
Software Requirements Specification for “Talk2Metro”

Prepared by

Md. Sajib Uddin

ID: ASH2125001M

Tarekul Islam Tusher

ID: ASH2125003M

Md. Imran Hasan

ID: MUH2125020

Nowshin Fabiha Ibnat

ID: BKH2125031F

**Institute of Information Technology
Noakhali Science and Technology University**

Table of Contents

1. Introduction.....	4
1.1 Problem Statement.....	4
1.2 Purpose.....	4
1.3 Project Scope	4
1.4 Glossary	5
1.5 References	5
1.6 Overview	5
2. Stakeholders and Characteristics	5
2.1 Commuters (End Users).....	6
2.2 Metro-Rail Authorities	6
2.3 Developers.....	6
2.6 Customer Support Team	6
3. Design and Implementation Constraints	6
3.1 Language.....	7
3.1.1 Flutter (Framework).....	7
3.1.2 Dart	7
3.1.3 Firebase	7
4. Requirement Specification	7
4.1. Functional Requirements	7
4.1.1 User Authentication.....	7
4.1.2 Route Mapping.....	7
4.1.3 Ticket Booking.....	8
4.1.4 Ticket Generation and Bar Code.....	8
4.1.5 User Interface	8
4.2. Non-Functional Requirements.....	8
4.2.1 Performance Requirements	8
4.2.2 Security Requirements	8
4.2.3 Usability Requirements.....	8
4.2.4 Scalability Requirements	8
4.2.5 Reliability Requirements.....	9
4.2.6 Compatibility Requirements	9
5. Requirement Engineering Process.....	9
5.1 Requirement Elicitation Techniques	9

5.1.1 Hold Interviews.....	9
5.1.2 Perform Document Analysis	10
5.2 Requirement Validation.....	11
5.2.1 Review the Requirements	11
5.2.2 Test the Requirements	11
5.2.3 Simulate the Requirements	11
6. Use Case Diagram	12
7. Use Case Descriptions	13
8. Activity Diagram	18
Figure:8.1 Book Ticket	18
Figure:8.2 Metro Map	19
Figure:8.3 Search Train.....	19
Figure:8.4 Current Location	20
Figure:8.5 OTP validation.....	20
Figure:8.6 Bar Code Generator	20
Figure:8.7 Download Ticket.....	21
Figure:8.8 Payment.....	21
9. Sequence Diagram	22
Figure:9.1 Book Ticket	22
Figure:9.2 Metro Map	22
Figure:9.3 Search Train.....	23
Figure:9.4 Current Location	23
Figure:9.5 OTP validation.....	24
Figure:9.6 Bar Code Generator	25
Figure:9.7 Download Ticket.....	25
Figure:9.8 Payment.....	26
10. Swimlane Diagram.....	27
11. Appendix	28
11.1.1 Three-level Scale.....	28
11.1.2 Prioritization of the requirements of Talk2Metro	28

1. Introduction

The Software Requirements Specification (SRS) introduction contains the SRS's policy, scope, references, and summary. This document's goal is to gather information about proposed system as we name it as “Talk2Metro” and is to give readers a greater understanding of it by outlining the issue statement in great detail. While defining the qualities of a high-quality product, it also emphasizes the advantages and requirements of the participants. Details on the “Talk2Metro” can be found on this document.

1.1 Problem Statement

Urban commuters often face significant challenges with traditional metro-rail ticket booking methods, such as long queues, limited ticket availability etc. In response to these issues, there is a pressing need for a more efficient, user-friendly solution. The problem is to develop a web-based mobile application that allows users to seamlessly book metro-rail tickets, displays nearby stations based on the user’s location, provides route maps to the selected station and facilitates secure online payments. This system must address key challenges such as ensuring accurate precise location services, secure transactions, and a scalable, intuitive user interface.

1.2 Purpose

The purpose of this project is to enhance the metro-rail ticket booking experience by developing a mobile application that leverages modern technologies to provide a seamless, efficient, and user-friendly process. By incorporating route mapping, the application aims to eliminate the inconveniences of traditional ticket booking methods. It will provide users with the convenience of booking tickets from anywhere, receiving instant confirmation through Bar code, and staying informed with real time journey updates. This project seeks to improve overall commuter satisfaction, reduce wait times, and streamline the ticketing process, ultimately contributing to a more efficient and accessible urban transportation system.

1.3 Project Scope

User Authentication

- Authentication via OTP.

Location-Based Services

- Automatic detection of the user’s current location
- Display of nearby metro stations based on user’s location

Ticket Booking & Bar Code Generation

- Generation of Bar code for ticket validation post-payment
- Storage and retrieval of ticket information

User Interface Design

- Responsive and intuitive UI/UX design for mobile devices
- Seamless navigation and user-friendly interface

System Security and Scalability

- Ensuring data security and privacy for users
- Designing the system to handle high user volumes efficiently

Testing and Quality Assurance

- Unit testing, integration testing, and user acceptance testing
- Continuous monitoring and feedback for improvements

Deployment and Maintenance

- Deployment of the app on relevant app stores and web hosting platforms
- Ongoing maintenance and updates based on user feedback and technological advancements.

1.4 Glossary

This section provides definitions for all document names, acronyms, and abbreviations. The application domain's terms and concepts are defined.

SRS – Software Requirement Specification

UI – User Interface

API – Application Programming Interface

1.5 References

IEEE. IEEE Std. 830 – 1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

1.6 Overview

In “Talk2Metro” project, we are excited to introduced some innovative features here. Experience the future of metro rail with Bar Code-Based Ticket Validation.

2. Stakeholders and Characteristics

The term “stakeholder” refers to the people or groups affected by a software development project. Stakeholders exist both within the organization and outside of it. They may be the end users, or

they might simply be affected by the process. Either way they have a vested interest in the final product. Here are the stakeholders of “Talk2Metro”:

2.1 Commuters (End Users)

Characteristics: Individuals who frequently use metro-rail services, including students, professionals, and tourists. They seek convenience, reliability, and ease of use in booking tickets and receiving journey updates.

Needs: User-friendly interface, quick and easy booking process.

2.2 Metro-Rail Authorities

Characteristics: Organizations responsible for managing metro-rail operations and services. They are interested in improving service efficiency and passenger satisfaction.

Needs: Efficient ticketing system, reduced congestion at ticket counters, accurate passenger data, and improved customer service.

2.3 Developers

Characteristics: Technical team responsible for designing, developing, testing, and maintaining the application. They require clear requirements, robust development tools, and efficient collaboration.

Needs: Well-defined project scope, access to necessary APIs and SDKs, support for integration with payment gateways, and clear communication with stakeholders.

2.6 Customer Support Team

Characteristics: Team responsible for assisting users with issues related to the application, payments, or travel updates. They require tools to manage user queries and feedback efficiently.

Needs: Access to user data and transaction history, efficient communication tools, and training on the application’s features and functionalities.

3. Design and Implementation Constraints

In order to ensure the project's success, we used design and implementation limitations. It can also refer to a tool that enables testers and developers to view and interact with the user interface (UI) components of an application.

3.1 Language

User interface Design, usually known as UI Design, is the visual organization of the parts of a website or technological product that a user could interact with. In other words, it is the visual layout of a website. On the other hand, the code that enables a computer program or application to run and cannot be viewed by a user is referred to as the back end. The back end of a computer system is where the majority of data and operating syntax are kept and accessed. Typically, the code is comprised of one or more programming languages.

3.1.1 Flutter (Framework)

Flutter is Google’s Mobile SDK to build native iOS and Android apps from a single codebase. It is a toolkit created by Google that lets developers build apps for mobile, web, and desktop using the same code. It uses the Dart programming language.

3.1.2 Dart

Dart is an open-source general-purpose programming language developed by Google. It supports application development on both the client and server side. However, it is widely used for the development of Android apps, iOS apps, IoT(Internet of Things), and web applications using the Flutter framework.

3.1.3 Firebase

Firebase is a comprehensive app development platform provided by Google. It offers a suite of tools and services that help developers build, deploy, and grow their mobile (Android/iOS) and web applications quickly and efficiently. The key features of Firebase include a real-time database, authentication, cloud storage, hosting, and cloud functions, which allow developers to focus on building great user experiences without worrying about the underlying infrastructure.

4. Requirement Specification

Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

4.1. Functional Requirements

Functional requirements define what a product must do and what its features and functions are. Nonfunctional requirements describe the general properties of a system. They are also known as quality attributes.

4.1.1 User Authentication

- Users must be able to register with their Mobile number.

4.1.2 Route Mapping

- The application must provide a visual route map from the user's current location to the selected station.
- The route map must update in real-time as the user moves.

4.1.3 Ticket Booking

- Users can book ticket maximum 10 days ago.
- After setting the destination and date, users can pay.

4.1.4 Ticket Generation and Bar Code

- Upon successful payment, the application must generate a QR code containing ticket details.
- The QR code must be unique and valid for the selected journey.

4.1.5 User Interface

- The application must have a responsive and intuitive user interface suitable for mobile devices.
- The interface must support both portrait and landscape orientations.

4.2. Non-Functional Requirements

These are requirements that specify criteria that can be used to judge the operation of the system rather than specific behaviors. They are usually architecturally significant requirements. The extent to which these factors are implemented varies from project to project. They are also known as non-behavioral requirements.

4.2.1 Performance Requirements

- The application must respond to user inputs within 2 seconds.
- The system should handle up to 10,000 concurrent users without performance degradation.

4.2.2 Security Requirements

- User data must be encrypted in transit and at rest.
- The application must comply with relevant data protection regulations.
- Payment transactions must be processed through secure, PCI-compliant gateways.

4.2.3 Usability Requirements

- The application must be easy to navigate with clear instructions and help options.
- The voice control system must have a high accuracy rate (95%+) in recognizing commands.

4.2.4 Scalability Requirements

- The application must be scalable to handle increasing user loads.
- The system architecture must support easy addition of new features and services.

4.2.5 Reliability Requirements

- The application must have an uptime of 99.9%.
- The system must recover gracefully from failures and provide meaningful error messages to users.

4.2.6 Compatibility Requirements

- The application must be compatible with major mobile operating systems (iOS and Android).
- The application must work on various screen sizes and resolutions.

5. Requirement Engineering Process

A systematic and strict approach to the definition, creation, and verification of requirements for a software system is known as requirements engineering. To guarantee the effective creation of a software product, the requirements engineering process entails several tasks that help in understanding, recording, and managing the demands of stakeholders.

5.1 Requirement Elicitation Techniques

Requirements elicitation involves the process of researching and gathering system requirements from various stakeholders, including users, customers, and administrators. Different techniques are employed to effectively gather these requirements for the “Talk2Metro” project.

5.1.1 Hold Interviews

We hold discussions that can be held individually or with a small group of participants. They are an effective way to access services without spending a lot of time with participants because we meet with people to discuss only certain important requirements of this program. Negotiations are useful for obtaining individual requirements for members in organizing workshops where those members of the program come together to resolve any issues or conflicts. We mainly perform our interview based on some specific criteria.

Sample of Requirements Collection:

Requirement Elicitation Techniques	Interviews, Surveys, Focus Groups
Collected From	Users of Metro rail

Findings	<p>1. User Authentication: Users must register with their Mobile number. The system will validate the Mobile number by sending an OTP.</p> <p>2.Ticket Generation and Bar Code: Upon successful payment, the application must generate a unique Bar code containing the ticket details. The Bar code should be scannable at the metro station for validation. The generated Bar code must be unique and valid only for the selected journey. It should include information such as the journey date, starting station, destination, and ticket class.</p> <p>3.Responsive and Intuitive User Interface: The application must have a responsive and intuitive user interface optimized for mobile devices. It should support smooth navigation and user-friendly interactions.</p>
-----------------	--

5.1.2 Perform Document Analysis

Document analysis is crucial for understanding existing systems, processes, and requirements. Here's how we can perform document analysis effectively for the "TALK2METRO" project:

- **Gather Existing Documentation** This step involves collecting any existing documentation related to the project or system under review. It may include technical specifications, user manuals, system architecture diagrams, requirements documents, and any other relevant materials. Gathering existing documentation provides insights into the project's background, functionality, and features, serving as a foundation for the review process.
- **Review Textual Content** In this step, the textual content of the documentation is carefully examined and analyzed. This includes reviewing descriptions, instructions, specifications, and other textual elements to understand the project's functionality, processes, and requirements. Reviewing textual content helps identify key information, clarify ambiguities, and ensure consistency and accuracy in documentation.
- **Identify Functionalities and processes** Here, the focus is on identifying the various functionalities and processes described in the documentation. This involves breaking down the system's capabilities and operations into distinct components, such as user registration, data management, authentication, ordering process, delivery logistics, etc. Each functionality and process are analyzed to understand its purpose, interactions, and

dependencies within the system.

- **Document Findings** Finally, the findings from the review process are documented systematically. This includes summarizing key points, highlighting significant observations or discrepancies, and recording any recommendations or actions needed to address identified issues. Documenting findings ensures that insights gained from the review process are captured effectively and can be used to guide future development, updates, or improvements to the project or system.

5.2 Requirement Validation

Requirement validation ensures the accuracy and quality of requirements for the “TALK2METRO” platform. To validate requirements effectively:

5.2.1 Review the Requirements

Conduct rigorous peer reviews to identify ambiguities and gaps in the requirements. Utilize a diverse team of reviewers to examine written needs, analysis models, and relevant information thoroughly.

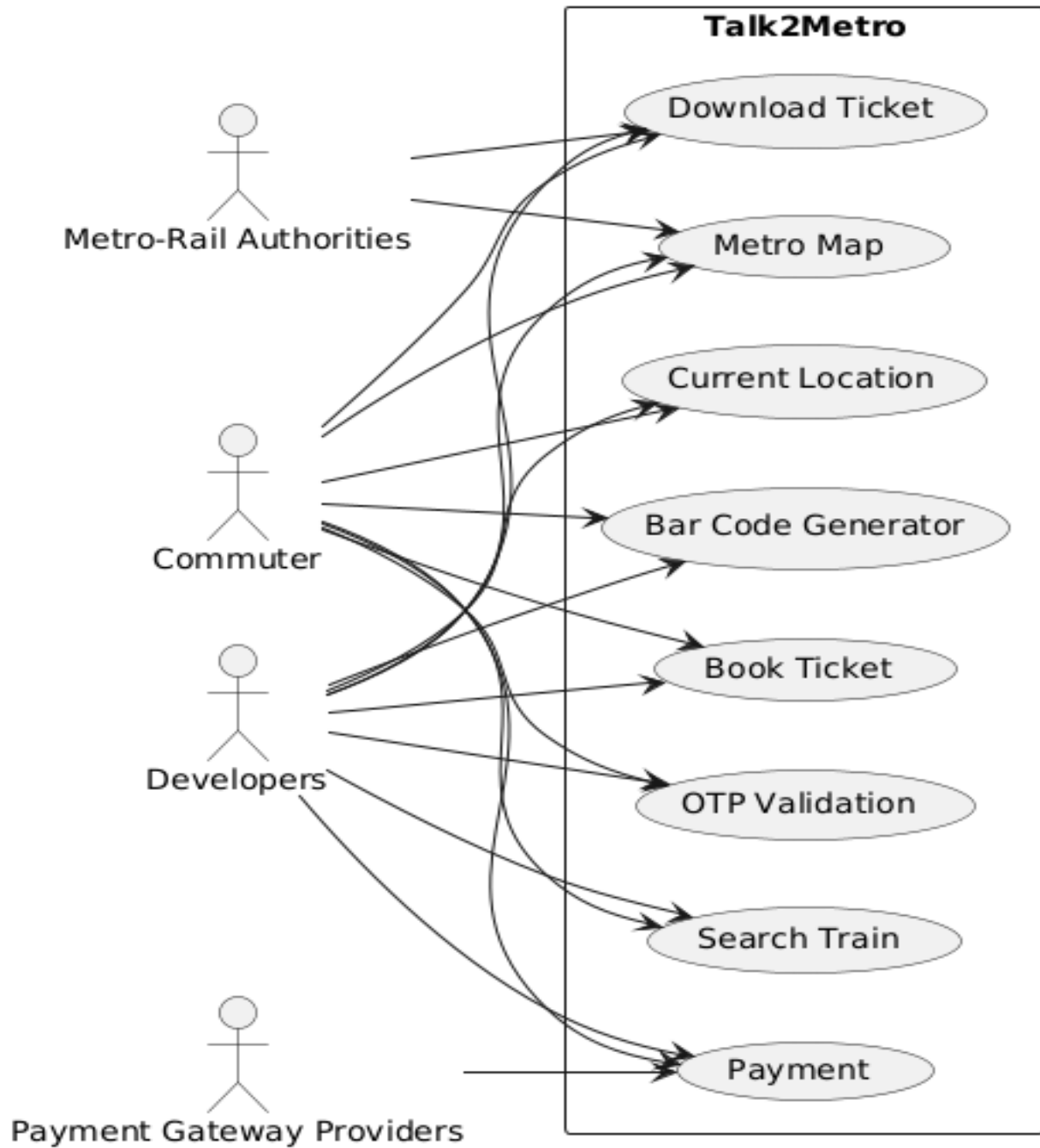
5.2.2 Test the Requirements

Create test cases to validate requirements and ensure they meet expected performance standards. Perform writing tests to verify the expected behavior of the “TALK2METRO” platform under specified conditions, including user needs and system functionality.

5.2.3 Simulate the Requirements

Utilize simulation tools to simulate the proposed system and add detail to written requirements. Prototyping and simulation techniques elevate requirement validation by providing a tangible representation of the “TALK2METRO” platform's functionality.

6. Use Case Diagram



7. Use Case Descriptions

Use Case No.	1	
Use Case	Book Ticket	
Goal	Allow users to select their destination station for ticket booking.	
Preconditions	User must be logged in.	
Success End Condition	User successfully selects a destination station.	
Failed End Condition	User is unable to select a destination station.	
Primary Actors:	User	
Secondary Actors:	None	
Trigger	Confirm	
Main Success Flows	Step	Action
	1	User initiates the ticket booking process.
	2	System displays a list of available metro stations.
	3	User selects the desired destination station.
	4	System confirms the selection and proceeds to the next step in booking.
Alternative Flows	Step	Branching Action
	1	If the list of stations fails to load: - System displays an error message and retries loading the list.
	2	If the selected station is not available: - System notifies the user and asks for a new selection.
Quality Requirements	Step	Requirement
	1	Station list should load within 3 seconds.
	2	The system should handle up to 1000 stations without performance issues.

Use Case No.	2	
Use Case	Metro map	
Goal	Provide users with a visual map of the metro system and route information.	
Preconditions	- User must be valid. - System must have access to up-to-date metro map data.	
Success End Condition	User views the metro map and route information without issues.	
Failed End Condition	Metro map fails to load or displays incorrect information.	
Primary Actors:	User	
Secondary Actors:	None	
Trigger	"Metro Map" Button	
Main Success Flows	Step	Action

	1	User selects the option to view the metro map.
	2	System retrieves and displays the metro map.
	3	User interacts with the map to view different routes and stations.
Alternative Flows	Step	Branching Action
	1	If the metro map fails to load: - System displays an error message and retries loading the map.
	2	If the map data is outdated: - System prompts for an update and reloads the map.
Quality Requirements	Step	Requirement
	1	Metro map should be interactive and responsive.
	2	Map data should be updated regularly to ensure accuracy.

Use Case No.	3	
Use Case	Search Train	
Goal	Provide users the available trains.	
Preconditions	- User must be valid. - System must have access to up-to-date metro map data.	
Success End Condition	User views the available trains without issues.	
Failed End Condition	App fails to load available trains or displays incorrect information.	
Primary Actors:	User	
Secondary Actors:	None	
Trigger	"Confirm" Button	
Main Success Flows	Step	Action
	1	User selects the onboarding location.
	2	User selects the destination location.
	3	User selects the date of departure.
Alternative Flows	Step	Branching Action
	1	If the app fails to load: - System displays an error message and retries loading the map.
Quality Requirements	Step	Requirement
	2	Map data should be updated regularly to ensure accuracy.

Use Case No.	4
Use Case	Current Location
Goal	To show the user's current location.

Preconditions	- User must be valid. - System must have access to up-to-date google map data.	
Success End Condition	User views the own location.	
Failed End Condition	App fails to load the correct location of user.	
Primary Actors:	User	
Secondary Actors:	None	
Trigger	"My Routes" Button	
Main Success Flows	Step	Action
	1	User selects the button.
	2	System should show the current location of users.
Alternative Flows	Step	Branching Action
	1	If the app fails to load: - System displays an error message and retries loading the map.
Quality Requirements	Step	Requirement
	2	Map data should be updated regularly to ensure accuracy.

Use Case No.	5	
Use Case	OTP validation	
Goal	Allow users to with a valid mobile number.	
Preconditions	- User must have access to a vali phone number.	
Success End Condition	User successfully validates.	
Failed End Condition	Not a valid phone number	
Primary Actors:	User	
Secondary Actors:	Railway Administration	
Trigger	"Send OTP" Button	
Main Success Flows	Step	Action
	1	Users type their valid phone number.
	2	System validates the information.
Alternative Flows	Step	Branching Action
	1	If the phone number is already in use: - System notifies the user and prompts for a different phone number.
	2	If the validation fails: - System displays an error message and requests correction.
Quality Requirements	Step	Requirement
	1	OTP validation should take less than 2 minutes.
	2	Validation should be accurate and secure.

Use Case No.	6	
Use Case	Bar Code Generator	
Goal	Generate Bar codes for ticket validation post-payment.	
Preconditions	- User must have successfully booked a ticket. - System must have Bar code generation capability.	
Success End Condition	Bar code is generated and sent to the user.	
Failed End Condition	Bar code generation fails.	
Primary Actors:	User	
Secondary Actors:	Railway Administration	
Trigger	Successful completion of ticket booking and payment.	
Main Success Flows	Step	Action
	1	User completes ticket booking and payment.
	2	System generates a Bar code for the ticket.
Alternative Flows	Step	Branching Action
	1	If Bar code generation fails: - System retries the process and notifies the user of any issues.
Quality Requirements	Step	Requirement
	1	Bar code generating time should be less than 109 second
	2	Bar code generator processing should be secure and reliable.

Use Case No.	7	
Use Case	Download Ticket	
Goal	Allow users to view ticket as pdf	
Preconditions	-User must be valid. -User must book a ticket. -System must store the ticket information.	
Success End Condition	-User successfully views the ticket pdf.	
Failed End Condition	User is unable to see the ticket pdf.	
Primary Actors:	User	
Secondary Actors:	None	
Trigger	"Download ticket" Button	
Main Success Flows	Step	Action
	1	User book a ticket.
	2	System retrieves the ticket and return as pdf.
Alternative Flows	Step	Branching Action
	1	If the system fails to retrieve the ticket: -System displays an error message and suggests retrying.
Quality Requirements	Step	Requirement
	1	Pdf should load within 3 seconds.

	2	Data displayed should be accurate.
--	---	------------------------------------

Use Case No.	8	
Use Case	Payment	
Goal	Facilitate secure and efficient payment for ticket booking.	
Preconditions	- User must be valid. - User must have a valid payment method.	
Success End Condition	Payment is processed successfully and ticket is booked.	
Failed End Condition	Payment fails and ticket is not booked.	
Primary Actors:	User, Payment Authentication	
Secondary Actors:	Railway Administration	
Trigger	"Pay" Button	
Main Success Flows	Step	Action
	1	User initiates payment process.
	2	System presents available payment methods.
	3	User selects a payment method and provides necessary details.
	4	System processes the payment through the selected gateway.
	5	Payment is confirmed and system generates a ticket.
Alternative Flows	Step	Branching Action
	1	If payment authentication fails: - System notifies the user and prompts to retry or use a different method.
	2	If payment gateway is down: - System suggests alternative payment methods or retry later.
Quality Requirements	Step	Requirement
	1	Payment processing time should be under 10 seconds.
	2	Payment details should be securely encrypted.

8. Activity Diagram

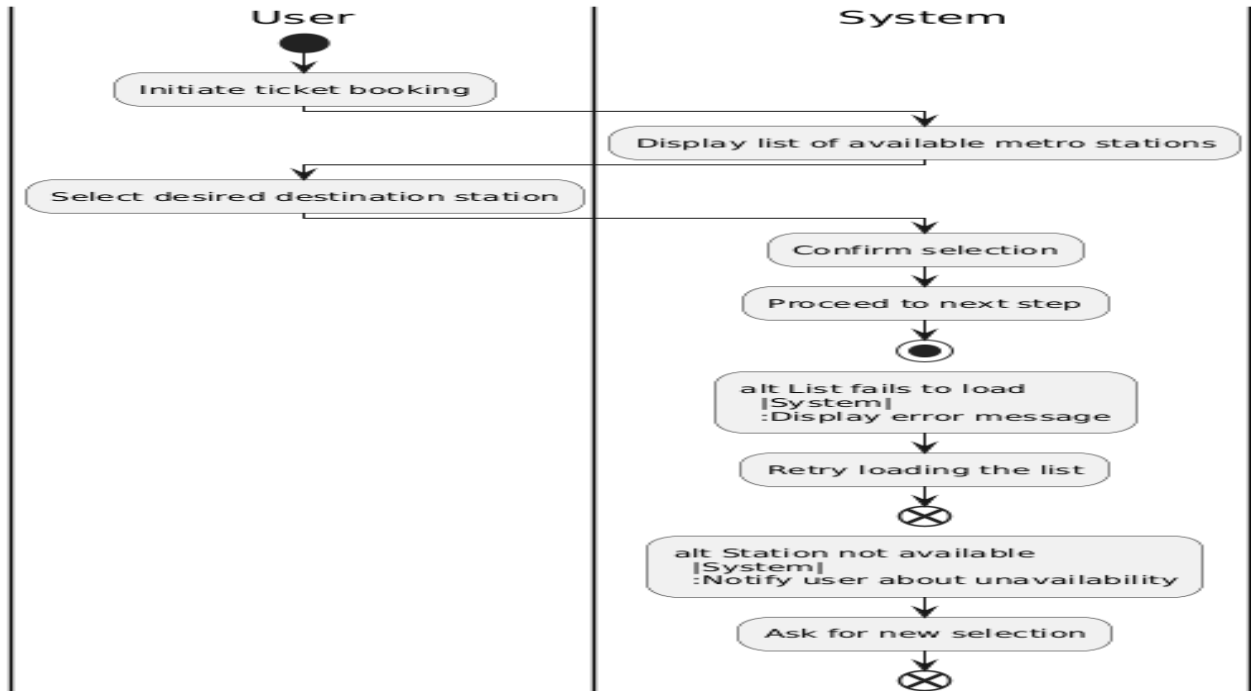


Figure:8.1 Book Ticket

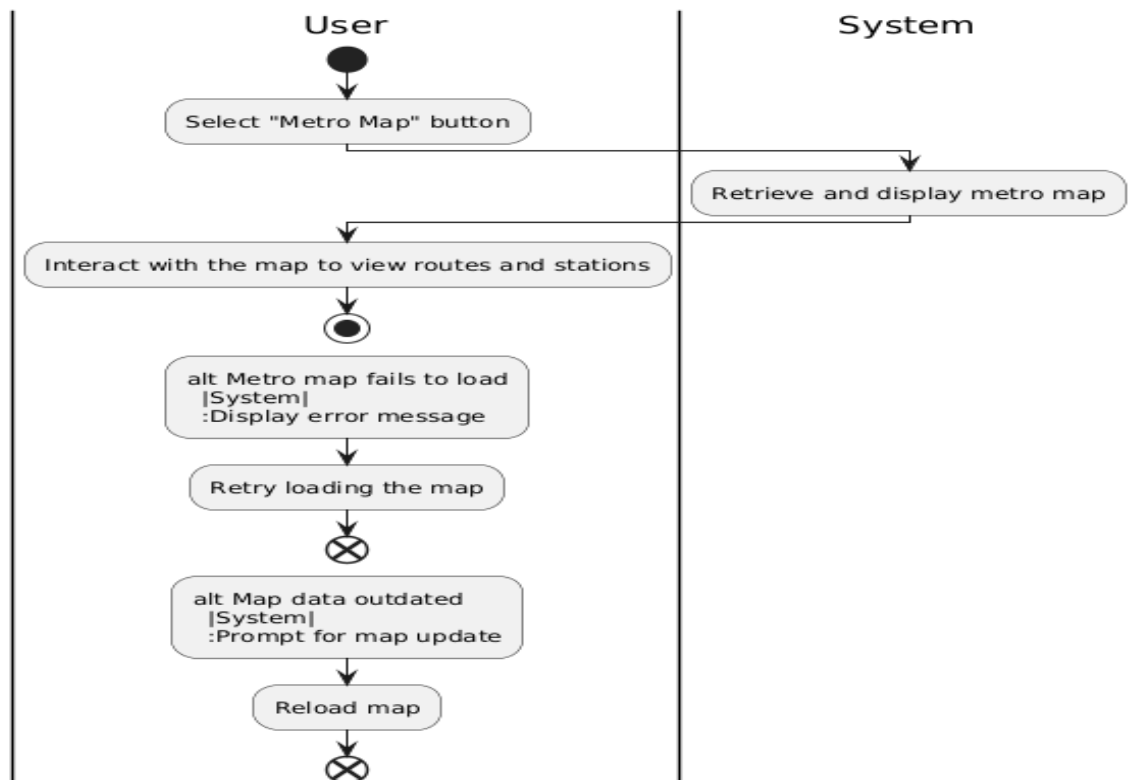


Figure:8.2 Metro Map

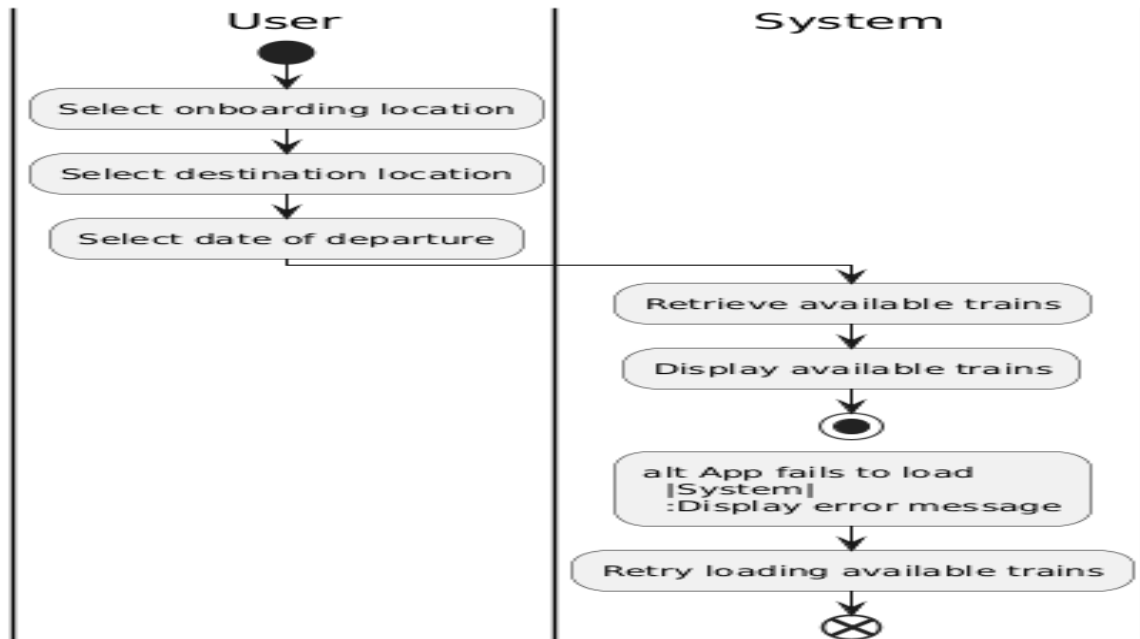


Figure:8.3 Search Train

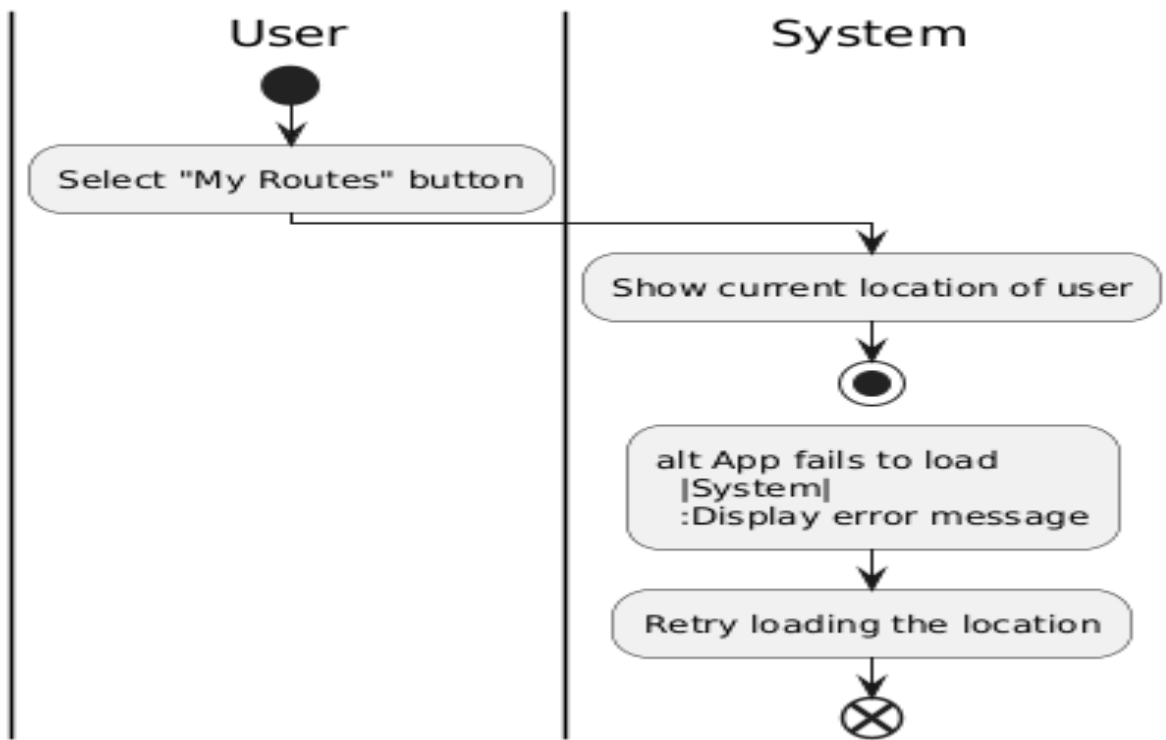


Figure:8.4 Current Location

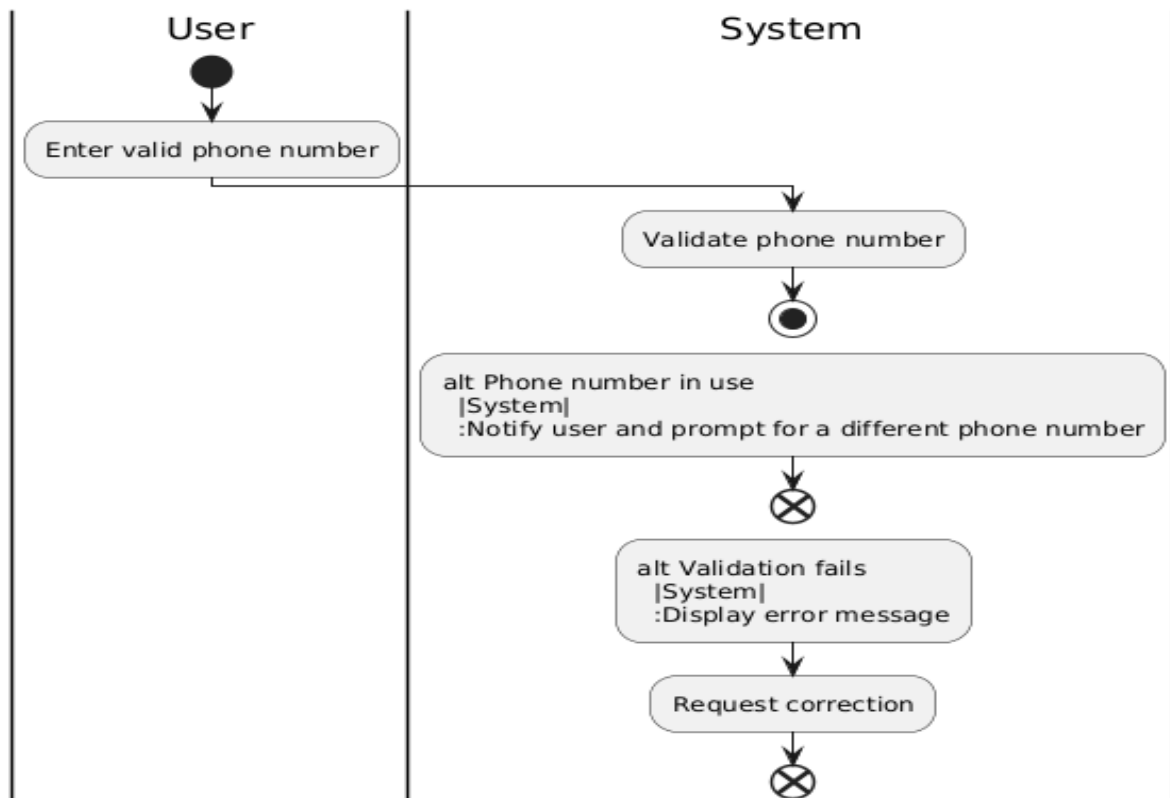


Figure:8.5 OTP validation

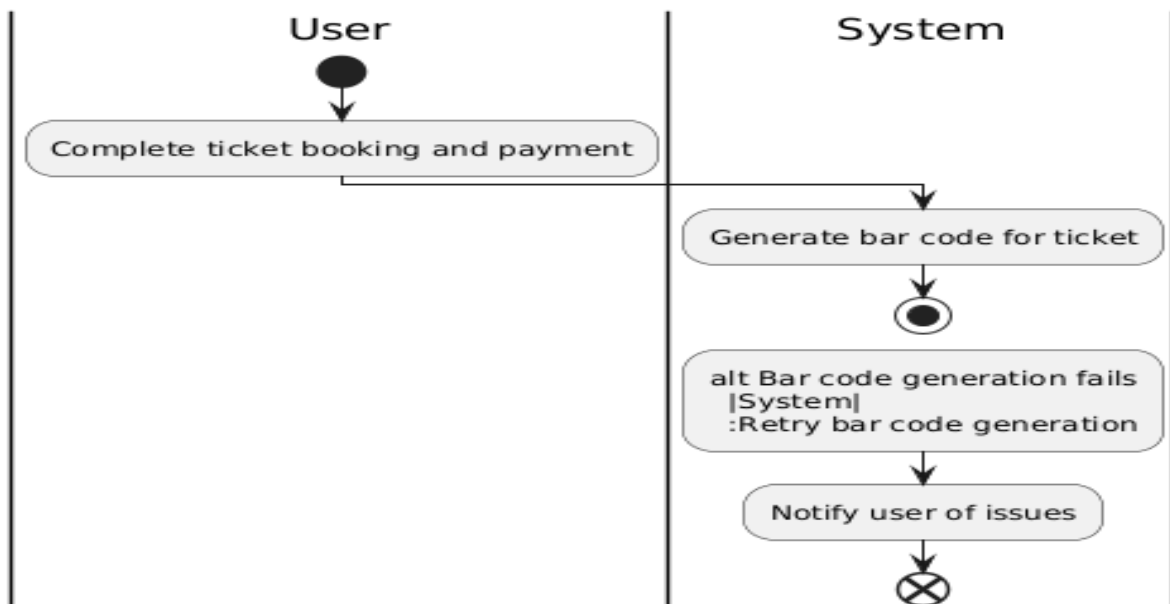


Figure:8.6 Bar Code Generator

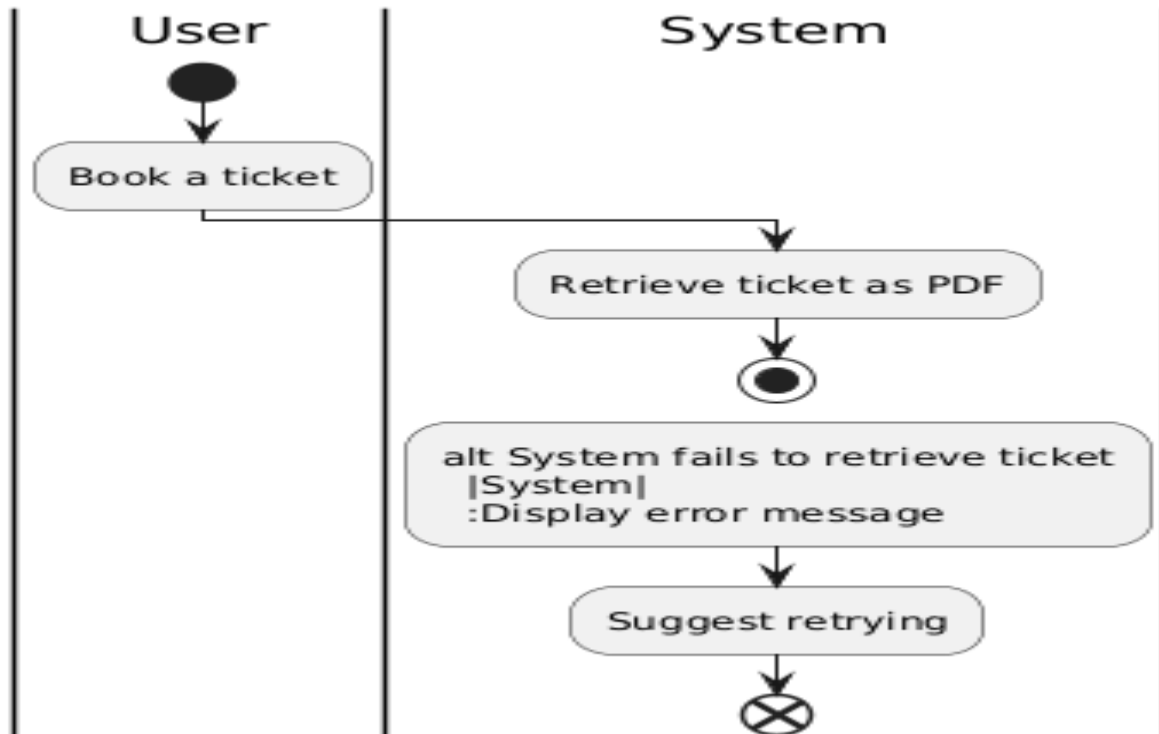


Figure:8.7 Download Ticket

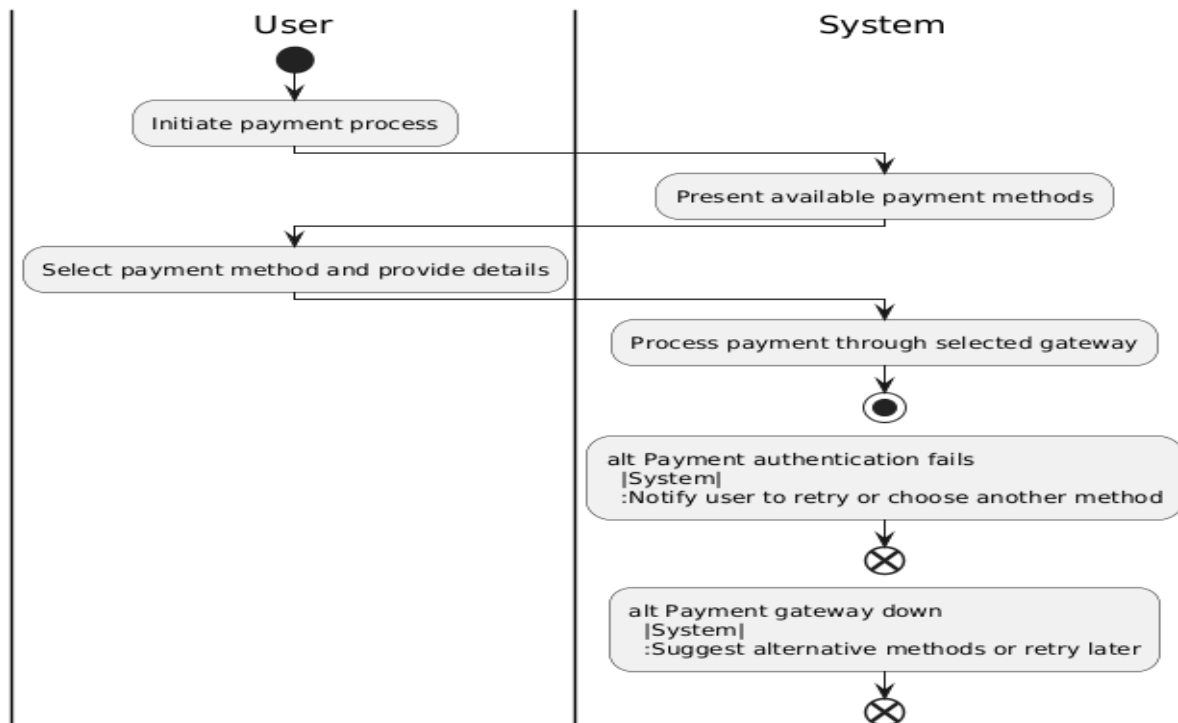


Figure:8.8 Payment

9. Sequence Diagram

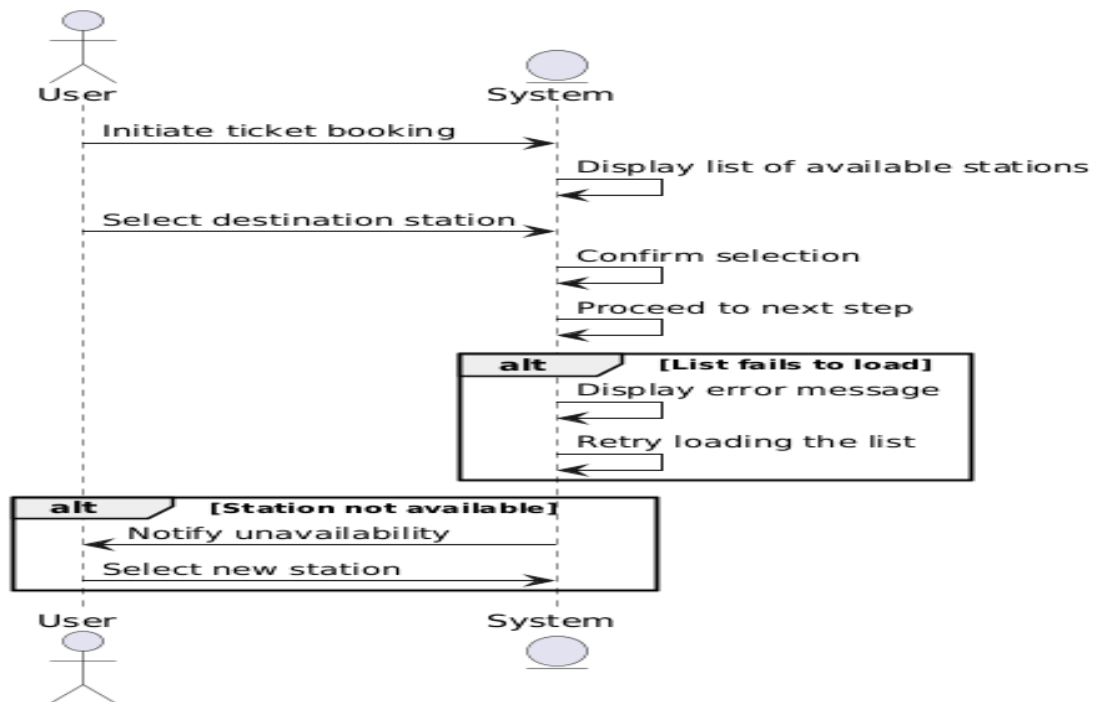


Figure:9.1 Book Ticket

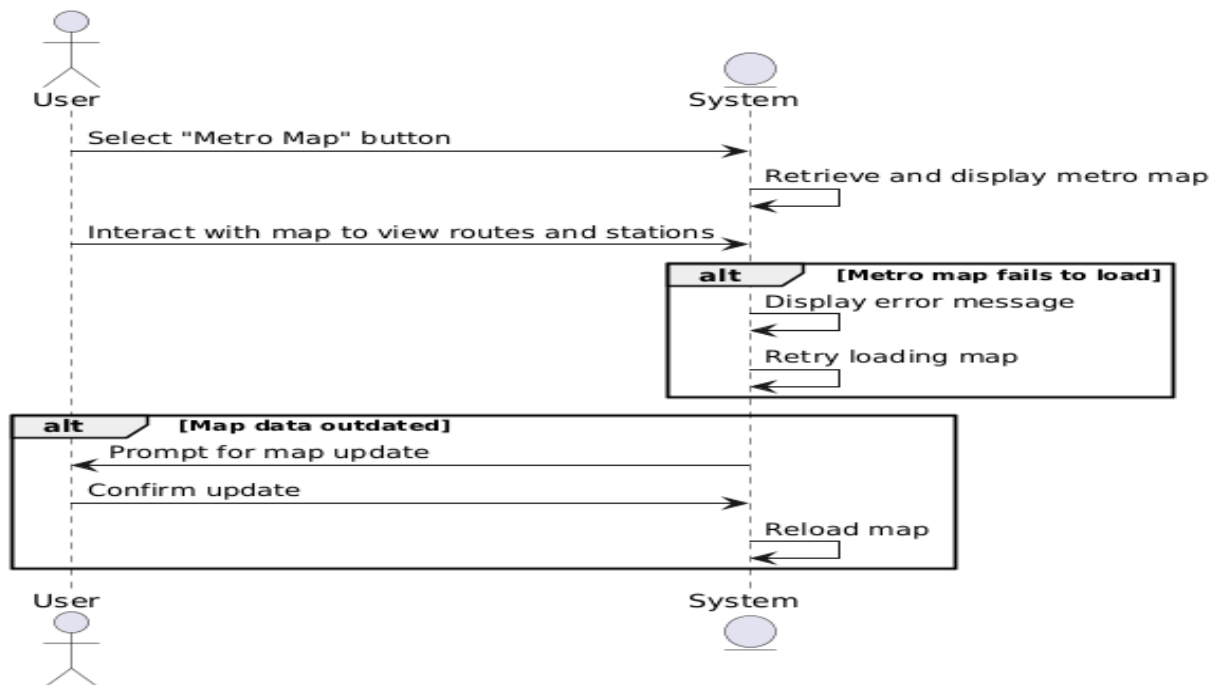


Figure:9.2 Metro Map

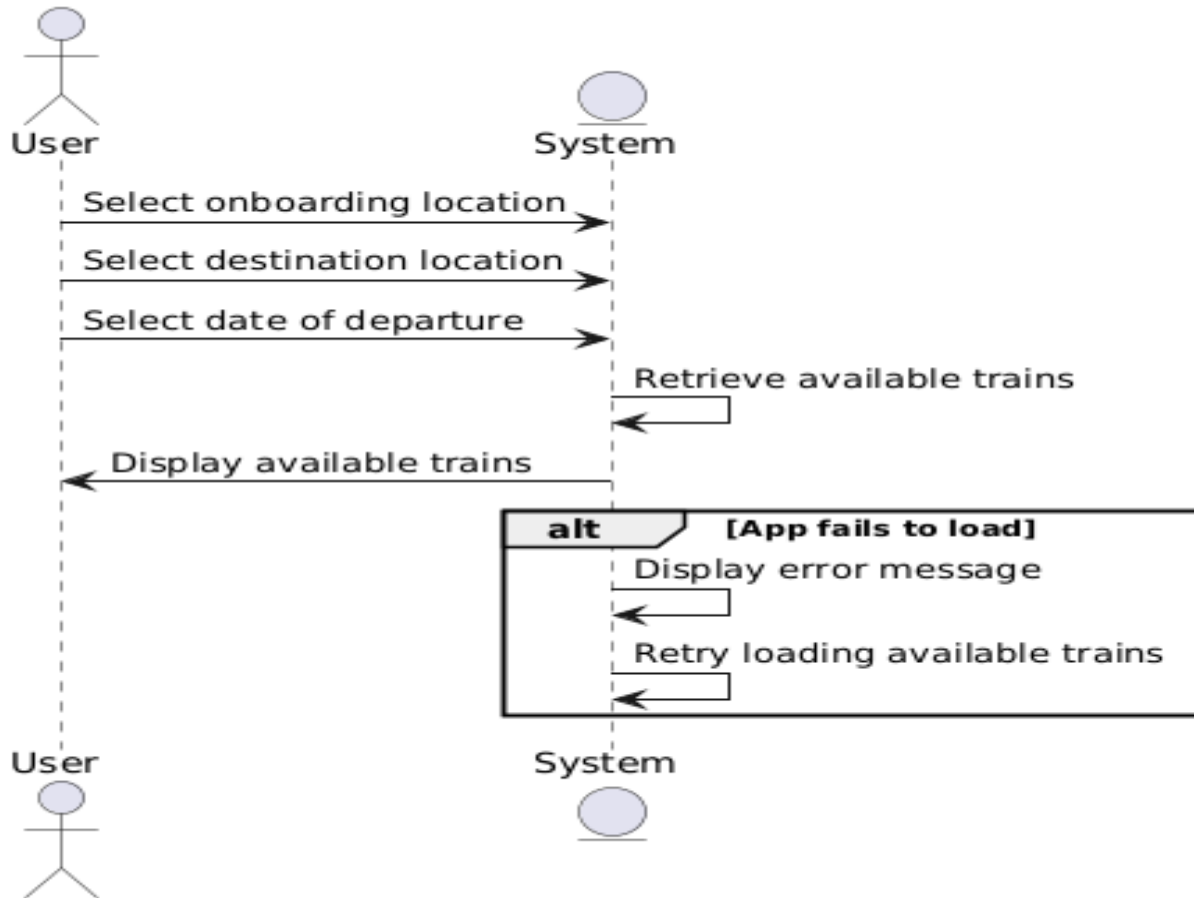


Figure:9.3 Search Train

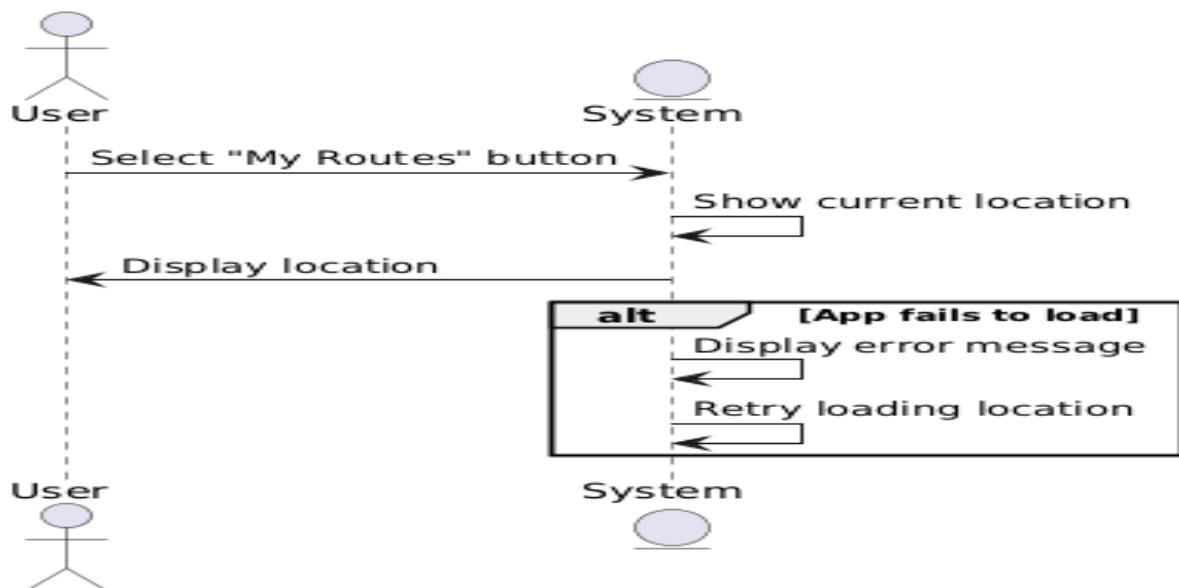


Figure:9.4 Current Location

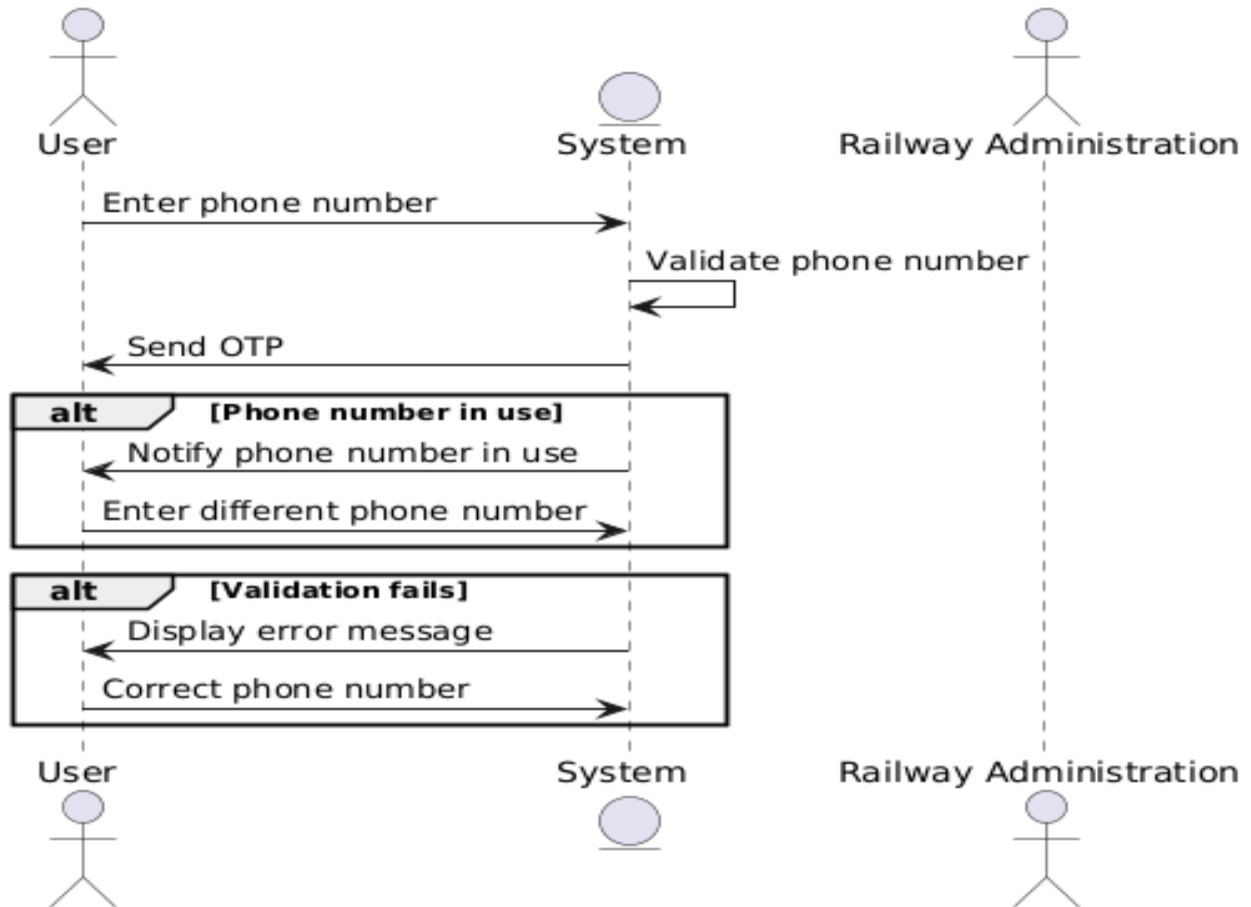


Figure:9.5 OTP validation

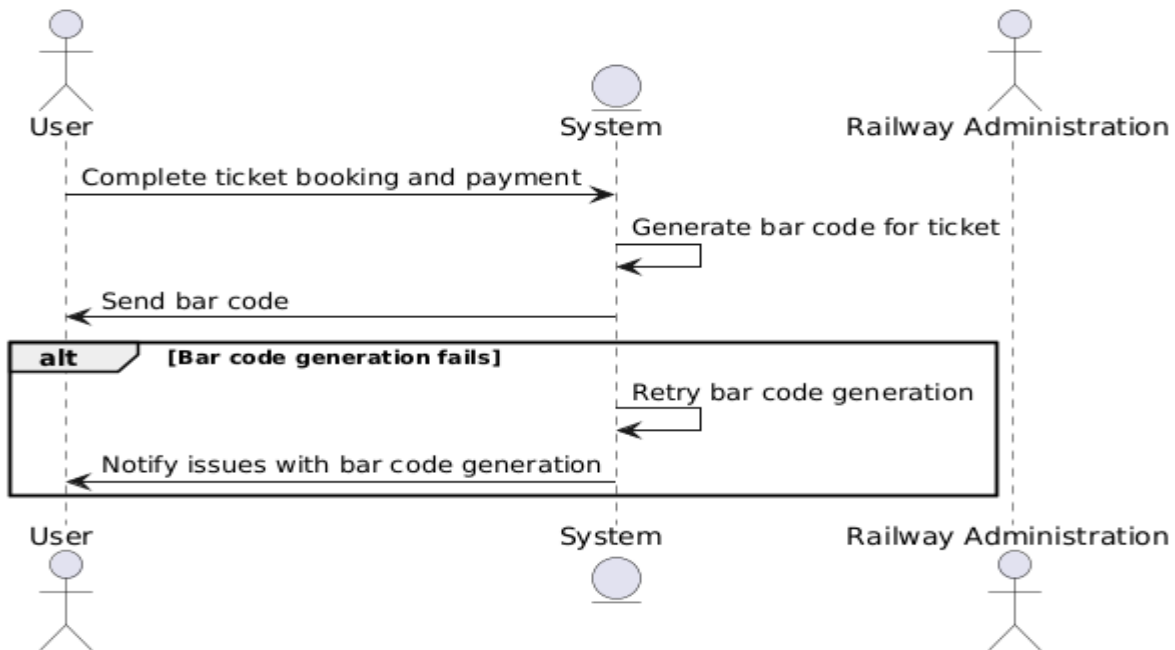


Figure:9.6 Bar Code Generator

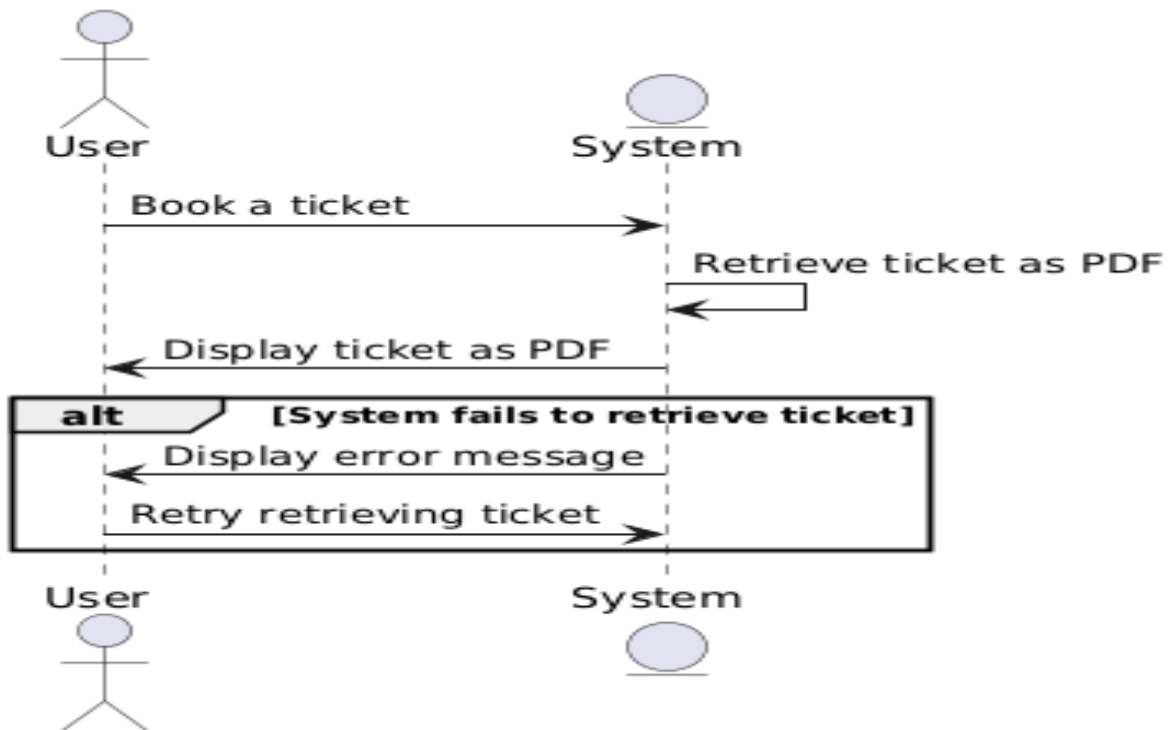


Figure:9.7 Download Ticket

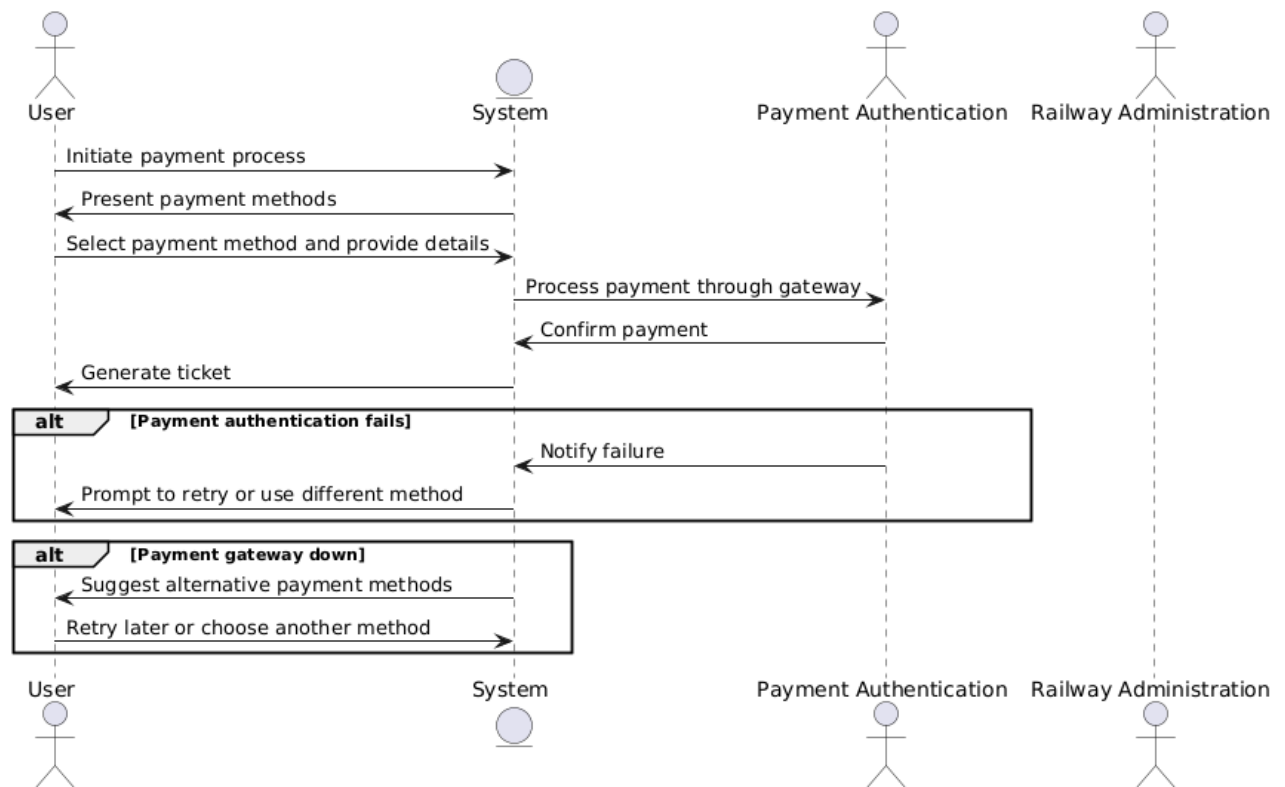
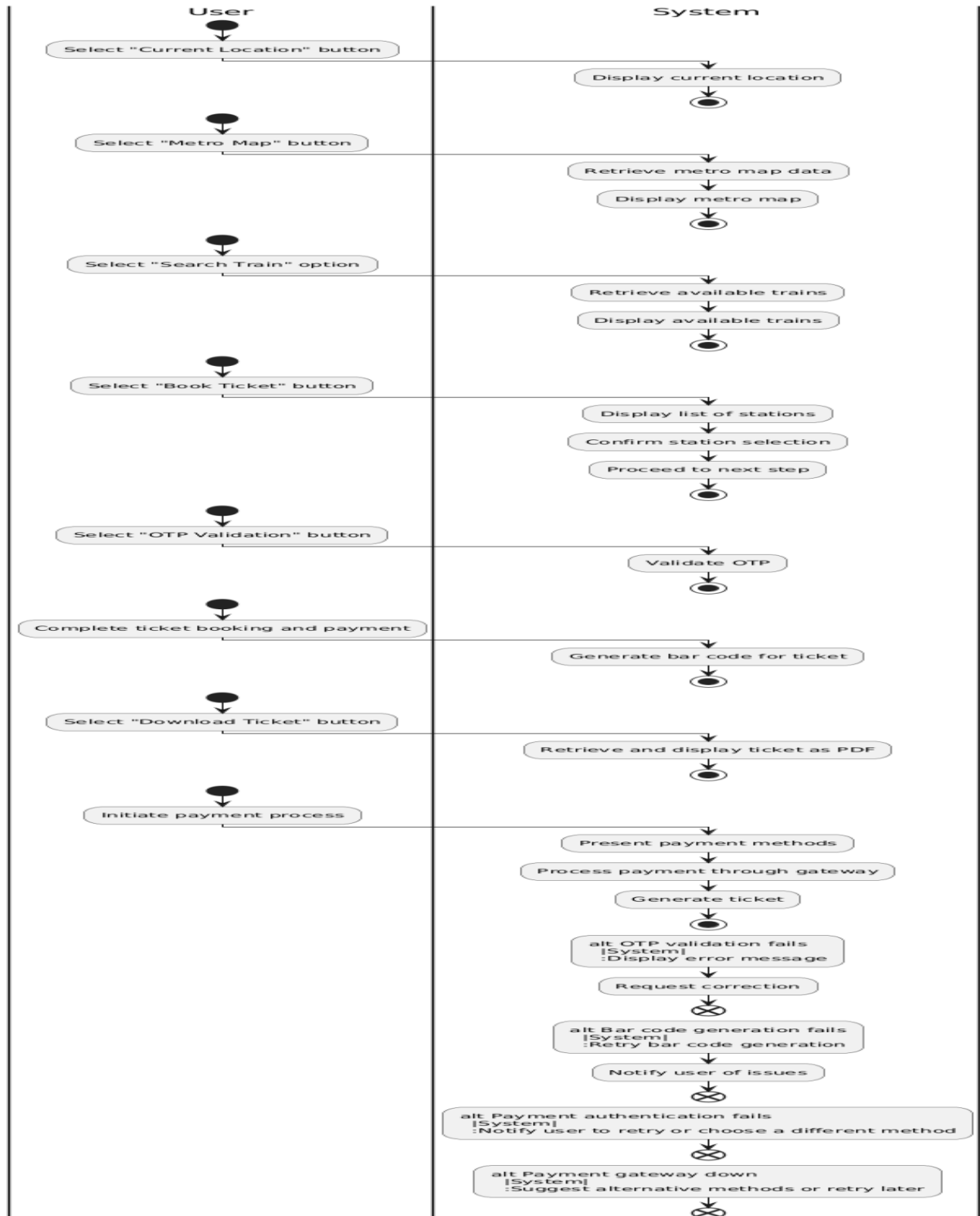


Figure:9.8 Payment

10. Swimlane Diagram



11. Appendix

11.1 Prioritization of requirements

We've prioritized the functional requirements by following Three-level Scale technique.

11.1.1 Three-level Scale

Three-level scale techniques are a type of measurement scale that uses three levels to measure a variable. The three levels are typically labeled as "low," "medium," and "high." This type of scale is often used when the variable being measured is subjective or difficult to quantify.

11.1.2 Prioritization of the requirements of Talk2Metro

F_R1 – High priority: Users should have the validation on the platform, providing providing a valid phone number

F_R2 – High priority: Users must be able to search for metro stations using various criteria such as station name or proximity, facilitating easy access to desired routes.

F_R3 – High priority: Users must be able to book metro-rail tickets seamlessly through the platform, specifying the origin, destination, and desired travel options.

F_R4 – High priority: The platform should offer secure payment processing mechanisms, enabling users to complete transactions using preferred payment.

F_R5 – High priority: The platform must generate a unique Bar code for each booked ticket, ensuring valid and secure ticket validation at metro stations.