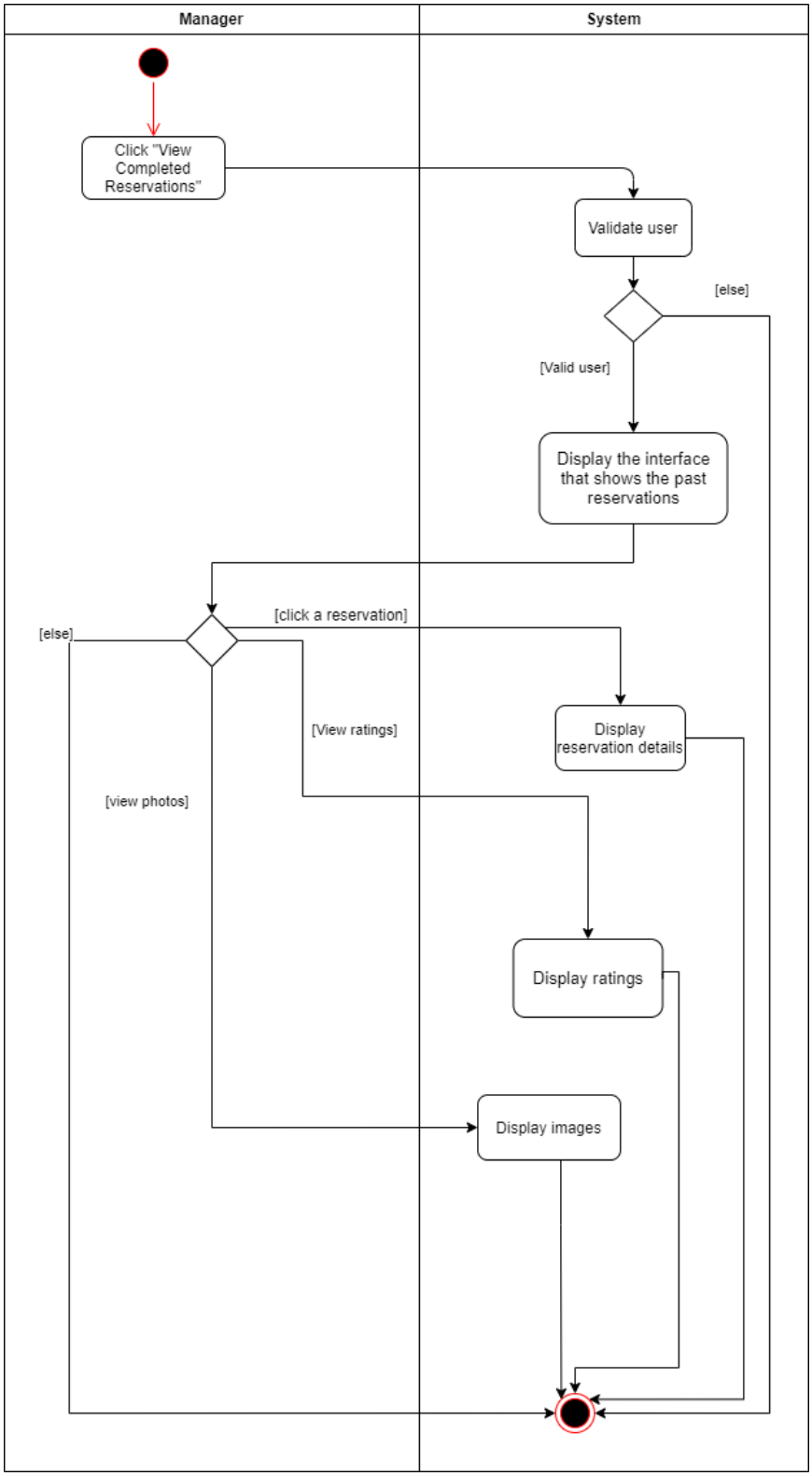


Figure 44: Activity Diagram - View Completed Reservations

View Dashboard

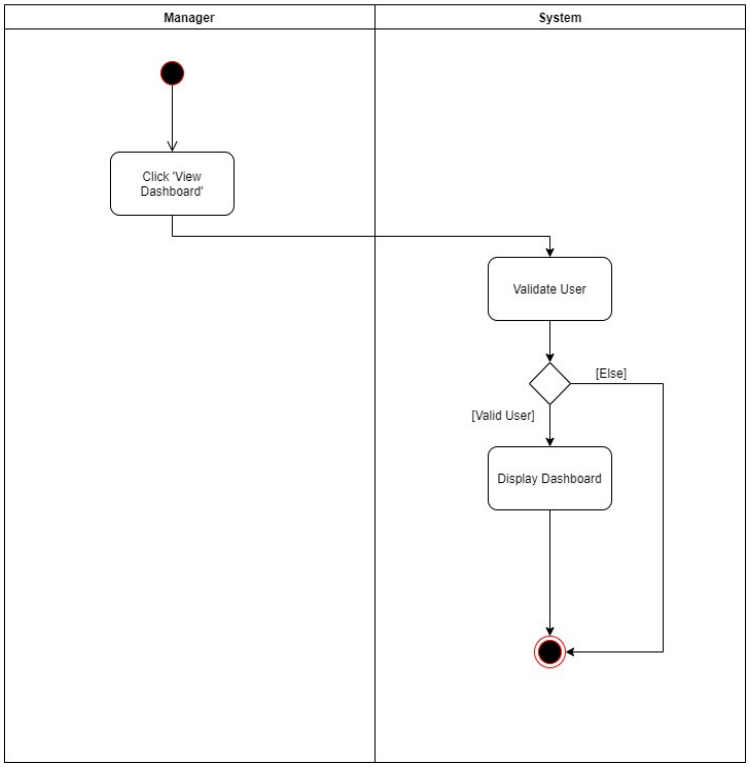


Figure 45: Activity Diagram - View Dashboard

View Service Dashboard

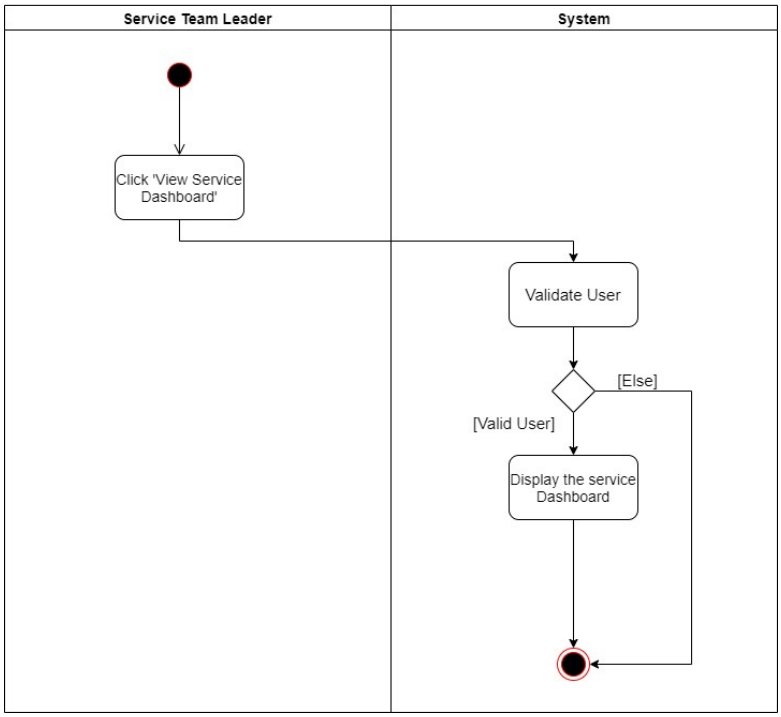


Figure 46: Activity Diagram - View Service Dashboard

Add/Update/Delete Employee

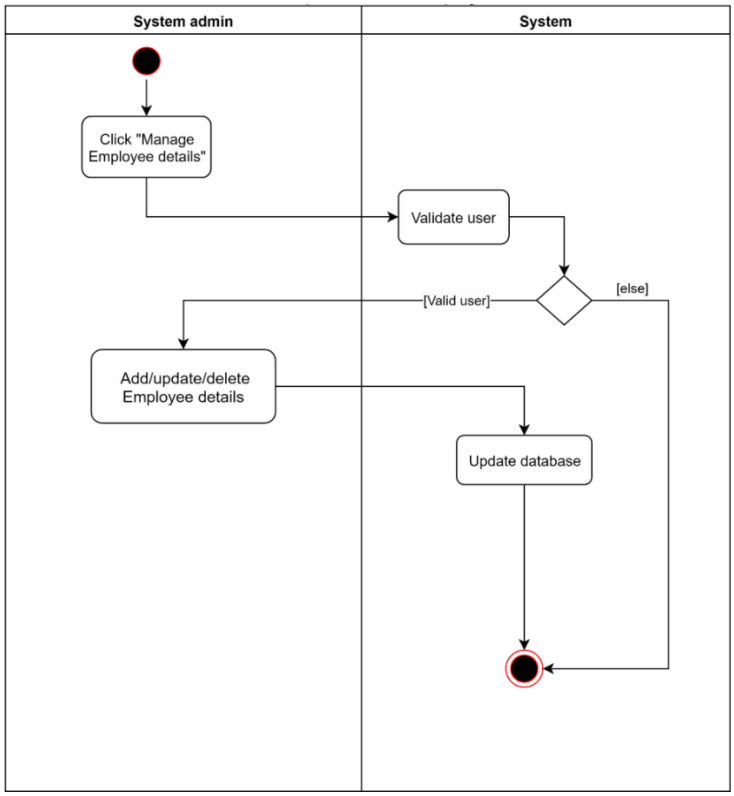


Figure 47: Activity Diagram - Add/Update/Delete Employee

Add/Update/Delete Service Team Leader

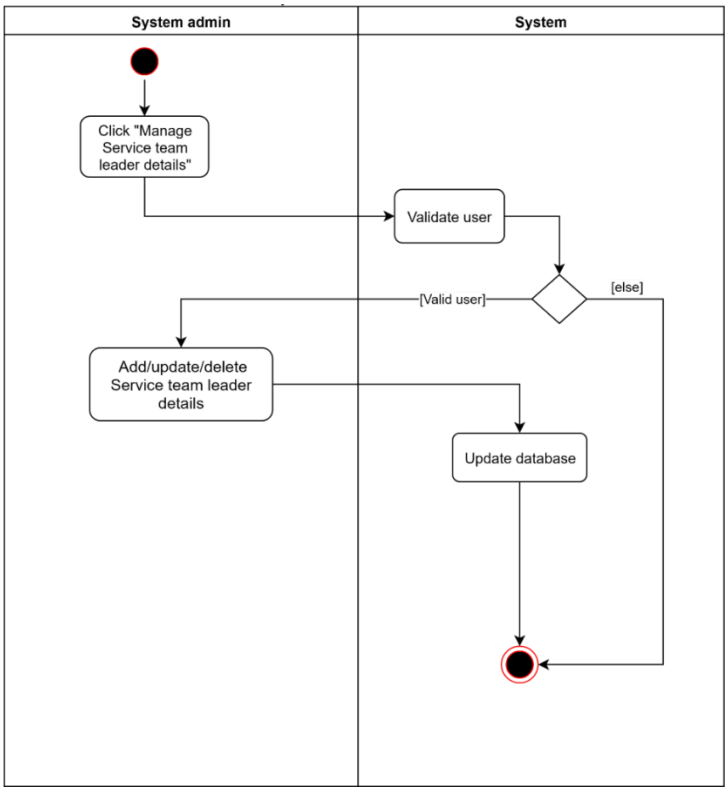


Figure 48: Activity Diagram - Add/Update/Delete Service Team Leader

View/Search Employee

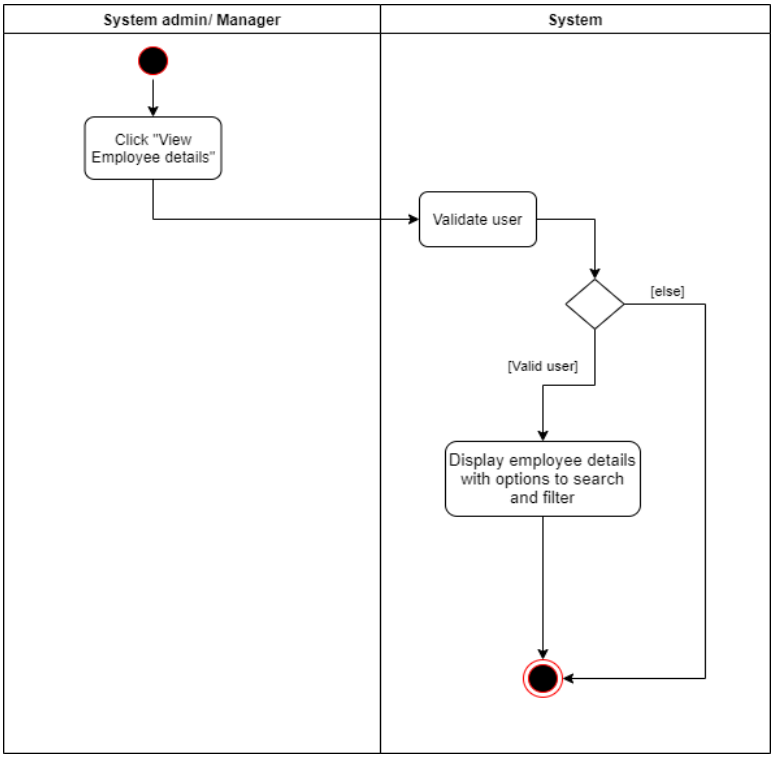


Figure 49: Activity Diagram - View/search Employee

View/Search Service Team Leader

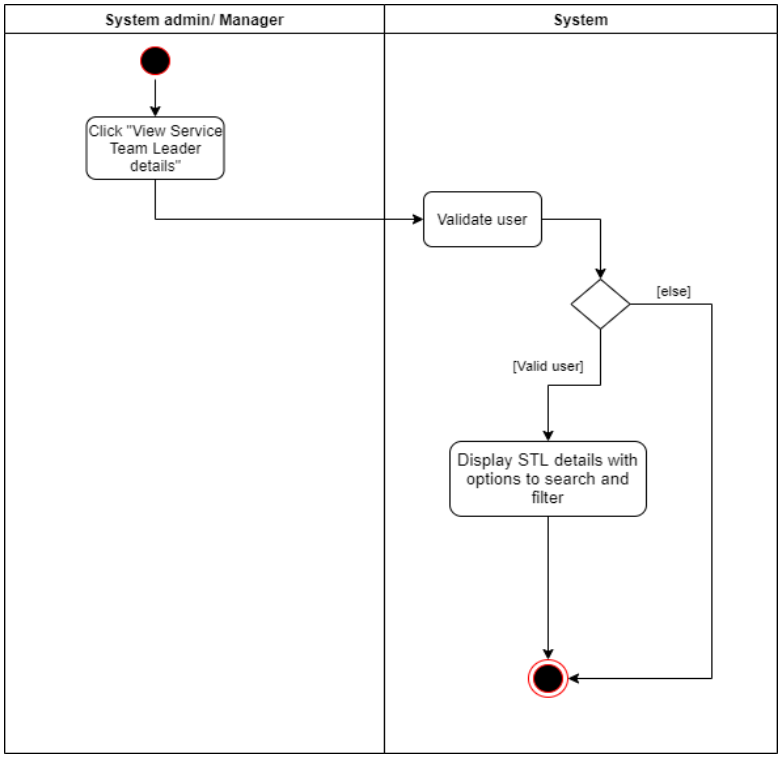


Figure 50: Activity Diagram - View/search Service Team Leader

Add/Update/Delete Service Details

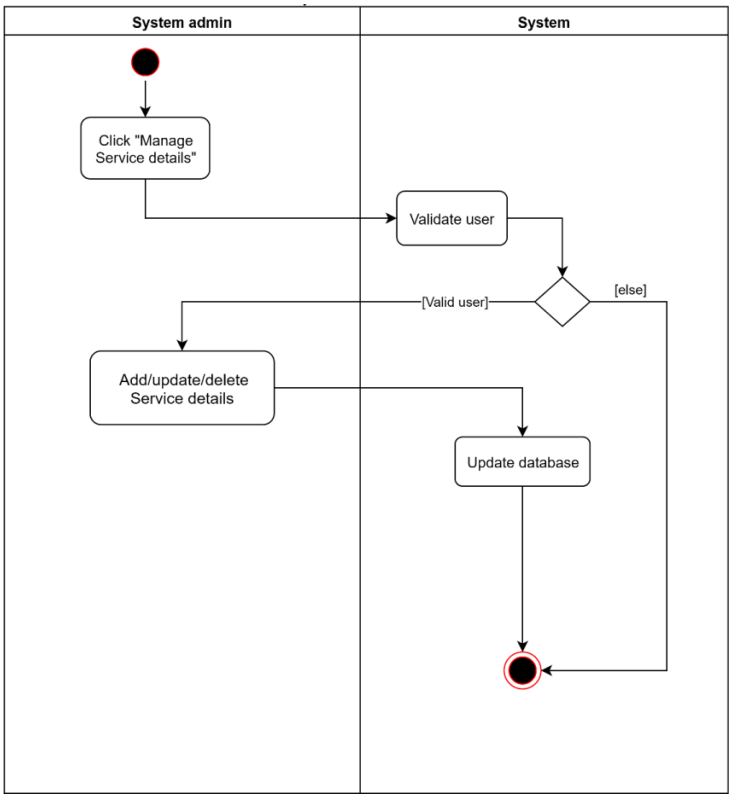


Figure 51: Activity Diagram - Add/Update/Delete Service Details

Add/Update/Delete Equipment Details

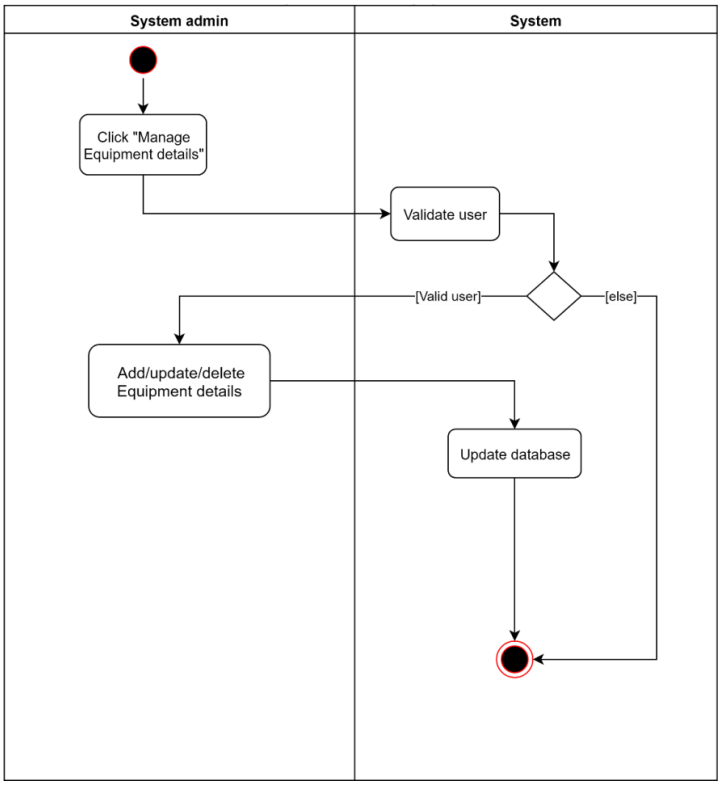


Figure 52: Activity Diagram - Add/Update/Delete Equipment

View/Search Service Details

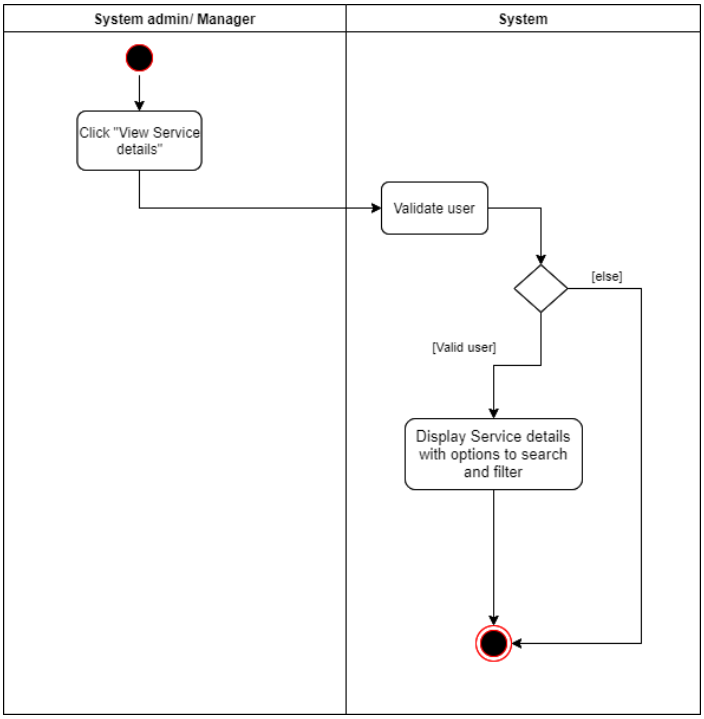


Figure 53: Activity Diagram - View/Search Service Details

View/Search Equipment

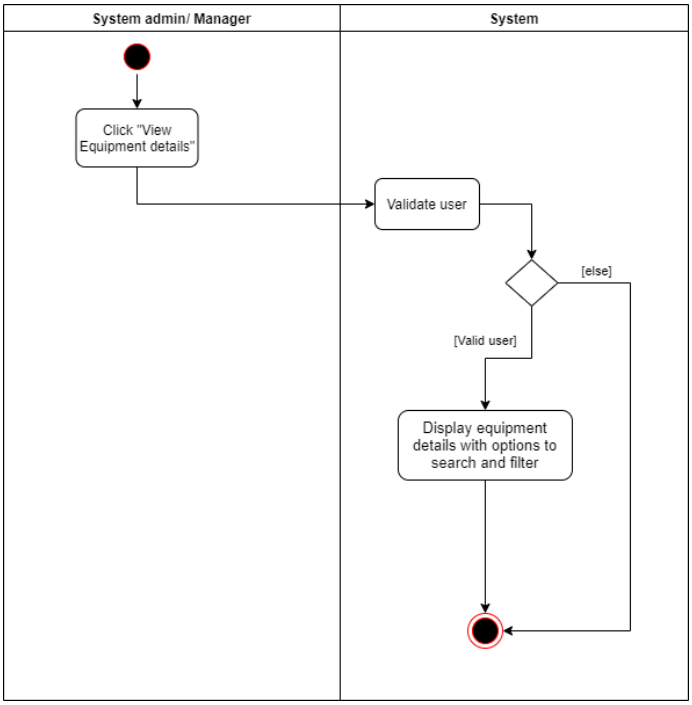


Figure 54: Activity Diagram - View/Search Equipment

View Calendar

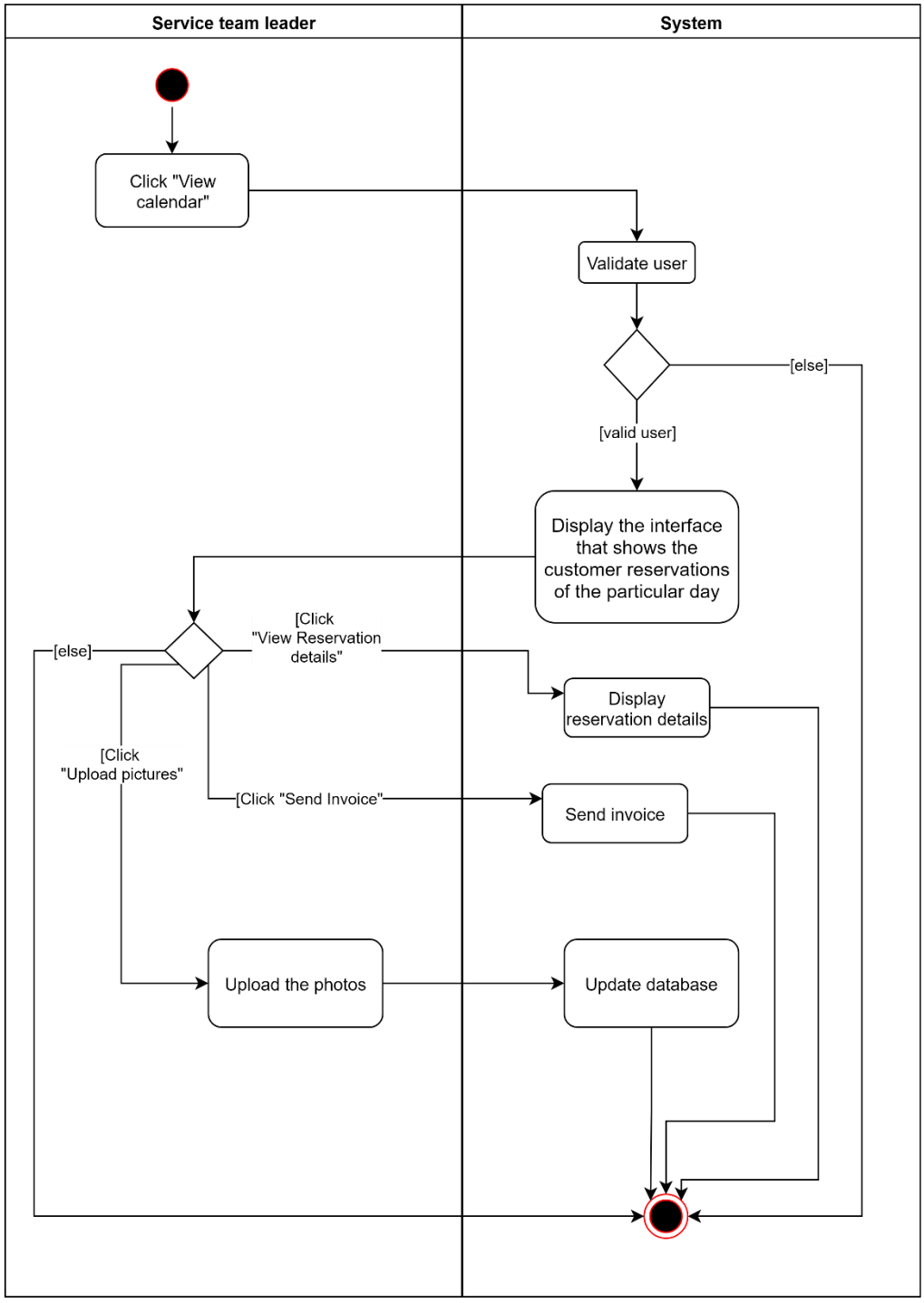


Figure 55: Activity Diagram - View Calendar

8.6 State Transition Diagrams

There are four dynamic components of the system. The triggers and the transitions regarding the four state machines are shown below.

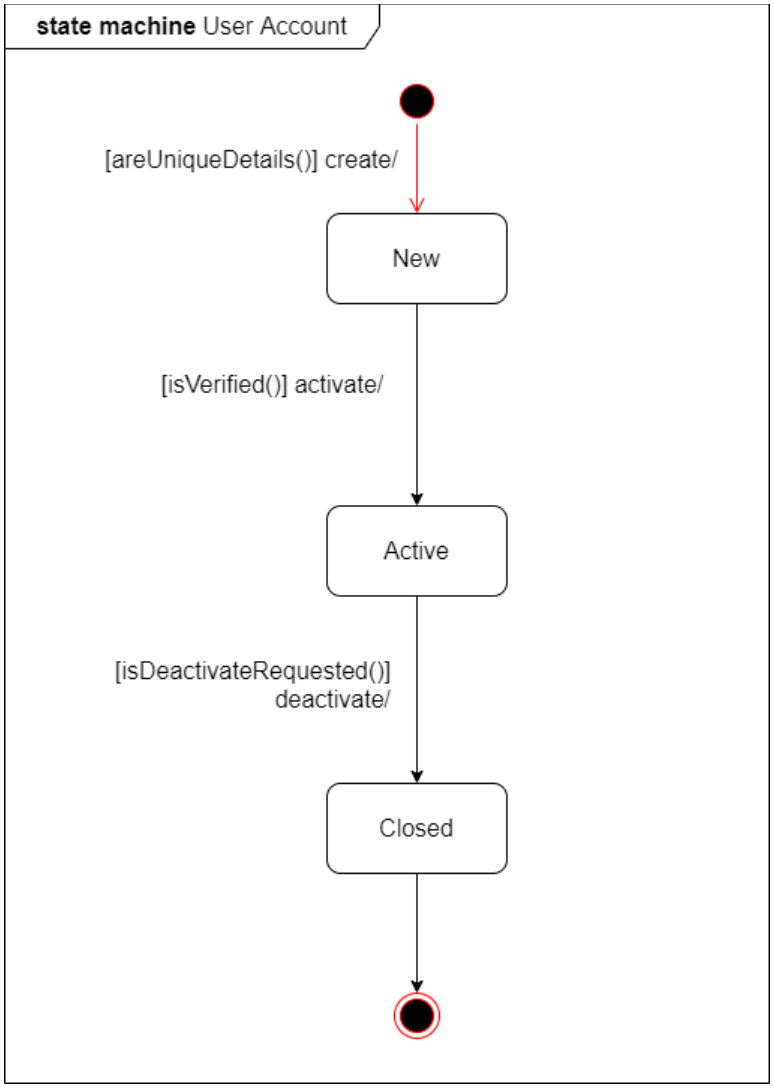


Figure 56: State Transition Diagram - User Account

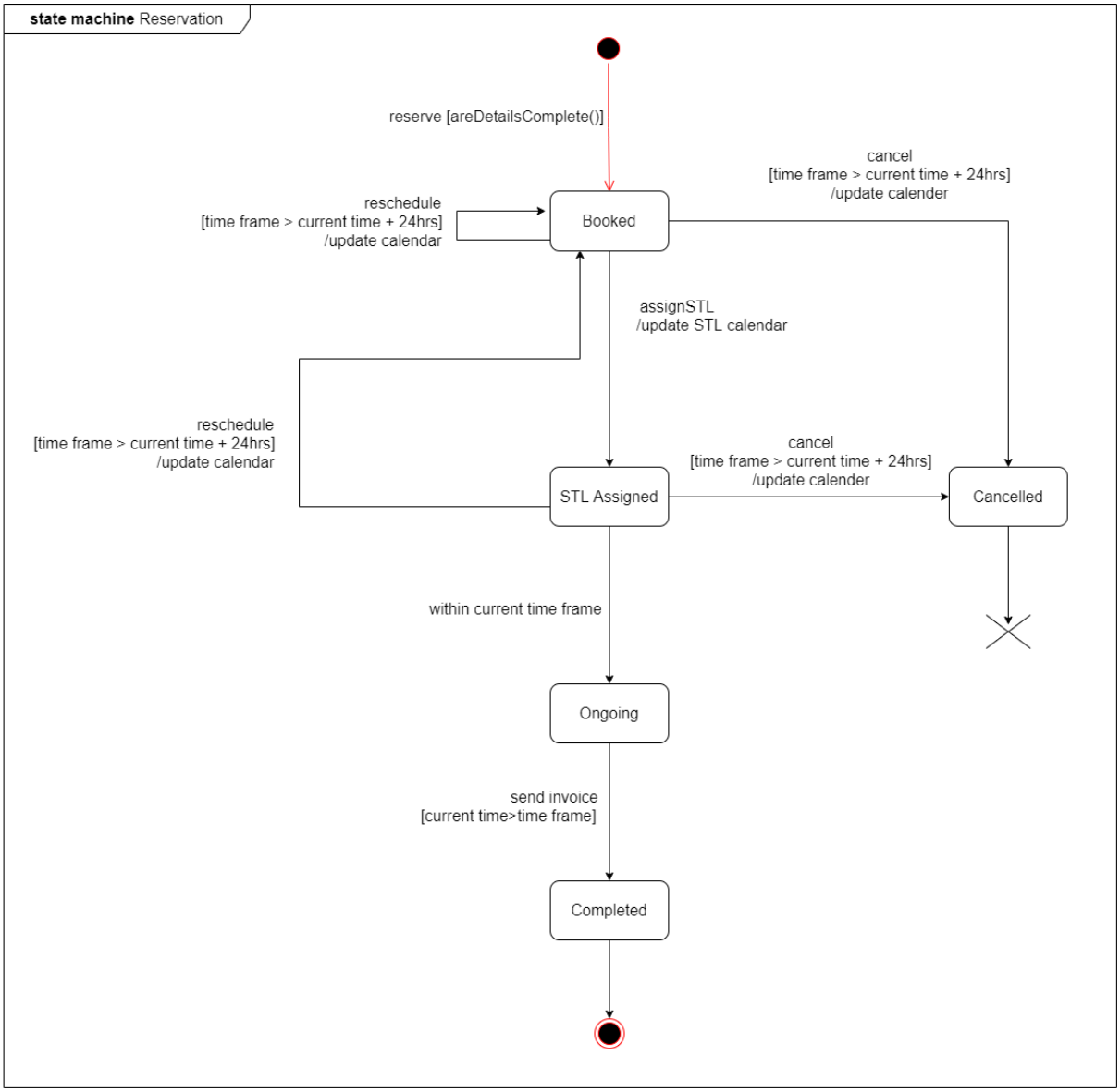


Figure 57: State Transition Diagram - Reservation

The below state machine depicts the states and transitions of a service team for a particular time slot.

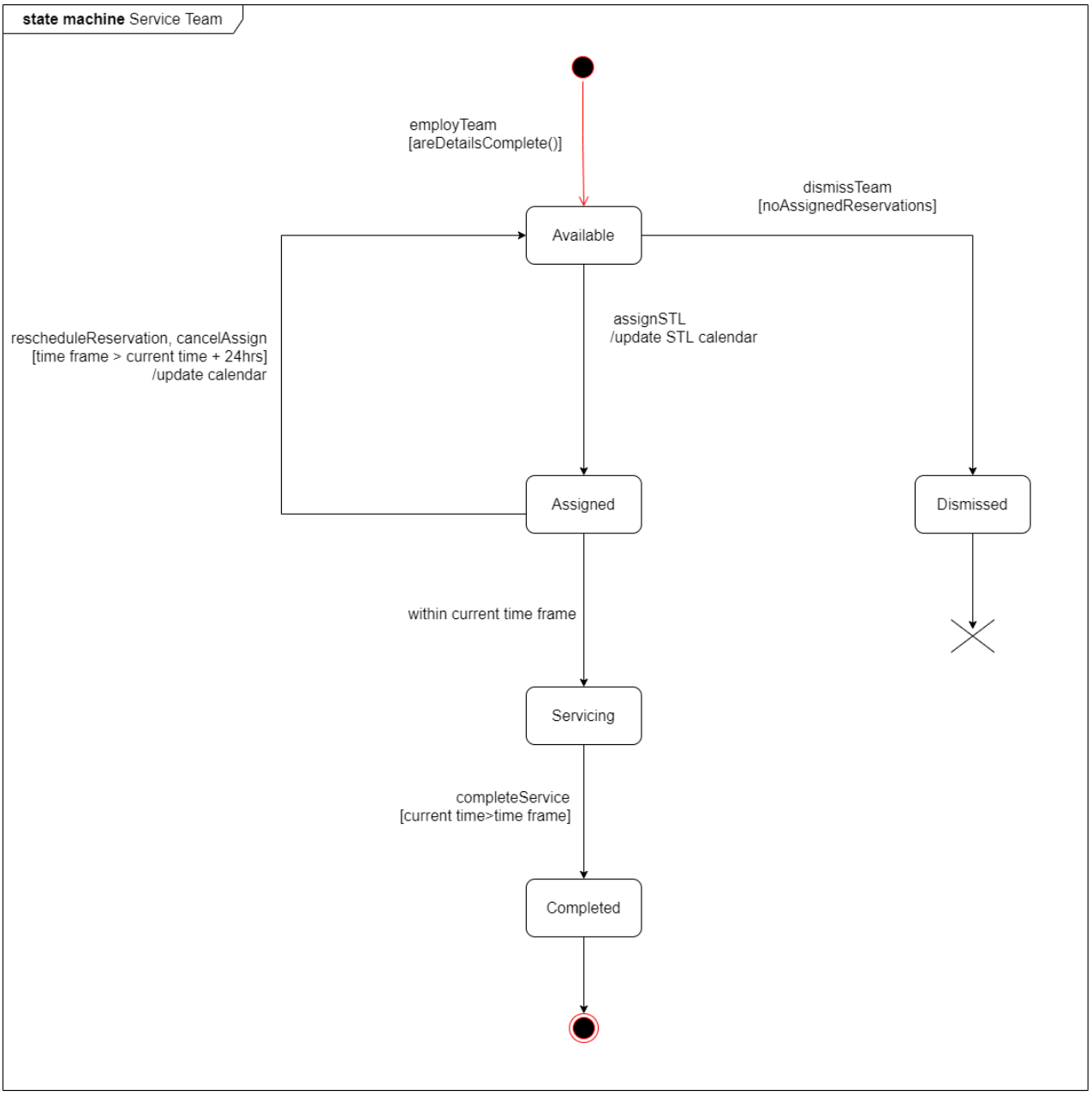


Figure 58: State Transition Diagram - Service Team

The below state machine depicts the states and transitions of equipment for a particular time slot.

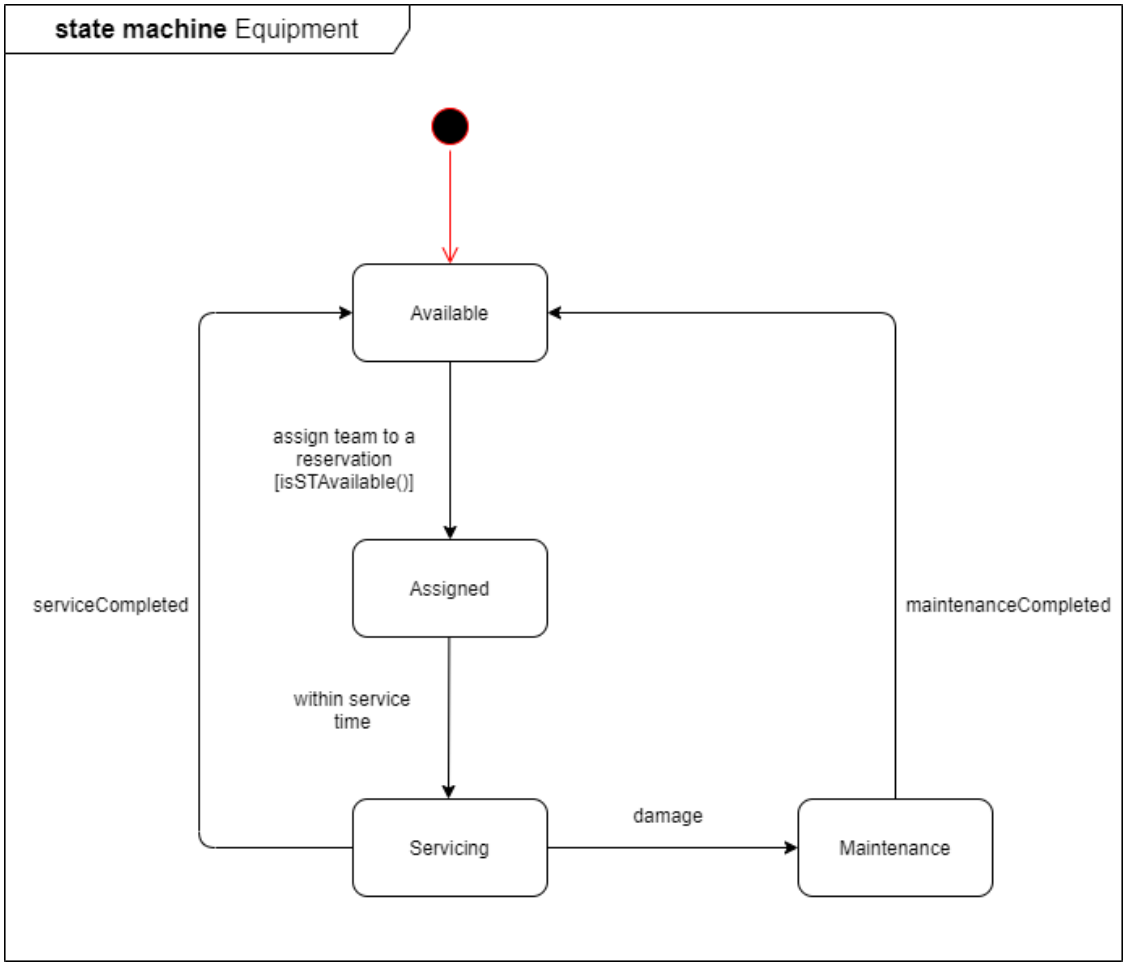


Figure 59: State Transition Diagram - Equipment

References

[1] L.Bass, P.Clements and R.Kazman, Software Architecture in Practice(3rd).

[2] I. Sommerville, Software Engineering(10th).

[3] "uml-diagrams," [Online]. Available: https://www.uml-diagrams.org/. [Accessed 2021].