Software
Engineering

University of Belize
CMPS4131 – Group 3

System Specification

University of Belize
Education Empowers a Nation

Azriel Baris Cuellar
Levi Coc
Daniel Eugenio Garcia
Victor Alexander Jr. Tillet
Floyd Jason Ack

Website Link
https://2019120154.wixsite.com/buscommutecompanion

Table of Contents

Glossary

Notes

- ***Terms defined in the lexicon are italicized when used in definitions within the lexicon.***
- ***When used in the lexicon, “entity” includes a natural person where the context requires.***

Term **Definition**

Access Control

Means to ensure that access to assets is authorized and restricted based on business and security requirements.

Source: ISO/IEC 27000:2018

Accountability

Property that ensures that the actions of n entity may be traced uniquely to that entity.

Source: ISO/IEC 2382:2015

API

An Application Programming Interface is a set of protocols and tools used to build software and allow different software systems to interact and communicate with each other.

Asset

Something of either tangible or intangible value that is worth protecting, including people, information, infrastructure, finances, and reputation.

Source: Adapted from NIST

Authentication

The process of verifying a user’s identity, usually through a username and password, to ensure that only authorized users can assess the software system.

Authenticity

Property that an entity is what it claims to be.

Source: ISO/IEC 27000:2018

Availability

Property of being accessible and usable on demand by an authorized entity.

Source: ISO/IEC 27000:2018

Compromise

Violation of the security of an *information system*.

Source: Adapted from ISO 21188:2018

Confidentiality

Property that information is neither made available nor disclosed to unauthorized individuals, entities, processes or systems.

Source: Adapted from ISO/IEC 27000:2018

Course of Action

An action or actions taken to either prevent or respond to a *cyber incident*. It may describe technical, automatable responses but can also describe other actions such as employee training or policy changes.

Source: Adapted from STIX

Cyber

Relating to, within, or through the medium of the interconnected information infrastructure of the interactions among persons, processes, data, and *information systems*.

Source: Adapted from CPMI-IOSCO (Citing NICCS)

Cyber Alert

Notification that a specific cyber incident has occurred, or a *cyber threat* has been directed at an organization's *information systems*.

Source: Adapted from NIST

Cyber Event

Any observable occurrence in an information system. *Cyber events sometimes provide indication that a cyber-incident is* occurring.

Source: Adapted from NIST (definition of "Event")

Cyber Incident

A cyber event that:

(i) Jeopardize the *cyber security* of an information system or the information the system processes, stores or transmits; or

(ii) Violates the security policies, security procedures or acceptable use policies, whether resulting from malicious activity or not.

Source: Adapted from NIST (definition of "Incident")

Cyber Security

Preservation of confidentiality, integrity, and availability of information and/or *information systems* through the *cyber* medium. In addition, other properties, such as authenticity, *accountability*, *non-repudiation, and reliability can also be* involved.

Source: Adapted from ISO/IEC 27032:2012

Data Analytics

The process of collecting, analysing, and interpreting data to gain insights and make informed decisions about the bus ticket system.

Encryption

A process of converting sensitive information into a code or cipher to protect it from unauthorized access and use.

Exploit

Defined way to breach the security of information systems through *vulnerability*.

Source: ISO/IEC 27039:2015

FURPS

A requirements analysis framework that stands for Functionality, Usability, Reliability, Performance, and Supportability.

GPS Tracking

A technology that uses satellite signals to determine the exact location of a bus, which can be used to track the bus's progress along its route and provide real-time updates to users.

Incident

An unplanned interruption to a service, a reduction in the quality of a service or event that has not yet impacted the service to the customer or user.

Source: ISO/IEC 20000-1

Information Systems

Set of applications, services, information technology *assets* or other information-handling components, which includes the operating environment.

Source: Adapted from ISO/IEC 27000:2018

Mobile Application

A software application that runs on mobile devices, allowing users to interact with the software system on their mobile phones or tablets.

Multi-Factor Authentication

The use of two or more of the following factors to verify a user's identity.

-- knowledge factor, "something an individual knows".

-- possession factor, "something an individual has".

-- biometric factor," something that is a biological and behavioural characteristic of an individual".

Source: Adapted from ISO/IEC 27040:2015 and ISO/IEC 283237:2017 (definition of "biometric characteristic")

Notification System A feature of the software system that sends alerts to users about any updates or changes to their bus ticket, such as delays or cancellations.

Payment Gateway A secure online service that processes payments from users for the purchase of bus tickets.

Protect (function) Develop and implement the appropriate safeguards to ensure delivery of services and to limit or contain the impact of *cyber incidents*.

Source: Adapted from NIST Framework

Real-time Updates Immediate information about changes or delays to the bus schedule or route, provided to the passenger through the software system.

Recover (function) Develop and implement the appropriate activities to maintain plans for *cyber resilience* and to restore any capabilities or services that were impaired due to a *cyber-incident*.

Source: Adapted from NIST Framework

Refund Policy The terms and conditions outlining the process for issuing refunds to users who have purchased a bus ticket but are unable to use it.

Reliability Property of consistent intended behaviour and results.

Source: ISO/IEC 27000:2018

Respond (function) Develop and implement the appropriate activities to take action regarding a detected *cyber-event*.

Source: Adapted from NIST Framework

Responsive Design A design approach that ensures the software system is optimized for different devices and screen sizes, providing a seamless passenger experience regardless of the device being used.

Use Case Diagram A graphical representation of the interactions between users and the system, showing the different use cases for the software system.

User Interface (UI) The visual and interactive components of the software system that users interact with to complete tasks.

Verification Confirmation, through the provision of objective evidence that can be exploited by one or more threats.

Source: ISO/IEC 27042:2015

Vulnerability A weakness, susceptibility or flaw of an asset or control that can be exploited by one or more threats.

Source: Adapted from CPMI-IOSCO and ISO/IEC 27000:2018

1. INTRODUCTION

1.1 In today's fast-paced world, people are looking for convenient and efficient ways to manage their travel needs. Bus companies are no exception and require a robust and user-friendly bus ticket management system to streamline their operations and provide their customers with a seamless experience.

1.2 This system aims to convert a traditional process, often done with pen and paper, into a digital experience. With the widespread use of smartphones, data and online banking, this system will serve as a management tool for bus conductors to keep track of the number of passengers on the bus.

1.3 The system will also provide the driver with information about where to stop in advance. Passengers will be able to use the system to purchase tickets and plan their bus stops before boarding the bus. This proposal outlines the development of a desktop application that will provide bus companies with a comprehensive solution for managing ticket sales, schedules, and real-time bus tracking.

2. INDIVIDUAL CONTRIBUTIONS BREAKDOWN

Responsibility Level	**Team Member Names**				
	Azriel Cuellar	**Daniel Garcia**	**Victor Tillet**	**Levi Coc**	**Floyd Ack**
Project Management	20%	20%	20%	20%	20%
Sec. 1: Customer Statement of Requirements	20%	20%	20%	20%	20%
Sec. 2: System Requirements	20%	20%	20%	20%	20%
Sec.3: Functional Requirements Specification	20%	20%	20%	20%	20%
Sec. 4: User Interface Specs	20%	20%	20%	20%	20%
Sec. 5: Domain Analysis	20%	20%	20%	20%	20%
Sec. 6: Plan of Work	20%	20%	20%	20%	20%

3. PROBLEM STATEMENT

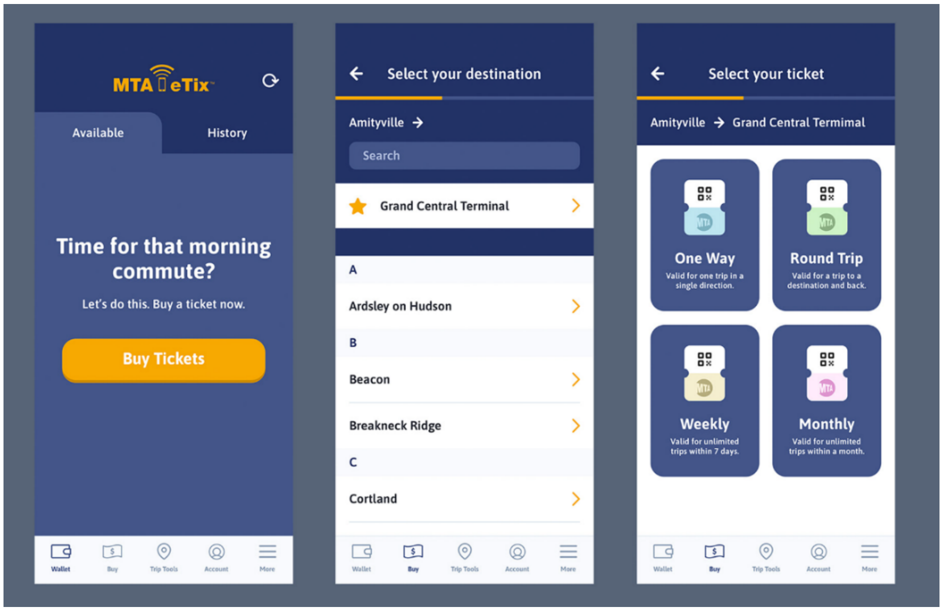
As a bus company, we understand the number of challenges passengers face when it comes to using bus services in Belize. One of the most significant issues is the lack of reliability in bus schedules at terminals. With so many factors that can impact the timeliness of buses, it can be difficult for passengers to plan their trips with confidence. Additionally, the current ticketing system is also prone to flaws, which can result in a frustrating experience for passengers.

For example, the issue of standees on buses is caused in part by a lack of available tickets or the over-selling of tickets at the terminal. In the past, passengers would simply have to line up and push their way onto the bus in order to secure a seat. This was not only dangerous in the event of an accident, but it also favoured able-bodied individuals over those who were elderly, disabled, or accompanied by young children. To remedy this, our company began selling tickets for sale. However, this solution has brought with it a new set of problems. The ticket seller at the terminal is not always aware of the number of available seats on the bus, and sometimes a purchased ticket does not guarantee a seat because someone has already pushed their way onto the bus.

The current system is also not very efficient from a management perspective. Our company does not have an accurate way of tracking the number of passengers on a specific bus on any given day. This information can only be estimated based on revenue, but there is always the possibility that money goes missing and we are not able to detect it. The conductor of the bus calls the terminal manager to report the number of passengers on board and the number of available seats, but this process can be prone to inaccuracies or even deceit. The counting of tickets sold is also not a fool-proof method, as some passengers may not purchase tickets in advance, or the conductor may fail to distribute them along the way.

On occasion, drivers or conductors may choose to let more passengers onto the bus, even if it means standing, which can be highly unsafe in the event of an accident. This practice is also unfair to those who are elderly, disabled, or parents with young children. Additionally, some ticket holders may be unable to find a seat due to other passengers pushing ahead of them, rendering their ticket useless until the next bus arrives. While bus conductors may attempt to help, not all of them do, causing passenger frustration and potential loss of revenue for bus companies. The current system also poses a risk of financial loss or robbery, as passengers must pay at the terminal, which can potentially be a dangerous location for employees. Moreover, even with the current system in place, some passengers still attempt to pay on the bus, causing further confusion and potential conflicts over seat availability.

To improve the experience for passengers and address these challenges, we are looking for a software solution that can streamline and optimize the bus ticketing and management system. The proposed bus ticket management system will enable individuals to buy bus tickets using their mobile phones, and the purchasing process could be as straightforward as placing the phone on a surface or making the purchasing online, akin to Amazon. The model could be comparable to the MTA eTix system, which permits the buying of train tickets before boarding. The MTA eTix framework will be used as a reference for this bus ticket managing system.



Some of the key requirements that we would like to see in this solution include:

1. An accurate and real-time tracking system for bus schedules and arrivals, to provide passengers with up-to-date information on the status of their bus.
2. A secure and efficient ticketing system that eliminates the possibility of overselling tickets or providing an inaccurate count of available seats. This should also be easily accessible for all passengers, regardless of their physical ability.
3. A management system that allows our company to track the number of passengers on each bus on a real-time basis, as well as a clear and accurate record of ticket sales.
4. Integration with existing payment methods, such as mobile payment platforms, to allow for a seamless and convenient experience for passengers.
5. A user-friendly interface for both passengers and bus conductors, to make the process of buying and using tickets as simple and straightforward as possible.
6. Robust security measures to protect passenger information and prevent unauthorized access to the system.

We understand that this is a complex project, but we believe that a well-designed and implemented bus ticket management system has the potential to significantly improve the experience of passengers and the efficiency of bus companies. We are looking forward to hearing from you about how you can help us achieve these goals.

4. SYSTEM REQUIREMENTS

4.1 Functional Requirements (REQ)

These requirements define the specific functions and features that the bus ticket management system must have in order to meet the needs of the bus company. They are the “what” of the system.

Identifier	**PW**	**Requirement**
REQ1	5	The system shall allow passengers to create and manage passenger profiles including passenger information and payment methods.
REQ2	4	The system shall enable passengers to purchase available tickets online or through a mobile application.
REQ3	4	The system shall allow users to view the details of previous and present bookings.
REQ4	5	The system shall provide passengers with information about their trips, including departure and arrival times, bus routes, and fare pricing.
REQ5	4	The system shall allow the bus company to easily manage and update bus schedules to prevent delays.
REQ6	3	The system shall allow bus companies to track ticket sales, passenger information, and revenue.
REQ7	5	The system shall allow users booking through a computer the ability to generate e-tickets or print their tickets with the system.
REQ8	5	The system shall accurately calculate fees based on routes, passengers, and trip dates.
REQ9	5	The system shall generate reports on ticket sales, passenger traffic, and revenue, which can be used to track performance and make informed business decisions.

Identifier	PW	Requirement
REQ10	2	The system shall allow users to be able to access it by logging in.
REQ11	1	The system shall allow users to modify the time or conditions under which a notification is activated.
REQ12	2	The system shall give users the ability to sign out of the system.
REQ13	2	The system shall allow users to be able to cancel their ticket purchase.
REQ14	3	The system shall allow users to change the date and time their ticket is valid.
REQ15	1	The system shall allow users to offer a comprehensive feedback of their journey experience.

4.2 Non-Functional Requirements (NONREQ)

These requirements define the quality attributes and performance characteristics that the bus ticket management system must meet. They define the "how" of the system.

Identifier	PW	Requirement
NONREQ1	5	The system must have a user-friendly interface that is easy to use and navigate, providing a seamless experience for passengers.
NONREQ2	5	The system must have strong security measures to protect passenger information, including encryption for sensitive data, safely notify users of details regarding the ticket purchased and multi-factor authentication for passenger accounts.
NONREQ3	5	The system shall be scalable to accommodate future growth and increased passenger traffic.

NONREQ4	3	The system shall have a fast response time and be able to handle high volumes of traffic, ensuring that tickets can be purchased, and reservations can be managed in real-time.
NONREQ5	3	The system shall be compatible with various platforms and devices, including desktop computers, laptops, tablets, and mobile devices.
NONREQ6	5	The system shall be available and reliable for use at all times.
NONREQ7	3	The system shall allow users to purchase a ticket without experiencing crashes or processing failures in it.
NONREQ8	1	The system must allow users to view their ticket even when they don't have an internet connection.
NONREQ9	2	The system shall apply a cut-off time and location limit beyond which it will not be possible to buy or cancel tickets.

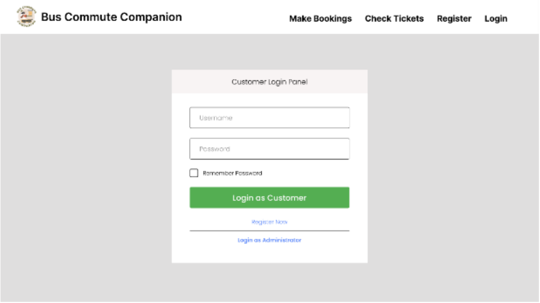
4.3 On-Screen Appearance Requirements (ONSREQ)

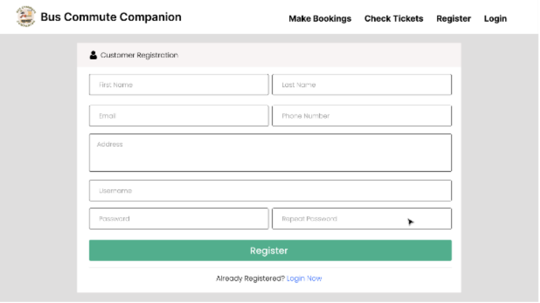
These requirements refer to the visual and graphical design elements of the bus ticket management system that determines how it appears on the passenger's screen.

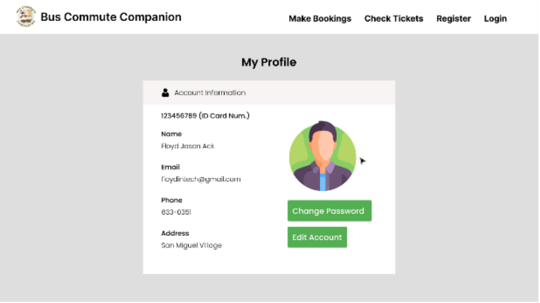
Identifier	**PW**	**Requirement**
ONSREQ1	5	The system shall have a clear display of available schedules and routes, along with real-time updates on any changes or delays.
ONSREQ2	5	The system shall give users the ability to select their desired route, departure time, and seat type quickly and easily, and view a summary of their ticket details before confirming the purchase.
ONSREQ3	3	The system shall have a secure payment gateway for users to securely enter their payment information and complete the transaction.

ONSREQ4	3	The system shall have a notification system to inform users of any updates or changes related to their ticket, such as schedule changes or delays.
ONSREQ5	3	The system shall give users the ability to view their ticket history and manage their account information, including updating their personal information, email address, and payment information.
ONSREQ6	5	The system shall have a responsive design that is optimized for different screen sizes and devices, including mobile devices, to provide a seamless experience for users.

4.4 On-Screen Appearance Requirements UI Design







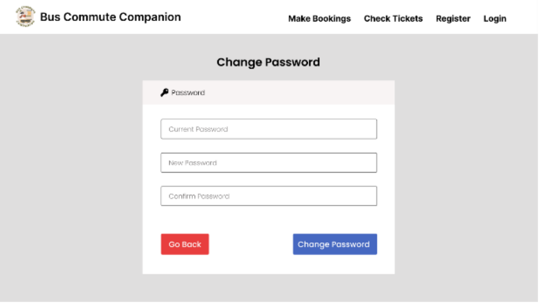
123456789 (ID Card Num.)

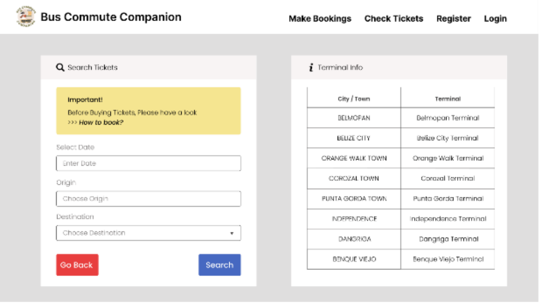
Name
Floyd Jason Ack

Email
floydintech@gmail.com

Phone
633-0351

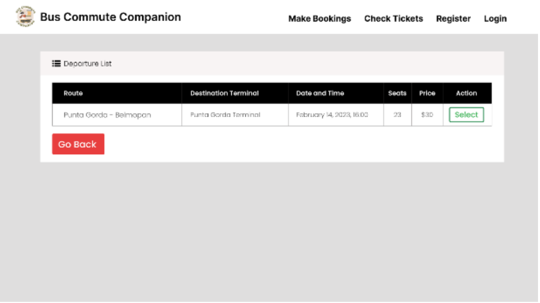
Address
San Miguel Village





Important!
Before Buying Tickets, Please have a look
>>> *How to book?*

City / Town	Terminal
BELMOPAN	Belmopan Terminal
BELIZE CITY	Belize City Terminal
ORANGE WALK TOWN	Orange Walk Terminal
COROZAL TOWN	Corozal Terminal
PUNTA GORDA TOWN	Punta Gorda Terminal
INDEPENDENCE	Independence Terminal
DANGRIGA	Dangriga Terminal
BENQUE VIEJO	Benque Viejo Terminal



Route	Destination Terminal	Date and Time	Seats	Price	Action
Punta Gorda - Belmopan	Punta Gorda Terminal	February 14, 2023, 16:00	23	$30	Select

4.5 FURPS Table

FURPS (Functionality, Usability, Reliability, Performance, Supportability) is a widely used model for categorizing software requirements. A FURPS table for this bus ticket management system looks like this:

Functionality	i. The system should allow users to purchase and manage their bus tickets through a user-friendly and intuitive interface. ii. It should enable them to quickly select their desired route, departure time, and seat type, view a summary of their ticket details, and securely complete the transaction. iii. The system should also provide real-time updates on available bus schedules and routes and any changes or delays. iv. Additionally, it should allow users to view their ticket history and manage their account information.
Usability	i. The user interface should be clear and easy to navigate, with clear display of available bus schedules and routes. ii. The payment gateway should be secure and allow for easy entry of payment information. iii. The notification system should inform users of any updates or changes related to their tickets. iv. The design should be optimized for different screen sizes and devices to provide a seamless experience for users.
Reliability	i. The system should be reliable, with minimal downtime and errors. ii. The real-time updates on bus schedules and routes should be accurate, and the payment gateway should securely process transactions without any security breaches. iii. The notification system should provide timely and accurate updates on any changes or delays.
Performance	i. The system should be fast and responsive, with quick processing times for ticket purchases and updates. ii. It should be able to handle a high volume of transactions at once.

Supportability	i. The system should be able to integrate with other systems, such as the bus companies' accounting systems, to provide real-time updates in ticket sales and revenue.

5. STAKEHOLDERS

Stakeholders in the context of an application refer to individuals, groups, or organizations that have a direct or indirect interest in the development, implementation, and use of the application. The stakeholders of Bus Commute Companion, classified as primary, secondary, or tertiary stakeholders are:

1. Primary Stakeholders:
    a. Passengers: who will use Bus Commute Companion to purchase tickets and manage their trips.
    b. Bus Company Staff: including ticket agents, customer service representatives, and operations managers who will use the system to manage passenger bookings and track bus schedules.

2. Secondary Stakeholders:
    a. Bus Company Management: who will oversee the system's implementation and use, as well as make decisions based on generated reports and data.
    b. Software Developers: responsible for designing, building, and maintaining the system.
    c. Payment Gateway Providers: who will facilitate online payments and transactions.
    d. Regulatory Authorities: who may need to approve the system's implementation and use and ensure compliance with local laws and regulations.

3. Tertiary Stakeholders:
    a. Third-Party Service Providers: who may provide additional services through the system, such as advertising, promotions, or travel insurance.
    b. Investors: who may have a financial stake in the bus company and the success of the system.

6. ACTOR AND GOALS

Based on Bus Commute Companion's stakeholders, the following are the actors and their goals:

1. Passengers

   Aim to register for an account, search, and view bus schedules, book a ticket, pay for a ticket, view booking details, cancel a booking, view bus and trip updates, and provide feedback and ratings through their smartphones.

2. Bus company management and staff

   Manage bus schedules, routes, availability, ticket sales, bookings, and passenger feedback through desktop or laptop computers.

3. Payment gateway provider

   Provide secure online payment processing services for passengers making payments through the Bus Commute Companion app using their own payment gateway devices.

4. Software developers

   Develop, test, and deploy the Bus Commute Companion app and ensure its compatibility with various operating systems and devices using their own software development devices.

5. Regulatory authorities

   Ensure compliance with local and national laws and regulations related to transportation services and payment processing, using their own regulatory devices.

6. Third-party service providers

   Provide additional services that complement the Bus Commute Companion app, such as in-app advertising, location-based services, or analytics, using their own devices.

7. Investors

   Invest in the bus company and the Bus Commute Companion app and provide financial support for the development and growth of the business, using their own investment devices.

7. USE CASES

Casual Description

1. **UC-1. Register:** This use case allows a passenger to register on the Bus Commute Companion website or application by providing their personal information, such as their name, email address, and contact number. After successful registration, the passenger can access the website's features that require login.

   - ➔ Functional Requirements: REQ1
   - ➔ Non Functional Requirements: NONREQ1, NONREQ2, NONREQ3, NONREQ4, NONREQ5, NONREQ6
   - ➔ On-Screen Appearance Requirements: ONSREQ6

2. **UC-2. Log in:** A registered passenger can log in to the website by entering their username and password. This use case verifies the user's credentials and allows them to access their account details and booking history.

   - ➔ Functional Requirements: REQ10
   - ➔ Non Functional Requirements: NONREQ1, NONREQ2, NONREQ4, NONREQ5, NONREQ6
   - ➔ On-Screen Appearance Requirements: ONSREQ6

3. **UC-3. View Profile:** This use case enables the passenger to view their profile details, such as their name, profile photo, email address, and contact number. The passenger can also view their booking history and other account information.

   - ➔ Functional Requirements: REQ1
   - ➔ Non Functional Requirements: NONREQ1, NONREQ4, NONREQ5, NONREQ6
   - ➔ On-Screen Appearance Requirements: ONSREQ5, ONSREQ6

4. **UC-4. Edit Profile:** This use case enables the passenger to edit their profile information, such as their name, profile photo, email address, and contact number. The passenger can update their personal details and save the changes to their profile.

   - ➔ Functional Requirements: REQ1
   - ➔ Non Functional Requirements: NONREQ1, NONREQ2, NONREQ4, NONREQ5, NONREQ6
   - ➔ On-Screen Appearance Requirements: ONSREQ5, ONSREQ6

5. **UC-5. Search Trip:** The use case enables the passenger to search for available bus trips by selecting the departure and arrival locations, dates, and other filters, such as the number of passengers and preferred travel class. The system displays the available options based on the search criteria.

- ➔ Functional Requirements: REQ4
- ➔ Non Functional Requirements: NONREQ1, NONREQ4, NONREQ5, NONREQ6
- ➔ On-Screen Appearance Requirements: ONSREQ1, ONSREQ6

6. **UC-6. Select Trip:** After searching for trips, the passenger can select a preferred option and view the trip details, including the route, schedule, and ticket price. Then the passenger begins the process of purchasing a ticket for the specific trip .

- ➔ Functional Requirements: REQ4
- ➔ Non Functional Requirements: NONREQ1, NONREQ4, NONREQ5, NONREQ6
- ➔ On-Screen Appearance Requirements: ONSREQ1, ONSREQ2, ONSREQ6

7. **UC-7. Select Seat:** After selecting a trip, the passenger can choose a seat from the available options on the bus and proceed to the payment page. This use case displays the seating layout and allows the passenger to select their preferred seat.

- ➔ Functional Requirements: REQ4
- ➔ Non Functional Requirements: NONREQ1, NONREQ4, NONREQ5, NONREQ6
- ➔ On-Screen Appearance Requirements: ONSREQ1, ONSREQ2, ONSREQ6

8. **UC-8. Choose Payment:** This use case enables the passenger to pay for the selected trip using a secure online payment gateway. The passenger enters their payment details, and the system verifies and processes the payment.

- ➔ Functional Requirements: REQ2, REQ8
- ➔ Non Functional Requirements: NONREQ1, NONREQ2, NONREQ4, NONREQ5, NONREQ6, NONREQ7, NONREQ9
- ➔ On-Screen Appearance Requirements: ONSREQ1, ONSREQ2, ONSREQ3, ONSREQ6

9. **UC-9. View Payment History:** This use case enables the passenger to view their payment history, including all payments made for past and current tickets.

- ➔ Functional Requirements: REQ2, REQ8

- Non Functional Requirements: NONREQ1, NONREQ2, NONREQ4, NONREQ5, NONREQ6, NONREQ7
- On-Screen Appearance Requirements: ONSREQ1, ONSREQ2, ONSREQ6

10. **UC-10. View Booking History:** This use case enables a passenger to view their booking history, including all past and current tickets, cancellations, and modifications.

- Functional Requirements: REQ3, REQ4
- Non Functional Requirements:NONREQ1, NONREQ2, NONREQ4, NONREQ5, NONREQ6
- On-Screen Appearance Requirements: ONSREQ1, ONSREQ5, ONREQ6

11. **UC-11. Cancel Ticket:** This use case enables the passenger to cancel their current ticket, which will cancel their ticket and refund any applicable fare based on the cancellation policy.

- Functional Requirements: REQ4, REQ13
- Non Functional Requirements: NONREQ1, NONREQ4, NONREQ5, NONREQ6, NONREQ9
- On-Screen Appearance Requirements: ONSREQ1, ONSREQ5, ONSREQ6

12. **UC-12. Reschedule Ticket:** This use case enables a passenger to modify the date, time, and destination of their ticket, subject to availability and any applicable change fees.

- Functional Requirements: REQ4, REQ14
- Non Functional Requirements: NONREQ1, NONREQ2, NONREQ4, NONREQ5, NONREQ6
- On-Screen Appearance Requirements: ONSREQ1, ONSREQ4, ONSREQ5, ONSREQ6

13. **UC-13. Generate e-Ticket:** This use case enables a passenger to generate an e-ticket for their current ticket, which they can print or save as a digital copy to their device.

- Functional Requirements: REQ7
- Non Functional Requirements:NONREQ1, NONREQ4, NONREQ5, NONREQ6
- On-Screen Appearance Requirements: ONSREQ6

14. **UC-14. Provide Feedback and Rating:** This use case enables the passenger to provide feedback and rating for their trip experience, including the bus condition, driver behaviour, and overall satisfaction. The passenger can rate various aspects of the trip and provide comments to help the bus company improve its services.

➔ Functional Requirements: REQ15
➔ Non Functional Requirements:NONREQ1, NONREQ4, NONREQ5
➔ On-Screen Appearance Requirements: ONSREQ6

15. **UC-15 Change Notification Settings:** This use case enables a passenger to set up notifications for their current ticket, including reminders of the trip date and time, and any changes or updates to the ticket.

➔ Functional Requirements: REQ11
➔ Non Functional Requirements:NONREQ1, NONREQ4, NONREQ5
➔ On-Screen Appearance Requirements: ONSREQ4

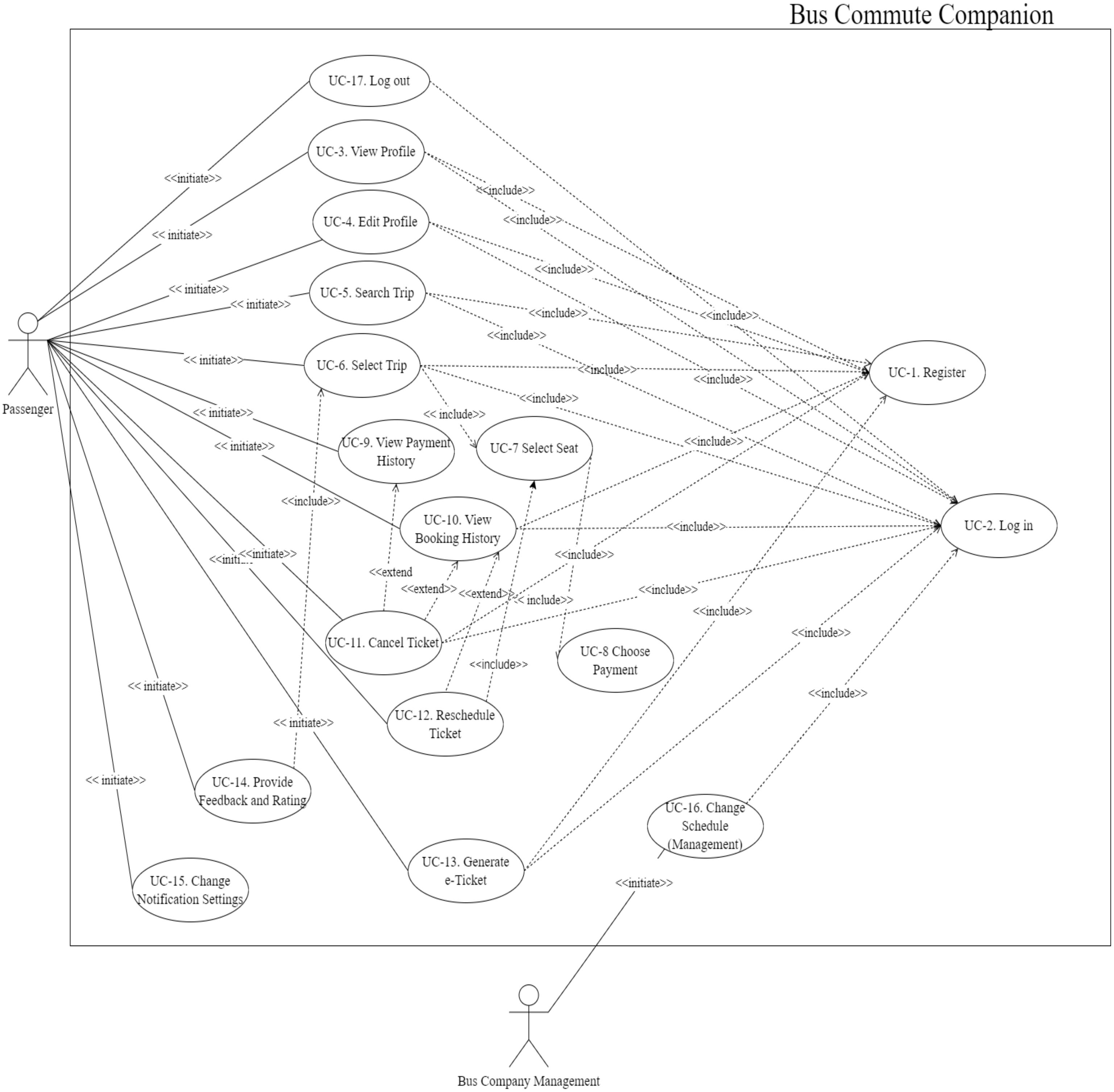
16. **UC-16. Change Schedule (Management):** This use case provides the bus company management to adjust the arrival time and departure time for the buses.

➔ Functional Requirements: REQ5
➔ Non Functional Requirements: NONREQ1, NONREQ2, NONREQ5, NONREQ6
➔ On-Screen Appearance Requirements:ONSREQ6

17. **UC-17. Log out:** This use case allows the passenger to log out of their account and end their session on the website or application. The passenger can log in again later to access their account and make new bookings.

➔ Functional Requirements: REQ12
➔ Non Functional Requirements: NONREQ1, NONREQ2, NONREQ4, NONREQ5, NONREQ6
➔ On-Screen Appearance Requirements: ONSREQ6

8. Case Diagram



9. Traceability Matrix

Requirements	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12	UC-13	UC-14	UC-15	UC-16	UC-17
REQ1	5		5	5													
REQ2								4	4								
REQ3										4							
REQ4					5	5	5			5	5	5					
REQ5																4	
REQ6																	
REQ7													5				
REQ8								5	5								
REQ9																	
REQ10		2															
REQ11															1		
REQ12																	2
REQ13											2						
REQ14												3					
REQ15														5			
NONREQ1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
NONREQ2	5			5				5	5	5		5	5			5	5
NONREQ3	5	5															
NONREQ4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3		3
NONREQ5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
NONREQ6	5	5	5	5	5	5	5	5	5	5	5	5	5			5	5
NONREQ7								3	3								
NONREQ8																	
NONREQ9								2			2						
ONSREQ1					5	5	5	5	5	5	5	5					
ONSREQ2						5	5	5	5								
ONSREQ3								3									
ONSREQ4												3			3		

ONSREQ5			3	3	3					5	3	3					
ONSREQ6	5	5	5	5		5	5	5	5	5	5	5	5	5		5	5
Total Weight	**36**	**28**	**29**	**34**	**29**	**36**	**36**	**55**	**48**	**43**	**37**	**45**	**31**	**21**	**15**	**27**	**28**

10. Fully Dressed Description

The fully-dressed description was performed in the use cases that obtained a high priority weight in the traceability matrix. These consist of UC-1, UC-6, UC-9, UC-10, & UC-12. UC-7. UC-7 & UC-8 were not done as they are required for UC-6.

<table>
<tr><th>Use Case #1</th><th>Register</th></tr>
<tr><td><b>Initiating Actor</b></td><td>Passenger</td></tr>
<tr><td><b>Actor's Goal</b></td><td>Register to the system</td></tr>
<tr><td><b>Participating Actor</b></td><td>System & Database.</td></tr>
<tr><td><b>Preconditions</b></td><td>1. The website is running
2. The device is connected to the internet.
3. The passenger has the website open.</td></tr>
<tr><td><b>Post Conditions</b></td><td>1. The user is now registered to the system.</td></tr>
<tr><td colspan="2"><b>Flow of the events for main success scenario</b>
□ 1. The passenger clicks the “Register Now ” button.
□ 2. The system displays the registration page.
□ 3. The passenger (a)fills out the necessary requirements in the form and (b) selects “Register”.
□ 4. The system (a) adds the user credentials in the database and (b) the passenger is now able to access the system. (c) The passenger is redirected to the login page.</td></tr>
<tr><td colspan="2"><b>Events for extensions (Alternate Scenarios):</b>
3a. Passenger fails to fill out required in formation
□ System (a) detects missing information, and (b) signals to the passenger of missing information
4a. System fails to upload credentials.
□System (a) signals error and (b) displays registration with note to try again.
□ The passenger fills out the necessary requirements in the form and selects “Register”.
□ The system (a) adds the user credentials in the database and (b) the passenger is now able to access the system. (c) The passenger is redirected to the login page.</td></tr>
</table>

<table>
<tr><td><b>Use Case #6</b></td><td><b>Select Trip</b></td></tr>
<tr><td><b>Initiating Actor</b></td><td>Passenger</td></tr>
<tr><td><b>Actor's Goal</b></td><td>Select a trip to purchase a ticket.</td></tr>
<tr><td><b>Participating Actor</b></td><td>System, & Bank.</td></tr>
<tr><td><b>Preconditions</b></td><td>1. The website is running.
2. The device is connected to the internet.
3. The passenger is on the website.
4. The passenger is already registered.
5. The passenger is already logged in.
6. The passenger has determined which ticket to purchase.</td></tr>
<tr><td><b>Post Conditions</b></td><td>1. The user has purchased a ticket</td></tr>
<tr><td colspan="2"><b>Flow of the events for main success scenario</b>
□1. The passenger clicks the “ Select” button.
□ 2. The system displays the “Select Seat “page.
□ 3. The passenger clicks the preferred seat.
□ 4. The system shows two payment options available: Credit Card or Pay on Arrival.
□ 5. The passenger selects the credit card option.
□ 6. The system prompts for credit card information.
□ 7. The passengers fills in credit card information and select submit.
□ 8. System passes card information and payment amount to the bank for authorization.
□ 9. Bank processes transactions.</td></tr>
<tr><td colspan="2"><b>Flow of events for extensions (Alternate Scenarios):</b>
4. Passenger selects Pay on Arrival
□ 1. System displays the total cost of the ticket and instructions on how to pay on arrival.
□ 2. Passenger closes prompt.</td></tr>
</table>

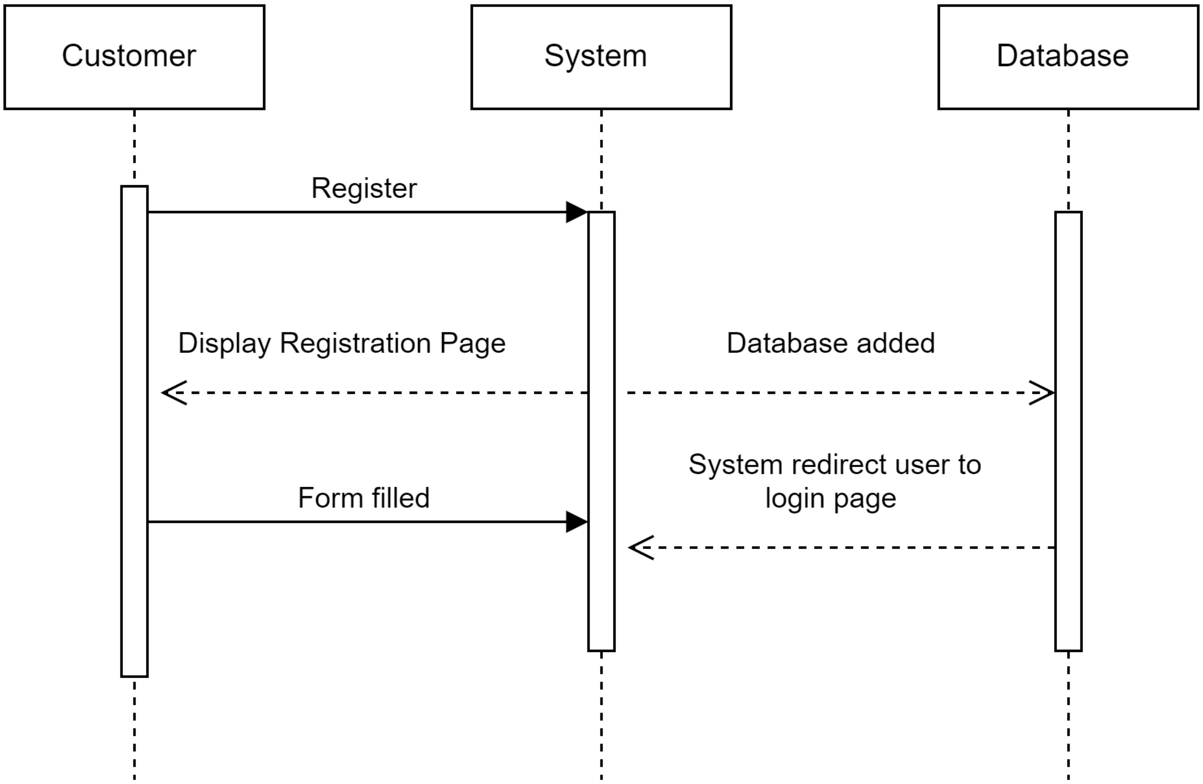
Use Case #9	**View Payment History**
Initiating Actor	Passenger
Actor's Goal	View previous and current payments made.
Participating Actor	System & Database
Preconditions	1. The website is running. 2. The device is connected to the internet. 3. The passenger is on the website. 4. The passenger is already registered. 5. The passenger is already logged in. 6. The passenger has purchased tickets previously
Post Conditions	1. The system shows payment history. 2. The system calculates total($) spent.
Flow of the events for main success scenario □ 1. The passenger selects the Account drop-down list and clicks “View History.” □ 2. The system displays the “View History” page that displays the passenger's transaction history(including the “View Booking History”).	
Flow of events for extensions (Alternate Scenarios): No alternate flow.	

Use Case #10	**View Booking History**
Initiating Actor	Passenger
Actor's Goal	View previous and current tickets purchased.
Participating Actor	System & Database
Preconditions	1. The website is running. 2. The device is connected to the internet. 3. The passenger is on the website. 4. The passenger is already registered. 5. The passenger is already logged in. 6. The passenger has purchased tickets before. 7. The passenger is before the time cutoff in which tickets are not purchasable
Post Conditions	1. The passenger is able to view their purchase history. 2. The system shall show the detail of present booking
Flow of the events for main success scenario □ 1. The passenger clicks the “View Booking History” button. □ 2. The system displays the passenger Booking History (including “Cancel Ticket). □ 3. The passenger views the pass and current purchase, cancellations, and modifications.	
Events for extensions (Alternate Scenarios): 3. Passenger selects “Cancel Ticket”.	

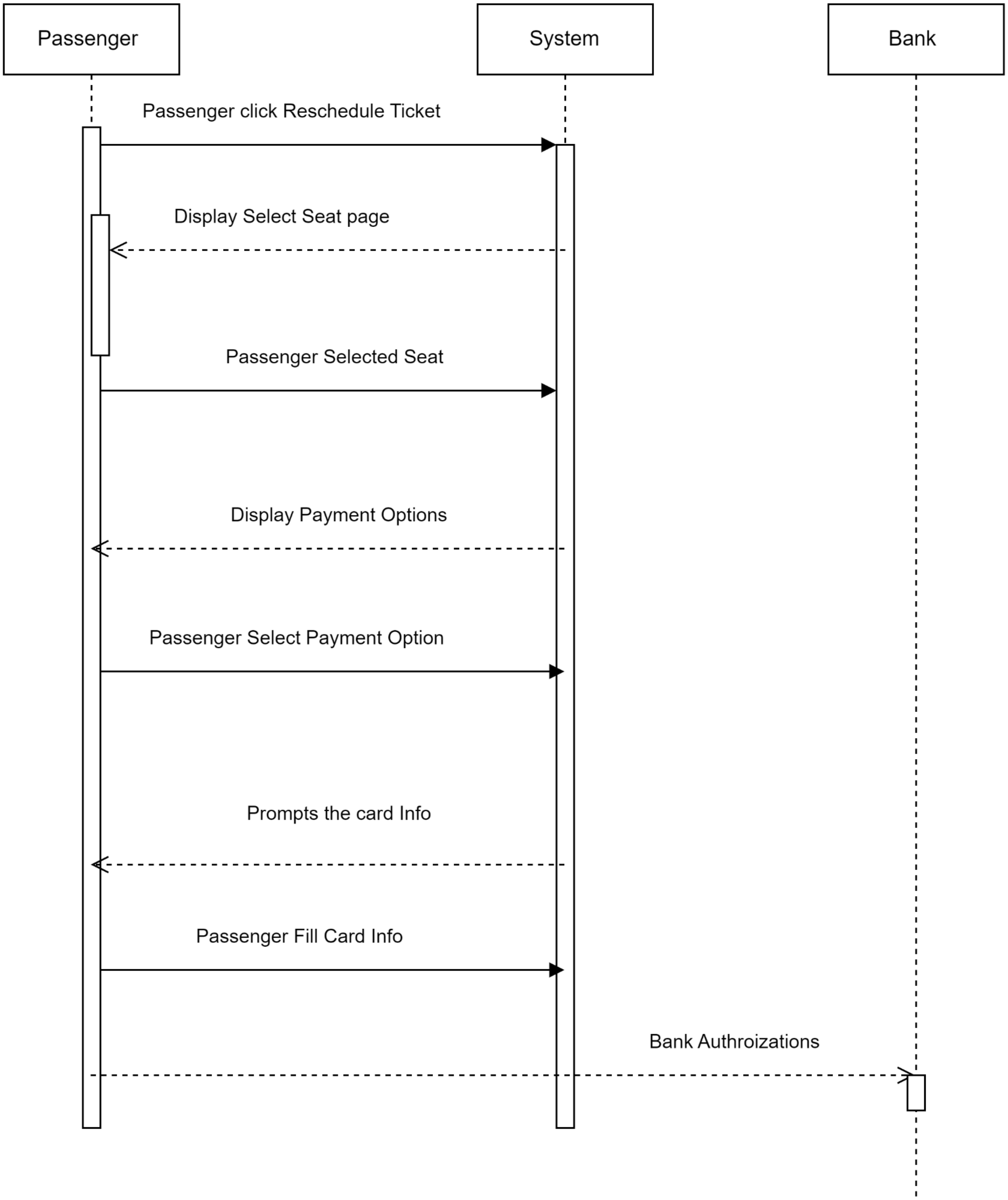
Use Case #12	**Reschedule Ticket**
Initiating Actor	Passenger
Actor's Goal	Reschedule Ticket
Participating Actor	System & Database
Preconditions	1. The website is running. 2. The device is connected to the internet. 3. The passenger is on the website. 4. The passenger is already registered. 5. The passenger is already logged in. 6. The passenger has purchased a ticket previously. 7. The passenger is before the time cutoff in which tickets are not purchasable 8. The passenger has decided on which ticket to reschedule.
Post Conditions	1. The user has successfully rescheduled the ticket. 2. The appropriate amount of funds have been added or refunded.
Flow of the events for main success scenario □ 1. The passenger clicks the “ Reschedule Ticket” button. □ 2. The system redirects to “Select Seat” page □ 3. The passenger clicks the preferred seat. □ 4. The system shows two payment options available: Credit Card or Pay on Arrival. □ 5. The passenger selects the credit card option. □ 6. The system prompts for credit card information. □ 7. The passengers fill in credit card information and selects submit. □ 8. System passes card information and payment amount to the bank for authorization. □ 9. Bank processes transactions.	
Events for extensions (Alternate Scenarios): □2a. The passenger selects the cancel option □ System displays confirm prompt □ Passenger confirms yes.	

11. System Sequence Diagrams

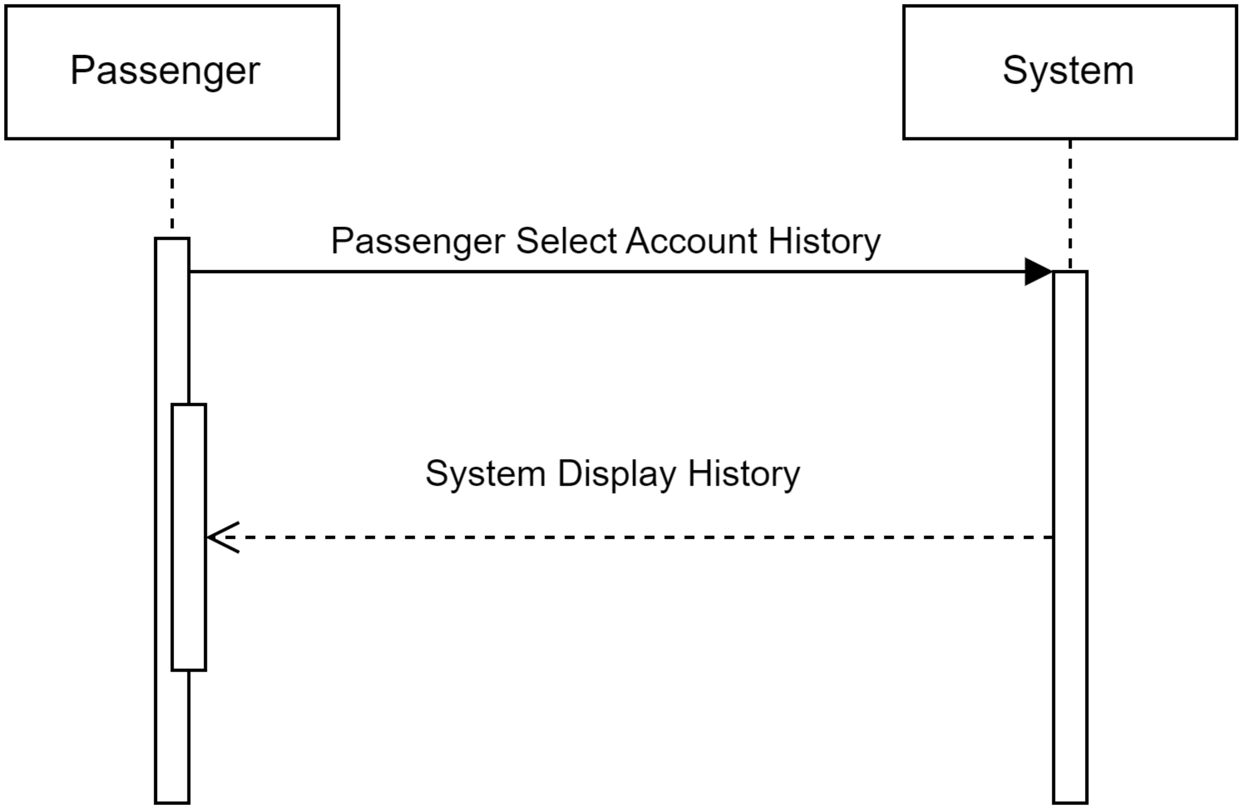
- Use case UC-1



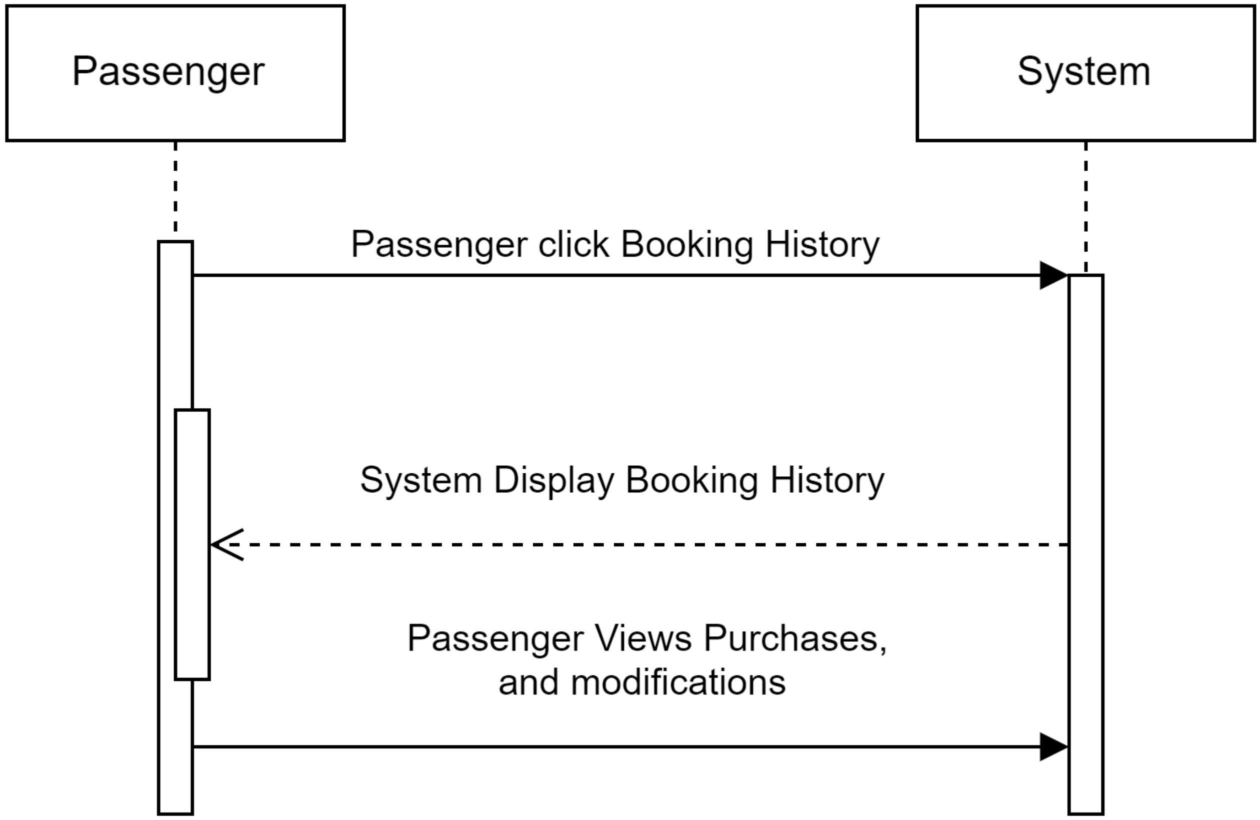
- Use Case UC-6


Passenger
System
Bank
Passenger click Reschedule Ticket
Display Select Seat page
Passenger Selected Seat
Display Payment Options
Passenger Select Payment Option
Prompts the card Info
Passenger Fill Card Info
Bank Authroizations

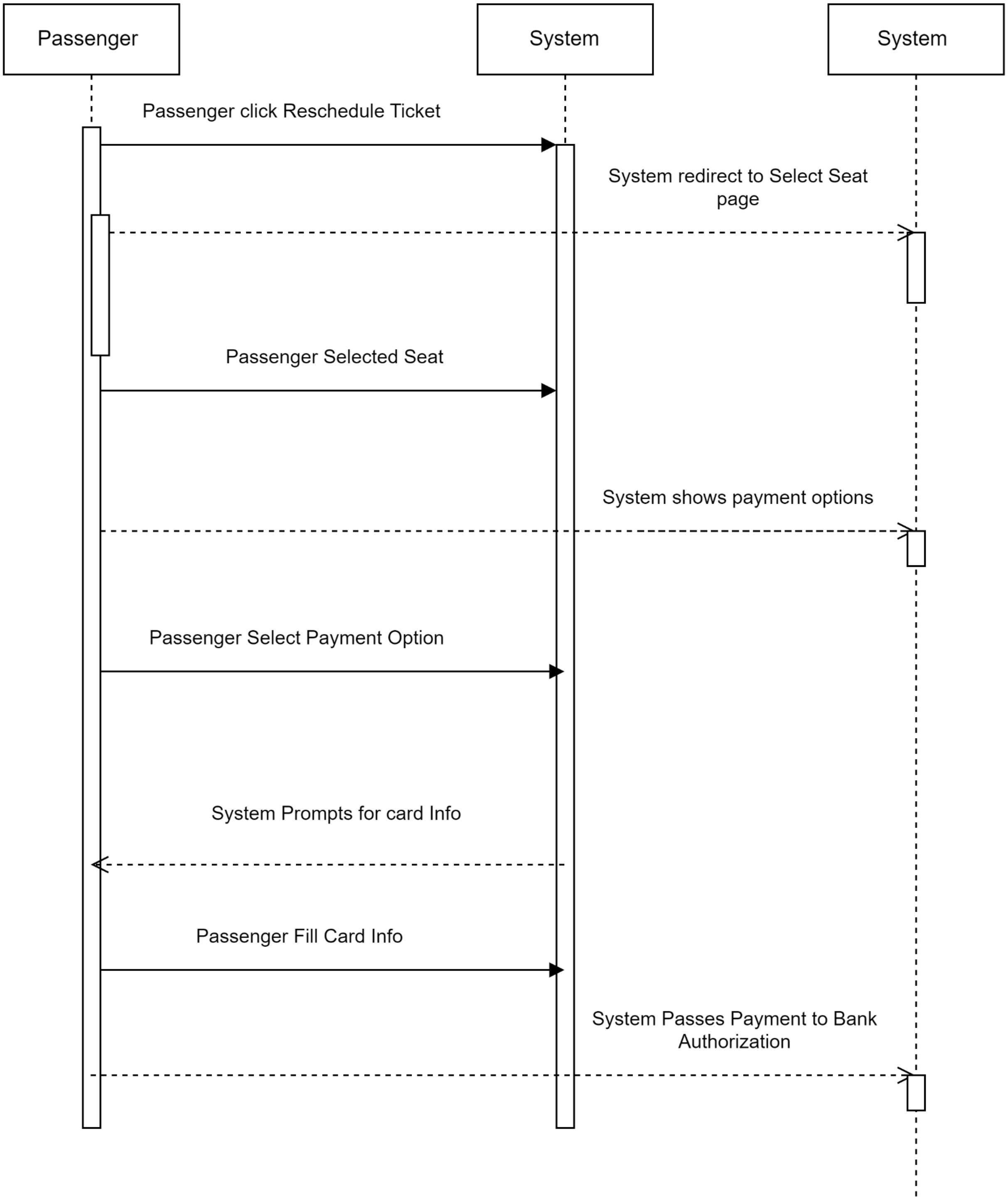
- Use Case UC-9


Passenger
System
Passenger Select Account History
System Display History

- Use Case UC -10

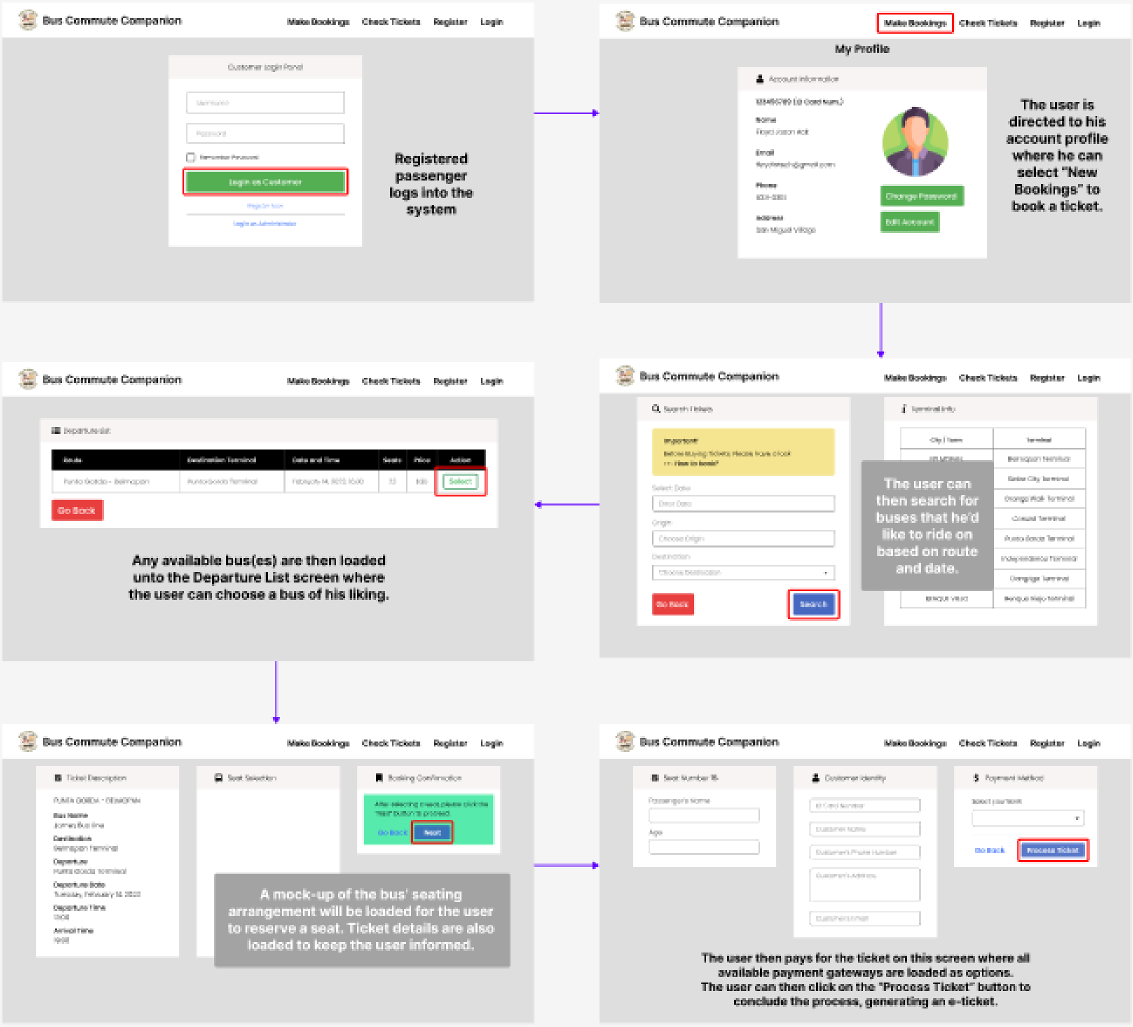

Passenger
System
Passenger click Booking History
System Display Booking History
Passenger Views Purchases,
and modifications

- Use Case UC -12


Passenger
System
System
Passenger click Reschedule Ticket
System redirect to Select Seat page
Passenger Selected Seat
System shows payment options
Passenger Select Payment Option
System Prompts for card Info
Passenger Fill Card Info
System Passes Payment to Bank Authorization

12. USER INTERFACE SPECIFICATION

Steps to Book a Ticket



Registered passenger logs into the system

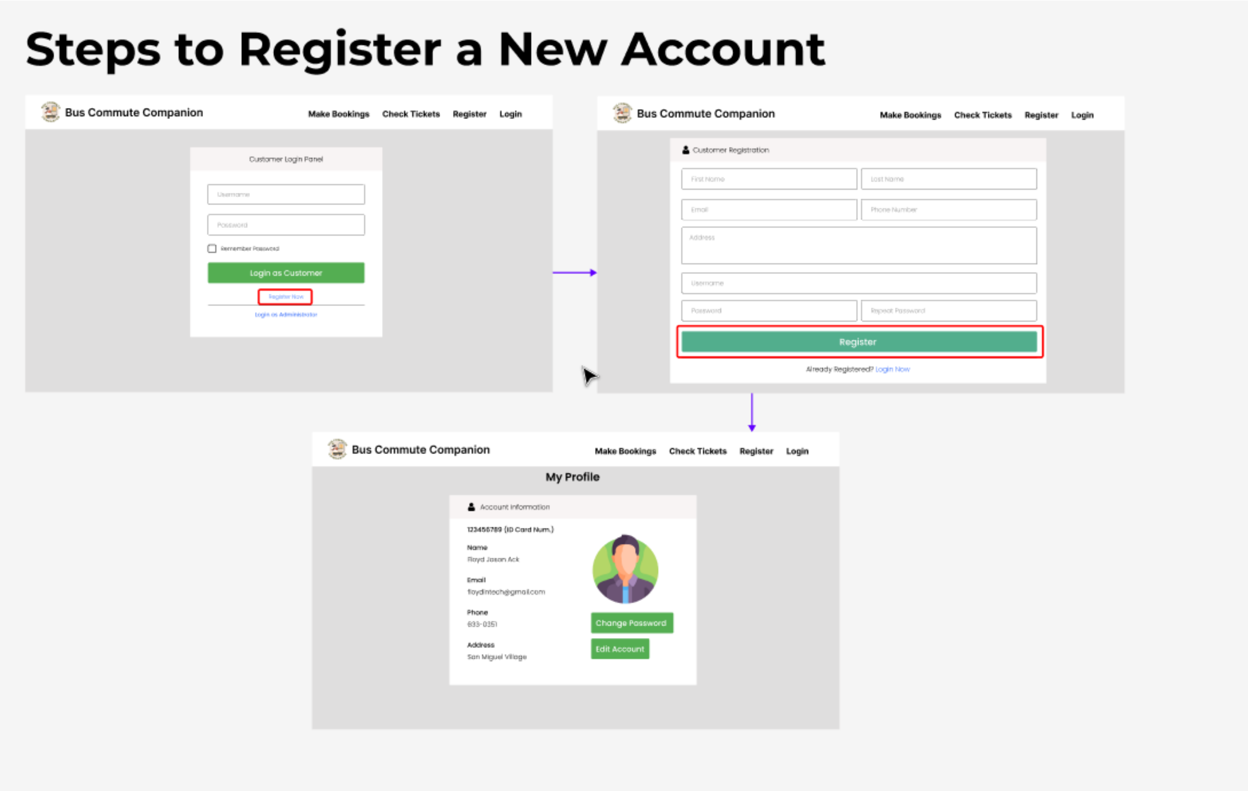
The user is directed to his account profile where he can select "New Bookings" to book a ticket.

The user can then search for buses that he'd like to ride on based on route and date.

Any available bus(es) are then loaded unto the Departure List screen where the user can choose a bus of his liking.

A mock-up of the bus' seating arrangement will be loaded for the user to reserve a seat. Ticket details are also loaded to keep the user informed.

The user then pays for the ticket on this screen where all available payment gateways are loaded as options.
The user can then click on the "Process Ticket" button to conclude the process, generating an e-ticket.



Steps to Register a New Account

123456789 (ID Card Num.)

Name
Floyd Jason Ack

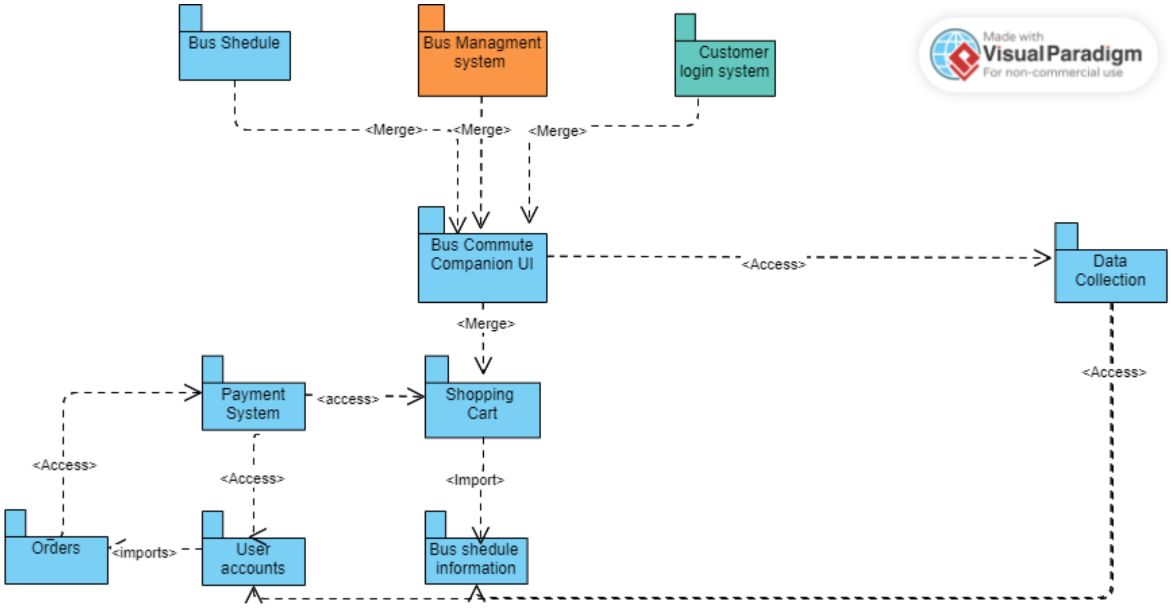
Email
floydintech@gmail.com

Phone
633-0351

Address
San Miguel Village

1. SYSTEM ARCHITECTURE

1.1. Identifying Subsystems



1.2. Architecture Styles

This bus ticketing system will implement three architecture styles: Software AS A Service, Real-time Computing, and Data-centric architectures.

Software As A Service(SAAS): This system will be hosted online so the passengers can use the web-based application on their devices. The server for the system will be hosted online. The users will only need an internet connection, and they will be able to use the ticketing system.

Data-Centric and real-time computing (RTC): This ticketing system is significantly data-driven, and all data will be saved on a database. Every detail will be saved on the database, and the database will be updated in real-time. There will not be any delays. It is crucial to the system that the database is updated as the user makes a change so that no clashes happen. This system needs to have high reliability and availability.

1.3. Mapping Subsystems to Hardware

The server will handle tickets that will be issued to the customers.

The server will also handle the flow of payments for the respected bus companies.

The server will also help gather Customer data(the routes they take, What times they usually take the bus) this will help the traffic department and by extension the government when planning new routes and the addition of new bus companies.

1.4. Connectors and Network Protocols

HTTPS: This application will be hosted on the web. Since our web application will be processing transactions and be a place where users log in and input their personal information, this website must be hosted on a secured website.

PHP: It will be used to establish the website and server communication and when someone is connecting via Computer/Phone via Browser.

AJAX PHP: When establishing the communication between the app and the server. When someone is connecting via a smartphone.

TCP: This service will be using the TCP protocol to have a reliable transmission of data over the internet when communicating between the server and the web application/ smartphone application.

1.5. Global Control Flow

Execution Orderness: For this bus ticketing system, the execution orderliness is event-driven. This will allow the users to use and execute any of the subsystems on the system, be it reserving a seat or canceling a seat. For this to work, the server will be in a loop, constantly waiting for the events to be executed by the users.

Time dependency: For this system, the time dependence is real-time dependency. As stated above, the server will always be waiting for any updates. This is the most important part of the system because the available seats must be updated as they become accessible, and this must be shown on the application.

1.6. Hardware Requirements

Mobile Requirements:

OS: Android 4.0 or greater ; Apple IOS 6.0 or greater

Storage : 2GB

Screen Resolution: Minimum Screen Resolution of 640 X 480

Network Bandwidth: 60 kbs

PC Requirements:

OS: Any os that can run the following software below

Web Browsers: Chrome, Version 4 or later
Firefox, Version 4 or later
Safari, Version 4 or later
Opera, Version 4 or later

Network: Broadband Internet connection is required

Plan of work

We divided our project into three main sections as a group. After submitting our initial report, our top priority will be to begin building the main interface, which is crucial for attracting users. All of our main pages, which act as the bus ticket management system's primary user interface, will be designed and built by us. Each group member will receive a particular page that they will be responsible for designing and building. As the database is essential for putting functionalities into action, we will begin designing and building it next. The final step will be to code and put into action all of our site's key features. Throughout the semester, we will divide the report into equal portions and continue to do so.

References

Hello Joe Weaver - JustRide App. (n.d.). https://hellojoeweaver.com/justride