



UNIVERSITY OF ECONOMICS AND LAW

ANALYSIS AND DESIGN INFORMATION SYSTEM

FINAL PROJECT

DEVELOP A MOBILE APPLICATION: AUTOBUS BUS BOOKING SYSTEM

Mrs: Le Thi Kim Hien

Team Members

NAME	ID
Tô Nguyễn Tường Vy	K224060853
Trần Thị Thùy Lợi	K224060840
Nguyễn Quế Anh	K224060828
Nguyễn Huỳnh Kim Sương	K224060848
Lê Đình Giáp	K224060831

ACKNOWLEDGEMENT

In an era marked by rapid advancements in technology and rising demand for streamlined passenger transportation services, the necessity for effective ticket-booking systems has grown. Recognizing this requirement, our team proceeded with the creation of an application that allows for the easy purchase of bus tickets for coach travel. Our initiative arose from a desire to address typical travel obstacles, such as time-consuming checkout processes, disorganization, and difficulties in accessing precise information about bus routes and tickets.

We extend our heartfelt thanks to our instructors—Ms. Le Thi Kim Hien - for their guidance, expertise, and unwavering support throughout this project. Your constructive feedback and encouragement have been instrumental in shaping our understanding of the theoretical frameworks and practical applications necessary for creating an effective management information system.

We also appreciate the collaborative efforts of our team members, whose dedication and creativity have contributed significantly to the success of this endeavor. Working together, we have developed a user-friendly platform that integrates essential features and enhances the user experience.

This project has not only strengthened our technical skills but also reinforced the importance of teamwork and effective communication in system design. We are grateful for the knowledge gained and look forward to applying these lessons in our future endeavors.

Thank you for your support and inspiration.

ABSTRACT

In response to the increasing demand for seamless, efficient transportation booking systems, this project focuses on developing a user-friendly bus ticket booking application tailored for the Vietnamese market. The application, named AutoBus, streamlines the ticketing process by providing essential features like real-time search for routes, seat selection, secure payment gateways, and customer support, thereby enhancing user convenience and accessibility. AutoBus addresses common issues in both online and offline ticketing systems, including time inefficiency, lack of transparency, and the manual nature of offline operations. By employing modern technologies and a multi-functional design, the application ensures secure transactions and personalized experiences for users. The implementation of this project follows the System Development Life Cycle (SDLC) framework, encompassing phases from planning to support. Through this project, we aim to provide a robust solution that contributes to the modernization of public transportation systems in Vietnam, fostering improved passenger mobility and customer satisfaction.

LIST OF ABBREVIATION

Terms / Acronyms	Definitions /Glossary
API	Application Programming Interface: A set of routines, protocols, and tools for building software and websites.
B2C	Business-to-Consumer: Refers to the transactions conducted directly between a business and consumers.
DB	Database: An organized collection of data, generally stored and accessed electronically from a computer system.
MongoDB	A document-oriented NoSQL database used for high-volume data storage.
SIT environment	SIT (System Integration Testing) environment: A phase in software testing where individual modules are combined and tested as a group to ensure they work together.
UX	User Experience: The overall experience of a person using a product, especially in terms of how easy or pleasing it is to use.
UI	User Interface: The means by which the user and a computer system interact, particularly the use of input devices and software.

DOCUMENT REFERENCES

Document Name	Reference
Complete Files	<u>GROUP 03_FINAL - Google Drive</u>
UX/UI Figma Design	<u>AUTOBUS – Figma</u>
BPMN	<u>BPMN</u>
DFD	<u>DFD</u>
Use Case	<u>USE CASE</u>
Sequence Diagram	<u>SEQUENCE</u>
ERD	<u>ERD MODEL</u>
Slides	<u>SLIDE - Google Drive</u>
Demo Video	<u>DEMO APP</u>
Work Roster	<u>WORK ROSTER</u>
Gantt Chart	<u>GANTT CHART</u>
Pert Chart	<u>PERT CHART</u>

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
ABSTRACT	ii
LIST OF ABBREVIATION	iii
DOCUMENT REFERENCES	iv
TABLE OF CONTENTS	v
LIST OF FIGURES.....	ix
LIST OF TABLES.....	xii
CHAPTER 1: PROJECT OVERVIEW.....	1
1.1. Executive Summary.....	1
1.1.1. Project Overview	1
1.1.2. Value.....	2
1.1.3. Introduction of Information Systems	3
1.1.4. Assumptions	4
1.1.5. Research Methodologies	5
1.1.6. Research Progress.....	6
1.1.7. Structure of the Project.....	7
1.2. Context.....	8
1.2.1. Background.....	8
1.2.2. Demands	15
1.2.3. Business Problems Statement/Business Needs	19
1.3. Problems and Solutions	22
1.4. Risks and Assumptions.....	23
1.4.1. Analyze risks and assumptions.....	23
1.4.2. Feasibility study.....	24
1.5. User Roles and Permissions	25
1.6. Business Function Diagram (BFD) of AutoBus.....	26
1.7. Tool Used	28
CHAPTER 2: THEORETICAL BACKGROUND	29
2.2. Terms/Acronyms and Definitions	29
2.2.1. BPMN - Business Process Modeling Notation:	29
2.2.2. DFD - Data Flow Diagram	29
2.2.3. Use case	30
2.3. SDLC - The system development life cycle.....	31
2.3.1. System Planning	31
2.3.2. System Analysis	33
2.3.3. System Design	36

2.3.4. System Implementation	41
2.3.5. System Security and Support.....	44
CHAPTER 3: USERS UNDERSTANDING	46
3.1. Market Analysis.....	46
3.1.1. FUTA - Phuong Trang	46
3.1.2. Redbus	50
3.1.3. Vexere.....	54
CHAPTER 4: OVERVIEW ANALYSIS	59
4.1. Business Requirements.....	59
4.1.1. Business Objectives	59
4.1.2. Functional Requirements.....	59
4.1.3. Non - functional Requirements	60
4.2. Fact-finding	60
4.3. Overall Flow	64
4.4. Cost-benefit Analysis	66
CHAPTER 5: DETAIL ANALYSIS.....	71
5.1. User Stories & Current Process Flow	71
5.1.1. Current Process Flow	71
5.1.2. User Stories.....	73
5.1.3. Future Process Flow	74
5.2. Main Function.....	75
5.3. Processes - Business Process Modelling Notation (BPMN)	80
5.3.1. Deliverables	80
5.3.2. Overall Process	81
5.3.3. Login Process	88
5.3.4. Payment Process	93
5.3.5. Rating Process	96
5.4. Data Flow Diagram (DFD).....	97
5.4.1. Deliverables	97
5.4.2. DFD Diagram	98
5.4.3. Specification	100
5.5. Use case	102
5.5.1. Deliverables	102
5.5.1. Overall	103
5.5.2. Account management	106
5.5.3. Booking	117
5.5.4. Transactions.....	118

5.5.5. Booking History	130
5.5.6. Support and Feedback	147
5.5.7. Notifications	153
5.6. Sequence Diagram.....	156
5.6.1. Deliverables	156
5.6.2. Account management	157
5.6.3. Booking	159
5.6.4. Transactions.....	161
5.6.5. Booking History	163
5.6.6. Support and Feedback	165
5.6.7. Notifications	167
5.7. Business Rule	168
5.8. Database design	178
5.8.1 Overview and purpose.....	178
5.8.2. Deliverables	178
5.8.3 Database.....	179
5.8.4. ERD Model.....	194
5.8.5 Data Normalization	198
5.8.6 Physical database in MongoDB.....	200
5.9. Data Dictionary.....	202
5.9.1. Data type definition	202
5.9.2. Data dictionary	203
CHAPTER 6: SYSTEM DESIGN	213
6.1. Develop Wireframe for the app	213
6.1.1. Account Management.....	213
6.1.2. Booking Ticket	216
6.1.3. Transaction	218
6.1.4. Booking History	222
6.1.5. Cancellation	225
6.2. Design UI Mockup screen	226
6.2.1. Design System	226
6.2.2. UI Screens of AutoBus	228
CHAPTER 7: SYSTEM IMPLEMENTATION AND TESTING	242
7.1. Prototype.....	242
7.1.1. Account Management.....	242
7.1.2. Booking	248
7.1.3. Transaction	249

7.1.4. Booking History	251
7.1.5. Cancellation	253
7.2. Test Scenarios and Test Case	254
7.2.1. Summary testing	254
7.2.2. Testing	255
CHAPTER 8: PROJECT PLAN.....	272
8.1. Work Breakdown Structure (WBS).....	272
8.2. Project Milestone	274
8.3. Pert Chart	275
8.4. Gantt Chart	275
CHAPTER 9: CONCLUSION	277
9.1. Practical significance	277
9.1.1. For user	277
9.1.2. For team	278
9.2. Conclusion	279
REFERENCES	280

LIST OF FIGURES

Figure 1.1. Research Progress	7
Figure 1.2. Global vehicular travel by mode, 1900–2016.....	9
Figure 1.3. Global vehicular travel as a function of global GDP, 1900–2016.....	10
Figure 1.4. Volume of freight (thousand tons) carried by types of transport in the period of 2007–2018	12
Figure 1.5. Percentage of domestic freight transport in Vietnam by mode of transport (2016).....	13
Figure 1.6. The growth in the number of passengers carried (mil. people) by types of transport in the period of 2007–2018.	14
Figure 1.7. Number of passengers carried (mil. people) by types of transport in the period of 2007–2018.....	14
Figure 1.8. Business Function Diagram of AutoBus.....	27
Figure 2.1. Overview of Feasibility	32
Figure 2.2. User Interface Design.....	37
Figure 3.1. Log In (FUTA).....	47
Figure 3.2. Search for travel trip (FUTA)	48
Figure 3.3. View selected bus trip (FUTA).....	49
Figure 3.4. Input booking information (FUTA)	50
Figure 3.5. Log in	51
Figure 3.6. Search for travel trips	52
Figure 3.7. View the selected trips	53
Figure 3.8. Input personal information for booking	54
Figure 3.9. Log in	55
Figure 3.10. Search for travel trips	56
Figure 3.11. View the selected trips	57
Figure 3.12. Input personal information for booking	58
Figure 5.1. Current process flow	71
Figure 5.2. Indirect ticket booking process via hotline	72
Figure 5.3. Future Process Flow	74
Figure 5.4. Overall BPMN	81
Figure 5.5. BPMN of Login process	88
Figure 5.6. BPMN of Updated personal information process	89
Figure 5.7. BPMN of Payment Process.....	93
Figure 5.8. BPMN of Rating process	96
Figure 5.9. DFD level 0.....	98
Figure 5.10. DFD level 1	99

Figure 5.11. Use case of Account Management.....	108
Figure 5.12. Use case of Booking	117
Figure 5.13. Use case of Transactions.....	118
Figure 5.14. Use case of Booking History	130
Figure 5.15. Use case of Support and Feedback	147
Figure 5.16. Use case of Notifications	153
Figure 5.17. Sequence diagram of Account management.....	158
Figure 5.18. Sequence diagram of Booking	160
Figure 5.19. Sequence diagram of Transactions	162
Figure 5.20. Sequence diagram of Booking History	164
Figure 5.21. Sequence diagram of Support & Feedback.....	166
Figure 5.22. Sequence diagram of Notifications	168
Figure 5.23. ERD Model	194
Figure 5.24. Data Normalization	198
Figure 5.25. Physical database in MongoDB	200
Figure 6.1. Account Management Process in AutoBus App.....	213
Figure 6.2. Log in /Sign up/ Forgot Password.....	214
Figure 6.3. Travel Booking and Management App UI Overview	214
Figure 6.4. Account Management	215
Figure 6.5. Booking Tickets Process in AutoBus App.....	216
Figure 6.6. Select pick up/drop off points for Single/Standard Bus.....	217
Figure 6.7. Select pick up/drop off points for Limousine Bus	217
Figure 6.8. Select pick up/drop off points	218
Figure 6.9. Transaction	218
Figure 6.10. Voucher	220
Figure 6.11. Choose Voucher and Payment Method.....	221
Figure 6.12. Payment.....	222
Figure 6.13. Booking History	223
Figure 6.14. View Unpaid/Paid/Upcoming Ticket and Invoice	224
Figure 6.15. Rating and Feedback	225
Figure 6.16. Cancellation.....	226
Figure 6.17. AutoBus' logo	226
Figure 6.18. AutoBus' color palette	227
Figure 6.19. Account Management in AutoBus App	228
Figure 6.20. Log in /Sign up/ Forgot Password.....	229
Figure 6.21. Travel Booking and Management App UI Overview	230
Figure 6.22. Account Management	231

Figure 6.23. Booking Tickets Process in AutoBus App.....	232
Figure 6.24. Select pick up/drop off points for Limousine Single/Standard Bus	232
Figure 6.25. Select pick up/drop off points	233
Figure 6.26. Transaction.....	234
Figure 6.27. Voucher	235
Figure 6.28. Choose Voucher and Payment Method.....	236
Figure 6.29. Payment.....	237
Figure 6.30. Booking History	238
Figure 6.31. View Unpaid/Paid/Upcoming Ticket and Invoice	239
Figure 6.32. Rating and Feedback.....	240
Figure 6.33. Rating and Feedback.....	241
Figure 7.1. Prototype of login, registration and forgotten password process.....	242
Figure 7.2 Prototype of searching process	244
Figure 7.3. Prototype of account management process	246
Figure 7.4. Prototype of Booking Process.....	248
Figure 7.5. Prototype of transaction process	249
Figure 7.6. Prototype of Booking History Process.....	251
Figure 7.7. Prototype of Cancellation Process	253
Figure 8.1. AutoBus's Booking System Project with 4 epics	273
Figure 8.2. Sprints, Tasks & SubTasks related to each epic	273
Figure 8.3. Project Milestone	274
Figure 8.4. Pert chart	275
Figure 8.5. Gantt chart.....	276

LIST OF TABLES

Table 1.1. Project Overview	1
Table 1.2. Values	2
Table 1.3. Demands	15
Table 1.4. Business Problems Statement.....	19
Table 1.5. Problems and Solutions	22
Table 1.6. Risks and assumptions.....	23
Table 1.7. User Roles and Permissions	25
Table 4.1. Question Types	61
Table 5.1. Current process flow	72
Table 5.2. Indirect booking process via hotline	73
Table 5.3. Main Function	75
Table 5.4. Deliverables	80
Table 5.5. High Level Requirements of Overall BPMN	82
Table 5.6. High Level Requirements of Login process.....	90
Table 5.7. High level requirements of Payment Process in AutoBus.	94
Table 5.8. Use Case Detail - Description: Register account of Account Management ..	109
Table 5.9. Flow of Events: Module Register Account of account management.....	110
Table 5.10. Use Case Detail - Description: Module Login account of account management	111
Table 5.11. Flow of Events: Module Login account of account management.....	111
Table 5.12. Use Case Detail - Description: Module Forgot password of account management.....	113
Table 5.13. Flow of Events: Module Forgot password of account management	113
Table 5.14. Use Case Detail - Description: Module Update personal information of account management.....	115
Table 5.15. Flow of Events: Module Update personal information of account management	116
Table 5.16. Choose Payment Method Specification	119
Table 5.17. Choose Payment Method Flow Analysis	119
Table 5.18. Make Payment Specification.....	120
Table 5.19. Make Payment Flow Analysis.....	121
Table 5.20. Apply Voucher Specification	122
Table 5.21. Apply Voucher Flow Analysis	122
Table 5.22. View Voucher Detail Specification.....	123
Table 5.23. View Voucher Detail Flow Analysis.....	123
Table 5.24. Payment Guideline Specification	124

Table 5.25. Payment Guideline Flow Analysis	125
Table 5.26. Process Payment Specification.....	126
Table 5.27. Process Payment Flow Analysis.....	126
Table 5.28. Update Status of Tickets Specification	127
Table 5.29. Update Status of Tickets Flow Analysis	127
Table 5.30. View Ticket-Booking Invoice Specification.....	128
Table 5.31. View Ticket-Booking Invoice Flow Analysis.....	128
Table 5.32. Use Case Description: View Paid ticket	131
Table 5.33. Use Case Description: View upcoming ticket.....	132
Table 5.34. Use Case Description: View Ongoing ticket.....	133
Table 5.35. Use Case Description: Update real-time information	134
Table 5.36. Use Case Description: View Complete ticket	135
Table 5.37. Use Case Description: View Unpaid ticket	136
Table 5.38. Use Case Description: View payment guidelines	137
Table 5.39. Use Case Description: Change pick-up/drop-off point.....	138
Table 5.40. Use Case Description: Change transit location	139
Table 5.41. Use Case Description: View Cancellation	140
Table 5.42. Use Case Description: Select tickets to cancel.....	141
Table 5.43. Use Case Description: View cancel regulations	143
Table 5.44. Use Case Description: Select reason to cancel.....	144
Table 5.45. Use Case Description: Confirm Cancellation	145
Table 5.46. Use Case Description: Contact support hotline.....	147
Table 5.47. Use Case Description: Rating.....	149
Table 5.48. Use Case Description: Submit feedback	150
Table 5.49. Use Case Description: View other Ratings and Review	151
Table 5.50. Use Case Detail - Description: Module View List of Notifications	153
Table 5.51. Flow of Events: Module View List of Notifications.....	154
Table 5.52. Use Case Detail - Description: Module View Tickets	154
Table 5.53. Flow of Events: Module View Tickets	155
Table 5.54. Use Case Detail - Description: Module View promotions.....	155
Table 5.55. Flow of Events: Module View promotions	156
Table 5.56. Deliverables	156
Table 5.57. Business Rule	168
Table 5.58. Deliverables	178
Table 5.59. Identify and describe entities.....	179
Table 5.60. Data type.....	181
Table 5.61. Relationship description table	195

Table 5.62. Database Configuration and Optimization	200
Table 5.63. Data Type Definition.....	202
Table 5.64. Data dictionary	203
Table 7.1. Summary testing.....	254
Table 7.2. Testing	255

CHAPTER 1: PROJECT OVERVIEW

1.1. Executive Summary

1.1.1. Project Overview

Table 1.1. Project Overview

Idea Name	A ticket-booking application that allows users to buy tickets for coach travel.
Introduction to the Company	Application name: AutoBus. AutoBus was designed with travelers in mind, aiming to streamline the entire ticketing process by allowing users to effortlessly book tickets for coach travel with just a few taps. Our mission is to transform how people plan their journeys by making ticket purchasing simple, fast, and accessible anytime, anywhere.
Category	Ticketing, Retail, Passenger Transport
Project Type	B2C
Problems of Services	Online: <ul style="list-style-type: none">- Wasting customers' time during checkout.- Requires better organization and a more professional appearance.- Customers struggle to locate contact information. Offline: <ul style="list-style-type: none">- Leads to fraud in bus ticket revenue payments.- Challenges in tracking revenue in real-time.- Requires more manual labor, increasing staff needs.- Various issues include record-keeping for items, seat availability, seat pricing, and invoice generation.
Project's Solutions	<ul style="list-style-type: none">- Streamline the ticket booking process.- An automated payment system.- An automated order tracking system.- Compare ticket prices.- Customer Support feature.

	<ul style="list-style-type: none"> - Encryption measures. - Clear information. - Multi-factor authentication options. - Expand the payment options. - User-friendly payment process with clear instructions and a secure environment. - Regularly update and enhance the security protocols.
Scope	<p>Bus management company in Viet Nam.</p> <p>End-users: Vietnamese customers who:</p> <ul style="list-style-type: none"> - love traveling. - are interested in traveling. - travel and use bus transportation services frequently. - love exploring various tourist attractions and new places. - have the habits and behaviors of using the Internet to access technological devices and digital platforms.
Purpose of the Project	<p>Develop automatic ticket booking system:</p> <ul style="list-style-type: none"> - Create a user-friendly app to book bus tickets. - Simplify the ticketing process and ensure real-time updates. <p>Enhance customer experience and convenience:</p> <ul style="list-style-type: none"> - Provide notifications on updates, seat assignments, and travel changes. <p>Comprehensive customer support:</p> <ul style="list-style-type: none"> - Offer troubleshooting, clear policies, and improve communication transparency.

1.1.2. Value

Table 1.2. Values

Customer	<ul style="list-style-type: none"> - Ability to quickly search for bus routes, compare prices, and book tickets from anywhere in Viet Nam. - Various options for choosing preferred seats, bus operators, and timings. - Assurance of secure transactions and booking confirmations. - Access to real-time updates on bus schedules, delays, and cancellations.
-----------------	---

	<ul style="list-style-type: none"> - Protection of user data and transactions. - User-friendly interface for diverse users. - Intuitive design and seamless user experience.
Application	<ul style="list-style-type: none"> - Fast, responsive, and error-free operation. - Higher usage and retention rates because of a user-friendly interface and features. - Growth of the user base by attracting new passengers and keeping current ones. - Ability to expand and manage more users. - Good reviews and promotion improve brand image.
Company	<ul style="list-style-type: none"> - Revenue generation through service offerings. - Building brand trust with quality service. - Expanding services and the user base. - Continuously improving application features and services. - Engaging with local communities for mutual benefits. - Reducing costs, time, and human resources. - Growing and dominating the online bus ticket booking market.

1.1.3. Introduction of Information Systems

The AutoBus information system is intended to revolutionize coach transport ticketing in Vietnam by enhancing its effectiveness, usability, and security. AutoBus' user-friendly interface makes it simple for users to look up routes, evaluate costs, and reserve seats, making it a convenient booking option for everyone. Assuring safe transactions and a variety of payment alternatives, the system's automated payment solution with multi-factor authentication and encryption promotes user convenience and confidence.

By automating order and revenue monitoring, decreasing manual labor, and increasing accuracy in controlling seat availability, pricing, and invoicing, AutoBus also tackles the operational difficulties associated with offline ticketing. In order to keep consumers informed and improve satisfaction, the system also offers real-time customer support and notifications for schedule updates.

Because AutoBus is scalable, it can grow both its user base and its services in response to demand, which helps the company achieve its objective of being the most popular online ticketing platform in Vietnam. AutoBus wants to change how consumers book and arrange coach travel by taking a contemporary, user-centered approach that will meet the changing needs of modern travelers.

1.1.4. Assumptions

AutoBus's design is predicated on a number of assumptions on users and their technology. To utilize the app for things like scheduling, ticket purchasing, and real-time payment, users must have reliable internet access. In order to properly use the app and its capabilities, it also assumes that users have a basic level of digital literacy.

Furthermore, AutoBus makes the assumption that the majority of users will access the app using contemporary devices running the most recent software that supports essential security features like multi-factor authentication and encryption. In order to handle real-time adjustments, like ticket availability and timetable modifications, the organization is required to possess a dependable data infrastructure.

These presumptions may restrict accessibility for users with outdated devices or inadequate internet connectivity, even while they simplify the app's design.

1.1.5. Research Methodologies

1.1.5.1. Basic Methodologies

- **Surveys:** conduct interviews to collect a wide range of perspectives and insights.
- **Information Synthesis:** systematically compose the collected data to form the fundamental background for research.
- **Evaluation:** scrutinize the amassed information for its practical relevance and applicability.
- **Comparative Analysis:** examine and contrast different alternatives to identify the best solutions.

1.1.5.2. Approach Methodologies

- **Customer Engagement:** interact with users of bus booking services in Ho Chi Minh City to obtain more insight into current trends and practical requirements.
- **Theoretical Inquiry:** conduct in-depth research by reviewing academic journals, reports, documents, and online sources from both domestic and international contexts, synthesizing relevant information and expanding upon previous research findings.
- **Observation and Analysis:** examine and analyze the operational workflows of ride-hailing services to gather primary insights.
- **Technology Exploration:** inspect essential programming languages for the project, such as Java and XML, as well as research database management systems and Web Services applications.
- **Interface Design:** analyze and formulate software interfaces specifically optimized for standard website platforms.
- **Database Management:** investigate and evaluate the NoSQL database management system, specifically MongoDB.

- **Software Testing:** perform extensive testing within the System Integration Testing (SIT) environment to ensure the software's functionality and reliability.

1.1.5.3. Data-collecting Methodologies

Obtain secondary data from reliable sources such as academic journals, official documents, reports, and studies available online, both from domestic and international sources.

1.1.6. Research Progress

The research process is built relying on the original research objectives. Initially, the authors conduct an investigation on consumer demands and their struggles. Then, essential literature is reviewed, and website programming languages, standard website operating systems, Web Services, and NoSQL databases are explored to establish a solid foundation for the research, development, and construction of a bus ticket booking website for customers. Finally, software testing is performed within the SIT environment to ensure readiness for market launch. Generally, we build our research progress based on **The SDLC model**, which includes five steps.



Figure 1.1. Research Progress

(Source: Author's source)

1.1.7. Structure of the Project

The project structure includes key sections:

- Introduction: providing an overview of the project, objectives, and scope
- Literature Review and Theoretical Background: covering information management systems in transportation and essential technology foundations
- System Planning: which involves feasibility studies, resource allocation, and timeline milestones
- System Analysis: focusing on functional and non-functional requirements, and business process modeling

- System Design: detailing system architecture, user interface, and data security
- System Implementation: including front-end and back-end development and payment integration
- Testing and Quality Assurance: with various testing phases and bug resolution
- Deployment and Maintenance: covering data migration, user support, and continuous improvements
- Conclusion: summarizing outcomes, future scalability, and lessons learned.

1.2. Context

1.2.1. *Background*

1.2.1.1. Global Passenger Transports

Global passenger transport encompasses travel by private and public road vehicles, rail, air, and non-motorized means, significantly contributing to economic growth. Forecasting demand in this sector is critical due to its economic, social, and environmental impacts.

Prior to the 2020 pandemic, projections anticipated steady growth, with OPEC in 2018 estimating an increase in passenger cars from 1133 million to 1969 million by 2040. By 2020, this estimate adjusted slightly to 1936 million for 2040, with a forecast of 2119 million by 2045, largely driven by non-OECD countries. Airbus projected a 4.7% annual growth in air travel globally until 2038.

In cities with high population densities, such as Tokyo, Hong Kong, and Seoul, public and non motorized transport modes are dominant, resulting in lower car usage and more reliance on rail due to reduced road space and slower average car speeds, especially in large urban centers.

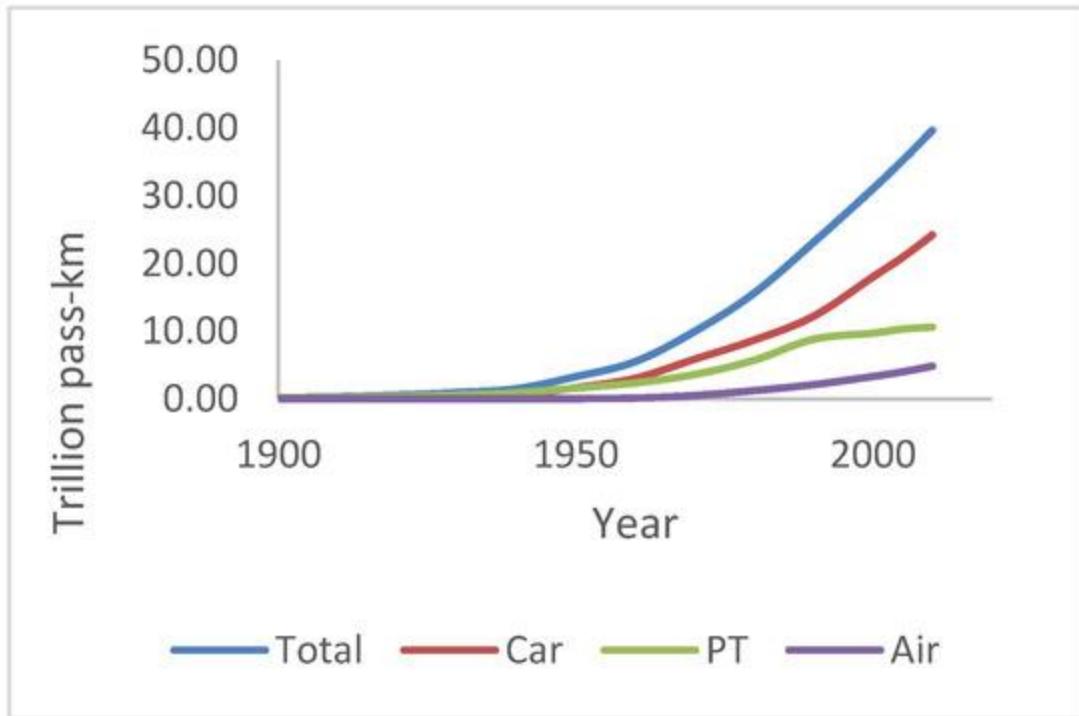


Figure 1.2. Global vehicular travel by mode, 1900–2016.

(Sources: <https://doi.org/10.3390/encyclopedia1010018>)

Two decades ago, Schafer and Victor [18] forecast the world's travel future out to 2050, mainly based on three assumptions. First, that on average, people everywhere allocate roughly 1.1 h per day for travel by all modes, including non-vehicular travel. Second, that at least in high-income countries, travel expenditures form a roughly constant share of household disposable income. Third, that global real GDP per capita would continue to grow at a constant rate of around 1.5% per annum. Given the second and third assumptions, it follows that total travel would continue to rise in line with total income. However, because of the first assumption of a daily travel time limit, faster modes would progressively replace the slower ones. In short, car travel would replace non-motorized modes and surface public transport, and air travel (together with very fast rail) would replace long-distance surface travel.

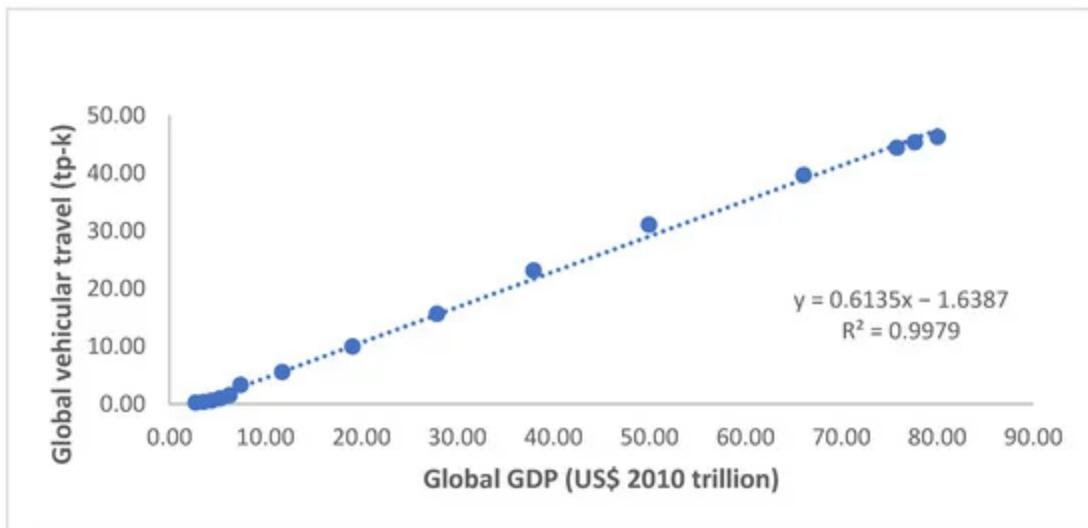


Figure 1.3. Global vehicular travel as a function of global GDP, 1900–2016

(Sources: <https://doi.org/10.3390/encyclopedia1010018>)

Current trends of global passenger transport:

Passenger transport plays a fundamental role in supporting economic growth, creating jobs and connecting people to essential services. As governments grapple with a mixture of old and new challenges to modernizing transportation systems, it is important to explore the emerging trends shaping the mobility industry as it enters the next era of exponential growth.

Trend 1: Improving product inventory and seat allocation

Ideally, a seat inventory allocation problem would consider that two or more transportation companies would be competing and operating on the same origin-to-destination route (Ammirato et al., 2020). Most of the seat allocation works are based on static problem formulation, where the transportation companies set a booking limit and once the sale of discounted tickets is closed, it is not reopened (Ammirato et al., 2020). Some authors focus on how firms can profit from offering multiple fares and limiting the availability of seats at low fares when they cannot segment the market (Ammirato et al., 2020).

Trend 2: Monitoring of industry data and events

Data are fundamental elements to design transportation offering, which translated in the scheduling of transportation services (Ammirato et al., 2020). In this sense, industry data is crucial to perform the vital activities of transportation, namely, schedule design, fleet assignment, bus maintenance routing and route scheduling. Strategic analysis of industry data is a key topic to avoid falling into strategic errors (Ammirato et al., 2020). Another crucial issue is the industry events, meaning those company changes (mergers, acquisitions, alliances) with impacts on the industry structure (Ammirato et al., 2020).

1.2.1.2. Vietnam Passenger Transport

In Vietnam, road transportation plays a dominant role. In 2019, road vehicles transported about 26.8% of goods (million tons.km) and about 63.1% of passengers (million passengers.km). In 2030, the market shares of inter-provincial cargo transportation by road, railway, and inland waterway will be 51.2%, 7.9%, and 30.9%, respectively. The market shares of inter-provincial passenger transportation by road and railway will be around 92% and 4.7%, respectively. In terms of urban traffic, with the high rate of urbanization and poor traffic infrastructure, meeting the mobility needs of urbanites becomes a major challenge.

The long-term state policies and orientations for urban transport are to build a highly connected and convenient public transport system. According to Decision No.355/QD-TTg and Decision No.318/QD-TTg, by 2020, the proportion of public passenger transport in major cities in Vietnam will reach 25–30%. However, the current public transport share in big cities only meets 9–15% of the mobility demand, far lower than the target. Because of poor public transport services, private vehicles, especially motorbikes, are preferable.

The prevalence of private motorcycles in Vietnam's urban traffic derives from many remarkable advantages including flexibility, suitability for short-distance travel, narrow road traffic condition, low land area occupation, and especially the purchase price and

operation cost, which are consistent with the living standard of the majority of people [4]. However, a large number and high density of private fossil fuel vehicles, of which motorcycles make up over 90%, put high pressure on traffic infrastructure, triggering traffic congestion as well as air and noise pollution in urban areas.

In recent years, with a young population structure (the median age in Vietnam is 32.5 years and the population aged 15–24 accounts for 70%) and improving average income per capita (Vietnam GNI per capita for 2019 was 2590 USD), there has been an emergence of the more comfortable, safer, and/or greener means of transport. A significant change in the private car consumption in Vietnam has been witnessed recently. However, the expensive purchase price, tax, fees, and high operation costs are hindering the accessibility of this vehicle type. In addition, traffic congestion, lack of car parking, as well as poor urban transport infrastructure (land fund reserved for transportation is only about 7.5–9.75%) are also barriers to private car adoption.

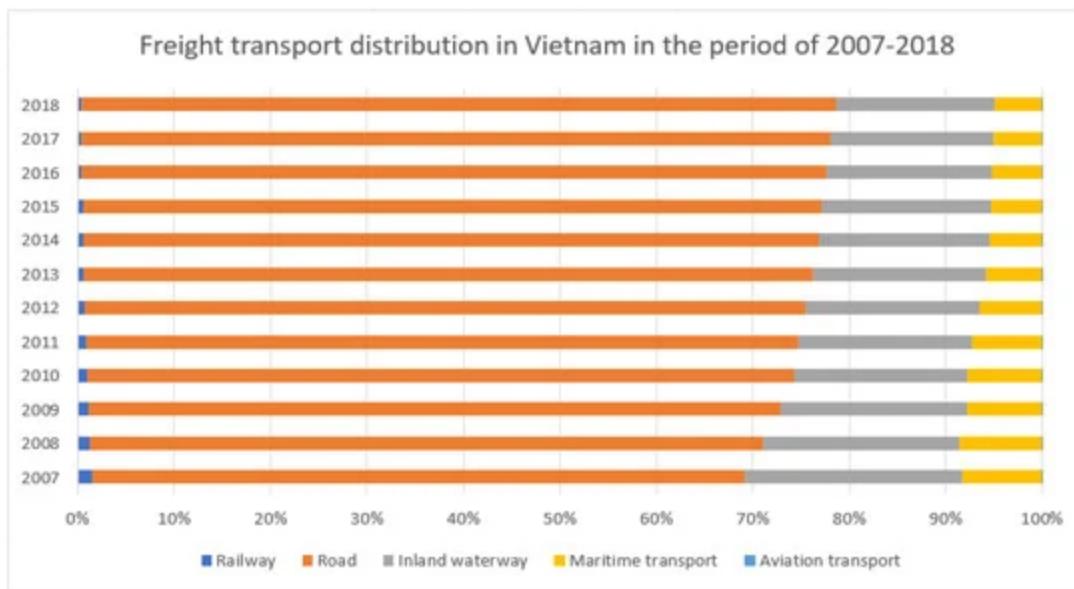


Figure 1.4. Volume of freight (thousand tons) carried by types of transport in the period of 2007–2018

(Sources: <https://www.mdpi.com/2071-1050/13/10/5577>)

Generally, road transport, inland waterways, and marine transport are the main modes of freight transport. However, road vehicles always express their salient role. In 2007, there was 403,361.8 tons of freight carried by road vehicles, followed by inland waterways with 135,282.8 tons. In 2018, roadways contributed up to 78.32% of the total volume of freight—a volume 4.78 times higher than the transportation capability of the second important transport mode—inland waterway. In the same period, a growth of 2.96 times in the volume of freight also could be seen in road transport, which highly outperforms the remaining transportations.

Mode	Tons Loaded (%)	Ton-Kilometers Performed (%)	Average Distance (Kilometers)
Road	77.2%	23.7%	50
Rail	0.4%	1.3%	615
Inland waterways	17.1%	18.9%	212
Coastal shipping	5.2%	55.7%	2046
Aviation	0.0%	0.3%	2333

Figure 1.5. Percentage of domestic freight transport in Vietnam by mode of transport (2016)

(Sources: <https://openknowledge.worldbank.org/entities/publication/b81bf00e-3c4d-570e-8ba0-69008e2d78f0>)

Road vehicles are the preferable choice for short-range transportation. Nevertheless, in spite of the shortest transport distance, road vehicles contributed up to 23.7% of total ton-kilometers of freight transport, reaching the second share of ton-kilometers transported.

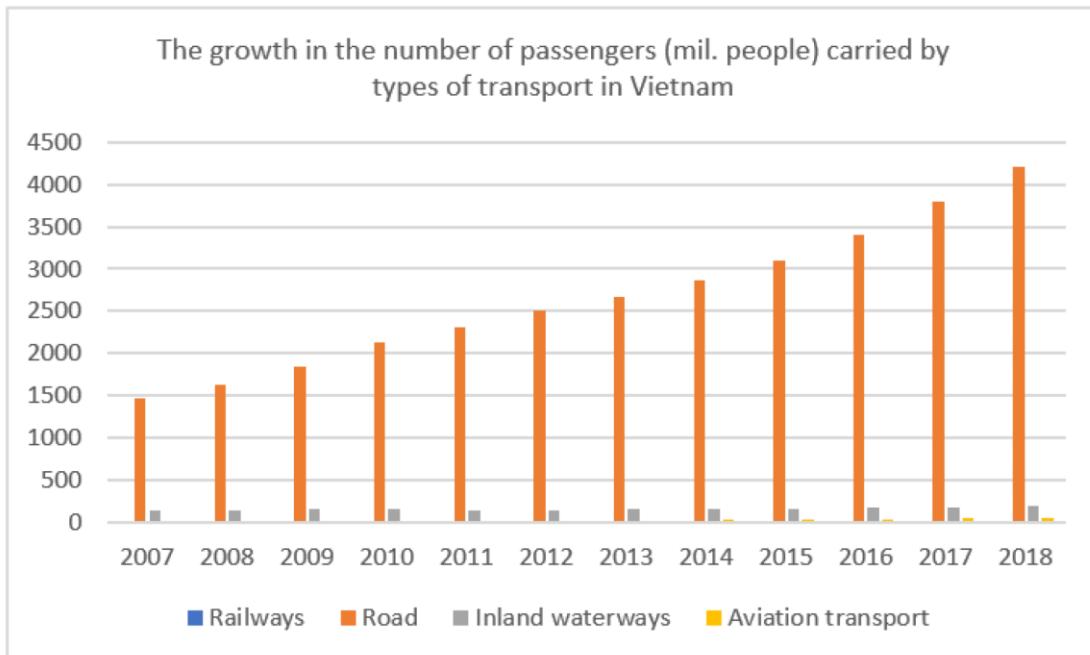


Figure 1.6. The growth in the number of passengers carried (mil. people) by types of transport in the period of 2007–2018.

(Sources: <https://www.mdpi.com/2071-1050/13/10/5577>)

Year	Total (Mil. People)	Railways (Mil. People)	Road (Mil. People)	Inland Waterways (Mil. People)	Aviation Transport (Mil. People)
2007	1638	11.6	1473	144.5	8.9
2008	1793.5	11.3	1629	143	10.2
2009	2016.9	11.1	1843.6	151.3	10.9
2010	2315.2	11.2	2132.3	157.5	14.2
2011	2476.1	11.9	2306.7	142.4	15.1
2012	2676.5	12.2	2504.3	145	15
2013	2839.9	12.1	2660.5	150.4	16.9
2014	3056.8	12	2863.5	156.9	24.4
2015	3310.5	11.2	3104.7	163.5	31.1
2016	3623.2	9.8	3401.9	172.9	38.6
2017	4027.1	9.5	3793.2	179.9	44.5
2018	4456.2	8.6	4206.6	192.1	48.9

Figure 1.7. Number of passengers carried (mil. people) by types of transport in the period of 2007–2018

(Sources: <https://www.gso.gov.vn/en/px-web/?pxid=E0901&theme=Transport%2C%20Postal%20Services%20and%20Telecommunications>)

With regard to passenger transport, the population growth, urbanization, business, commerce, and tourist activity have led to a rising demand for passenger mobility. In correlation between other modes, roadways accounted for 94.4% of total passengers carried in 2018.

1.2.1.3. The rise of bus booking website development

The travel industry has rapidly digitized, especially with online booking platforms and mobile apps. Bus booking apps, in particular, have made ticket booking and trip planning more convenient. Digital tourism started with automated reservation systems (CRS), which evolved into global distribution systems (GDS) that support B2B and B2C travel bookings. Today's bus ticket websites offer user-friendly features like easy registration, real-time bus availability, routes, schedules, and delay updates, making travel seamless. These systems are now essential, meeting rising demand in countries like Vietnam for efficient bus booking services.

1.2.2. *Demands*

Table 1.3. Demands

Demands		Descriptions	Evidences
Ease of use	Page loading speed	Page loading speed is fast	A 31% improvement in LCP (which measures a page's loading performance) led to a 15% improvement in visitor-to-site conversion rates. It resulted in an 11% improvement in cart-to-site ratio and an 8% increase in sales (According to rentracks.com)
	Intuitive interface	Customers require an app with many functions but still convenient and easy to use.	The website has an intuitive interface with a high conversion rate of 7% (Marketsplash, 2023).
	Cross-platform access	Make sure the app runs on many different operating systems.	According to some studies, there are approx 70-80% of users Prefer applications that can run on multiple platforms ¹² . This is due to its convenience and accessibility from many different devices, helping users have a seamless and consistent experience (According to Mangoads.vn)
Real-time information	Real information about tickets as well as the itinerary	- Customers want information about the	Several studies and reports show that integrating real-time features can improve customer retention rate by 20% to 40%

		<ul style="list-style-type: none"> remaining tickets to easily book tickets - Sharing itineraries directly makes it convenient for customers to follow. 	
User personalization	User profile	Easily analyze customer needs as well as optimize schedules and promotions	Up to 49% of customers buy products that they had no intention of buying before just because they experienced good customer service due to personalization (According to subiz.com)
Ticket booking process	Effective filter	Users who want a quick booking experience should optimize their filters and search.	Effective filtering results in 64% of customers enjoying travel and 26% finding the same joy in the booking process.
	Easy process	The booking process is simplified based on customer data.	39% of respondents feel more satisfied when bookings can be made quickly (FounderJar, 2023).
Transparent pricing policy	Clear ticket price analysis	Provide full information about prices, daily fees for each trip or fixed prices for customers to easily understand.	72% of users are excited about booking tickets with clear ticket prices (FounderJar, 2023).
	Cancellation and refund policy, ticket refund	Be transparent about pricing policies on platforms such as	Flexible cancellation increases customer satisfaction by 40%

		websites and applications for customers to conveniently follow.	
Customer reviews		Allowing customers to rate both shows professionalism in service work and demonstrates the desire to increasingly improve the service.	However, some studies show that approx 30-40% of users Be prepared to provide feedback when reasonably requested and without causing inconvenience (According to: Approi.com)

1.2.3. Business Problems Statement/Business Needs

Table 1.4. Business Problems Statement

Problem space

Why do we do this?	Problem statement: Offline	Impact of this issue:
<p>The need to use public transportation tends to increase due to convenience as well as world trends. Besides the need to enhance customer experience, improve service, and increase interaction with customers, web apps are an effective solution for businesses to solve many different problems at a reasonable cost.</p>	<ul style="list-style-type: none">• Cost of printing tickets and hiring staff.• Difficulty in paying bus fare.• Inconvenient for customers.• Lack of detailed information on bus services, amenities and priority seating.• Lack of interaction with customers and taking feedback.	<ul style="list-style-type: none">• Difficulties in management and inventory work.• Lost revenue.• Reduced customer loyalty.• Data accuracy is not guaranteed.
<p>How do we measure success?</p> <ul style="list-style-type: none">• Investigate and analyze problems on existing systems and provide solutions to customers and management by having a stable system that individuals can operate within a short period of time.• Identify the relevant features of the various components and methods required for an Online Bus Booking System Website.• Deploying bus ticket sales on the web and website helps customers receive many benefits through the system, reducing waiting time and making online payments.• Improve the user interface and ensure that their platform is accessible, secure, and responsive.• Implementing strong security protocols, optimizing your website, offering flexible booking options, and providing clear and comprehensive information are important steps to enhance customer satisfaction and trust. user.		

Authentication

What do we know?

Vinbus App, Uni pass are outstanding examples of online car booking systems in Vietnam. It offers a wide range of services, including:

- Transparency of information: Price, schedule, time, customer care policy...
- Offering a wide range of promotions and discounts, new users can often find special offers, covering a variety of routes across the country, making it possible to book tickets to almost any destination In Vietnam. The platform is designed to be intuitive, allowing users to easily search for bus tickets, compare schedules and book their journey.
- Convenient payment with many different forms.
- Updating specific routes helps customers conveniently track their journey.

What do we need to answer?

What do customers want?

Is it easy to use?

Are the app's features easy to understand?

How simple is the user interface to understand?

What is the average time for each operation on the app?

1.3. Problems and Solutions

Table 1.5. Problems and Solutions

Problem	Solution
Acceptability	Encourage customers to use the app through discount campaigns.
Lack of recognition	Increase advertising and PR to bring the app into users' daily habits.
The booking process is inefficient	<ul style="list-style-type: none">- Ticket booking process: Optimize the ticket booking process with a simplified user interface, allowing users to easily select travel dates, routes and seats.- Automatic payment system: Integrated automatic payment system, providing many different options such as credit cards, loyalty cards, banks, e-wallets
Lack of real information	<ul style="list-style-type: none">- Provide real information to customers at a specific time.
Compare ticket prices	<ul style="list-style-type: none">- Compare fares: Develop a feature that allows users to compare fares between different time slots and bus operators.
Lack of interaction	<ul style="list-style-type: none">- Customer support feature: Integrate customer support feature right in the website, allowing users to

	access support through 3rd platforms such as Messenger, Zalo...
Security	<ul style="list-style-type: none"> - Encryption measures: Use security measures, update information regularly. - Announce privacy policies on platforms - Multi-factor authentication options: Introducing multi-factor authentication options to add an extra layer of security to user accounts.

1.4. Risks and Assumptions

1.4.1. Analyze risks and assumptions

Table 1.6. Risks and assumptions

Risks	Assumptions
Feasibility	Target users may not accept the app as expected or due to customers' direct ticket purchasing habits.
Competitors	The appearance of similar web apps/apps can make it difficult to attract customers
Server capacity is not enough to handle peak booking times	Server infrastructure may not be capable of handling large numbers of simultaneous bookings during peak travel seasons.

Demand for buses decreased	In many places, the demand for buses has decreased because the habit of using personal vehicles still exists.
Security	Security breaches or cyber attacks targeting the app website could lead to unauthorized access to sensitive user data, financial information, or personal information.
System interruption	Due to unforeseen technical issues, such as software errors, server errors, or compatibility issues with new device models or operating system updates.
Feedback and improvement	Use user feedback to identify areas for improvement and prioritize feature development. Ensure a streamlined feedback mechanism and actively solicit user input to drive continuous iteration and enhance the app's functionality and user experience.

1.4.2. Feasibility study

- **Technical Feasibility:** The system is technically feasible with existing technologies. The authentication and security protocols integrate and enhance data security, ensuring that the technical requirements are achievable.
- **Economic Feasibility:** The project appears economically feasible because its benefits outweigh the costs. It can enhance customer satisfaction and reduce expenses associated with manual tasks and paper tickets. Additionally, the costs include hosting, database maintenance, and minor software expenses have to be ongoing.

- **Operational Feasibility:** The project emphasizes a user-friendly design and includes intuitive features like a streamlined booking flow, clear navigation, and real-time updates. These are beneficial for end users, who can use the system without extensive training, addressing a broad audience of users familiar with digital platforms.

1.5. User Roles and Permissions

Table 1.7. User Roles and Permissions

User/Role/Actor	Example	Actor Description	Frequency of Use	Features Used	Permissions
Normal User	Passengers	Individuals who reserve coach tickets for trips, manage bookings, and access travel details.	Occasional, depending on trip planning	<ul style="list-style-type: none"> - Account management - Search for coach routes and schedules - Book tickets - Monitor ongoing trips - Pay unpaid tickets - Access booking history - View ticket details - Cancel bookings - Ratings - View promotions - Push Notifications 	<ul style="list-style-type: none"> - Access and update personal details. - Link payment methods: Bank accounts/ E-wallets. - View and purchase tickets. - View ticket's notification. - View drivers' information. - View Invoices. - Complete payments. - Request refunds.

					<ul style="list-style-type: none"> - Review purchase history. - Cancel booking tickets. - Apply discounts or promotions to purchases. - Search for coach routes based on demands.
Administrative User	Managers, system administrators	Responsible for managing the booking system, including routes, schedules, fares, and user accounts.	Daily or frequently	<ul style="list-style-type: none"> - Manage coach routes. - Manage promotions. - Track sales and analyze performance. - Customer relationship management. - Customer information management. 	<ul style="list-style-type: none"> - Manage bus routes and schedules (add, edit, delete). - View booking statistics. - Manage user accounts (create, edit, delete). - Handle customer support and issue resolution.

1.6. Business Function Diagram (BFD) of AutoBus

The diagram includes several main modules that work together to provide various functionalities, each broken down into specific tasks or features:

- Account Management: Allows users to register, log in, recover passwords, manage profiles, and update personal information.
- Booking Ticket: Enables route search, sorting/filtering options, seat selection, route details, and choice of pickup/drop-off points.
- Payment: Offers payment method selection, order review, voucher application, and payment completion.
- Invoice: Users can view invoices, total amount, and detailed booking information.
- Booking History: Access to booking history, with options to view paid/unpaid tickets, cancel bookings, and view completed ticket details.
- Support & Feedback: Provides feedback options, star rating, and customer service hotline access.
- Notifications: Manages system notifications for updates, promotions, and reminders.

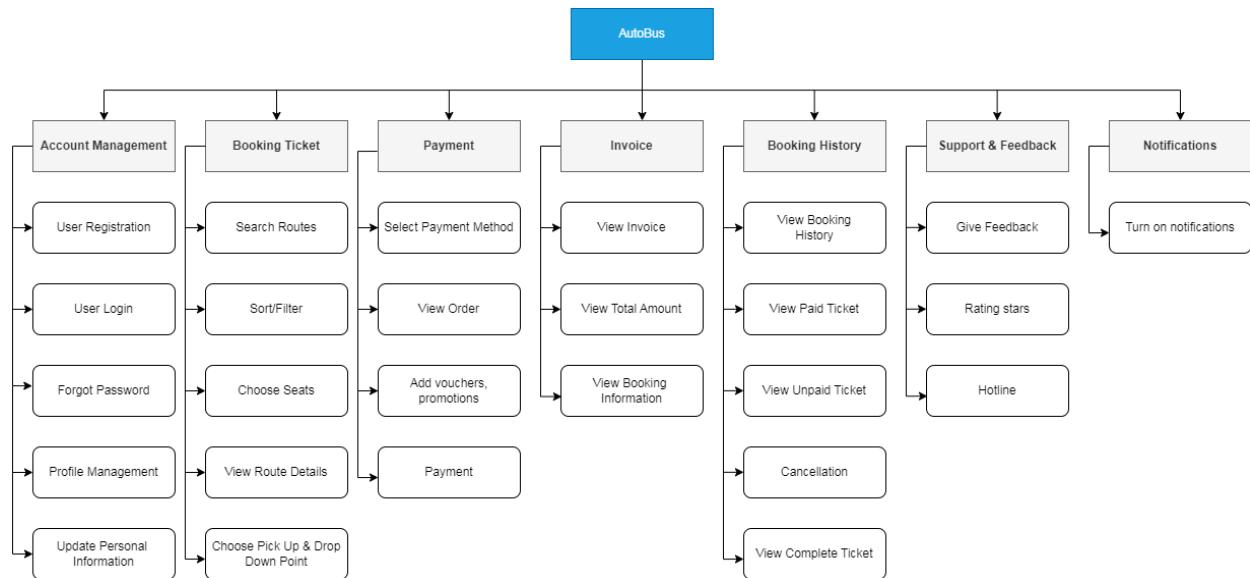


Figure 1.8. Business Function Diagram of AutoBus

(Sources: Authors' Sources)

1.7. Tool Used

Figma

Figma is a collaborative interface design tool that allows teams to create, prototype, and collaborate on designs in real-time, streamlining the user experience (UX) and user interface (UI) website design process before publishing to the frontend on Visual Studio Code.

Draw.io

Draw.io, now known as diagrams.net, is a free and flexible tool for creating diagrams such as flowcharts, network designs, and ERDs. Teams can work together and exchange ideas because to its user-friendly UI and cloud connection with programs like Google Drive and OneDrive. It adjusts to different presentation needs with output options like PNG, PDF, and SVG. Draw.io is a vital tool for planning and brainstorming in both technical and non-technical domains since it improves communication by graphically clarifying difficult ideas.

SQL Server

Microsoft created SQL Server, a dependable Relational Database Management System (RDBMS) for managing, storing, and analyzing data. While tools like SSRS and SSAS facilitate reporting and data analytics, its sophisticated T-SQL language offers strong querying capabilities. Data integrity is guaranteed by its robust security features, which include encryption and data masking. From small startups to huge corporations, SQL Server is a top option for managing data-intensive applications due to its scalability and connectivity with Azure.

MongoDB

MongoDB is an open source NoSQL database management program. NoSQL (Not only SQL) is used as an alternative to traditional relational databases. NoSQL databases are quite useful for working with large sets of distributed data. MongoDB is a tool that can manage document-oriented information, store or retrieve information.

CHAPTER 2: THEORETICAL BACKGROUND

2.2. Terms/Acronyms and Definitions

2.2.1. BPMN - Business Process Modeling Notation:

The Business Process Modeling Notation (BPMN) is visual modeling language for business analysis applications and specifying enterprise process workflows, which is an open standard notation for graphical flowcharts that is used to define business process workflows. It is a popular and intuitive graphic that can be easily understood by all business stakeholders, including business users, business analysts, software developers, and data architects.

BPMN allows us to capture and document business processes of an organization in a clear and consistent way that ensures relevant stakeholders, such as, process owners and business users are involved in the process. Thus, the team can respond to any issues identified in the processes more effectively. BPMN provides comprehensive and yet rich notations that can easily be understood by both technical and non-technical stakeholders.

In BPMN, the processes are described by means of diagrams with a series of graphic elements. Such visual presentation makes it easy for the users to understand the logic of a process.

BPMN has been primarily developed to design and read both simple and complex diagrams of business processes. For that, the BPMN standard classifies the graphic elements by categories: as a result, the elements are easily recognized by the users who work with business process diagrams.

2.2.2. DFD - Data Flow Diagram

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish.

- Context Diagram (Level 0)
- Decomposition (Level 1)
- Deeper Dives (Level 2)
- Increasing Complexity (Level 3)

2.2.3. Use case

Use Case is a technique commonly used in systems and software engineering to help users understand the functional requirements of the system. In addition, Use Case is also a technique that can describe the interaction between the system and the user in the same environment and for the same specific purpose. This interaction can be one of the following two cases:

- How external user objects interact with the system.
- How this system interacts with other systems.

A use case document should establish and identify a few key components — these are:

- System
- Actors
- Scenario
- Use case

Types of Use cases:

Use cases come in two forms: business and system. A system use case is a detailed look at how users interact with each part of a system. It highlights how unique inputs and contexts cause the system to reach different outcomes. This level of detail highlights how a system's individual functions work in any scenario.

Business use cases paint a more general picture of how a user might interact with your business to reach their goals. Instead of focusing on technical detail, it's a cause-and-effect description of different inputs.

2.3. SDLC - The system development life cycle

2.3.1. System Planning

The planning phase in the Systems Development Life Cycle (SDLC) is crucial, as it lays the groundwork for the entire project. This phase focuses on identifying the system's objectives, defining its scope, setting timelines, and allocating the necessary resources. Effective planning aligns the development process with the organization's goals, guiding the project in a clear and structured direction.

Known as the feasibility stage, this phase helps developers thoroughly plan for the upcoming project. It includes defining the problem and the scope of any existing systems while determining the objectives for the new system. By developing a detailed outline for the project, potential problems can be identified early, before they affect the overall development.

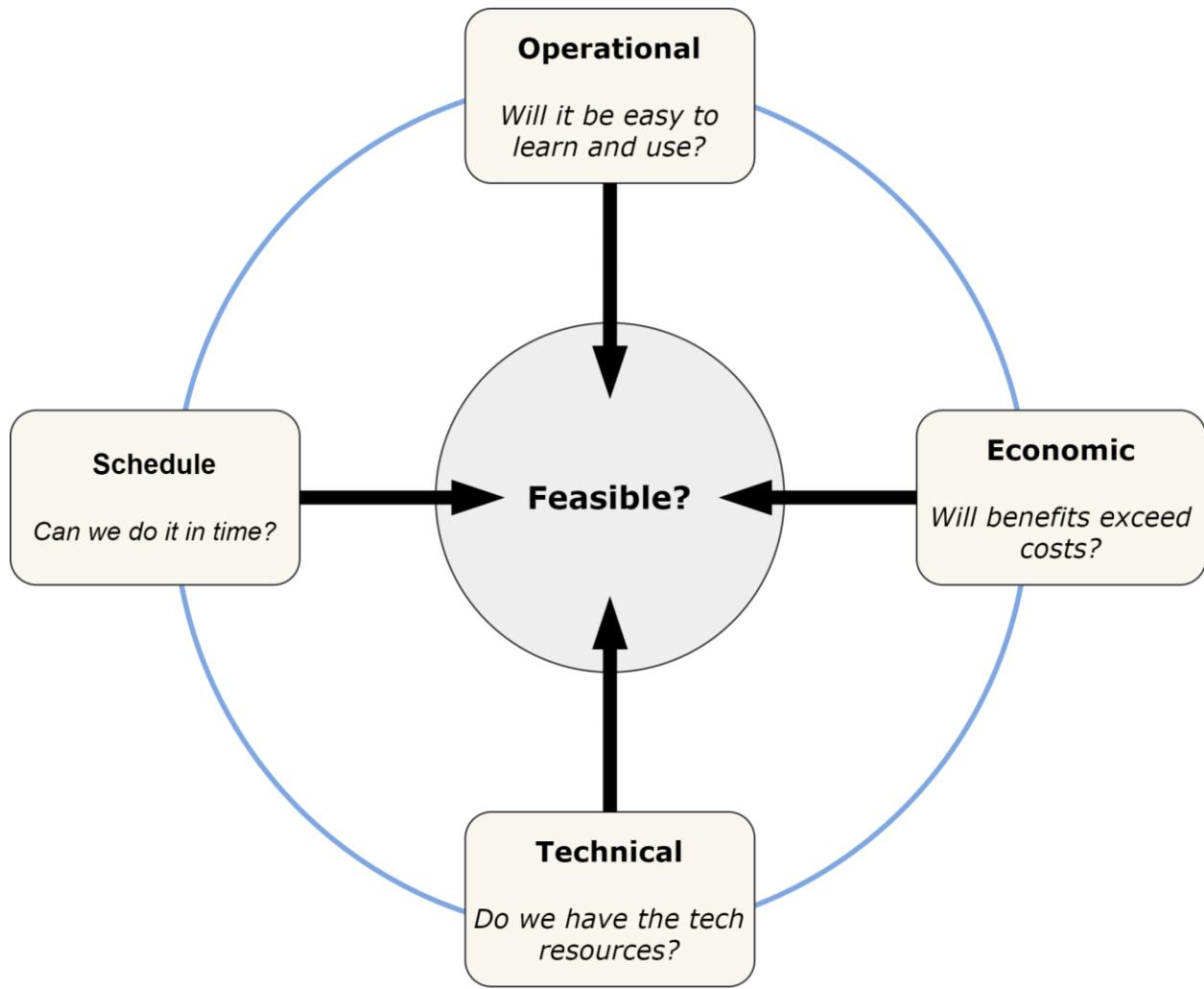


Figure 2.1. Overview of Feasibility

(Source: Authors' source)

The planning phase also ensures that the team secures the funding and resources needed to execute the project successfully. An essential outcome of this stage is the project schedule, which becomes critical when development is geared towards a commercial product that must meet a specific market deadline.

The process begins with a systems request, which describes problems or desired changes, acting as the starting point for further evaluation. The next step involves conducting a

preliminary investigation, which is a vital step that includes fact-finding techniques, feasibility studies, and defining the project scope and constraints.

Analysts must carefully consider the company's mission, objectives, and IT needs. After a thorough investigation, they compile a report for management that outlines the problem or opportunity, the project's scope, constraints, and the feasibility of moving forward.

Project management is also a significant aspect of the planning phase. By planning and scheduling each stage of the project, the development process is kept on track, and deadlines are met. During this stage, developers may also use advanced tools such as predictive analysis and AI simulations to assess early-stage ideas and test their validity. These tools help project managers build a comprehensive picture of the resources needed for long-term success, estimate costs, and identify potential obstacles.

At its core, the planning phase answers a fundamental question: how can a specific problem be solved with a software solution? This phase involves analyzing the resources, costs, and timeline required for successful project completion. It serves as a blueprint for the entire SDLC, allowing project managers to set clear deadlines and timeframes, ensuring that the product is ready for the market within the set timeline.

In conclusion, the planning phase is **essential** for the successful execution of the SDLC. It provides the strategic framework, identifies the necessary resources, defines the scope, and sets timelines that ultimately pave the way for the smooth development and successful launch of the final product.

2.3.2. System Analysis

The **Systems Analysis** phase in the Systems Development Life Cycle (SDLC) is a critical stage that sets the foundation for building a logical model of the system. It focuses on understanding the current system through fact-finding techniques and defines requirements for the new system. The output of this phase is the **System Requirements Document (SRS)**, a comprehensive guide for developers as they proceed with system development.

The key tasks in this phase include requirements modeling, data and process modeling, object modeling, and the exploration of development strategies.

Requirements Modeling

In the first step, developers engage in fact-finding activities to describe the current system and gather the needs for the new system. This is essential in ensuring that the system supports business objectives and fulfills user expectations. Requirements modeling involves interviewing stakeholders, conducting surveys, observing processes, and reviewing existing documentation to identify what the new system must achieve. These insights help form the basis of the system requirements.

Data and Process Modeling

Once the requirements are gathered, developers move to **data and process modeling**. This stage involves graphically representing the system's data flows, entities, and processes, helping stakeholders visualize how information will move through the system. Tools such as data flow diagrams (DFDs) and entity-relationship diagrams (ERDs) are often used to clarify how data is captured, stored, processed, and used.

Object Modeling

Another essential aspect of the Systems Analysis phase is **object modeling**, where developers create objects that represent real-world entities, transactions, and events that interact with the system. Object models help in designing the system in a way that reflects how users interact with it in the real world. By defining the attributes and behaviors of these objects, the system's structure becomes clearer, laying the groundwork for future development stages.

Development Strategies

As the analysis phase continues, various **development strategies** are considered to determine how the system will be developed. This might include evaluating current software trends, exploring different development methodologies (e.g., agile vs. waterfall), or deciding whether to outsource certain parts of the project. These strategic decisions ensure that the system is developed efficiently, aligns with the project's goals, and stays within the allotted budget and timeframe.

Fact-Finding Techniques

A central component of systems analysis is the use of **fact-finding techniques**. These techniques are employed to gather detailed information about the system's requirements and its operational environment. Common techniques include interviews, questionnaires, document reviews, and direct observations. Each method provides valuable insights into the system's current performance and areas for improvement. For example, interviews can offer in-depth qualitative feedback, while surveys can provide broad quantitative data from a larger sample.

Fact-finding is organized around the **Who, What, Where, When, How, and Why** questions to get a full picture of the system's requirements. These questions help identify business functions, user needs, system performance requirements, and other critical aspects of the project.

Documentation and the SRS

The culmination of the analysis phase is the creation of the **Software Requirements Specification (SRS)**. This document outlines all the specifications for the system, including hardware, software, and network requirements. The SRS serves as a blueprint for the rest of the project, ensuring that developers have clear guidelines to follow and preventing scope creep. It also sets specific performance, security, and control requirements to ensure the system meets user needs and functions efficiently in its operating environment.

The **System Analysis** phase is an interactive process that provides a strong foundation for the system's development. Although SDLC models such as the waterfall method depict this phase as sequential, it often requires constant feedback and iterations to ensure that the system's design aligns with business goals.

2.3.3. *System Design*

The Design phase translates the requirements gathered during Analysis into a detailed technical blueprint. This includes designing the system's architecture, database models, user interfaces, and defining system components. The outcome of this phase provides the technical structure needed to guide the upcoming development and implementation activities. This step includes the following:

2.3.3.1. *Interface design*

The goal of this phase is to define the set of interface objects and actions i.e., control mechanisms that enable the user to perform desired tasks. Indicate how these control mechanisms affect the system. Specify the action sequence of tasks and subtasks, also called a user scenario. Indicate the state of the system when the user performs a particular task. Always follow the three golden rules stated by Theo Mandel. Design issues such as response time, command and action structure, error handling, and help facilities are considered as the design model is refined. This phase serves as the foundation for the implementation phase.

User Interface (UI) Design

User Interface is one of the most common front-end app view and direct human-computer interactions in which users can manipulate and control software as well as hardware. It can include all methods and devices used to accommodate interaction between machines and users. User interface can take out many forms, but always accomplishes two fundamental tasks :

- Communicating information from the machine to the user.
- Communicating information from the user to the machine.

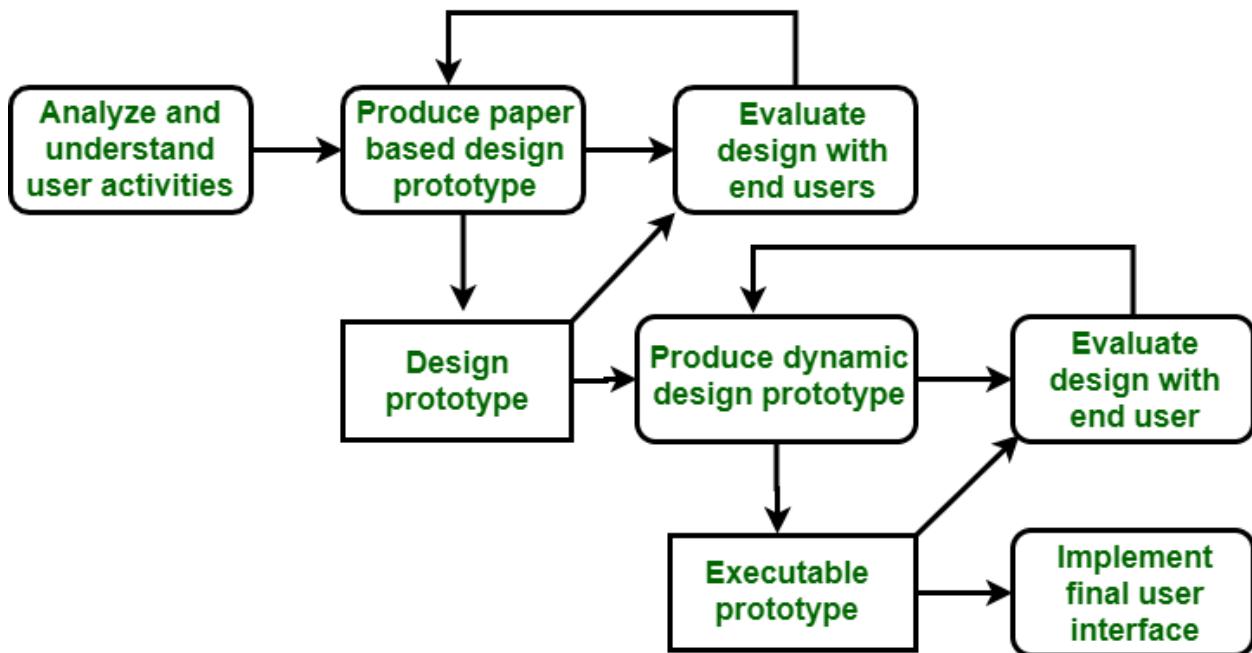


Figure 2.2. User Interface Design

(Source: [Qualities of Good User Interface Design - GeeksforGeeks](#))

The flowchart illustrates a user-centered design process that begins with analyzing and understanding user activities. Based on this, a paper-based design prototype is created and evaluated with end users to gather feedback. After refinement, a more detailed design prototype is produced, leading to the development of a dynamic, interactive version. This dynamic prototype is then tested again with users for further evaluation. Following this, an executable prototype is developed, which eventually leads to the implementation of the final user interface. Throughout the process, iterative feedback and refinement ensure that the design evolves in response to user needs and insights.

User Experience (UX) Design

User Experience (UX) Design is a crucial step because it focuses on creating products that are easy to use, meet user needs, and provide a positive experience. A well-designed UX enhances user satisfaction, leading to increased retention and word-of-mouth recommendations, while also boosting conversion rates by simplifying the user journey. Investing in UX early reduces development costs by addressing issues before they become expensive problems post-launch. Additionally, good UX improves accessibility, ensuring that products are usable by a wider audience, including those with disabilities. It also supports SEO by enhancing site performance and usability, leading to higher search engine rankings. Ultimately, great UX fosters brand loyalty by creating a positive emotional connection between users and the brand. Below are Key Components of UX Design:

- **Usability:** Ensures that a product is easy to use and understand.
- **Accessibility:** Makes products usable for people with a wide range of abilities.
- **Functionality:** Ensures the product performs its intended tasks effectively.
- **Aesthetics:** Involves creating a visually pleasing interface.
- **Interaction Design:** Focuses on creating engaging interfaces with logical and thought-out behaviors.
- **User Research:** Involves understanding the needs, behaviors, and pain points of users through interviews, surveys, and usability testing.

Interaction Design

Interaction design focuses on crafting meaningful and engaging interactions between users and products. It involves determining how users will interact with the interface, ensuring that every button, control, and gesture works seamlessly to meet user expectations. Good interaction design revolves around creating intuitive pathways for users to achieve their goals with minimal friction, paying close attention to how tasks flow within the interface and how users respond to different feedback mechanisms such as animations, sounds, or visual cues.

By focusing on logical and thoughtful behavior of each interface element, interaction design helps make the experience feel natural and easy. Elements such as button placements, gestures, navigation flows, and response to user actions are carefully considered to enhance the user's overall engagement. The ultimate goal of interaction design is to create a fluid, cohesive experience that keeps users immersed in the task without unnecessary confusion or frustration. It's closely tied to both usability and accessibility, ensuring that users of all types can engage with the product effortlessly.

2.3.3.2. Data design

The data design phase in SDLC is vital for laying the groundwork for how data will be organized, managed, and secured within the system. By addressing data structures, data flow, storage, security, and integration, data design ensures that the system can handle its data needs effectively, supporting both current functionality and future growth. This involves creating a detailed data model of the system, including:

Data Structures: A key component of data design is defining the data structures that will be used to organize and store information. This involves creating detailed data schemas that map out the structure of databases, including the creation of tables, fields, and the relationships between different pieces of data. These structures ensure that data is efficiently stored and can be easily retrieved, manipulated, and updated. A well-designed data structure helps in optimizing system performance and managing large amounts of data.

Data Flow Diagrams (DFDs): To visualize how data moves through the system, Data Flow Diagrams (DFDs) are created. DFDs illustrate the flow of information between inputs, processes, and outputs, making it easier to understand how data interacts with different parts of the system. By mapping the flow of data, DFDs help identify any potential bottlenecks or inefficiencies in data processing and ensure that data is correctly processed from one stage to another.

Data Storage: In the data design phase, deciding on appropriate data storage solutions is crucial. This involves selecting the right type of database (e.g., relational or non-relational), data warehouses, or other storage mechanisms based on the system's scalability and performance requirements. The storage design must ensure that the system can handle the expected data volume while maintaining quick access and retrieval times, even as the system grows.

Data Security: Another critical aspect of data design is ensuring that data is secure. Measures need to be implemented to protect the integrity, confidentiality, and availability of the data. This includes encryption, access control mechanisms, and data backup strategies to prevent data loss or unauthorized access. Proper data security protocols also ensure compliance with regulations and standards, such as GDPR or HIPAA, depending on the nature of the system and the industry it operates in.

Data Integration: Modern systems often need to integrate data from various sources, including external systems or legacy databases. In this context, data design must include strategies for data integration, ensuring that information from different systems can be combined, standardized, and used effectively within the new system. This could involve creating interfaces or data pipelines that facilitate the smooth exchange of data between different platforms.

2.3.3.3. System architecture

In the System Development Life Cycle (SDLC), the system design phase is a critical stage that establishes the architecture of the system by transforming requirements gathered during the analysis phase into a comprehensive blueprint for development. This phase begins with defining system requirements, where essential hardware, software, and network components are specified to ensure effective functionality. Architectural design follows, focusing on identifying major system components, outlining user interfaces, designing data flow, and planning necessary security measures to protect data integrity. Detailed design then delves into the specifics of each component, including database

structure, application logic, and integration strategies to ensure seamless interaction between different parts of the system. Additionally, modeling and diagramming tools, such as class diagrams, sequence diagrams, and flowcharts, are employed to visualize the system's architecture and interactions. Finally, prototyping plays a pivotal role in this phase, allowing developers to create preliminary models of the system to gather feedback from stakeholders and make iterative improvements based on user input. Overall, the system design phase is essential for ensuring that all requirements are meticulously planned and that the final architecture is robust, efficient, and user-friendly.

2.3.4. System Implementation

The Implementation phase involves deploying the developed system into a live environment. Key activities include system installation, migrating data, training users, and configuring infrastructure. This phase requires thorough planning to ensure a smooth transition from the existing system to the new one with minimal disruptions.

Design to Coding (Dependent on Design Method)

The process of moving from design to coding varies based on the chosen design methodology (e.g., Waterfall, Agile, etc.). In traditional methodologies, developers meticulously follow the design specifications to write code, ensuring each aspect of the design is implemented accurately. In contrast, Agile methodologies may involve iterative cycles where coding, testing, and design evolve concurrently, allowing for more flexibility and adaptability to changing requirements.

Coding

During the coding phase, developers translate the design specifications into actual code using appropriate programming languages and tools. This involves writing, reviewing, and refining code to ensure that it meets the defined requirements and adheres to coding

standards. Developers may work collaboratively in teams, often employing version control systems to manage changes and maintain code integrity throughout the process.

Testing the System

Once coding is complete, the system undergoes rigorous testing to identify and rectify defects. This phase typically includes various types of testing, such as unit testing (testing individual components), integration testing (ensuring components work together), system testing (validating the complete system), and user acceptance testing (gathering feedback from end-users). The goal is to ensure that the system functions as intended and meets all specified requirements.

Documentation

Documentation is a crucial part of the development process, providing a comprehensive record of the system's design, functionality, and usage. This includes technical documentation for developers, user manuals for end-users, and system architecture documentation. Proper documentation ensures that future maintenance and upgrades can be carried out efficiently and that users understand how to effectively utilize the system.

Management Approval

Before proceeding to the next phases, the system must receive management approval. This typically involves presenting the system to stakeholders for review and ensuring that it aligns with business objectives and user needs. Management approval serves as a checkpoint, confirming that the project is on track and meets the expectations set forth during the planning and design phases.

Testing in Operational Environment

After receiving management approval, the system is tested in a real operational environment. This phase ensures that the system performs effectively under real-world

conditions, including assessing its performance, reliability, and security. Testing in the operational environment helps identify any issues that may not have been apparent in earlier testing phases and validates that the system is ready for deployment.

Training

Once the system has been thoroughly tested and is deemed operational, training sessions are conducted for end-users and administrators. Effective training ensures that users are familiar with the system's functionality, features, and best practices for utilization. This may involve workshops, hands-on sessions, and the provision of training materials to support users in adapting to the new system.

Data Conversion

Data conversion is the process of migrating existing data from previous systems into the new system. This step involves careful planning to ensure that data is accurately transformed and integrated, minimizing the risk of data loss or corruption. Data cleansing may also be necessary to ensure the quality and integrity of the data being imported into the new system.

System Changeover

The final step is the system changeover, where the new system is fully implemented and replaces the old system. This can occur through various methods, such as direct changeover (immediate switch), phased changeover (gradual implementation), or parallel running (running both systems simultaneously for a period). The choice of changeover method depends on factors like risk tolerance, system complexity, and user readiness. During this phase, close monitoring is essential to address any issues that arise and ensure a smooth transition to the new system.

2.3.5. System Security and Support

For information systems to be maintained and protected, system support and security management is essential. Backup and recovery procedures, maintenance management, system performance, security, and user assistance are important elements. It also addresses a system's end of life and foresees upcoming IT possibilities and challenges.

2.3.5.1. User Support

This include answering user questions, updating documentation, providing help desk services, and training users. System usability and user satisfaction depend on efficient user support. Training through manuals, films, or in-person instruction is necessary for both passengers and system operators. While operators should have specialized internal support, a strong help desk guarantees that travelers can resolve ticketing difficulties across several platforms.

2.3.5.2 Maintenance Management

For effective ticketing operations, frequent software updates and hardware checks are essential. Maintaining hardware, such as scanners, lowers downtime and increases reliability, while scheduled software upgrades fix bugs, enhance features, and guarantee security. Effective maintenance management increases safety, boosts productivity, and reduces operating expenses.

2.3.5.3. System Performance Management

This involves proactive management strategies including resource allocation, bottleneck identification, and issue monitoring to maximize performance. While performance and workload data enable real-time modifications, automatic monitoring and fault management guarantee timely issue resolution. Planning for capacity gets the system ready for periods of high demand.

2.3.5.4. System Security

Data protection and defense against cyberattacks are two aspects of system security. Regular code reviews with SaaS/PaaS providers, employee training, encryption, and IP filtering are countermeasures against common dangers including malware, DoS attacks, and direct access.

2.3.5.5. Backup and Recovery

This entails making and keeping backups of important data in case of calamities or data loss. Full, differential, and transaction log backups are among the backup types, and the data is kept in several safe places.

CHAPTER 3: USERS UNDERSTANDING

3.1. Market Analysis

3.1.1. FUTA - *Phuong Trang*

User Flows

Flow 1: Log in

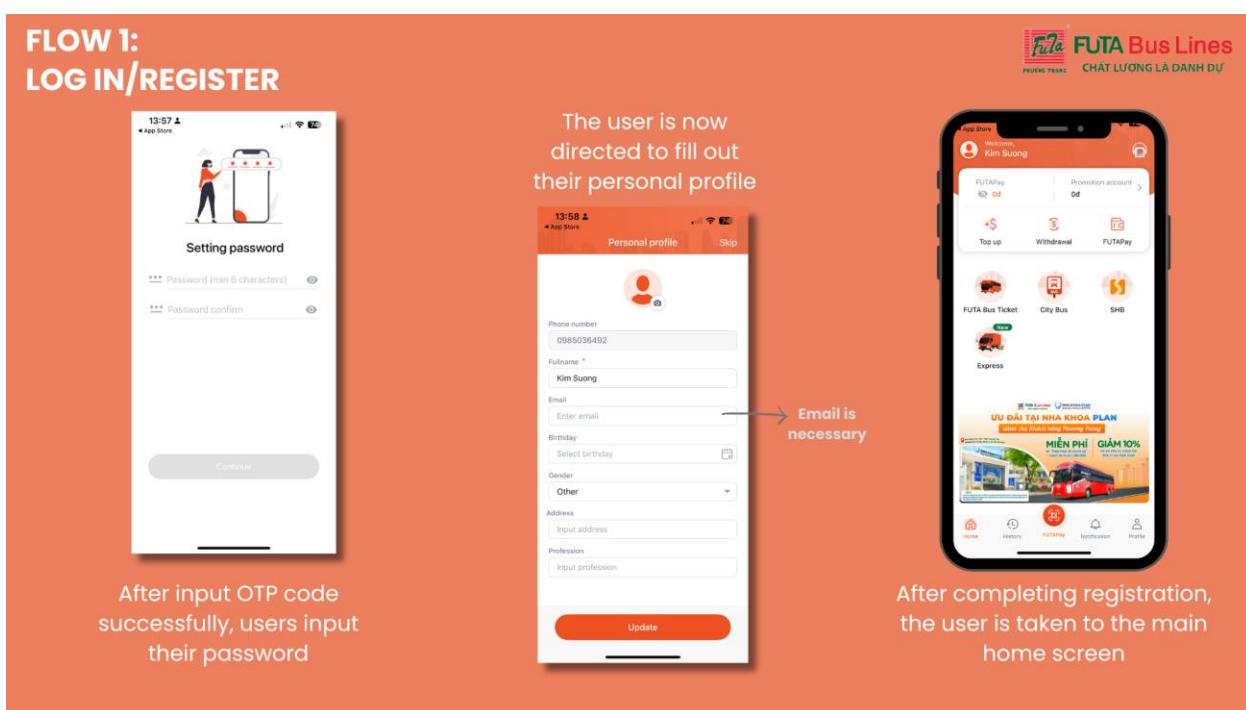
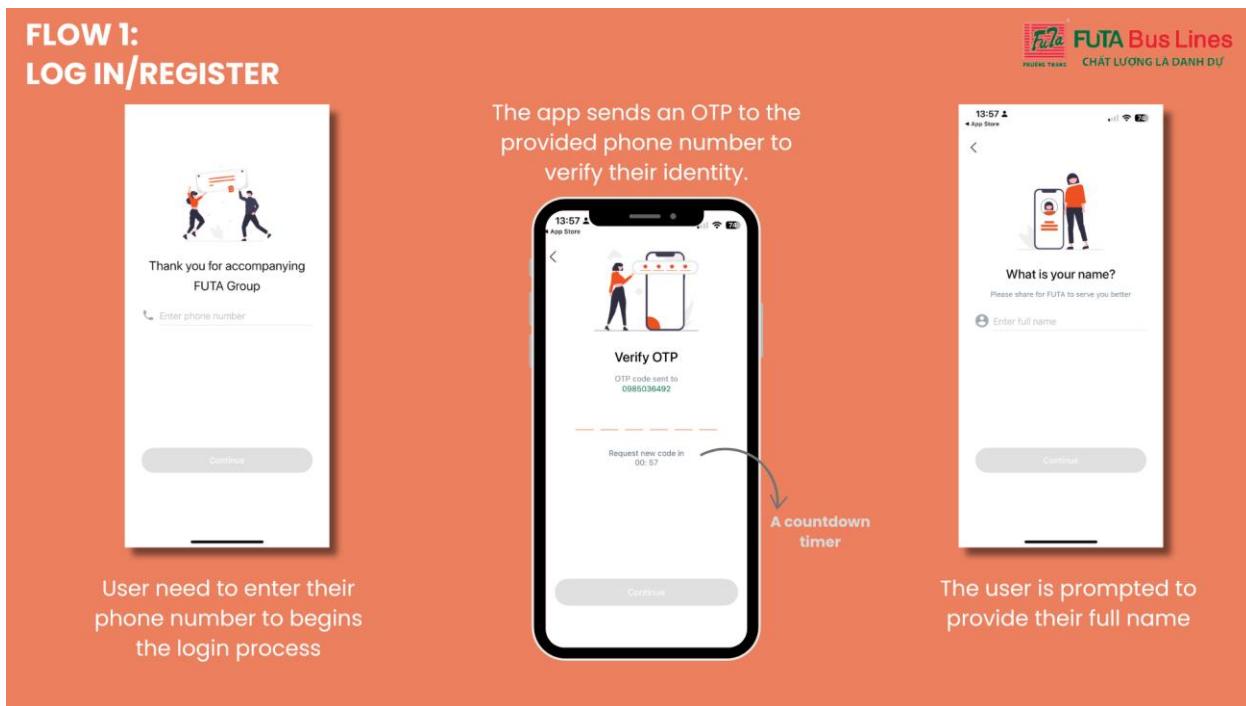


Figure 3.1. Log In (FUTA)

(Source: Authors' source)

It starts with the user entering their phone number to initiate the login process. The app then sends an OTP to verify the user's identity, featuring a countdown timer. After verification, the user is prompted to provide their full name. Following this, the user sets their password and fills out their personal profile, including an email address, which is mandatory. Once registration is complete, the user is redirected to the main home screen of the app, showcasing various features and options. The flow is designed to ensure a smooth and secure onboarding process.

Flow 2: Search for travel trip

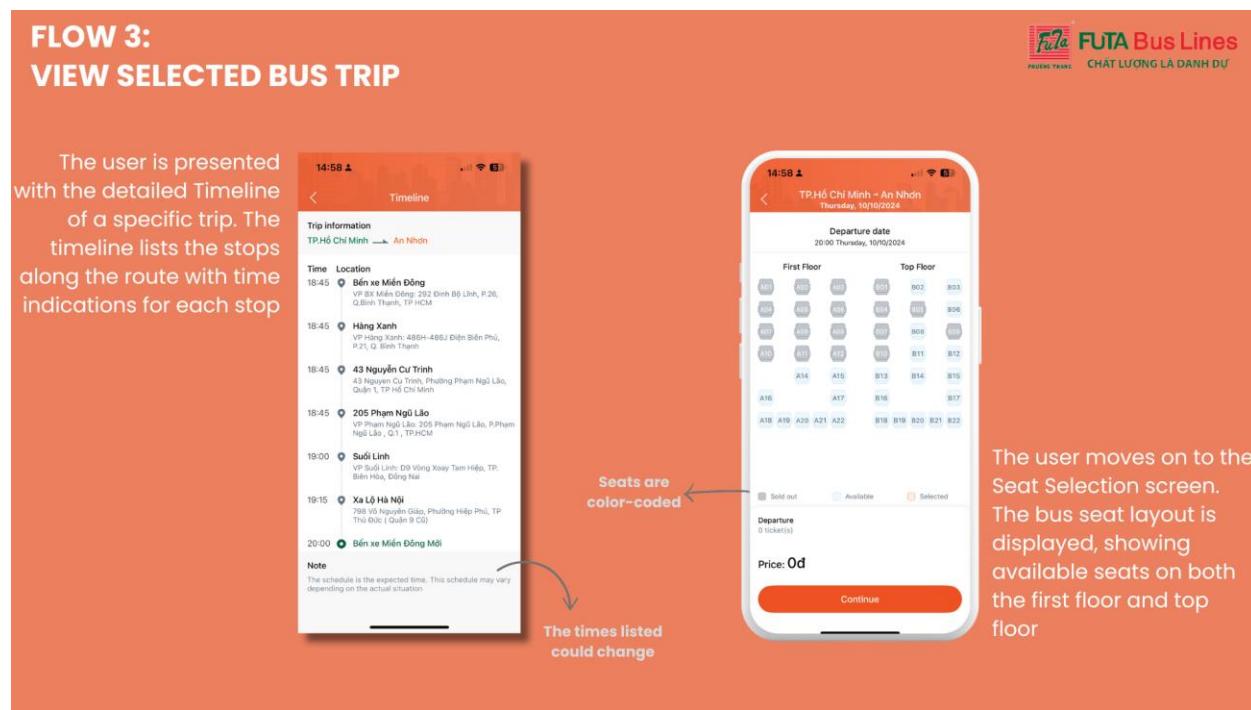


Figure 3.2. Search for travel trip (FUTA)

(Source: Authors' source)

The user begins by viewing a detailed timeline of the selected bus trip, which lists all the stops along the route with specific time indications for each stop. The timeline provides clear, color-coded information to enhance usability. Next, the user proceeds to the seat selection screen, where the bus seat layout is displayed, showing available seats on both

the first floor and the top floor. The interface allows users to choose their preferred seat with real-time price updates before continuing the booking process.

Flow 3: View selected bus trip

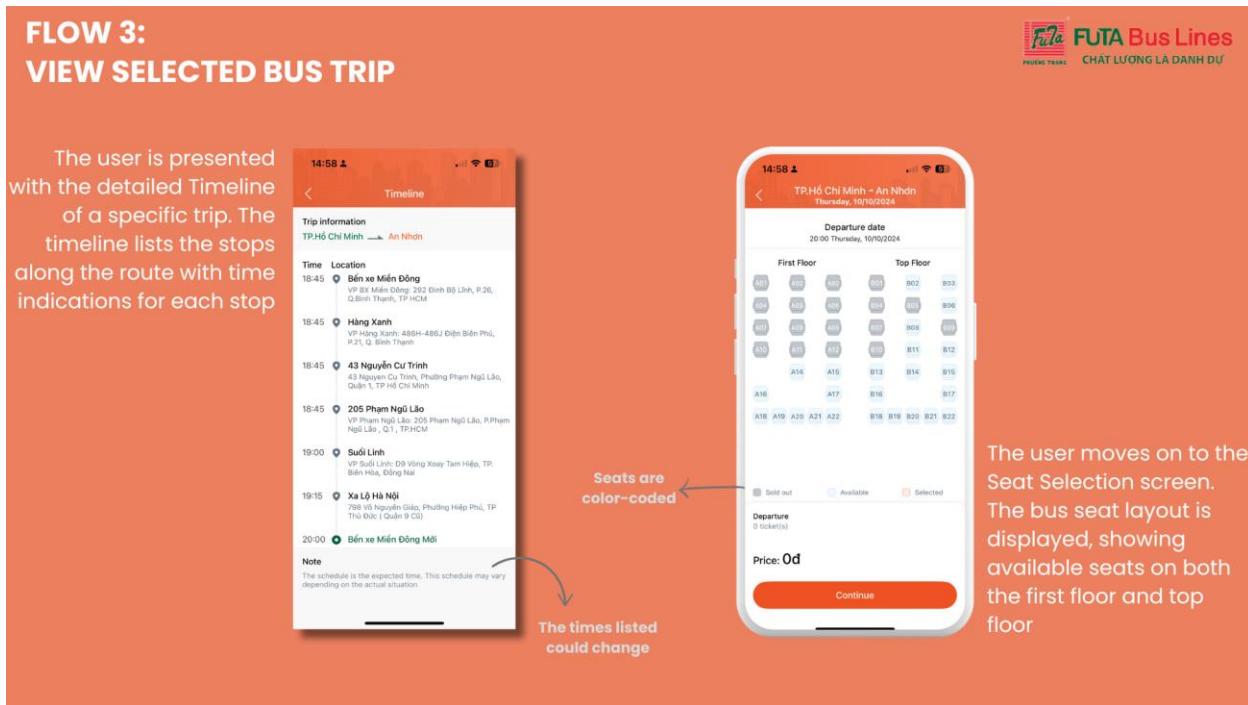


Figure 3.3. View selected bus trip (FUTA)

(Source: Authors' source)

Initially, the user is presented with a detailed timeline of the trip, which lists all stops along the route along with their respective time schedules. These times may change, and the timeline provides a clear, color-coded design to enhance understanding. Following this, the user navigates to the seat selection screen, where the layout of the bus is displayed, including available seats on both the first and second floors. This step allows the user to select their preferred seat before proceeding with the booking process.

Flow 4: Input booking information

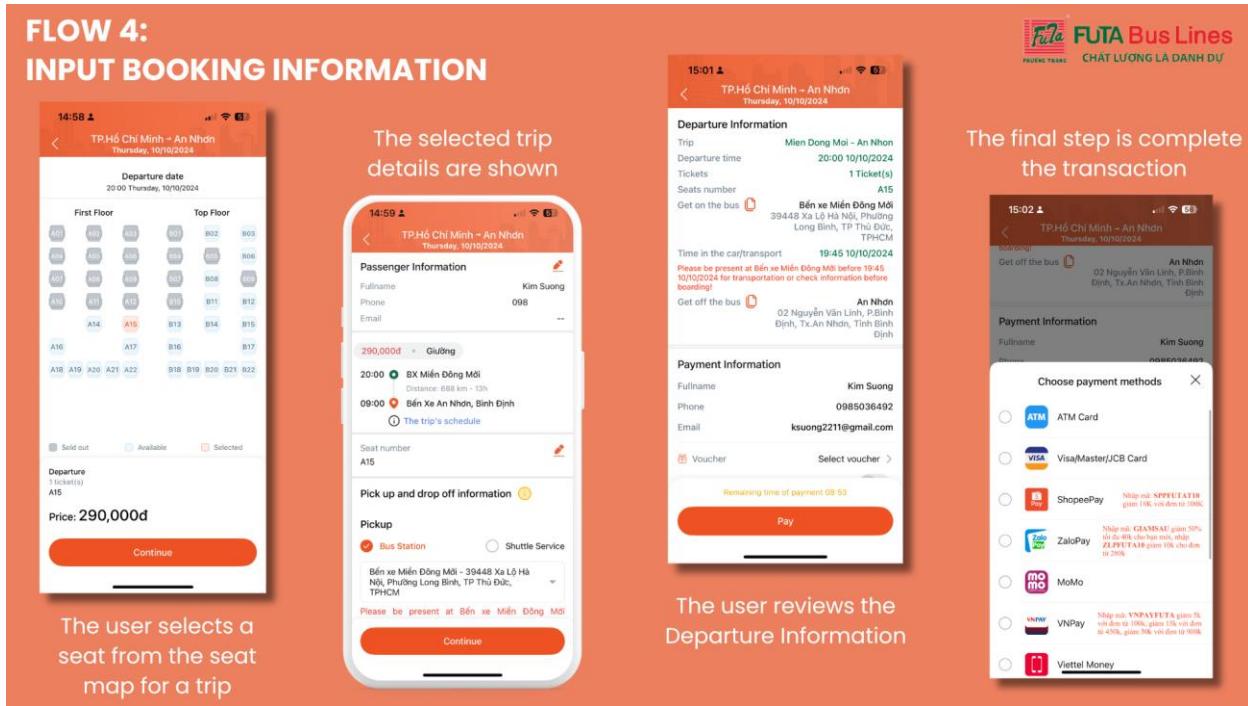


Figure 3.4. Input booking information (FUTA)

(Source: Authors' source)

The process begins with the user selecting a seat from the bus's seat map for their chosen trip. Next, the app displays the trip details and prompts the user to fill in passenger information, including pick-up and drop-off points. The user then reviews the departure details, ensuring the information is correct before proceeding. Finally, the payment screen offers multiple methods for completing the transaction, allowing users to finalize their booking securely and efficiently.

3.1.2. Redbus

User Flow

Flow 1: Log in

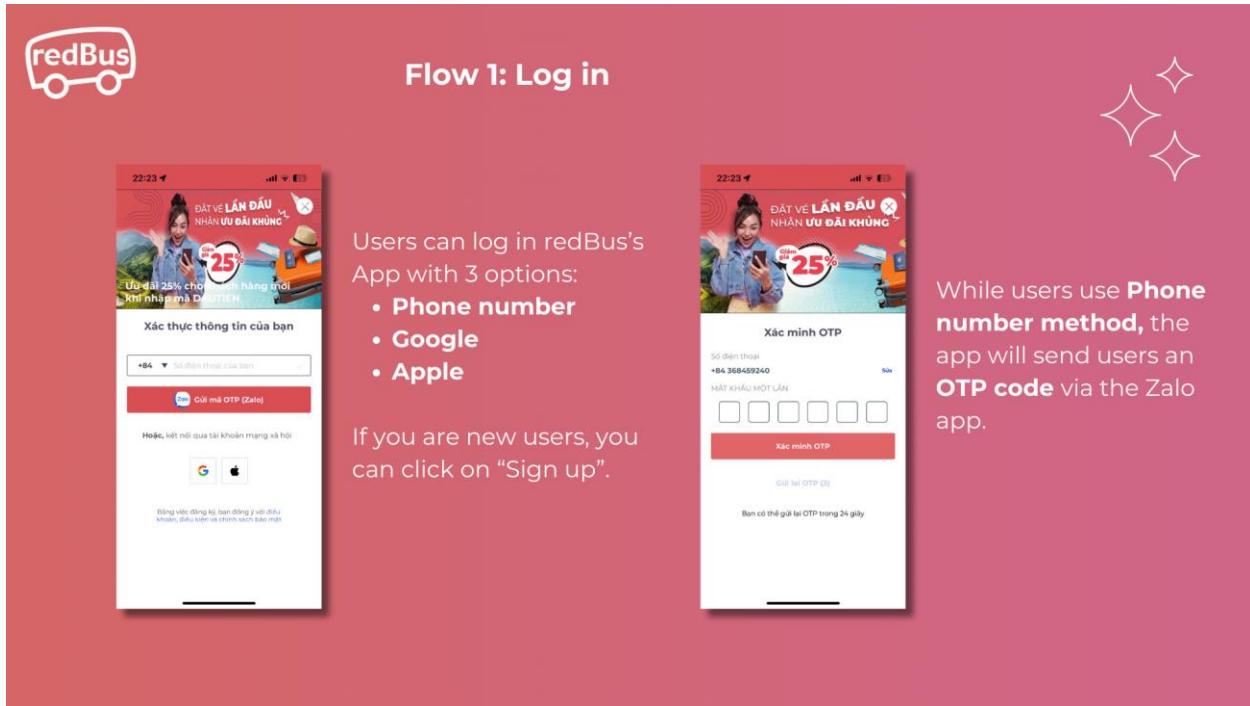


Figure 3.5. Log in

(Sources: Authors' Source)

Users can log in using three options: phone number, Google, or Apple. New users can click on the "Sign up" button to create an account. When using the phone number method, the app sends an OTP (One-Time Password) via the Zalo app to verify the user's identity, ensuring a secure and straightforward login experience.

Flow 2: Search for travel trips

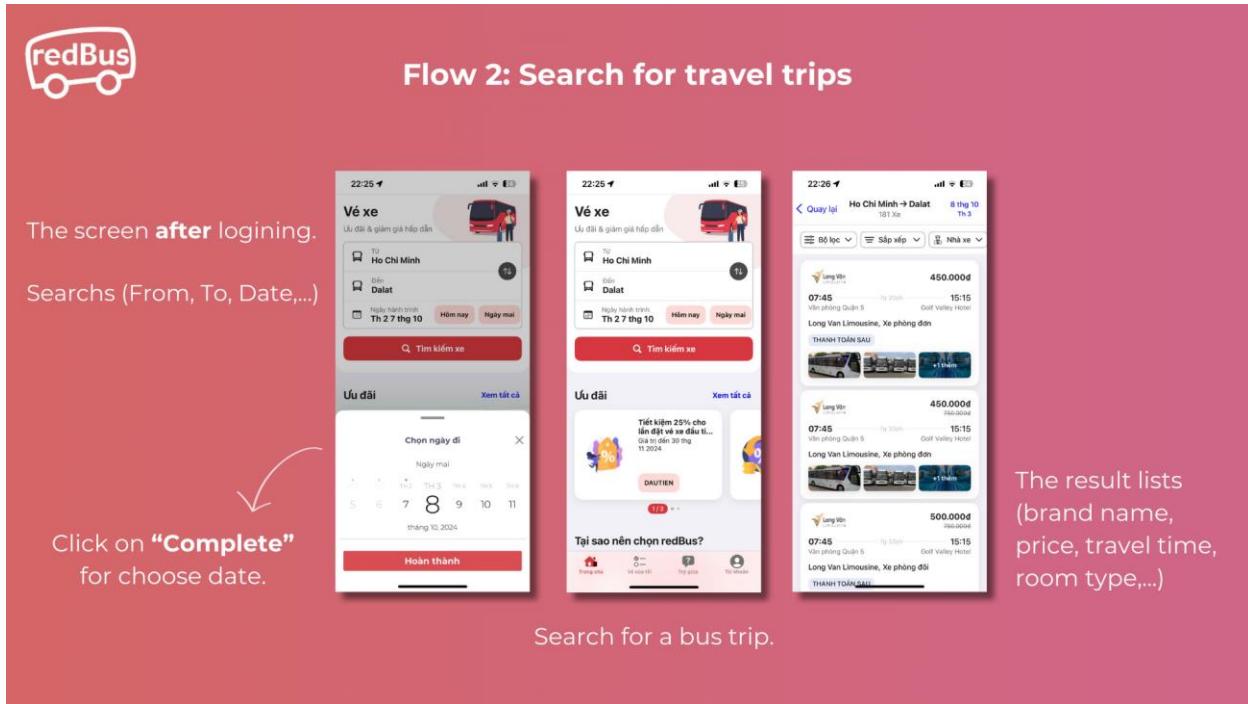


Figure 3.6. Search for travel trips

(Sources: Authors' Source)

After logging in, users can search for bus trips by entering details such as departure and destination locations, and travel dates. A "Complete" button confirms the selected date. The app displays search results that include key details such as bus brand, ticket price, travel time, and room type, allowing users to compare options efficiently.

Flow 3: View the selected trips

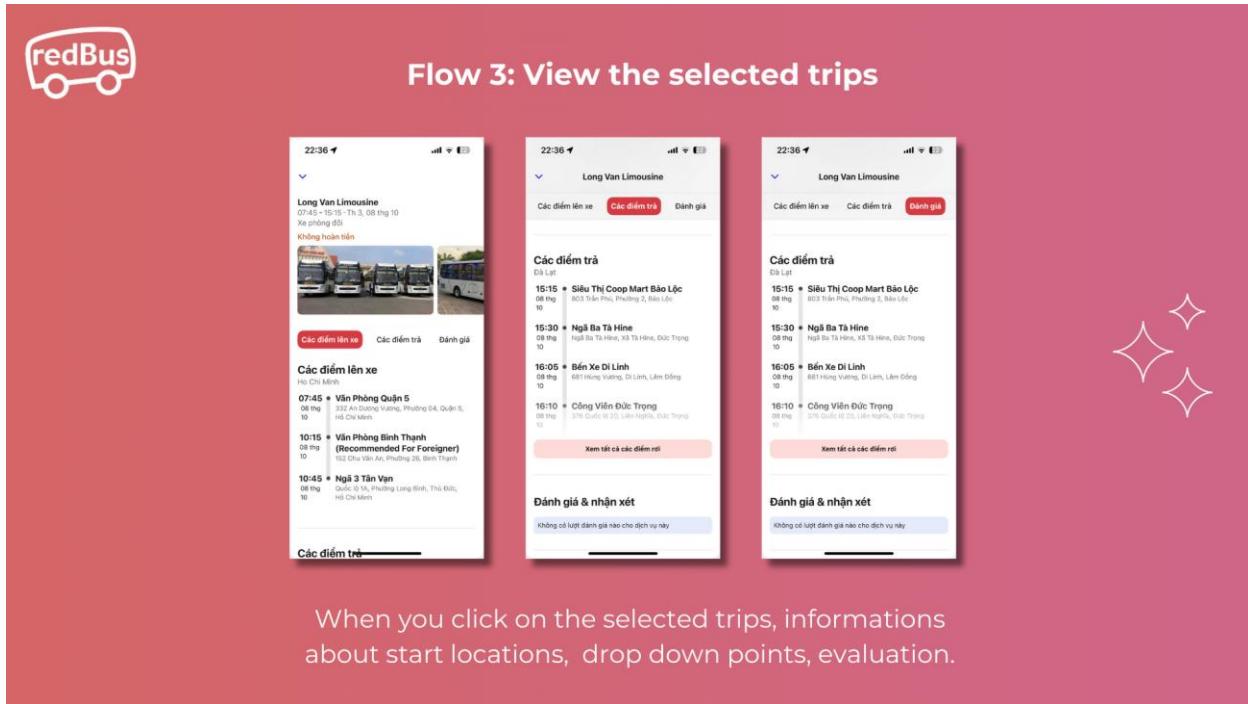


Figure 3.7. View the selected trips

(Sources: Authors' Source)

When users select a specific trip, they are provided with comprehensive information, including starting locations, drop-off points, and additional details like reviews and ratings. This helps users make informed decisions about their travel plans.

Flow 4: Input personal information for booking

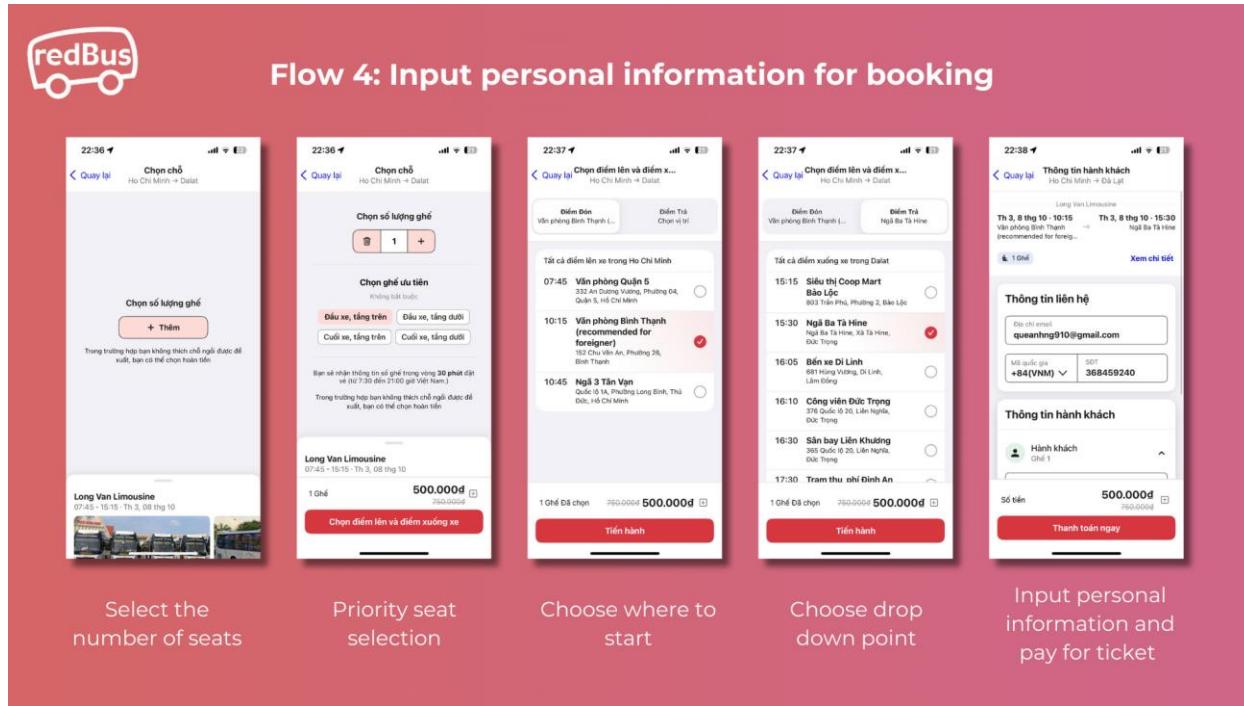


Figure 3.8. Input personal information for booking

(Sources: Authors' Source)

This flow guides the user through the steps to finalize a bus booking. It begins with selecting the number of seats, followed by priority seat selection. The user then chooses the starting location and the drop-off point. Finally, they input personal details, such as contact information, and proceed to payment to complete the booking.

3.1.3. Vexere

User Flow

Flow 1: Log in

VEXERE



Flow 1: Log in

Users can log in Vexere's App with 3 options:

- Phone number
- Facebook
- Google

If they are new users, they can click on "Sign up"

When users use **Phone Number method**, the app will send users an **OTP code**

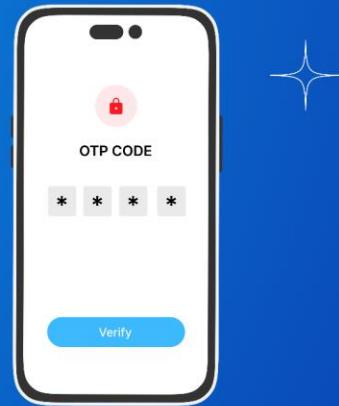


Figure 3.9. Log in

(Sources: Authors' Source)

In the VeXeRe app, users can log in using one of three options: phone number, Facebook, or Google. New users can sign up if needed. When using the phone number method, the app sends an OTP (One-Time Password) to the user for identity verification, ensuring

Flow 2: Search for travel trips

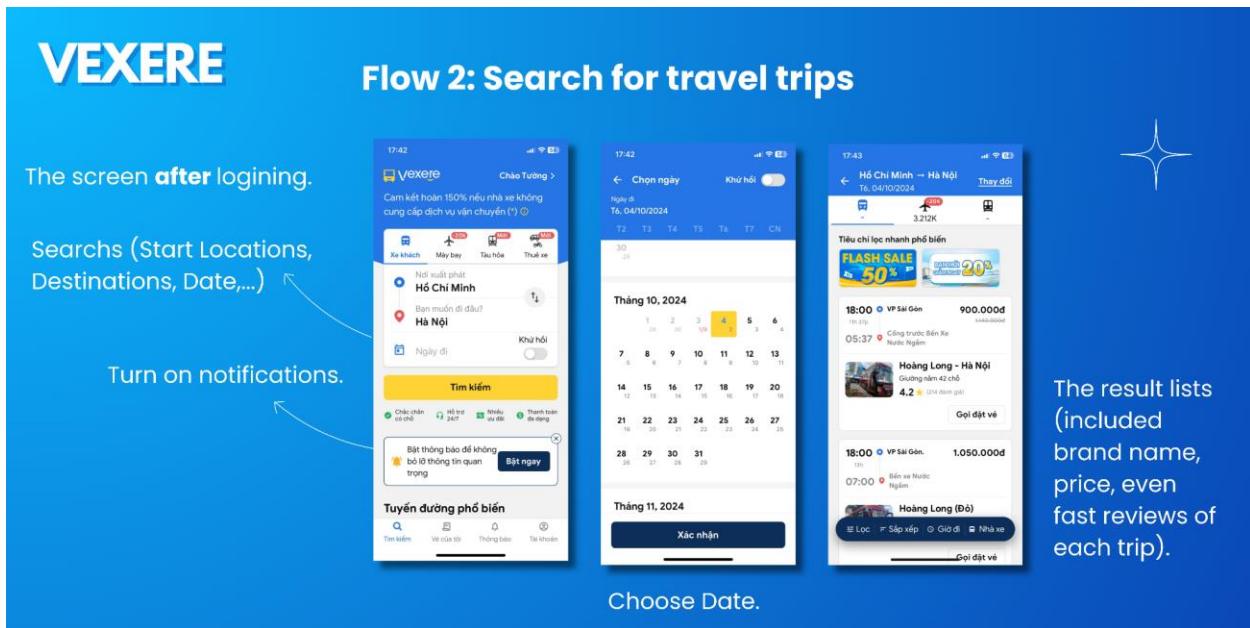


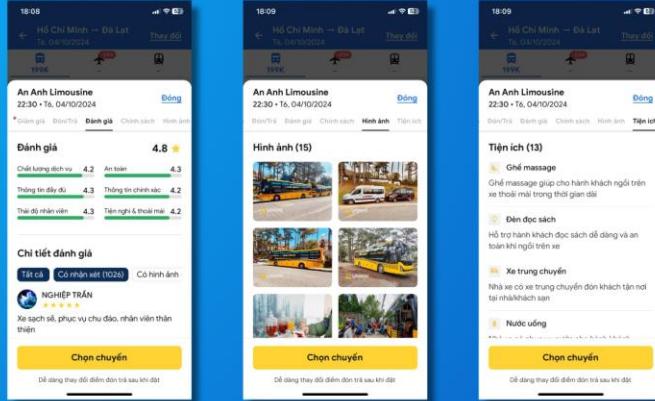
Figure 3.10. Search for travel trips

(Sources: Authors' Source)

After logging in, users can search for bus trips by entering details such as start locations, destinations, and dates. The interface includes a calendar for selecting travel dates, and notifications can be enabled for updates. The search results display comprehensive trip information, including brand names, prices, and quick reviews for easy comparison.

Flow 3: View the selected trips

Flow 3: View the selected trips



When you click on the selected trips, informations about rating, reviews, policy, images about this trip will be appeared.

Figure 3.11. View the selected trips

(Sources: Authors' Source)

Upon selecting a specific trip, the app provides detailed information about the journey. This includes ratings, reviews, policies, and images related to the trip. These details help users evaluate and choose the most suitable travel option, ensuring a well-informed decision-making process.

Flow 4: Input personal information for booking

Flow 4: Input person information for booking

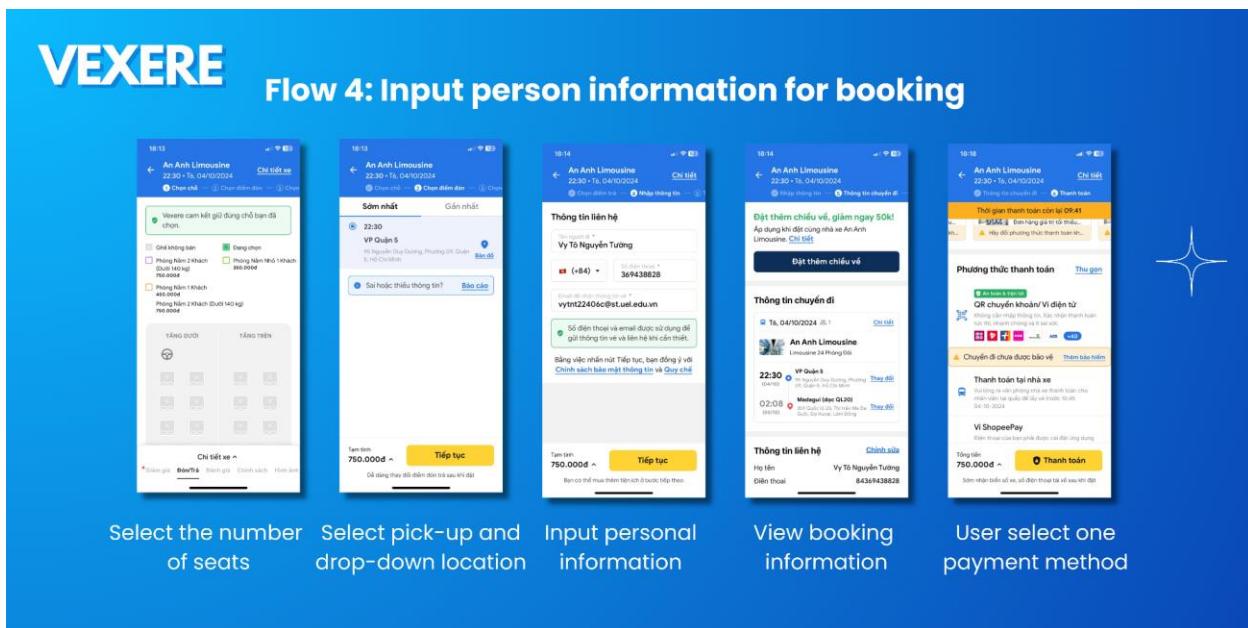


Figure 3.12. Input personal information for booking

(Sources: Authors' Source)

The process begins with the user selecting the number of seats needed for the trip. Next, they choose the pick-up and drop-down locations from available options. The user then inputs personal information, such as contact details, to proceed. After reviewing the booking summary, including trip details and costs, the user selects a payment method to finalize the booking. This flow ensures a straightforward and organized process for completing a bus trip reservation.

CHAPTER 4: OVERVIEW ANALYSIS

4.1. Business Requirements

4.1.1. *Business Objectives*

Enhance User Experience: Provide an easy-to-use platform where passengers can quickly book tickets and access route information.

Increase Sales Efficiency: Automate the bus ticket booking process to reduce manual work for bus companies, allowing them to process bookings more efficiently.

Provide Secure Transactions: Ensure all payment transactions are processed securely, offering a variety of payment options such as credit cards, e-wallets, and bank transfers.

Improve Customer Satisfaction: Offer features like real-time updates on seat availability, ticket confirmation, bus schedules, and the ability to track the bus journey.

4.1.2. *Functional Requirements*

User Login and Registration: Users must be able to create new accounts, log in, and log out securely using email or social media credentials.

Route Search: Users should be able to search for buses by entering departure and arrival locations, travel dates, and filters (e.g., price, bus operator).

Booking Process: Users must be able to select their seats, input personal details, and confirm their bookings.

Payment Integration: Users must be able to pay through **various payment gateways, including credit cards and e-wallets.**

Ticket Management: After booking, users should be able to view their booked tickets, check trip details, and receive updates.

Admin Dashboard: Bus operators need access to an admin panel where they can manage routes, schedules, bookings, and payments.

4.1.3. Non - functional Requirements

Security: Implement SSL encryption for all data transmissions, secure storage for user data, and multi-factor authentication for user accounts.

Scalability: The system must be able to handle large volumes of users, particularly during peak travel seasons.

Performance: The app must load within 2 seconds and respond to user actions promptly. The booking process should take no more than 5 minutes.

Usability: The app must have an intuitive and user-friendly interface.

Compliance: The app must comply with local regulations, including data privacy laws like GDPR, and industry standards for payment processing.

4.2. Fact-finding

Interview Plan for End-Users (Passengers) for Developing a Mobile Bus Booking Application.

Step 1: Determine the People to Interview

Identify Key End-User Profiles:

- Frequent Commuters: Those who use buses daily or weekly for work or study.

- Occasional Travelers: Passengers who use the bus system for intercity travel or occasional trips.
- Senior Users and Students: Specific demographics that may have unique needs or preferences.

Choose Interview Format:

- Individual Interviews: To allow for in-depth, personal insights from each user.
- Group Interviews: Optional for gathering collective feedback and observing group discussions about common challenges or expectations.

Step 2: Establish Objectives for the Interview

Define Key Areas to Discuss:

- Understand common difficulties with the current booking system and what features they expect from a new app.
- Gather ideas on specific functionalities, like seat selection, real-time notifications, and payment preferences.
- Learn how users navigate current mobile or online platforms, focusing on ease of use and accessibility.

Facts to Gather:

- Big Picture: How users generally perceive the bus booking process.
- Specifics: Details like preferred booking methods, critical features, and expectations for user experience.

Step 3: Develop Interview Questions

Table 4.1. Question Types

Question Type	Question	Response Options/Notes
Open-Ended Question	<i>What challenges do you face when booking a bus ticket?</i>	<i>Open response</i>
Open-Ended Question	<i>What do you like or dislike about the current booking system?</i>	<i>Open response</i>
Open-Ended Question	<i>If you could add one feature to a bus booking app, what would it be?</i>	<i>Open response</i>
Close-Ended Question	<i>Do you often book your ticket before arriving at the bus station?</i>	<i>[Yes/No]</i>
Close-Ended Question	<i>Do you need a reminder feature for your departure time?</i>	<i>[Yes/No]</i>
Range of Response Question	<i>On a scale of 1 to 10, how convenient do you find the estimated arrival time notification feature?</i>	<i>[1-10]</i>
Close-Ended Question	<i>What is your preferred payment method when booking a ticket?</i>	<i>[Credit Card/E-Wallet/Cash/Other]</i>
Range of Response Question	<i>How well does the current app meet your booking needs?</i>	<i>[Not Satisfied, Average, Very Satisfied]</i>
Close-Ended Question	<i>Do you often find it challenging to search for bus schedules?</i>	<i>[Yes/No]</i>
Open-Ended Question	<i>What experience do you expect from a bus booking app?</i>	<i>Open response</i>

Range of Response Questions (for feedback with a measurable scale):

- *"On a scale of 1 to 10, how important is the option to choose your seat during booking?"*
- *"How would you rate the current bus booking experience: low, medium, or high?"*

Question Structure:

Funnel Approach: Start with open-ended questions to get broad insights into user expectations and challenges, then move to close-ended questions for specific details.

Step 4: Prepare for the Interview

Preparation Essentials:

- Confirm Interview Details: Schedule a time, set a place (online or in-person), and limit each interview to around one hour. Confirm all details via email.
- Gather Required Materials: Ask users to recall specific experiences or challenges with bus booking systems, so they come prepared to share concrete examples.

Step 5: Conduct the Interview

Structure and Execution:

- Introduction: Begin by introducing, briefly describing the project, and explaining the interview objectives.
- Engaged Listening: Actively listen, ask follow-up questions, and give users time to respond fully to each question.
- Summarize and Confirm: Conclude the interview by summarizing the main points discussed, seeking confirmation for accuracy, and asking if there's anything else the interviewee would like to add.

Step 6. Document the interview

Write down or record the main insights, ideas, and pain points shared by the user. Focus on responses that highlight user needs, feature preferences, and common issues with the current booking process. After conducting the interview, record the information quickly.

4.3. Overall Flow

BPMN include: application, search, seat selection, payment, ticket confirmation, check-in, security and safety checks, customer support.

Customer Accesses the Application

- The customer opens the bus ticket booking app or website.
- They can either register for an account or log in if they already have one.

Search and Select Bus Route

- The customer inputs the departure location, destination, and preferred date/time of travel.
- The system retrieves available bus options based on the search criteria.
- The customer selects a suitable route and can view details such as bus information, timing, fare, and available seats.

Seat selection

- After choosing a bus route, the customer is presented with a seating chart.
- The customer selects their desired seat.

Payment

- The customer reviews their booking details (route, seat, fare).
- They choose a payment method (credit card, e-wallet, bank transfer, etc.).
- The system processes the payment and confirms it.

Ticket confirmation

- After successful payment, the system sends an e-ticket to the customer via the app or email.
- The ticket includes trip details, seat assignment, and a ticket QR code.

Check-in at Departure Point

- The customer can check in through the app before arrival.
- At the bus terminal, the customer scans their ticket QR code or shows the e-ticket to the staff for check-in.
- After verification, the customer boards the bus and sits in the reserved seat.

Security and Safety Checks

- If enhanced security is in place, the customer may need to pass through a security check.
- This could involve weighing luggage on a smart scale or scanning the ticket through a security system.

Journey and Real-time Updates

- During the journey, the customer may receive notifications about the trip, stops, or any changes through the app.
- The route management system provides real-time updates to the driver, including traffic conditions.

Trip Completion

- Once the journey is completed, the customer is notified of the trip's conclusion.
- The customer can provide feedback or rate the service directly within the app.

History and Customer Support

- The customer can view their booking history and past trips.
- They may use any rewards points for future bookings if the system supports loyalty programs.
- Customer support is available through various channels (live chat, hotline, etc.).

4.4. Cost-benefit Analysis

Step 1: Identifying Costs

Fixed Costs:

- **System Development Cost:** This includes costs for hiring developers, analysts, and UI/UX designers.
- **Software and Technology Cost:** The cost of purchasing licenses for development tools, database management software (e.g., MongoDB).
- **Server and Data Storage Cost:** The cost of setting up server infrastructure for running the application.

Maintenance Costs:

- **System Maintenance Cost:** Costs related to updating the system, bug fixes, and software upgrades.
- **Database Management Cost:** Ongoing expenses for maintaining and securing data on servers.
- **Operational Staff Cost:** Salaries for technical support staff, customer service, and day-to-day operations management.

Step 2: Identifying Benefits

Measurable Benefits:

- **Increased Revenue:** The potential to attract more users, which will lead to increased revenue from online ticket sales.
- **Reduced Operational Costs:** Automating the booking and payment processes reduces the need for hiring additional staff.
- **Improved Management Efficiency:** Automating ticket and customer data management helps save time and effort, making the process more efficient.

Intangible Benefits:

- **Better Customer Experience:** Users can book tickets quickly and easily, with instant notifications about their trips.
- **Enhanced Transparency:** Information about ticket prices, seat availability, and processes is more transparent, allowing customers to make informed decisions.
- **Increased Brand Credibility:** Providing a convenient and professional online booking service improves the company's image and credibility.

Step 3: Assign Monetary Values

Costs:

1. **Fixed Costs (Development Costs):**

- **System Development Cost:** \$2,500 (includes hiring developers and UI/UX designers for a small-scale system).
- **Software and Technology Cost:** \$500 (licenses for tools like MongoDB and other technologies).
- **Server and Data Storage Cost:** \$1,000 (cost for setting up basic cloud hosting infrastructure, such as AWS or Firebase).
- **Total Fixed Costs: \$4,000.**

2. **Maintenance Costs (Yearly Costs):**

- **System Maintenance Cost:** \$300/year (system updates and bug fixes).

- **Database Management Cost:** \$200/year (ensuring data security and availability).
- **Operational Staff Cost:** \$500/year (customer support and handling operational issues).
- **Total Maintenance Costs: \$1,000/year.**

Benefits:

1. Measurable Benefits:

- **Increased Revenue:** \$3,000/year (from online ticket sales).
- **Reduced Operational Costs:** \$1,000/year (less dependency on manual ticketing staff).
- **Improved Management Efficiency:** Saves \$500/year by automating tasks like ticket and seat availability management.

2. Intangible Benefits:

- **Better Customer Experience:** Customers can book tickets faster with instant confirmation.
- **Enhanced Transparency:** Easier access to pricing and seat availability.
- **Increased Brand Credibility:** A professional system improves trust and reliability.

Step 4: Calculate Net Benefit

Net Benefit Formula:

Net Benefit = Total Benefits - Total Costs

1. First Year:

- **Total Costs:**
 - Fixed Costs: \$4,000.
 - Maintenance Costs: \$1,000.

- **Total First-Year Costs: \$5,000.**
- **Total Benefits:**
 - Increased Revenue: \$3,000.
 - Reduced Costs: \$1,000.
 - Improved Efficiency: \$500.
 - **Total First-Year Benefits: \$4,500.**
- **Net Benefit (First Year):**

$$\$4,500 - \$5,000 = -\$500$$
 (slight loss due to upfront development costs).

2. Subsequent Years:

- **Total Costs:** Maintenance Costs only: **\$1,000/year**.
- **Total Benefits:** **\$4,500/year**.
- **Net Benefit (Subsequent Years):**

$$\$4,500 - \$1,000 = \$3,500/year$$
 (positive impact starting from year 2)

Step 5: Assess Risks

1. Potential Risks:

- **Technical Failures:** Bugs or crashes could disrupt the service.
- **Budget Overruns:** Unforeseen costs during development.
- **Customer Adoption:** Users may not be comfortable with the online system initially.
- **Competitor Advantages:** Larger companies offering similar systems with more features.

2. Mitigation Strategies:

- **Regular Testing:** Implement thorough testing during and after development.
- **Budget Management:** Allocate a contingency fund for unexpected expenses.
- **User Training and Support:** Provide clear tutorials and responsive support for new users.

- **Unique Value Proposition:** Offer features like loyalty points or discounts to stand out.

Step 6: Make the Decision

1. Feasibility Assessment:

- The initial investment may result in a slight deficit in the first year (-\$500), but starting from the second year, the system provides a net benefit of \$3,500/year.
- Intangible benefits, such as improved customer satisfaction and brand credibility, also justify the investment.

2. Decision:

- **Proceed with the system implementation.**

Despite the initial development costs, the project is financially viable in the long run and offers sustainable improvements in efficiency and customer experience.

CHAPTER 5: DETAIL ANALYSIS

5.1. User Stories & Current Process Flow

5.1.1. Current Process Flow

5.1.1.1 Direct Booking Process (At Bus Station)

The current bus ticket booking process is mostly manual through steps such as ticket issuance, payment collection, and inventory. This process is relatively suitable for small scale, less dependent on technology, and employees do not require high technical qualifications. However, as the number of passengers increases rapidly as well as the current trend of modernization, this process is no longer suitable. On the contrary, it brings a lot of trouble, takes a lot of time, and is difficult to manage, which is not convenient for customers, leading to many problems.

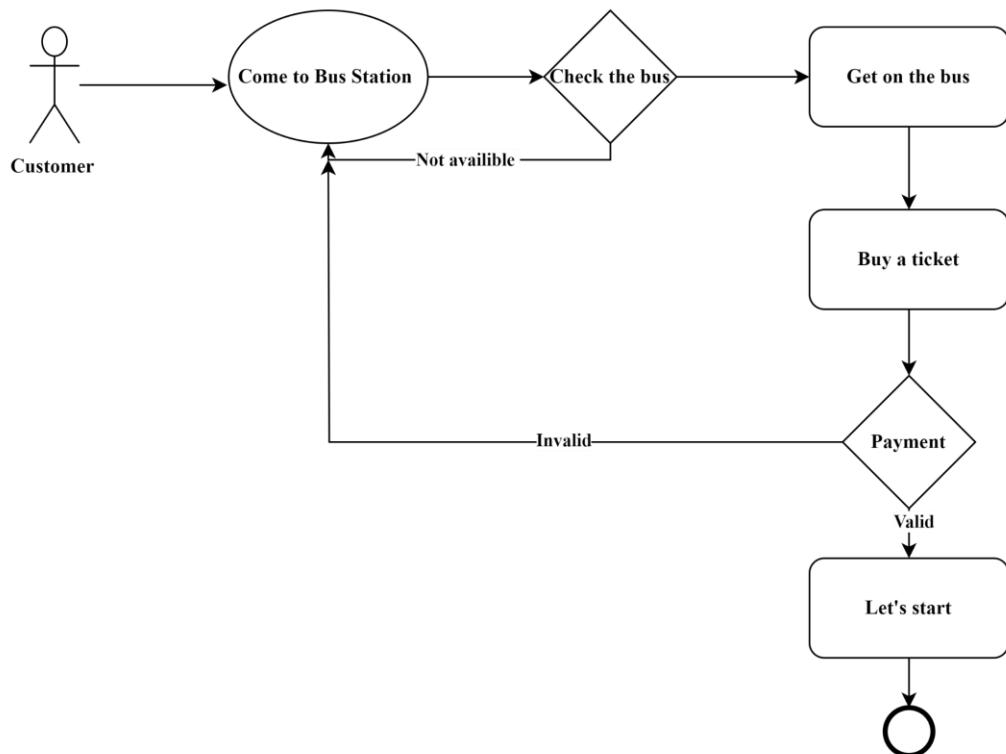


Figure 5.1. Current process flow

(Source: Authors' source)

Table 5.1. Current process flow

Face	Representation
Multiple operations	Many manual operations, dependent on humans, take a lot of time.
Tickets	Printed paper tickets are easily damaged, lack flexibility, and are easily lost.
Lack of technology	The process lacks modern technological integration, hindering efficiency and convenience.

5.1.1.2. Indirect Booking Process (Via Hotline)

It is made very clear that using a hotline to buy tickets entails contacting a specific phone number in order to speak with a customer support agent who helps you obtain the tickets you want for bus routes. Usually, the procedure entails giving pertinent information over the phone, including payment information, seat preferences, and bus travel data. While the hotline booking process has served its purpose, a transition to a digital process is required to meet the growing needs and expectations of today's tech-savvy consumers now.

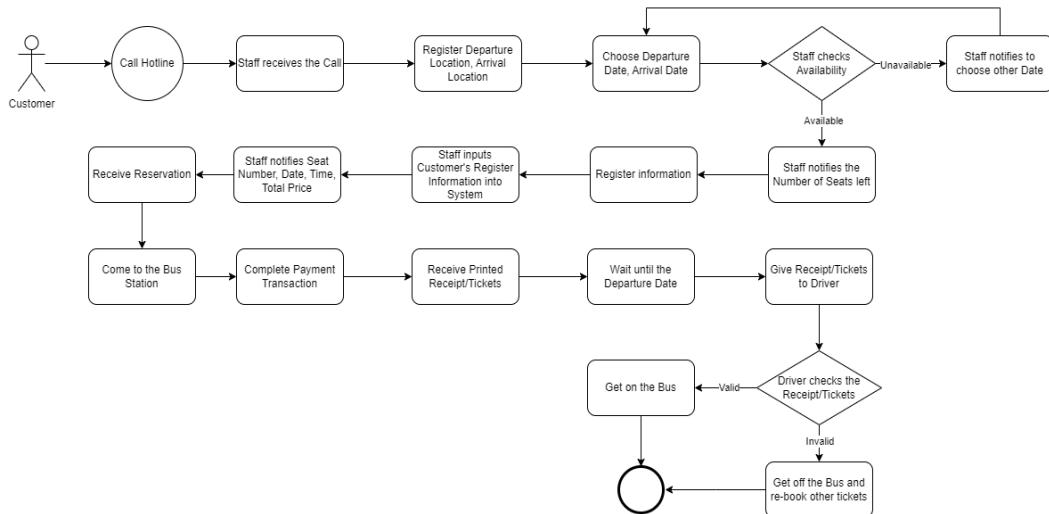


Figure 5.2. Indirect ticket booking process via hotline

(Source: Authors' source)

Table 5.2. Indirect booking process via hotline

Face	Representation
Wasting time	Spend a lot of time doing different operations.
Enter data manually	All operations are done manually and depend entirely on humans, making mistakes easily.
Lack of Information	Lack of information to customers, information is not updated visually and clearly.

To improve the ticket booking experience as well as enhance the customer experience, in this project we create a payment process system that is both convenient, easy to use and meets current needs now.

5.1.2. User Stories

A customer desires an application that allows them to quickly and conveniently purchase bus tickets online, helping them reach their destination smoothly. To provide the best solution for this customer, the app's process flow should focus on simplicity and security.

Below is the proposed process flow:

- First, users quickly register using their phone number, and the app will verify via OTP and request basic information like name and email address.
- Users search for bus trips based on options such as departure and destination points, departure time, etc.
- The app will then display available trips with detailed information, such as departure time, ticket price, bus type, and more.
- Users select their desired seat after reviewing the trip information.
- After checking all details, users choose a payment method and proceed to payment.
- The app sends a booking confirmation and provides ticket information to the user.
- Finally, users can manage purchased tickets and receive notifications, and if there are any questions, they can contact customer support directly within the app.

5.1.3. Future Process Flow

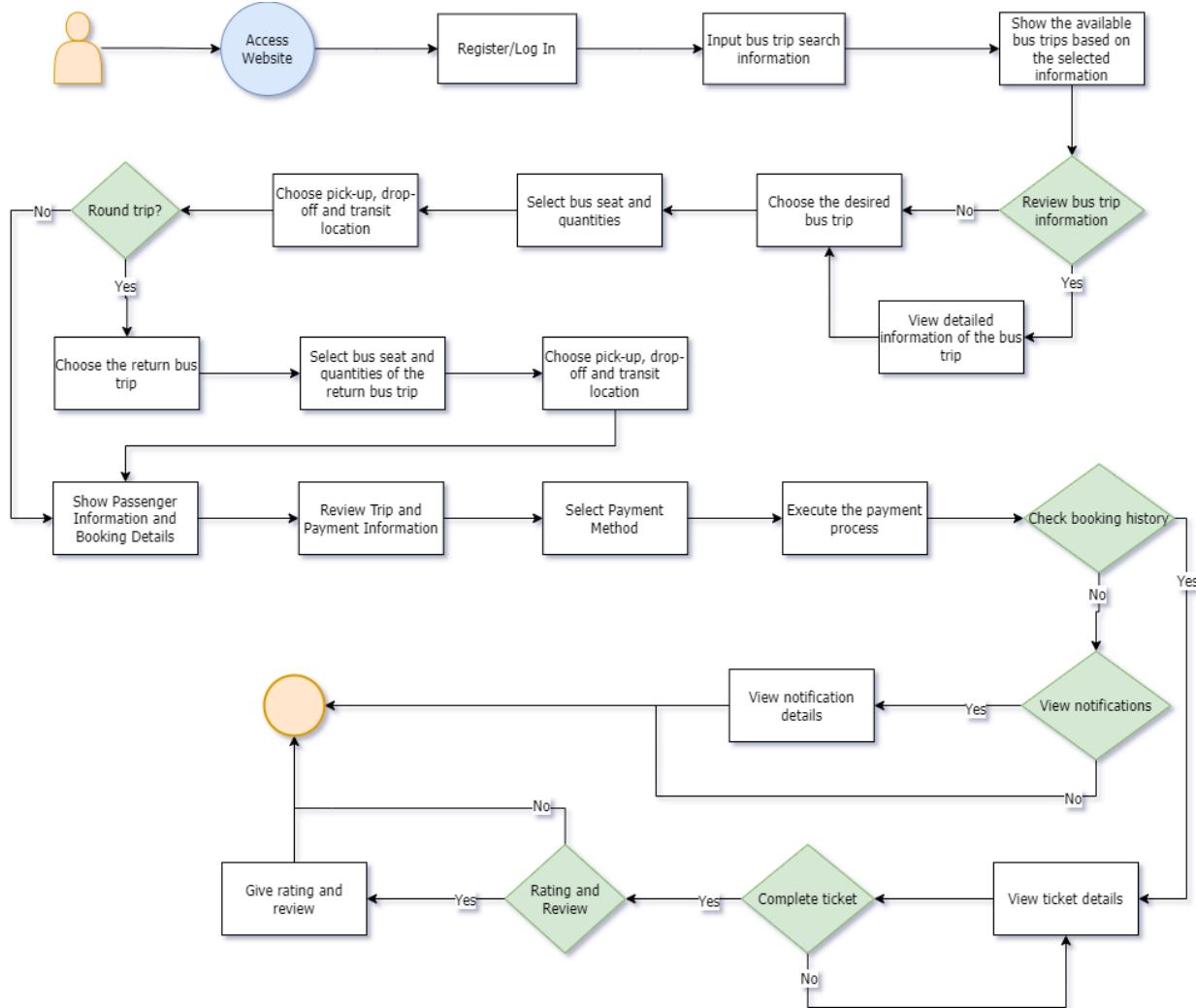


Figure 5.3. Future Process Flow

(Source: Authors' source)

This flowchart outlines the end-to-end process of a bus booking system, starting with the user accessing the website and logging in or registering. Users input trip details, review available bus options, and choose their desired trip, with the option to add a return trip if needed. After selecting pick-up/drop-off locations and bus seats, they provide passenger details, review booking information, and complete the payment. Post-booking, users can

check their booking history, view notifications or ticket details, and provide feedback through ratings and reviews, creating a streamlined and user-friendly journey.

5.2. Main Function

Table 5.3. Main Function

Function	Feature	Description	Input	Output
Account Management	<i>User Registration</i>	<p>New users can sign up for an account using email and password, and provide personal information such as name and phone number.</p> <ul style="list-style-type: none"> • Input registration information: Name, email, password, phone number. • Verify email via OTP code or confirmation link. • Save user information to the database. 	Personal information, Email, Password,..	Success/Failure, User Session Token

	<i>User Login</i>	<p>Registered users can log in to the system using email and password.</p> <ul style="list-style-type: none"> • Input login information: Email and password. • Verify login information with the database. • Create a user session after successful login. 		
	<i>Forgot password</i>	Ensures that users can recover their account if they forget their login credentials, through either a reset link or OTP.	User's email (the email they registered with).	<p>Success message if the email is valid and reset instructions are sent.</p> <p>Failure message if the email is not found in the system.</p>
	<i>Profile Management</i>	Allows users to maintain and update their personal information, change their password or email, and manage preferences like notifications and profile pictures. Both features are essential for ensuring a smooth and secure	User's session or login token (to identify which profile to retrieve).	<p>Display of user's personal information, such as name, email, phone number, and booking history; success/failure message,...</p>

		user experience in the bus booking system.		
Route Search	<i>Search for bus routes</i>	Users can search for available bus routes by entering travel details such as departure and arrival locations.	Departure location, Destination, Date/Time	List of bus routes matching the search criteria
Booking	<i>Booking</i>	Users can choose a round-trip ticket or not, select the departure date, number of tickets, pick-up and drop-off information, and apply a voucher that suitable with user's desire	Round-trip ticket, Departure date, Number of tickets, Pick-up and drop-off information, Voucher.	Booking Confirmation, E-ticket, Trip Information
Seat Selection	<i>Choose Seat</i>	Users can select their preferred seat based on the bus seat map	Trip information, Seat Map	Seat selected, seat status updated
Booking Confirmation	<i>Confirm booking</i>	Users confirm their booking details, including seat and personal information before finalizing the reservation.	Personal information, ticket details	Ticket successfully booked, confirmation sent via SMS/email.
Payment Processing	<i>Payment</i>	Users complete ticket payment via credit card, e-wallet, or bank transfer in a secure environment.	Successful payment, e-ticket and payment receipt	
View Booking history	<i>Booking history</i>	Users view their booking history, including previous trip	User account	Display list of user's booking history.

		details and payment status.		
Cancel booking	<i>Ticket cancellation</i>	Allows users to cancel a previously booked ticket, and the system updates seat availability accordingly.	Ticket information, Cancellation reason	Ticket successfully canceled, cancellation notification
Customer Support	<i>Customer Service</i>	Users can contact support via chat or hotline for issues related to booking or other inquiries.	User support request	Resolution provided, support response
Rating and Feedback	<i>Customer Feedback</i>	After completing a trip, users can rate and give feedback on the service and experience.	Rating, Comments	Feedback stored, updated in bus operator's review profile

5.3. Processes - Business Process Modelling Notation (BPMN)

5.3.1. Deliverables

Table 5.4. Deliverables

Section	Process Name	Description	Purpose
1	Overall Process	The overall process describes the entire flow of the bus booking application, from when the user accesses the app, logs in, searches for a bus, selects a seat, makes a payment, receives the ticket, and checks in.	Provides a comprehensive view of how all key processes are interconnected, helping stakeholders understand the complete system operation from start to finish.
2	Login Process	The login process outlines the steps users take to log into the system. This includes entering login credentials, account authentication, and displaying the login result.	Ensures user security and authentication, prevents unauthorized access, and protects customer personal information.
3	Payment Process	The payment process shows the steps users follow to pay for their bus ticket. It includes selecting a payment method, confirming details, processing the payment, and confirming the transaction.	Ensures a secure, efficient, and seamless payment process, accommodating various payment methods (credit card, e-wallet, bank transfer).
4	Rating Process	The rating process allows users to rate and provide feedback on the service after completing their trip. It includes selecting a star rating, entering feedback, and submitting the review.	Collects user feedback to improve service quality, enhance customer experience, and build customer loyalty through responsive service improvements.

5.3.2. Overall Process

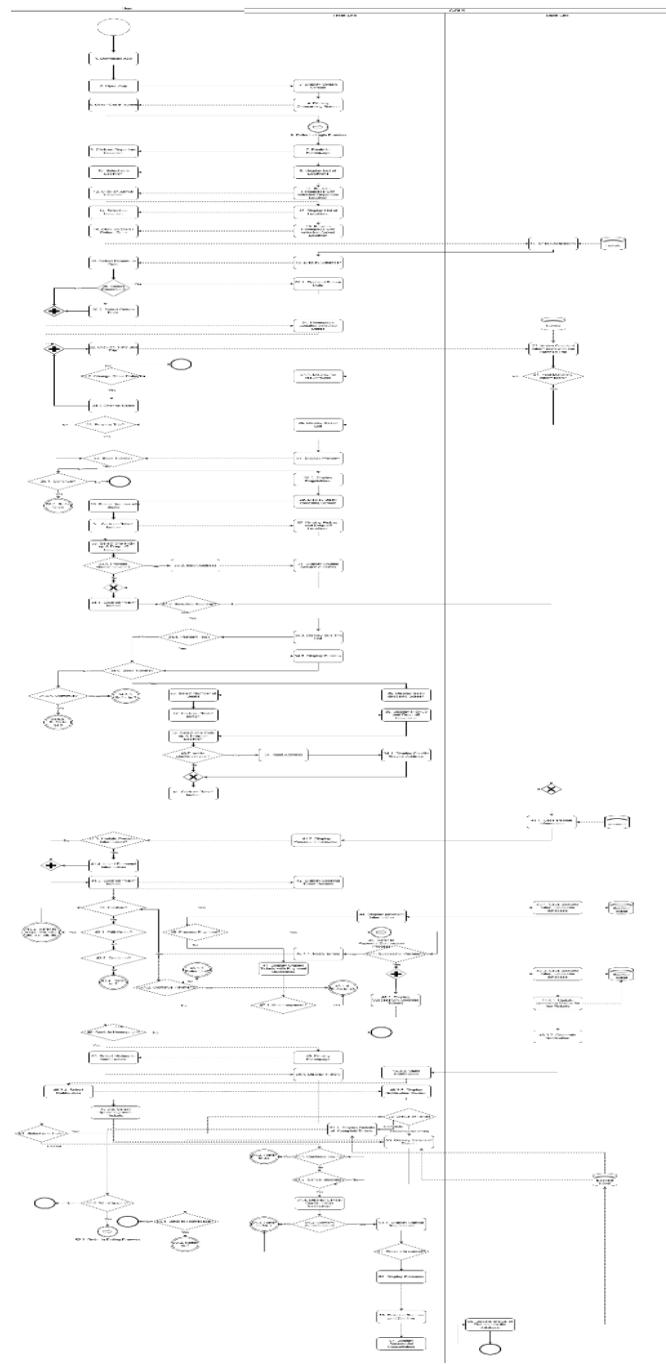


Figure 5.4. Overall BPMN

(Source: Authors' source)

It covers the user journey from opening the app to completing the booking and post-booking activities. The process includes user actions like logging in, selecting routes, choosing seats, inputting passenger information, and making payments. It also outlines system functions such as verifying user details, updating the database, and generating tickets. Additional flows include managing errors, sending notifications, and processing reviews. The detailed swimlanes divide responsibilities among users, the system, and external services, ensuring clarity and process efficiency.

Table 5.5. High Level Requirements of Overall BPMN

ID	Feature	Request Name	Description	Status	Priority (MoSCo w)
BRE-01	Log In	Input Phone Number / Email	<p>As a customer, I want the website to display a Log In Screen where I can sign in by either my Phone Number or Email, allowing me to begin my ticket-booking process efficiently.</p> <p>The Log In Screen with Phone Number/Email will include:</p> <ul style="list-style-type: none"> • Log In Field with Phone Number/Email • Sign Up button 	Approved	Must Have
BRE-02		Input Password	As a customer, I want the website to display the Password Field after entering my Phone Number or Email, so that I can log in to my account successfully.	Approved	Must Have

			<p>The Input Password will include:</p> <ul style="list-style-type: none"> • Password Field • Forgot Password button 		
BRE-03	Sign Up	Sign Up	<p>As a customer, I want the website to provide a sign-up option, allowing me to create a new user account.</p> <p>The Sign Up will include:</p> <ul style="list-style-type: none"> • Sign Up Field with with Phone Number/Email • OTP/Verification Code Field • Input Password Field 	Approved	Must Have
BRE-04	Forgot Password	Forgot Password	<p>As a customer, I want the website to include a Forgot Password feature with an OTP/Verification Code Input Field, so that I can recover access to my account.</p> <p>The Forgot Password will include:</p> <ul style="list-style-type: none"> • OTP/Verification Code Field • Submit button • Time Estimation • Reset Password Field 	Approved	Must Have
BRE-05	Bus Searching	Search Bus Trips	<p>As a customer, I want the website to display a section to search for bus trips, so that I can find the trips traveling to my desired destinations.</p>	Approved	Must Have

			<p>The section will include:</p> <ul style="list-style-type: none"> ● Departure Locations ● Arrival Locations ● Departure Date ● Round Trip Switch Option ● Return Date 		
BRE-06	Display Result List		<p>As a customer, I want the website to show a Result List of Bus Trips, so that I can choose the best option for my journey.</p> <p>The Result List of Bus Trips will include:</p> <ul style="list-style-type: none"> ● Sort, filter and time customization ● Result of Bus Trips 	Approved	Must Have
BRE-07	Display Details of Bus Trips		<p>As a customer, I want the website to display the details of a bus trip, so that I can confirm which option is best suited for my needs.</p> <p>The Details of Bus Trips will include:</p> <ul style="list-style-type: none"> ● Pick-up and Drop-off Locations ● Ratings and Reviews ● Amenities ● Bus Features 	Approved	Could Have
BRE-08	Booking	Book tickets	As a customer, I want the website to provide a ticket booking feature, so that I can	Approved	Must Have

			reserve seats for my desired trips.		
BRE-09		Display Seat Mapping and Positions	As a customer, I want the website to present the Seat Mapping and Positions, so that I can easily select the most suitable seats and specify the quantity of seats required for my journey.	Approved	Must Have
BRE-10		Display Pick-up and Drop-off Locations	As a customer, I want the website to display Pick-up and Drop-off Locations, so that I can determine the starting and ending points of my journey.	Approved	Must Have
BRE-11		Select the shuttle service	As a customer, I want the website to offer options for choosing the shuttle service when booking tickets, so that I can choose the transportation mode that suits my preferences.	Approved	Should Have
BRE-12		Display Personal Information	As a customer, I want the website to automatically input my personal information in the booking process, so that I can save time from the manual insertion.	Approved	Must Have
BRE-13		Display Booking Review	As a customer, I want the website to display the booking review before making payments so that I can verify my booking information.	Approved	Must Have
BRE-14	Transaction	Payment	As a customer, I want the website to facilitate payment	Approved	Must Have

			transactions, so that I can securely complete purchases for booked tickets.		
BRE-15		Display Payment Selection	<p>As a customer, I want the website to support a diversity of payment methods, so that I can easily connect to the one that I want to process payment.</p> <p>The Payment Selection shall include</p> <ul style="list-style-type: none"> • Promotions • Payment Selection 	Approved	Must Have
BRE-16		Display Unpaid Tickets with Guidelines	As a customer, I want the website to display unpaid tickets with guidelines, so that I can continue processing payment if I want.	Approved	Could Have
BRE-17	Ticket History	Display Ticket History	<p>As a customer, I want the website to display ticket history, so that I can centrally manage all of my booking tickets.</p> <p>The Ticket History shall include:</p> <ul style="list-style-type: none"> • Status of tickets: Upcoming/Ongoing/Complete Tickets • Categorized Tickets: Unpaid/Paid/Canceled Tickets • Review Suggestion 	Approved	Must Have

BRE-18	Cancellations	Cancellations	<p>As a customer, I want the website to allow me to make booking cancellations, so that I can cancel tickets that I do not want to continue traveling.</p>	Approved	Must Have
BRE-19	Notifications	Notifications	<p>As a customer, I want the website to send me notifications, so that I can keep up with the news relevant to my booked trips.</p> <p>The Notifications shall include 2 types of notifications:</p> <ul style="list-style-type: none"> • Updates about users' tickets. • Updates about promotions and offers. 	Approved	Must Have
BRE-20	Account management	User Profile	<p>As a customer, I want the website to provide me with the feature of managing my personal profile, so that I can easily change any information relevant to it.</p>	Approved	Could Have
BRE-21	Support and Feedback	Input ratings and reviews	<p>As a customer, I want the website to allow me to input ratings and reviews, so that I can provide feedback on my experience with the service.</p>	Approved	Must Have
BRE-22		Hotline	<p>As a customer, I want the website to provide me with the company's hotline, so that I can contact them for help if necessary.</p>	Approved	Should Have

5.3.3. Login Process

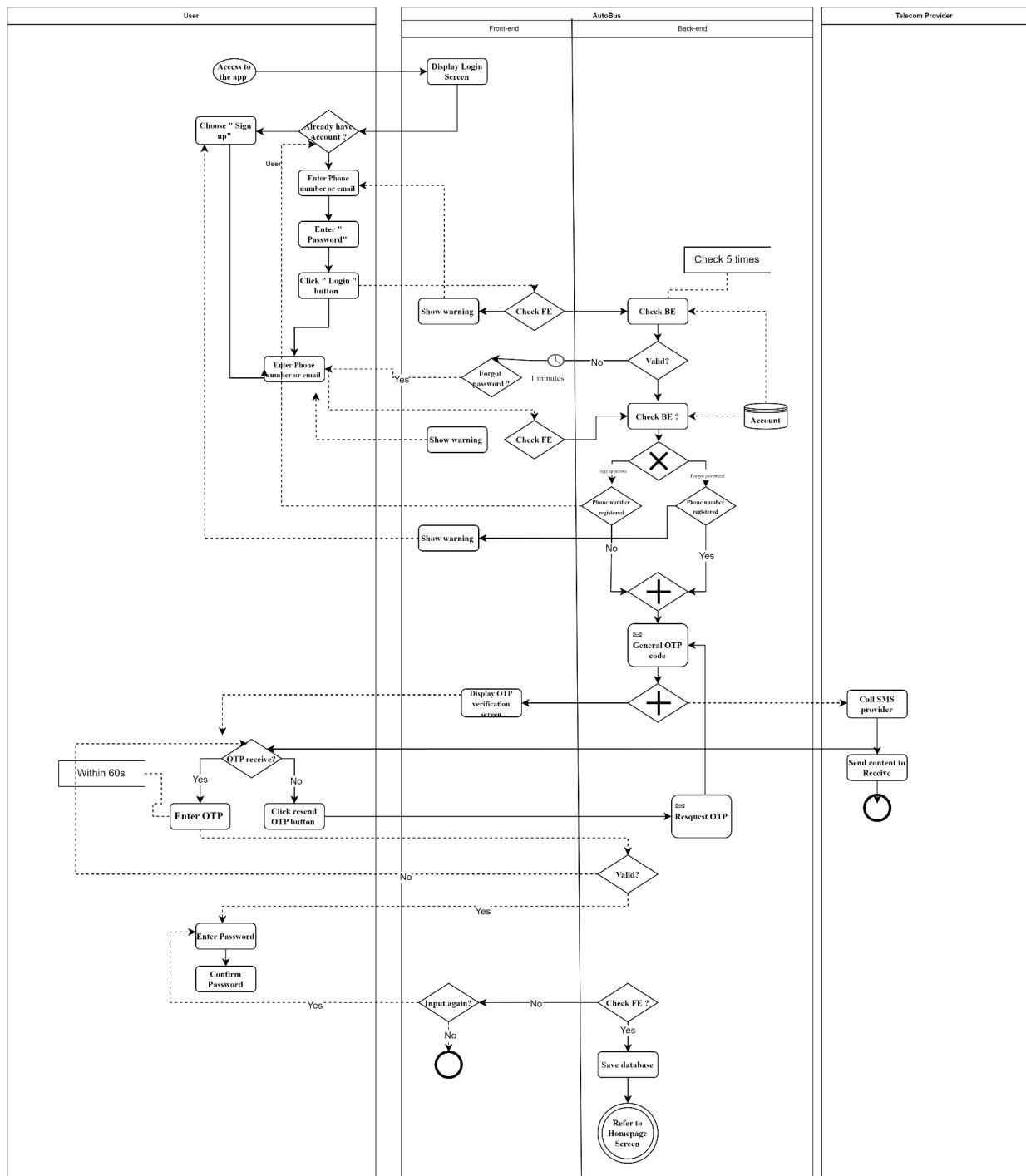


Figure 5.5. BPMN of Login process

(Source: Authors' source)

It begins with accessing the login screen, where users can enter their credentials or use the phone number for OTP-based authentication. The flow includes checks for account existence, password validation, and OTP verification. If users forget their credentials or fail login attempts, warnings are displayed, and OTPs are generated via SMS providers. Once validated, users can proceed to the homepage.

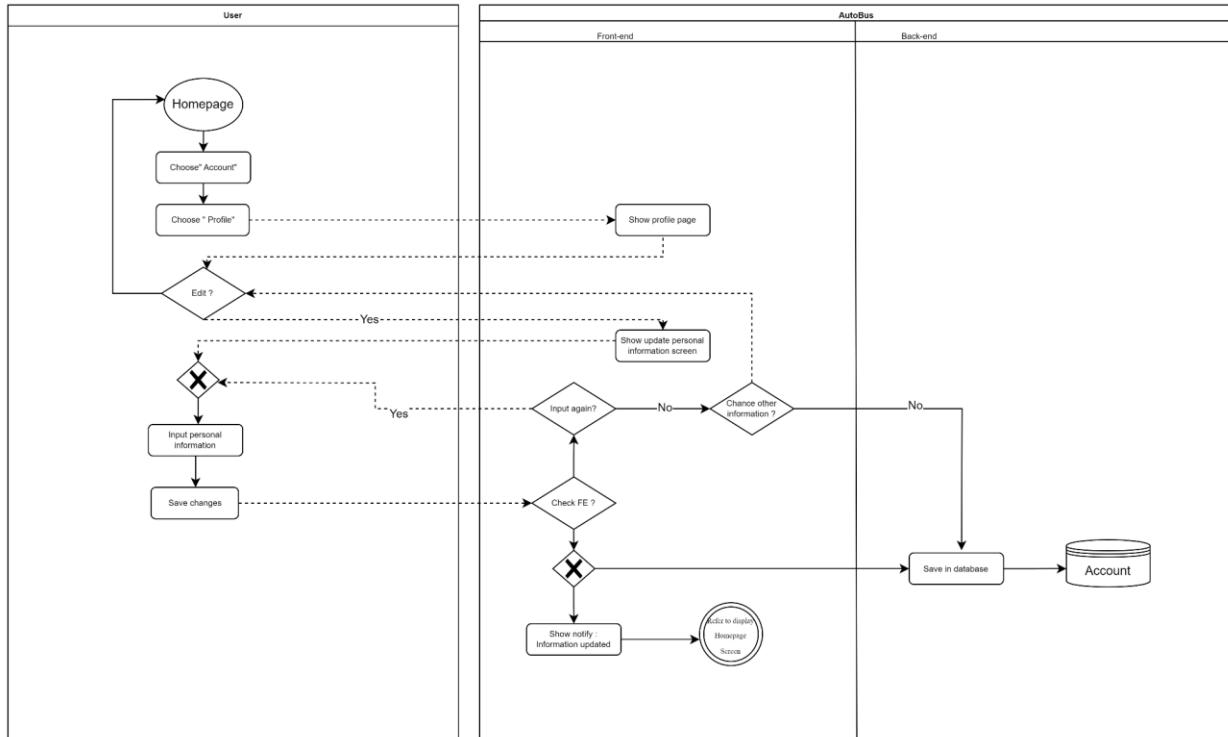


Figure 5.6. BPMN of Updated personal information process

(Source: Authors' source)

Users select "Account," navigate to their profile, and choose to edit their personal details. After inputting changes, the system validates the data and updates the database. Notifications confirm successful updates, and users are redirected to the homepage, ensuring smooth management of personal information.

Table 5.6. High Level Requirements of Login process

ID	Feature	Request Name	Description	Owner	Status
BRE - 0001	Log In	Login	As a customer, I want to declare my Phone Number and Password so that I can connect to my existing account and start booking bus tickets.	Approved	Must have
BRE - 0002	Forgot Password	Enter Phone Number/Email	As a customer, I want to provide my Email/Phone Number so that the system can find my existing account and reset Password.	Approved	Must have
BRE - 0003	Forgot Password	Enter OTP/Verification Code	As a customer, I want to declare OTP via phone number or Verification Code via email so that I can verify my Phone Number to register. The OTP/Verification Code shall include: <ul style="list-style-type: none"> ● Phone number received OTP/Verification Code ● 4-digit OTP Codes/Verification Codes ● Available time ● Resend request 	Approved	Must have

BRE - 0004	Forgot Password	Enter Password	As a customer, I want to enter a new Password so that I can get my account back.	Approved	Must have
BRE - 0005	Forgot Password	Confirm Password	As a customer, I want to reconfirm my Password so that I can verify the right Password once before submission.	Approved	Must have
BRE - 0006	Sign Up	Enter Phone Number/Email	As a customer, I want to provide my Phone Number/Email so that I can create a new Account and start booking tickets.	Approved	Must have
BRE - 0007	Sign Up	Enter OTP	As a customer, I want to declare OTP via phone number so that I can verify my Phone Number to register. The OTP shall include: <ul style="list-style-type: none"> • Phone number received OTP • 4-digit OTP Codes • Available time • Resend request 	Approved	Must have
BRE-0008	Sign Up	Verification Code	As a customer, I want to declare Verification Code via email so that I can verify my Email to register. The OTP shall include:		

			<ul style="list-style-type: none"> ● Email received Verification Code ● 4-digit Verification Codes ● Available time ● Resend request 		
BRE - 0009	Account Management	Update Account	<p>As a customer, I want to update my personal account so that I can change my information from this website.</p> <p>This will include:</p> <ul style="list-style-type: none"> ● Update Personal Information ● Update Password ● Promotions ● Update Payment Management ● Trip rating ● Promotional invitation ● Feedback ● Change language ● Support via hotline ● Settings 	Approved	Nice to Have

5.3.4. Payment Process

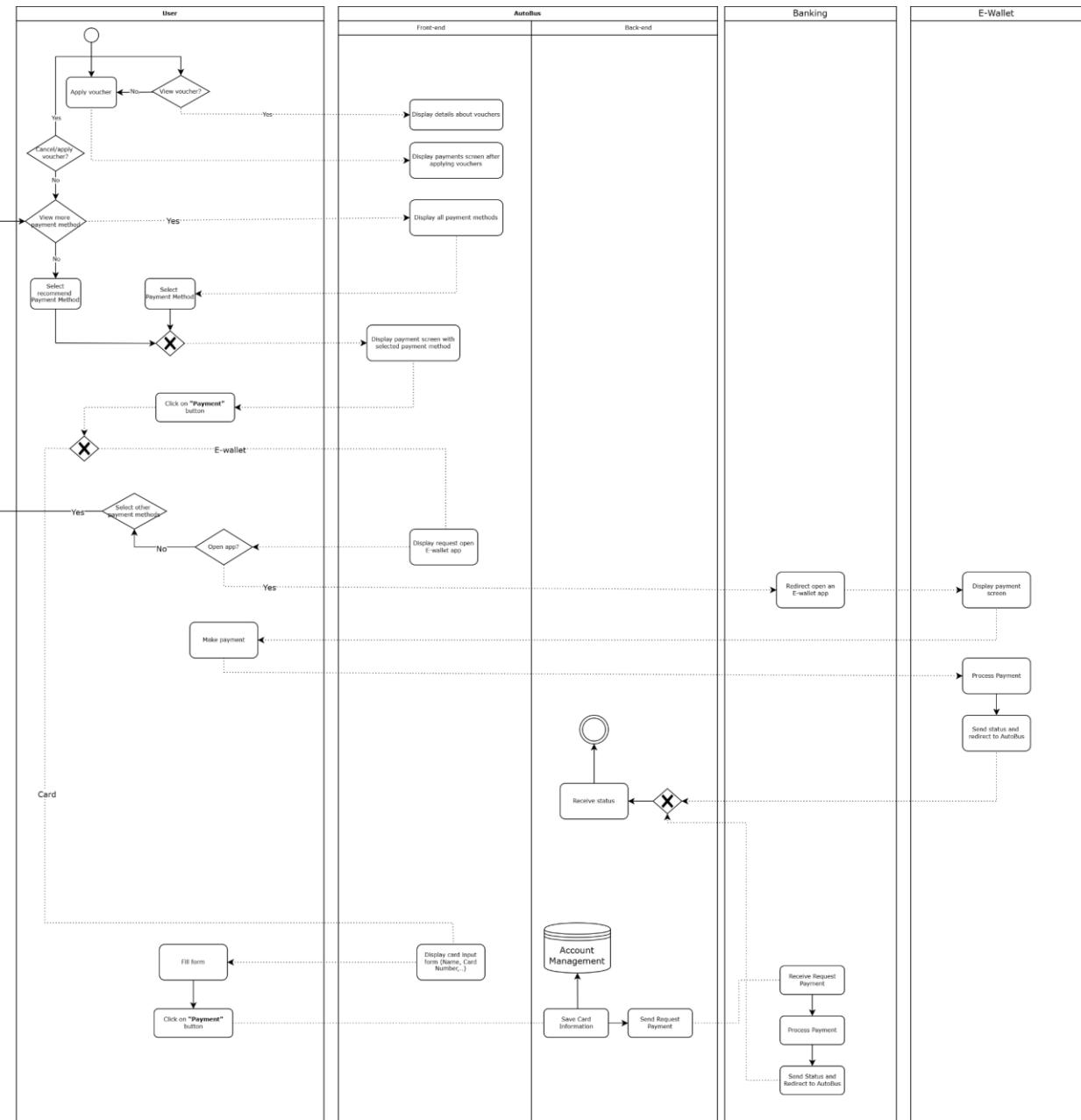


Figure 5.7. BPMN of Payment Process

(Sources: Authors' Sources)

The user begins by applying vouchers (if applicable) or directly selecting a payment method. The system displays the payment options and processes based on the user's

selection. For e-wallet payments, the system integrates with an external e-wallet app, allowing the user to confirm the transaction, after which the payment status is updated. For card payments, the user inputs card details, and the backend communicates with the banking system to process the payment securely. The system updates the payment status and confirms it to the user, completing the transaction. This flow ensures flexibility and security in payment handling.

Table 5.7. High level requirements of Payment Process in AutoBus.

ID	Feature	Request Name	Description	Status	Priority (MoSCoW)
BRE - 0001	Promotions & Discounts	Apply Vouchers	<p>As a customer, I want the website to offer voucher options, allowing me to reduce the ticket prices.</p> <p>The vouchers should include:</p> <ul style="list-style-type: none"> • Automatically applied vouchers when the user's account meets specific conditions. • Manually applied vouchers. 	Approved	Must Have
BRE - 0002	Payment	View Payment Methods	As a customer, I want the website to display all payment methods, so that I can easily choose the most suitable payment methods for me	Approved	Must Have

		<p>to proceed with the transaction.</p> <p>The View Payment Methods shall include:</p> <ul style="list-style-type: none"> • Recommendation of payment methods. • List of all payment methods. 		
BRE - 0003	View Transaction Status/Result	<p>As a customer, I want the website to display the result of the transaction after I proceed the payment, so that I can know whether my payment is successful or failed.</p> <p>The Transaction Status/Result shall include:</p> <ul style="list-style-type: none"> • Successful Payment. • Unsuccessful Payment. 	Approved	Must Have

5.3.5. Rating Process

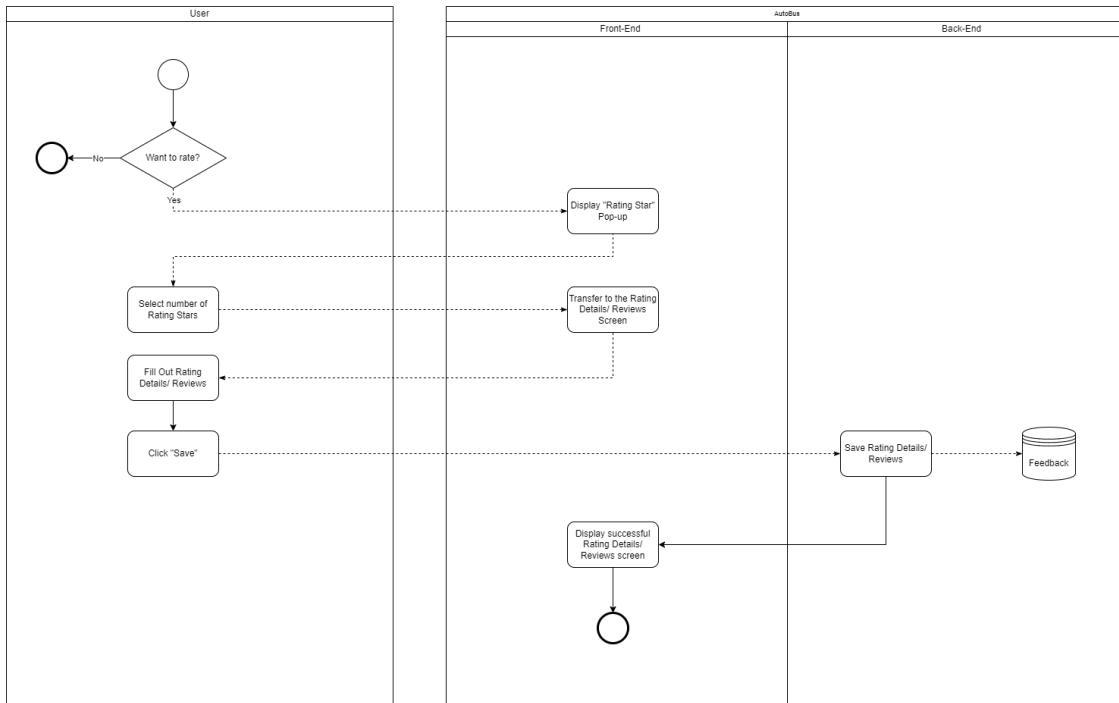


Figure 5.8. BPMN of Rating process

(Sources: Authors' Sources)

It starts with the user deciding whether to provide a rating. If they choose to rate, the system displays a pop-up for selecting the number of stars. The user then proceeds to a screen where they can fill out detailed feedback or reviews. After completing the input, the user clicks "Save," and the system stores the rating and review details in the feedback database. A confirmation screen notifies the user of the successful submission, completing the process. This flow ensures user feedback is collected efficiently and stored securely.

ID	Feature	Request Name	Description	Status	Priority (MoSCoW)
BRE - 0001	Rating & Review	Rating	As a customer, I would like to provide a rating feature, so that I can	Approved	Must Have

			express my satisfaction level after my journey.		
BRE - 0002	Rating & Review	Review	As a customer, I would like the website to offer a review feature, so I can provide my feedback and share my experiences.	Approved	Must Have
BRE - 0003	Rating & Review	View rating & review	As a customer, I would like the website to show ratings and reviews, allowing me to make well-informed decisions about future trips based on the experiences of others.	Approved	Must Have

5.4. Data Flow Diagram (DFD)

5.4.1. Deliverables

	Description	Purposes
Data Flow Diagram	The DFD visually represents the flow of data within the bus booking system, depicting the relationships between external entities, processes, and data stores. This includes Level 0 (context diagram) to give a high-level view, and Level 1.	Provides a structured overview of how data moves through the system, illustrating interactions between components and identifying data inputs, outputs, and transformations.
Specification	A comprehensive document detailing the elements represented in the DFD, including descriptions for each process, data store, data flow, and external entity. The specification	Ensures a clear understanding of each component and data interaction within the DFD, serving as a reference for developers and stakeholders to maintain consistency in design

	<p>includes definitions and purpose for each element, explaining their roles and interactions within the system.</p>	<p>and implementation.</p>
--	--	----------------------------

5.4.2. DFD Diagram

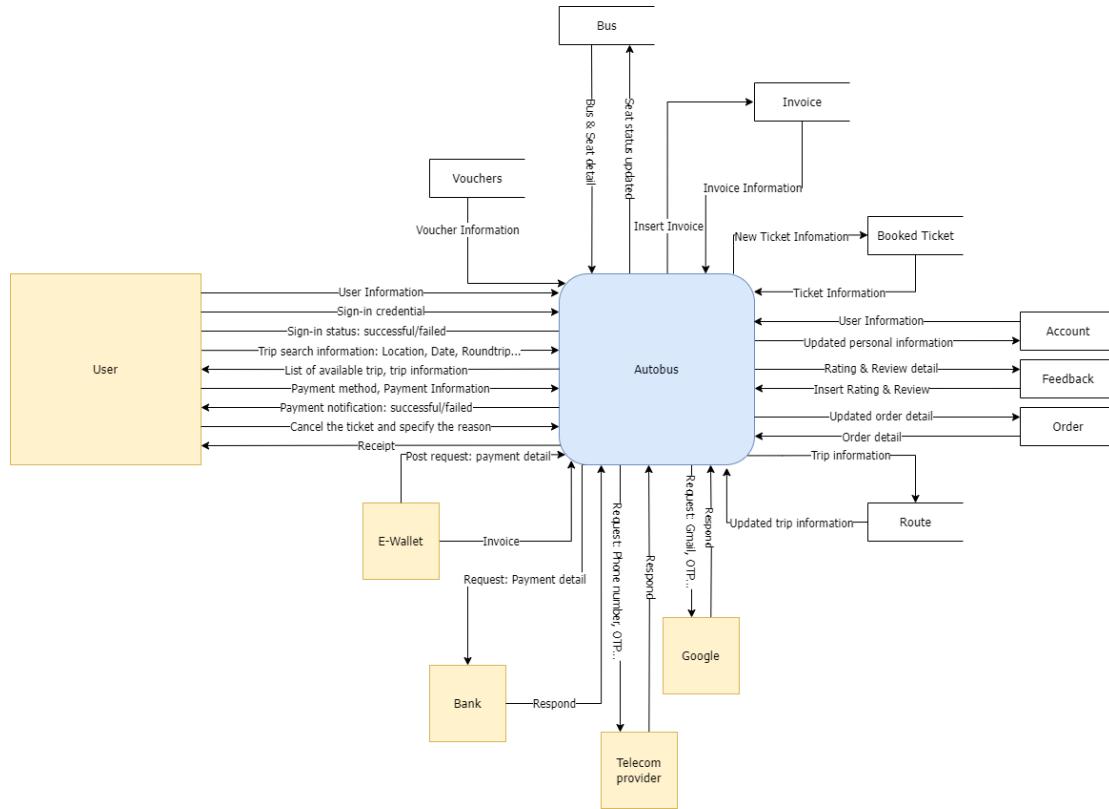


Figure 5.9. DFD level 0

(Source: Authors' source)

This diagram shows the connections between the user, the Autobus system, and external entities like e-wallets, banks, Google, telecom providers, and buses. It highlights key processes such as logging in, booking tickets, processing payments, and providing feedback. User inputs, such as trip details and payment methods, flow into the system, while outputs, like booked tickets, receipts, and notifications, are returned to the user. External systems handle additional tasks like OTP verification, payment processing, and email communications.

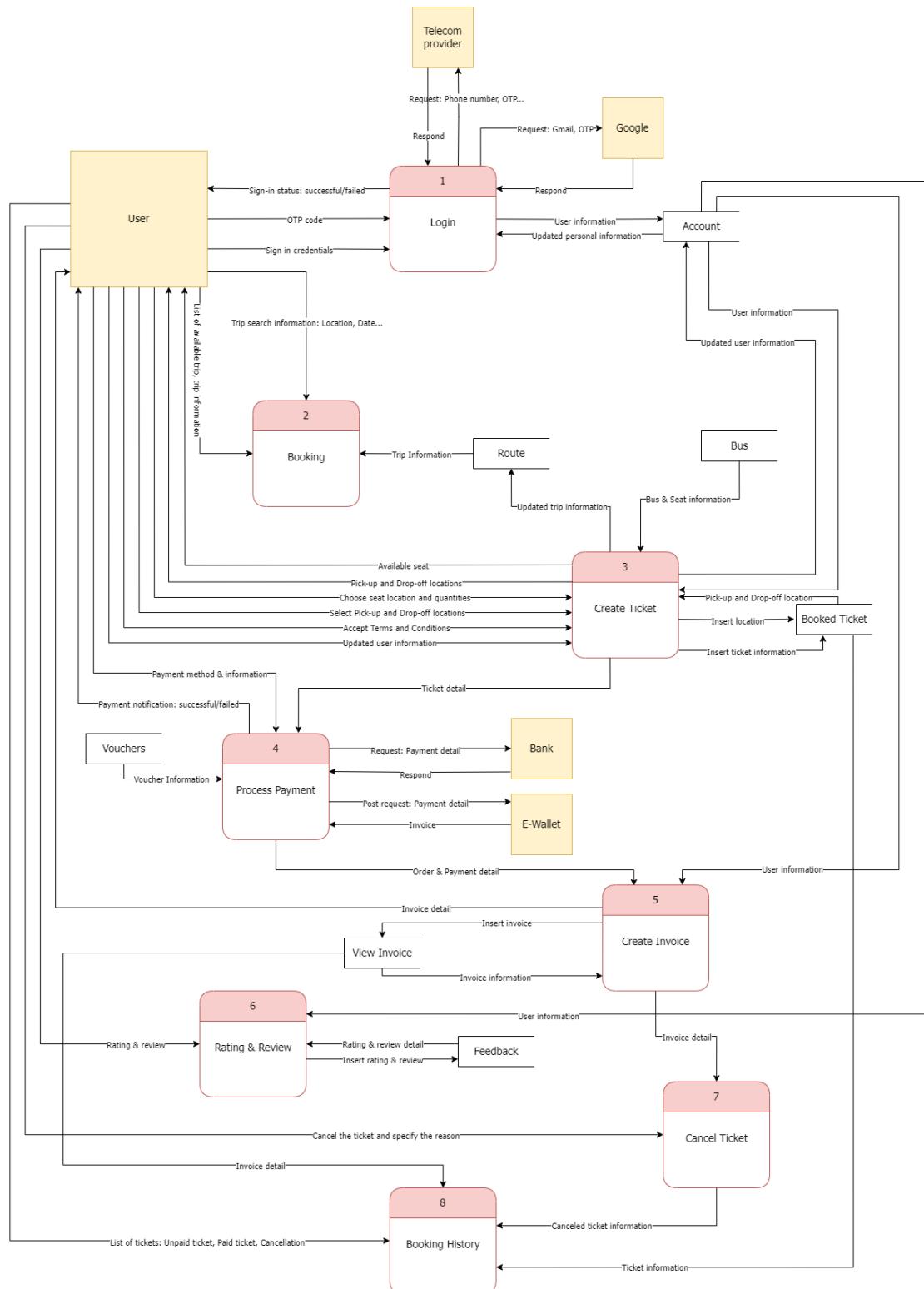


Figure 5.10. DFD level 1

(Source: Authors' source)

This representation breaks the system into modular processes, including login, booking, ticket creation, payment processing, invoice generation, and rating & review. It demonstrates how each module interacts with external systems and databases. Data flows between these modules are clear, ensuring a seamless user experience while keeping the system organized and efficient.

5.4.3. Specification

DFD specification

Process	Name	Specification
Process 1	Login	Customers can safely access the system by utilizing their login credentials through the login process. Customers can log in using either their phone number or email address. The system contacts Google or the telecom provider when a user logs in using their phone number or email address in order to validate their credentials and guarantee the security of the login procedure. After successfully logging in, clients can move on to the following phase.
Process 2	Booking	Customers choose their desired vacation from the list of alternatives in the second step, booking. Communication with the trips entity, which maintains data about the various trips available. Customers set the groundwork for the next steps in the ticket purchase process by providing pertinent information to complete their reservation.
Process 3	Create ticket	The third step, Create ticket, generates a ticket for the customer by using data from the chosen trip, customer information, and bus specifics. The produced ticket must be stored by the ticket booking entity and linked to the customer's individual booking information. This guarantees that the system and the consumer have an accurate record of the trip that was scheduled.
Process 4	Make payment	Following the creation of the ticket, the fourth process, Make payment, facilitates the secure transaction of funds. The customer initiates the payment process, and the system

		communicates with the bank and e-wallet platforms to securely process the transaction. Once payment is completed, the customer moves to the next step, triggering the creation of an invoice.
Process 5	Create invoice	The system creates an invoice outlining the transaction in the fifth step, Create invoice. A thorough record of all financial transactions within the system is provided by the invoice entity, which keeps pertinent data about the generated invoices. The customer's booking process is completed in this phase, guaranteeing that the system and the consumer have accurate records of the entire transaction.
Process 6	Ratings and Reviews	Customers can use the method to offer feedback about their trip and booking experiences. The Ratings and Reviews entity stores the ratings and reviews that customers provide. This information helps other customers make well-informed travel booking decisions and enhances the quality of services. This data may also be used by the system for reporting and analytics.
Process 7	Cancel Ticket	The process enables customers to cancel their booked tickets. Customers initiate the cancellation by providing relevant details, such as ticket booking ID and reason for cancellation. The system communicates with the ticket booking entity to locate and update the status of the booked ticket. If the cancellation is within the allowed timeframe, the system proceeds with canceling the ticket, and the ticket entity is updated accordingly. If the cancellation process is successful, the customer receives confirmation of the canceled ticket, and any applicable refunds are processed through communication with the bank or e-wallet.
Process 8	Booking History	This process enables users to access their booking history, displaying details of paid tickets, unpaid tickets, and cancellations. Users can view a list of all their transactions, including invoice information and ticket details retrieved from the respective data stores. This feature enhances transparency and provides users with easy access to their past bookings.

5.5. Use case

5.5.1. Deliverables

Table. Use Case Overall

	Description	Purposes
Use Case Diagram	<p>The Use Case Diagram visually illustrates the interactions between users (actors) and the system, highlighting the primary functions such as booking tickets, viewing schedules, managing bookings, and making payments. It represents all user interactions with the mobile app in a high-level view.</p>	<p>Provides a clear understanding of the system's scope by identifying the key interactions and functionalities that the mobile app must support. It also helps developers and stakeholders comprehend the user requirements and app functionality.</p>
Specification	<p>A detailed document specifying each use case, including descriptions of actors, preconditions, triggers, main and alternate flows, and postconditions for each use case. Each use case explains the specific functionality from a user perspective, detailing user actions and system responses.</p>	<p>Ensures all functionalities are documented for clarity and consistency in development. This serves as a reference for developers, testers, and stakeholders, making it easier to validate that each function meets the specified requirements.</p>

5.5.1. Overall

Use Case Name	Main Actor
Account Management	
UC-01: Login	Customer
UC-02: Register an account	Customer
UC-03: Update personal information	Customer
UC-04: Forgot password	Customer
UC-05: Create OTP	Telecom Provider
UC-05.1: Authenticate OTP	Telecom Provider
UC-05.2: Send OTP	Telecom Provider
UC-06: Create Authentication Code	Gmail Provider
UC-06.1: Verify Authentication Code	Gmail Provider
UC-06.2: Send Authentication Code	Gmail Provider
Booking	
UC-01: Find Bus Trip	Customer
UC-01.1: Choose Departure Point	Customer
UC-01.2: Choose Destination Point	Customer
UC-01.3: View Location List	Customer
UC-01.4: Select Start Date	Customer
UC-01.5: Choose Roundtrip	Customer
UC-01.6: Select Return Date	Customer
UC-01.7: View Schedule	Customer

UC-02: View Bus List	Customer
UC-02.1: Filter/Sort Buses	Customer
UC-03: View Bus Details	Customer
UC-03.1: View Pick-up and Drop-off Locations	Customer
UC-03.2: View Ratings and Review	Customer
UC-03.3: Check Available Amenities	Customer
UC-03.4: View Bus Features	Customer
UC-04: Book Tickets	Customer
UC-04.1: Check Bus Rules	Customer
UC-04.2: Select Bus Seats	Customer
UC-04.3: View Bus-seat Positions	Customer
UC-04.4: Select Pick-up and Drop-off Locations	Customer
UC-04.4.1: Select shuttle bus service	Customer
UC-04.5: View List of Pick-up and Drop-off Points	Customer
UC-04.6: Input Personal Information	Customer
UC-04.7: View Ticket-Booking Review	Customer
Transactions	
UC-01: Choose Payment method	Customer
UC-02: Make Payment	Customer

UC-02.1: Apply voucher	Customer
UC-02.2: Check voucher details	Customer
UC-02.3: Payment Instructions	Customer
UC-03: Process Payment	E-wallet Banking
UC-03.1: Modify ticket status	Systems
UC-04: View Ticket-Booking Invoice	Customer
Booking History	
UC-01: View ticket status	Customer
UC-02: View Paid ticket	Customer
UC-02.1: Check Upcoming ticket	Customer
UC-02.2: Check Ongoing ticket	Customer
UC-02.3: Check Complete ticket	Customer
UC-02.4: Update real-time information	Systems
UC-03: View Unpaid ticket	Customer
UC-03.1: Change pick-up/drop-off point	Customer
UC-03.2: Modify transit location	Customer
UC-03.3: View Payment Instructions	Customer
UC-04: View Cancellation	Customer

UC-04.1: Choose tickets to cancel	Customer
UC-04.2: View cancel regulations	Customer
UC-04.3: Choose reason to cancel	Customer
UC-04.4: Confirm Cancellation	Customer
Support and Feedback	
UC-01: Contact support hotline	Customer
UC-02: Rating	Customer
UC-02.1: Submit feedback	Customer
UC-02.2: View other Ratings and Review	Customer
Notifications	
UC-01: View List of Notifications	Customer
UC-02: View Tickets	Customer
UC-03: View Promotions	Customer

5.5.2. Account management

The Account Management Module included:

- **Main Use Cases:**
 - UC-01: Login Account
 - UC-02: Register Account
 - UC-03: Update Personal Information
 - UC-04: Forgot Password

- **Sub Use Cases (included from the main Use Cases):**
 - **UC-05: Create OTP**
 - UC-05.1: Verify OTP
 - UC-05.2: Send OTP
 - Linked with the **Telecom Provider** actor to send OTP via telecom service.
 - **UC-06: Create Verification Code**
 - UC-06.1: Verify Verification Code
 - UC-06.2: Send Verification Code
 - Linked with the **Gmail Provider** actor to send verification code via email.

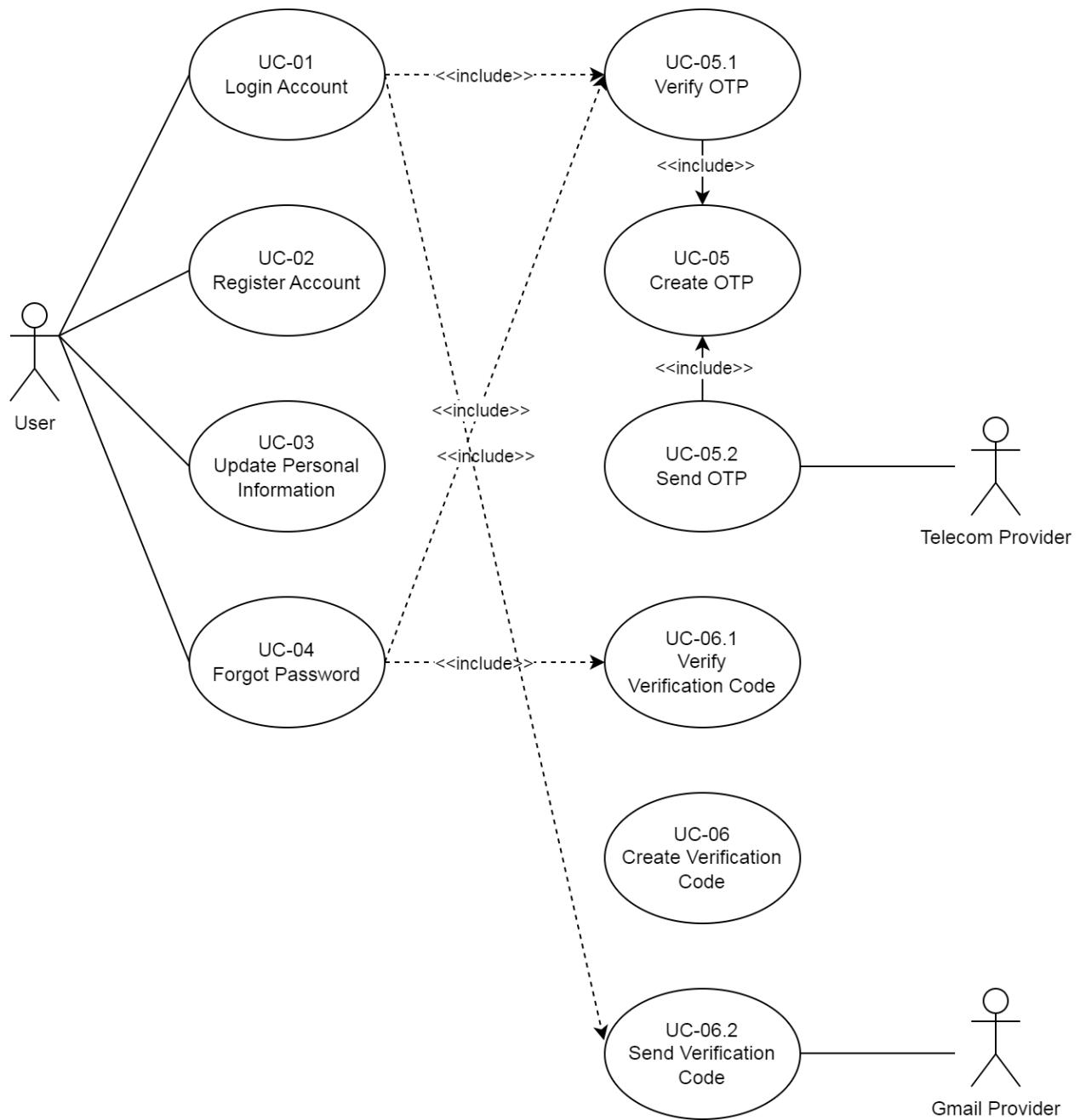


Figure 5.11. Use case of Account Management

(Source: Authors' source)

Use Case: Register an account

- Overall:

Table 5.8. Use Case Detail - Description: Register account of Account Management

Name of Use case	Register Account
Actors	Customer, Telecom provider, Gmail Provider.
Description	This use case describes how a user registers an account to access a website.
Successful completion	<ul style="list-style-type: none"> • The user is successfully registered and logged in. • The Homepage is shown with the user's account active.
Alternative	<ul style="list-style-type: none"> • If registration fails due to invalid input, the user is prompted to correct errors (e.g., incorrect email format, weak password). • If the database connection fails, an error message is displayed, and the user is asked to try again later. • If the user's information already exists in the database, the system suggests logging in instead of registering.
Precondition(s)	<ul style="list-style-type: none"> • The website has been accessed on the user's device. • The device is connected to the Internet. • The user's registration details are not yet stored in the database.
Trigger	Click on the “ Register ” button.
Postcondition(s)	<ul style="list-style-type: none"> • The user registered information is saved in the database. • The website shows the Homepage.
Assumptions	<ul style="list-style-type: none"> • The user has valid information (e.g., email, phone number) for registration. • The Telecom Provider ensures Internet connectivity throughout the process. • The system enforces data validation and error handling for all fields in the registration form.

- **Flow of Events:**

Table 5.9. Flow of Events: Module Register Account of account management

Step	Actor	Action
Flow of Events (BF)		
BF-01	Customer	The customer clicks on the "Register" button.
BF-02	System	The "Registration" screen is displayed.
BF-03	Customer	The user enters their phone number or email address.
BF-04	System	The system generates a One-Time Password (OTP) or verification code.
BF-05	Telecom provider/ Gmail provider	The OTP/verification code is sent to the user.
BF-06	Customer	The user receives and inputs the OTP/verification code.
BF-07	System	The system verifies the OTP/verification code.
BF-08	Customer	The user sets and confirms their password.
BF-09	System	The system validates and saves the user's information in the database.
BF-10	System	The "Homepage" screen is displayed.
Exceptional Flow (EF)		
EF-01	Customer	Input an invalid phone number/email
EF-02	System	Display pop-up notify "Invalid phone number/Email"
EF-03	Customer	Retry in BF-03
EF-04	Customer	Input an invalid OTP
EF-05	System	Display pop-up notify "Invalid OTP"

EF-06	Customer	Retry in BF-05
EF-07	Customer	Confirm an invalid password
EF-08	System	Display pop-up notify “Invalid password”
EF-09	Customer	Retry in BF-03

Use Case: Login

- Overall:

Table 5.10. Use Case Detail - Description: Module Login account of account management

Name of Use case	Login Account
Actors	Customer, Telecom provider, Gmail Provider.
Description	This use case describes how a user registers an account to access a website .
Successful completion	<ul style="list-style-type: none"> • The user is successfully registered and logged in. • The Homepage is shown with the user's account active.
Alternative	<ul style="list-style-type: none"> • If registration fails due to invalid input, the user is prompted to correct errors (e.g., incorrect email format, weak password). • If the database connection fails, an error message is displayed, and the user is asked to try again later. • If the user's information already exists in the database, the system suggests logging in instead of registering.
Precondition(s)	<ul style="list-style-type: none"> • The website has been accessed on the user's device. • The device is connected to the Internet. • The user's registration details are not yet stored in the database.

- Flow of Events:

Table 5.11. Flow of Events: Module Login account of account management

Step	Actor	Action	Note/References
Flow of Events			
BF-01	Customer	Access to the website	
BF-02	System	Display the “Login” Screen	
BF-03	Customer	Input Phone number/Email and password	
BF-04	System	Authenticate Phone number/Email and password	
BF-05	System	Display Homepage screen	
Exceptional Flow			
EF-01	Customer	Input an invalid phone number/Email	
EF-02	System	Display pop-up notify “Incorrect Phone Number/Email”	
EF-03	Customer	Retry in BF-03	
EF-04	Customer	Input an invalid password	
EF-05	System	Display pop-up notify “Incorrect Password”	
EF-06	Customer	Retry in BF-03	

Use Case: Forgot password

- Overall:

Table 5.12. Use Case Detail - Description: Module Forgot password of account management

Module	Forgot password
Description	This use case describes how a user get their account back if they forgot password
Actors	Customer, Telecom provider
Precondition(s)	<ul style="list-style-type: none"> • The website is downloaded. • The user's device connects to the Internet.
Trigger	Click on “Forgot password”
Postcondition(s)	<ul style="list-style-type: none"> • The user account is saved in the database. • The website shows the Homepage.

- **Flow of Events:**

Table 5.13. Flow of Events: Module Forgot password of account management

Step	Actor	Action	Note/References
Flow of Events			
BF-1	Customer	Access to the website	
BF-2	System	Display the “Login” Screen	
BF-3	Customer	Choose “Forgot password”	
BF-4	System	Display “Forgot password” Screen	
BF-5	Customer	Input Phone number/Email	
BF-6	System	Authenticate phone number/Email	
BF-7	System	Generate OTP/Verification Code	

BF-8	Telecom provider/ Gmail provider	Send OTP/Verification Code	
BF-9	Customer	Receive and enter the OTP code/Verification Code	
BF-10	System	Authenticate the valid OTP/Verification Code	
BF-11	System	Display “Create new password” screen	
BF-12	Customer	Input new password and confirm password	
BF-13	System	Authenticate and Save user information in the database	
BF-14	System	Display “Homepage” screen	
Exceptional Flow			
EF-01	Customer	Input an invalid phone number/Email	
EF-02	System	Display pop-up notify “Invalid phone number/Email”	
EF-03	Customer	Retry in BF-5	
EF-04	Customer	Customer click “Resend” OTP Code/Verification Code	
EF-05	System	Generate new OTP Code/Verification Code and send to customers	
EF-06	Customer	Customer input an invalid OTP/Verification Code	

EF-07	System	Display pop-up authenticate results with “Invalid OTP Code/Verification Code”	
EF-08	Customer	Retry in BF-8	
EF-09	Customer	Input less than 8 characters in text password fields or unmatching characters	
EF-10	System	Display the error “Password invalid”	
EF-11	Customer	Retry in BF-11	

Use Case: Update personal information

- Overall:

Table 5.14. Use Case Detail - Description: Module Update personal information of account management

Module	Update account
Description	This use case describes the process by which a user updates the information to a personal account.
Actors	Customer
Precondition(s)	<ul style="list-style-type: none"> • The user’s device connects to the Internet. • The user already has an account.
Trigger	Click on the “Profile” button.
Postcondition(s)	<ul style="list-style-type: none"> • The user-updated information is saved in the database.

- Flow of Events:

Table 5.15. Flow of Events: Module Update personal information of account management

Step	Actor	Action	Note/References
Flow of Events			
Step	Actor	Action	Note/References
BF-01	Customer	Click on the “Profile” button	
BF-02	System	Display “Profile” Screen	
BF-03	Customer	Click on the section to update the information	
BF-04	System	Display screen based on customer preference	

5.5.3. Booking

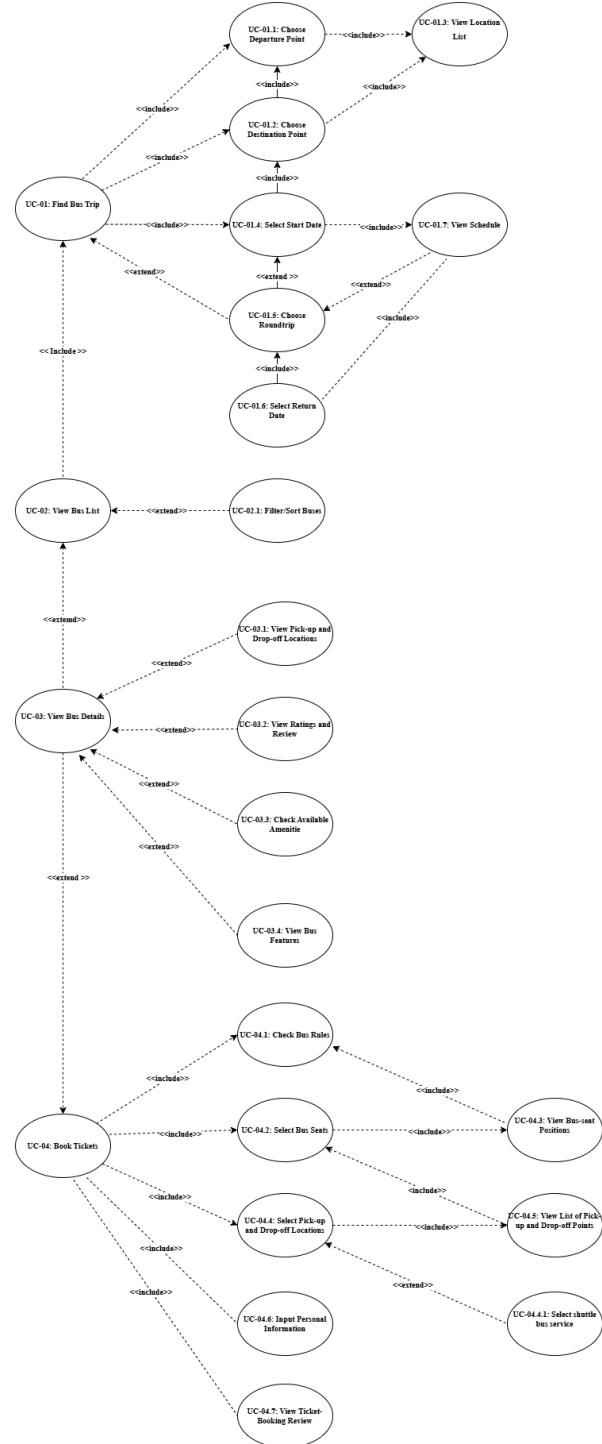


Figure 5.12. Use case of Booking

(Source: Authors' source)

Starting with finding a bus trip by selecting departure and destination points, start and return dates, and viewing schedules and locations. Users can then view the bus list with options to filter and sort results. Detailed bus information, such as pick-up/drop-off points, amenities, ratings, reviews, and rules, is accessible before booking. The ticket booking process involves selecting seats, pick-up/drop-off points, and optional shuttle services, followed by inputting personal details and reviewing ticket information, ensuring a comprehensive and user-friendly booking journey.

5.5.4. Transactions

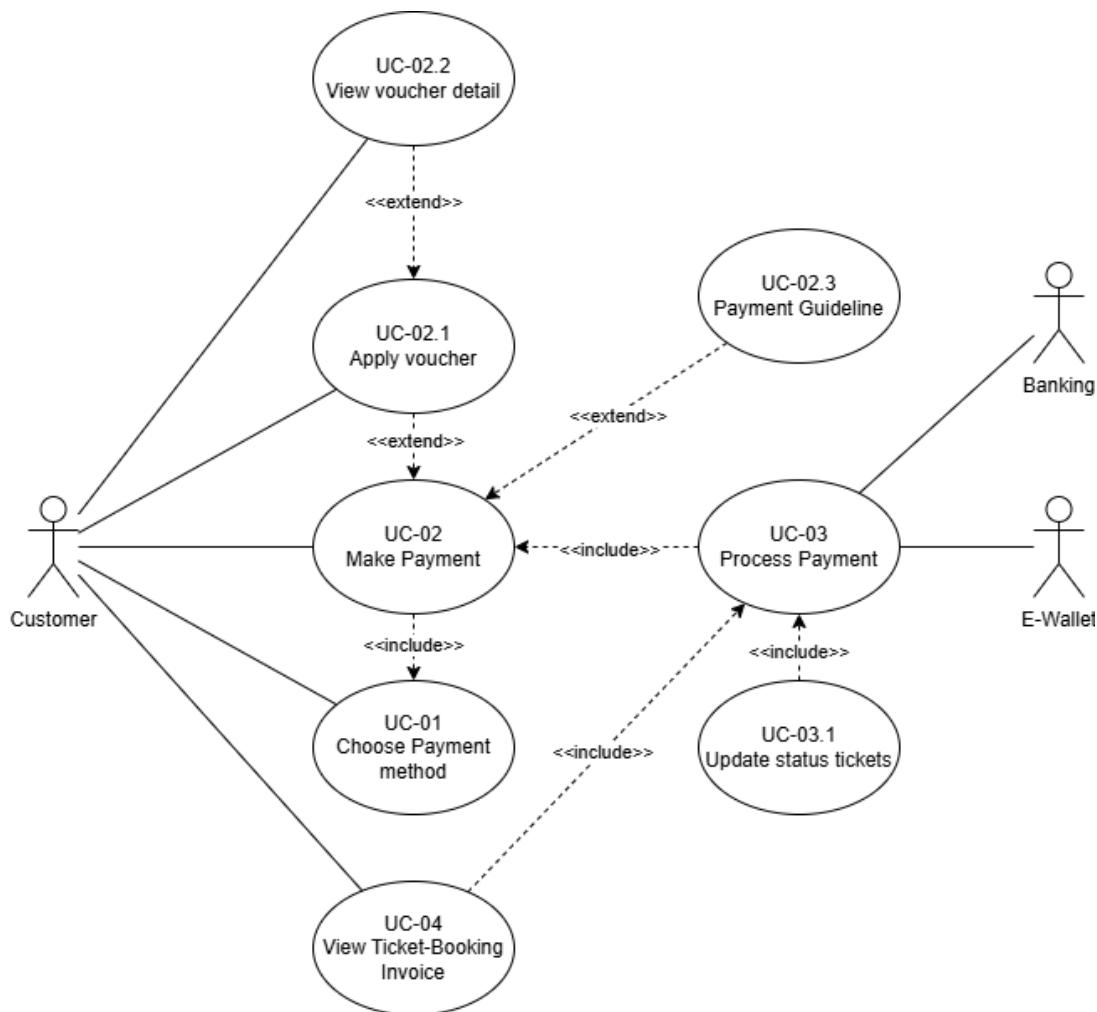


Figure 5.13. Use case of Transactions

(Source: Authors' source)

This use case diagram focuses on the payment and invoicing process of a bus booking system. It begins with the customer choosing a payment method (**UC-01**) and proceeding to make a payment (**UC-02**), which includes applying vouchers (**UC-02.1**) and viewing voucher details (**UC-02.2**). Customers can also refer to payment guidelines (**UC-02.3**) during the process. Payment processing (**UC-03**) involves interactions with external systems like banking and e-wallets, updating ticket statuses (**UC-03.1**) upon successful transactions. Finally, the system allows customers to view the ticket booking invoice (**UC-04**), completing the payment workflow in a structured and user-friendly manner.

UC-01: Choose Payment method

- **Overall:**

Table 5.16. Choose Payment Method Specification

Description	This Use Case describes the process by which the user chooses a suitable payment method to implement the payment.
Actors	Customer
Precondition(s)	The user's device connects to the Internet.
Trigger	The user selects one payment method and inserts required information of the chosen payment method.
Postcondition(s)	<ul style="list-style-type: none"> • One payment method is selected. • The user payment information is applied to the transaction.

- **Flow of Events:**

Table 5.17. Choose Payment Method Flow Analysis

Step	Actor	Action	Note/References
Basic Flow			

BF-1	Customer	Click on the “Confirm” button on the Ticket-booking Review.	
BF-2	System	Navigate to the Payment Screen.	
BF-3	Customer	Click on the recommended payment method.	
BF-4	System	Require the user to input the information relevant to the selected payment method.	
BF-5	Customer	Input required payment information.	
BF-6	System	Apply the information of the selected payment method to the transaction.	

Alternative Flow

AF-1	Customer	Click on the “Confirm” button on the Ticket-booking Review.	
AF-2	System	Navigate to the Payment Screen.	
AF-3	Customer	Click on the “View All” on the Payment Method section.	
AF-4	System	Display list of Payment Methods.	
AF-5	Customer	Select one payment method.	
AF-6	System	Require the user to input the information relevant to the selected payment method.	
AF-7	Customer	Input required payment information.	
AF-8	System	Apply the information of the selected payment method to the transaction.	

UC-02: Make Payment

- Overall:

Table 5.18. Make Payment Specification

Description	This use case describes the process by which the user makes the payment.
Actors	Customer
Precondition(s)	<ul style="list-style-type: none"> • The user's device connects to the Internet. • All payment information is already filled out.
Trigger	The user clicks on the “Pay” button on the Payment Screen.
Postcondition(s)	<ul style="list-style-type: none"> • The transaction is completely paid. • The “Successful Payment” is displayed.

- **Flow of Events:**

Table 5.19. Make Payment Flow Analysis

Step	Actor	Action	Note/References
Basic Flow			
BF-1	Customer	Click on the “Pay” button on the Payment Screen.	
BF-2	System	Navigate to the third-party payment platform to implement the payment.	
BF-3	Customer	Pay for the transaction.	
BF-4	System	Display “Successful Payment” Screen.	
Exceptional Flow			
EF-1	Customer	Click on the “Pay” button on the Payment Screen.	
EF-2	System	Navigate to the third-party payment platform to implement the payment.	
EF-3	Customer	Pay for the transaction.	
EF-4	System	Unable to verify the transaction.	

EF-5	User	Retry at	
EF-6	System	Display “Unsuccessful Payment” Screen.	

UC-02.1: Apply voucher

- Overall:

Table 5.20. Apply Voucher Specification

Description	This use case describes the process by which a user applies one voucher to the transaction.
Actors	Customer
Precondition(s)	<ul style="list-style-type: none"> • The user’s device connects to the Internet. • The user account meets the conditions of the voucher. • The voucher enables the user to select to apply.
Trigger	User selects one enable-to-click voucher.
Postcondition(s)	<ul style="list-style-type: none"> • The selected voucher is applied to the transaction. • The total price of the transaction is discounted based on the voucher regulations.

- Flow of Events:

Table 5.21. Apply Voucher Flow Analysis

Step	Actor	Action	Note/References
Basic Flow			
BF-1	Customer	Click on the “Confirm” button on the Ticket-booking Review.	
BF-2	System	Navigate to the Payment Screen.	
BF-3	Customer	Select one enable-to-click voucher.	
BF-4	System	Apply the voucher to the transaction and discount the total price based on the voucher regulations.	

Alternative Flow			
AF-1	Customer	Click on the “Confirm” button on the Ticket-booking Review.	
AF-2	System	Navigate to the Payment Screen.	
AF-3	Customer	Select “View All” on the Vouchers section.	
AF-4	System	Display the list of vouchers.	
AF-5	Customer	Select one enable-to-click voucher.	

UC-02.2: Check voucher details

- **Overall:**

Table 5.22. View Voucher Detail Specification

Description	This Use Case describes the process by which a user views the details of one voucher.
Actors	Customer
Precondition(s)	The user’s device connects to the Internet.
Trigger	User clicks on the “Condition” button on one voucher.
Postcondition(s)	The conditions of the voucher are displayed.

- **Flow of Events:**

Table 5.23. View Voucher Detail Flow Analysis

Step	Actor	Action	Note/ References
Basic Flow			

BF-1	Customer	Click on the “Confirm” button on the Ticket-booking Review.	
BF-2	System	Navigate to the Payment Screen.	
BF-3	Customer	Click on the “Condition” button on one voucher.	
BF-4	System	Navigate to the details of voucher conditions.	

Alternative Flow

AF-1	Customer	Click on the Account task on the bottom bar.	
AF-2	System	Navigate to the Account Management Screen.	
AF-3	Customer	Click on the “Promotions” section.	
AF-4	System	Display the list of vouchers.	
AF-5	Customer	Click on the “Condition” on one voucher.	
AF-6	System	Navigate to the details of voucher conditions.	

UC-02.3: Payment Instructions

- Overall:

Table 5.24. Payment Guideline Specification

Description	This use case describes the process by which the user views Payment Guideline when they process the payment of Unpaid Tickets again.
Actors	Customer
Precondition(s)	<ul style="list-style-type: none"> • The user's device connects to the Internet.

	<ul style="list-style-type: none"> • The user continues paying for the payment of the Unpaid Tickets. • The ticket is displayed on the “Unpaid” tab on the History Screen. • The Unpaid Tickets still in the 10-minute countdown progress for the payment.
Trigger	The user clicks on the “View Payment Guideline” button on the Payment Screen.
Postcondition(s)	<ul style="list-style-type: none"> • Display the Payment Guidelines.

- **Flow of Events:**

Table 5.25. Payment Guideline Flow Analysis

Step	Actor	Action	Note/References
Basic Flow			
BF-1	Customer	Click on the “Back” button on the Payment Screen.	
BF-2	System	Display the dialog including the Payment Guidelines.	
Alternative Flow			
AF-1	System	Display Unsuccessful Payment Screen.	
AF-2	Customer	Click on the “Back To Homepage” button.	
AF-3	System	Display a Dialog/Pop-up.	

AF-4	Customer	Click on “View Payment Guideline” button.	
AF-5	System	Display “Payment Guideline” Screen.	

UC-03: Process Payment

- **Overall:**

Table 5.26. Process Payment Specification

Description	This Use Case describes the process by which the user proceeds the payment with the Banking system or the E-Wallet Platforms.
Actors	Customer, Banking, E-Wallets,
Precondition(s)	<ul style="list-style-type: none"> • The user’s device connects to the Internet. • The user fully applies all required information relevant to their Banking/E-Wallet Account. • The user’s Banking/E-Wallet Account is valid.
Trigger	The user confirms the payment through the Banking or E-Wallet Platforms.
Postcondition(s)	<ul style="list-style-type: none"> • The payment transaction is successful. • The user’s Banking/E-Wallet Account has been deducted.

- **Flow of Events:**

Table 5.27. Process Payment Flow Analysis

Step	Actor	Action	Note/References
Basic Flow			

BF-1	Customer	Click on the “Pay” button on the Payment Screen.	
BF-2	System	Navigate to the Banking/E-Wallets Platform to implement the payment.	
BF-3	Customer	Pay for the transaction.	
BF-4	System	Display “Successful Payment” Screen.	

UC-03.1: Modify ticket status

- Overall:

Table 5.28. Update Status of Tickets Specification

Description	This use case describes the process by which the system updates the status of the tickets.
Actors	System, customer
Precondition(s)	<ul style="list-style-type: none"> • The user’s device connects to the Internet. • The user proceeds the payment.
Trigger	The system updates the status of the tickets.
Postcondition(s)	<ul style="list-style-type: none"> • The status of tickets is updated. • The tickets are displayed on the “History” in a suitable tab according to their status.

- Flow of Events:

Table 5.29. Update Status of Tickets Flow Analysis

Step	Actor	Action	Note/References
Basic Flow			
BF-1	Customer	Process Payment.	
BF-2	System	Update the status of the tickets.	

UC-04: View Ticket-Booking Invoice

- Overall:

Table 5.30. View Ticket-Booking Invoice Specification

Description	This use case describes the process by which the user views their ticket-booking invoice after payment.
Actors	Customer
Precondition(s)	<ul style="list-style-type: none"> • The user's device connects to the Internet. • The user has already paid for the tickets.
Trigger	The user clicks on the “View Invoice” button on the “Successful Payment” Screen.
Postcondition(s)	The Invoice is displayed.

- Flow of Events:

Table 5.31. View Ticket-Booking Invoice Flow Analysis

Step	Actor	Action	Note/References
Basic Flow			

BF-1	Customer	Process payment.	
BF-2	System	Display “Successful Payment” Screen.	
BF-3	Customer	Click on the “View Invoice” button.	
BF-4	System	Navigate to the “Ticket-booking Invoice” Screen.	

5.5.5. Booking History

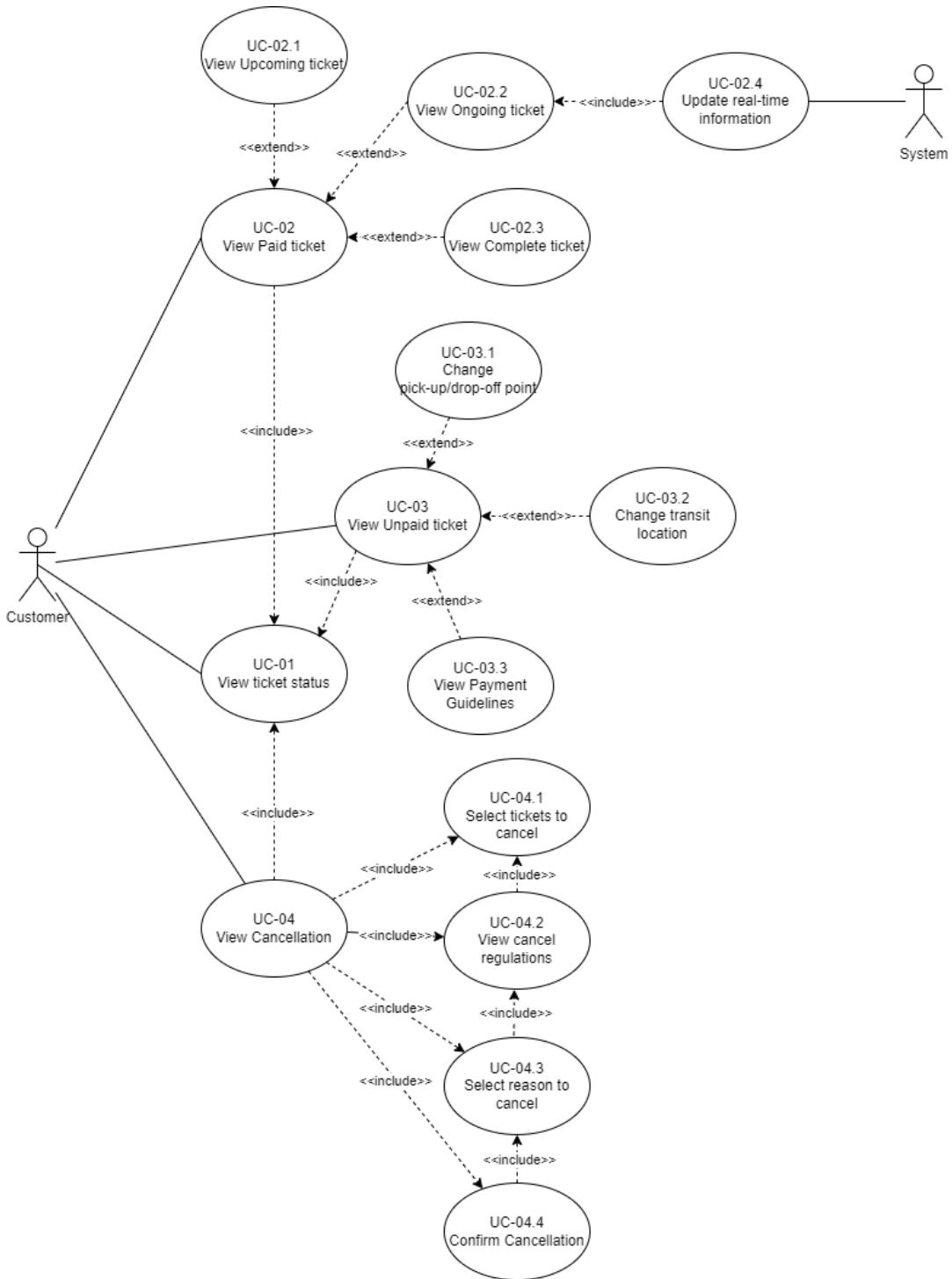


Figure 5.14. Use case of Booking History

(Sources: Author's Sources)

It starts with the customer viewing ticket status (**UC-01**) and accessing detailed options such as viewing paid tickets (**UC-02**) and unpaid tickets (**UC-03**). For paid tickets, customers can see upcoming (**UC-02.1**), ongoing (**UC-02.2**), or completed tickets (**UC-02.3**) and receive real-time updates (**UC-02.4**). Unpaid ticket management includes changing pick-up/drop-off points (**UC-03.1**) or transit locations (**UC-03.2**) and viewing payment guidelines (**UC-03.3**). The cancellation process (**UC-04**) allows customers to select tickets to cancel (**UC-04.1**), view cancellation regulations (**UC-04.2**), provide a reason for cancellation (**UC-04.3**), and confirm the cancellation (**UC-04.4**). This flow ensures comprehensive ticket management for users.

Use case: View Paid ticket

Table 5.32. Use Case Description: View Paid ticket

Name of Use Case	View Paid Ticket
Actors	Customer
Description	This use case describes the process by which a user reviews the list of paid tickets.
Flow of Events	This use case describes the process by which a user reviews the list of paid tickets.
Successful Completion	The list of paid tickets is successfully displayed to the user.
Alternative	If the internet connection is unavailable, an error message is shown, prompting the user to reconnect and retry. If no paid tickets exist, the system displays a message indicating that there are no tickets to show.
Precondition(s)	The user's device is connected to the Internet. The user has at least one paid ticket in the database.

Trigger	The user clicks on the “Paid Ticket” button.
Postcondition(s)	The system displays the list of paid tickets.
Assumptions	<p>The user is logged in and has the necessary credentials to view their paid tickets.</p> <p>The system has all the required data and functionality to retrieve and display the ticket list.</p> <p>Internet connectivity is stable during the process.</p>

Use case: View upcoming ticket

Table 5.33. Use Case Description: View upcoming ticket

Name of Use Case	View Upcoming Ticket
Actors	Customer
Description	This use case describes the process by which a user views details of an upcoming ticket.
Flow of Events	<p>The user selects an upcoming ticket by clicking on it.</p> <p>The selected forthcoming ticket's details are retrieved and displayed by the system.</p>
Successful Completion	The user is successfully presented with the selected forthcoming ticket's details.
Alternative	<p>The user is prompted to reconnect and try again if an error message indicates that the internet connection is not accessible.</p> <p>The system alerts the user that the ticket cannot be shown if the ticket data are not available.</p>
Precondition(s)	The Internet is linked to the user's device.
Trigger	The user selects a certain forthcoming

	ticket by clicking on it.
Postcondition(s)	The selected forthcoming ticket's details are shown.
Assumptions	<p>The user can view their forthcoming tickets because they are logged in and have proper credentials.</p> <p>All the information needed to retrieve and show the ticket details is in the system.</p> <p>Throughout the process, there is consistent internet connectivity.</p>

Use case: View Ongoing ticket

Table 5.34. Use Case Description: View Ongoing ticket

Name of Use Case	View Ongoing Ticket
Actors	Customer
Description	The procedure by which a user examines the specifics of an open ticket is explained in this use case.
Flow of Events	The user selects a certain open ticket. The selected continuing ticket's details are retrieved and displayed by the system.
Successful Completion	The user is successfully presented with the specified ongoing ticket's details.
Alternative	The user is prompted to reconnect and try again if an error message indicates that the internet connection is not accessible. The system alerts the user that the ticket cannot be shown if the ticket data are not available.
Precondition(s)	The Internet is linked to the user's device.
Trigger	The user selects a particular open ticket.
Postcondition(s)	The user selects a particular open ticket.
Assumptions	To view their open tickets, the user must

	be logged in and have valid credentials. All the information needed to retrieve and show the ticket details is in the system. Throughout the process, there is consistent internet connectivity.
--	--

Use case: Update real-time information

Table 5.35. Use Case Description: Update real-time information

Name of Use Case	Update Real-Time Information
Actors	Admin
Description	The method by which an administrator updates information in real-time to notify clients of current details is the subject of this use case.
Flow of Events	The administrator modifies and refreshes the current data. The change is processed by the system, which guarantees that the real-time data is successfully updated and modified.
Successful Completion	The real-time data is successfully updated to give clients the most recent information.
Alternative	The system alerts the administrator of the unsuccessful update and requests a retry if the internet connection is not accessible. - If a system fault occurs, the administrator is alerted so they can either manually fix the problem or try again later.
Precondition(s)	The administrator's device has an Internet connection, and the user is linked to at least one open ticket.
Trigger	The real-time information updating is started by the administrator.
Postcondition(s)	The most recent information for clients is reflected in the successfully updated real-time information.

Assumptions	The administrator has the right credentials and rights to update data in real time. Updates can be processed in real time by the system. Throughout the process, there is consistent internet connectivity.
--------------------	---

Use case: View Complete ticket

Table 5.36. Use Case Description: View Complete ticket

Name of Use Case	View Complete ticket
Actors	Customer
Description	This use case describes the process by which a user views the details of a complete ticket.
Flow of Events	The user clicks on a specific complete ticket. The system retrieves and displays the details of the selected complete ticket.
Successful Completion	The details of the selected complete ticket are successfully displayed to the user.
Alternative	The user is prompted to reconnect and try again if an error message indicates that the internet connection is not accessible. The system alerts the user that the ticket cannot be shown if the ticket data are not available.
Precondition(s)	The Internet is linked to the user's device.
Trigger	The user selects a particular full ticket.
Postcondition(s)	The selected complete ticket's details are shown.
Assumptions	The user can read all of their tickets because they are logged in and have valid credentials. All the information needed to retrieve and

	show the ticket details is in the system. Throughout the process, there is consistent internet connectivity.
--	--

Use case: View Unpaid ticket

Table 5.37. Use Case Description: View Unpaid ticket

Name of Use Case	View Unpaid Ticket
Actors	Customer
Description	The procedure by which a user examines the list of unpaid tickets is explained in this use case.
Flow of Events	The Booking History panel is accessed by the user. The default list of unpaid tickets appears on the Booking History screen.
Successful Completion	The user is successfully presented with the list of unpaid tickets.
Alternative	The user is prompted to reconnect and try again if an error message indicates that the internet connection is not accessible. - The system indicates that there are no tickets to show if there are no unpaid tickets.
Precondition(s)	The Internet is linked to the user's device.
Trigger	The "Unpaid Ticket" button is clicked by the user.
Postcondition(s)	The list of tickets that have not been paid for is shown.
Assumptions	The user can check their booking history because they are logged in and have valid credentials. All the information needed to retrieve and show the ticket list is already in the system.

	Throughout the process, there is consistent internet connectivity.
--	--

Use case: View payment guidelines

Table 5.38. Use Case Description: View payment guidelines

Name of Use Case	View Payment Guidelines
Actors	Customer
Description	This use case describes the process by which a user views payment guidelines before making a payment for a ticket
Flow of Events	<ol style="list-style-type: none"> 1. An unpaid ticket is clicked by the user. 2. The unpaid ticket's details are shown by the system. 3. To make a payment, the user selects "Pay Now." 4. The payment instructions are shown by the system
Successful Completion	The user is successfully presented with the payment instructions for the chosen ticket.
Alternative	The user is prompted to reconnect and try again if an error message indicates that the internet connection is not accessible. The user is notified by the system that the information cannot be presented if there are no payment instructions available.
Precondition(s)	The user's device has an Internet connection, and the user is linked to at least one unpaid ticket.
Trigger	The user selects a particular unpaid ticket by clicking on it.
Postcondition(s)	The chosen ticket's payment instructions are shown.
Assumptions	The user can check their unpaid tickets

	<p>because they are logged in and have valid credentials.</p> <p>All the information needed to access and show the payment guidelines is already in the system.</p> <p>Throughout the process, there is consistent internet connectivity.</p>
--	---

Use case: Change pick-up/drop-off point

Table 5.39. Use Case Description: Change pick-up/drop-off point

Name of Use Case	Change Pick-up/Drop-off Point
Actors	Customer
Description	The procedure by which a user modifies the pick-up or drop-off location for an unpaid ticket is explained in this use case.
Flow of Events	<ol style="list-style-type: none"> 1. An unpaid ticket is clicked by the user. 2. The unpaid ticket's details are shown by the system. 3. The "Change pick-up/drop-off point" option is chosen by the user. 4. The list of pick-up and drop-off locations is shown by the system. 5. The user saves modifications after choosing a new pick-up and drop-off location. 6. The system shows the selected ticket's most recent details.
Successful Completion	The chosen unpaid ticket's pick-up or drop-off location is successfully changed and shown.
Alternative	<p>The user is prompted to reconnect and try again if an error message indicates that the internet connection is not accessible.</p> <p>The user is prompted to choose a</p>

	different pick-up/drop-off location if the system determines that the new point is invalid or unavailable.
Precondition(s)	The user's device has an Internet connection, and the user is linked to at least one unpaid ticket.
Trigger	The user selects a particular unpaid ticket by clicking on it.
Postcondition(s)	The chosen unpaid ticket's pick-up or drop-off location has been successfully updated.
Assumptions	To make changes to their ticket, the user must be logged in and possess valid credentials. A database of authorized pick-up and drop-off locations is available through the system. Throughout the process, there is consistent internet

Use case: Change transit location

Table 5.40. Use Case Description: Change transit location

Name of Use Case	Change Transit Location
Actors	Customer
Description	The procedure by which a user modifies the transit location for an unpaid ticket is explained in this use case.
Flow of Events	<ol style="list-style-type: none"> 1. An unpaid ticket is clicked by the user. 2. The unpaid ticket's details are shown by the system. 3. The user chooses "Edit" for the location of pickup and drop-off. 4. The edit pickup/dropoff location screen is displayed by the system. 5. To modify the transportation location, the user chooses "Edit."

	<p>6. To save changes, the user enters the new transportation location and clicks "Confirm."</p> <p>7. The system shows the selected ticket's most recent details.</p>
Successful Completion	The chosen unpaid ticket's transit location has been successfully updated and shown.
Alternative	<p>The user is prompted to reconnect and try again if an error message indicates that the internet connection is not accessible.</p> <p>- The system alerts the user and asks them to enter a new transit location if the current one is invalid or unavailable.</p>
Precondition(s)	The user's device has an Internet connection, and the user is linked to at least one unpaid ticket.
Trigger	The user selects a particular unpaid ticket by clicking on it
Postcondition(s)	The chosen unpaid ticket's transportation location has been successfully updated.
Assumptions	<p>To make changes to their ticket, the user must be logged in and possess valid credentials.</p> <p>The system allows transportation locations to be changed.</p> <p>Throughout the process, there is consistent internet connectivity.</p>

Use case: View Cancellation

Table 5.41. Use Case Description: View Cancellation

Name of Use Case	View Cancellation
Actors	Customer
Description	The procedure by which a user examines the list of canceled tickets is explained in this use case.
Flow of Events	<ol style="list-style-type: none"> 1. The "Cancellation" button is clicked by the user. 2. The list of canceled tickets is shown by the system.
Successful Completion	The user is successfully presented with the list of canceled tickets.
Alternative	<p>The user is prompted to reconnect and try again if an error message indicates that the internet connection is not accessible.</p> <p>The system indicates that there are no cancellations if there are no canceled tickets.</p>
Precondition(s)	<p>The Internet is linked to the user's device.</p> <p>The user has at least one ticket linked to them.</p>
Trigger	The "Cancellation" button is clicked by the user.
Postcondition(s)	The canceled ticket list is shown.
Assumptions	<p>In order to access their ticket cancellations, the user must be logged in and possess valid credentials.</p> <p>Accurate and current cancellation information is available in the system.</p> <p>Throughout the process, there is consistent internet connectivity.</p>

Use case: Select tickets to cancel

Table 5.42. Use Case Description: Select tickets to cancel

Name of Use Case	Select Tickets to Cancel
Actors	Customer
Description	The procedure by which a user chooses which tickets to cancel is explained in this use case.
Flow of Events	<ol style="list-style-type: none"> 1. The "Booking History" page is displayed to the user. 2. The system shows a list of tickets together with the booking history. 3. The user chooses which particular ticket to cancel. 4. The selected ticket's details are shown by the system. 5. The "Cancel Ticket" button is clicked by the user. 6. A confirmation prompt for the cancellation request is shown by the system. 7. By selecting the "Confirm" option, the user validates the cancellation.
Successful Completion	The selected tickets are marked for cancellation and the user is notified.
Alternative	<p>The user is prompted to reconnect and try again if an error message indicates that the internet connection is not accessible.</p> <ul style="list-style-type: none"> - An appropriate error message is displayed by the system if the ticket cannot be canceled.
Precondition(s)	<p>The Internet is linked to the user's device.</p> <p>The user has at least one ticket linked to them.</p>
Trigger	The ticket is canceled by the user.
Postcondition(s)	The chosen tickets have been marked as canceled.

Assumptions	<p>The individual has legitimate login credentials and is able to view and edit their tickets.</p> <p>The system has the ability to cancel tickets.</p> <p>Throughout the process, there is consistent internet connectivity.</p>
--------------------	---

Use case: View cancel regulations

Table 5.43. Use Case Description: View cancel regulations

Name of Use Case	View Cancel Regulations
Actors	Customer
Description	In this use case, a consumer examines the cancellation policies prior to confirming a cancellation.
Flow of Events	<ol style="list-style-type: none"> 1. To cancel, the client clicks the "Confirm" button. 2. The system shows the specifics of the cancellation policies.
Successful Completion	The cancellation policies are successfully shown for the client to examine.
Alternative	The user is prompted to reconnect and try again if an error message indicates that the internet connection is not accessible.
Precondition(s)	The Internet is linked to the user's device. The user has at least one ticket linked to them.
Trigger	To cancel, the client clicks the "Confirm" button.

Postcondition(s)	The details of cancellation regulations are displayed.
Assumptions	To view their ticket details, the user must be logged in and possess valid credentials. The system has up-to-date, reliable information on cancellation regulations. Throughout the process, there is consistent internet connectivity.

Use case: Select reason to cancel

Table 5.44. Use Case Description: Select reason to cancel

Name of Use Case	Select Reason to Cancel
Actors	Customer
Description	The procedure where a customer chooses a cancellation reason before confirming it is explained in this use case.
Flow of Events	<ol style="list-style-type: none"> 1. The customer is prompted by the system to choose a cancellation reason. 2. The client chooses the relevant explanation from the list of possibilities (e.g., "Change of plans," "Found a better alternative").
Successful Completion	The system correctly records the chosen cancellation reason.
Alternative	The consumer is prompted to try again or get in touch with assistance if the system is unable to load the cancellation reasons.

Precondition(s)	The user's gadget has an Internet connection. The user's account is linked to at least one ticket.
Trigger	After reading the cancellation policies, the client chooses a reason for the cancellation.
Postcondition(s)	The system logs the reason for cancellation that was chosen.
Assumptions	<p>The individual has legitimate login credentials and is able to access tickets.</p> <p>The system provides current and accurate cancellation reasons.</p> <p>Throughout the procedure, there is steady internet connectivity.</p>

Use case: Confirm Cancellation

Table 5.45. Use Case Description: Confirm Cancellation

Name of Use Case	Confirm Cancellation
Actors	Customer
Description	This use case describes how a customer verifies that a ticket has been canceled.
Flow of Events	<ol style="list-style-type: none"> 1. The customer is prompted by the system to confirm that the chosen ticket has been canceled. 2. By clicking the "Confirm" button, the client validates the cancellation. 3. The system guarantees a refund and shows a notification that the cancellation procedure is being processed.

Successful Completion	The customer is guaranteed a refund after the system successfully completes the cancellation request.
Alternative	The customer is informed and instructed to try again or get in touch with assistance if the system finds an error during the cancellation process.
Precondition(s)	The Internet is connected to the user's device. The user has at least one ticket linked to them.
Trigger	The client decides to move on with the process after selecting a ticket for cancellation.
Postcondition(s)	The cancellation request has been verified, recorded, and handled.
Assumptions	The user's account and related tickets are active. Requests for cancellations can be accurately processed by the system. Throughout the procedure, a steady internet connection is maintained.

5.5.6. Support and Feedback

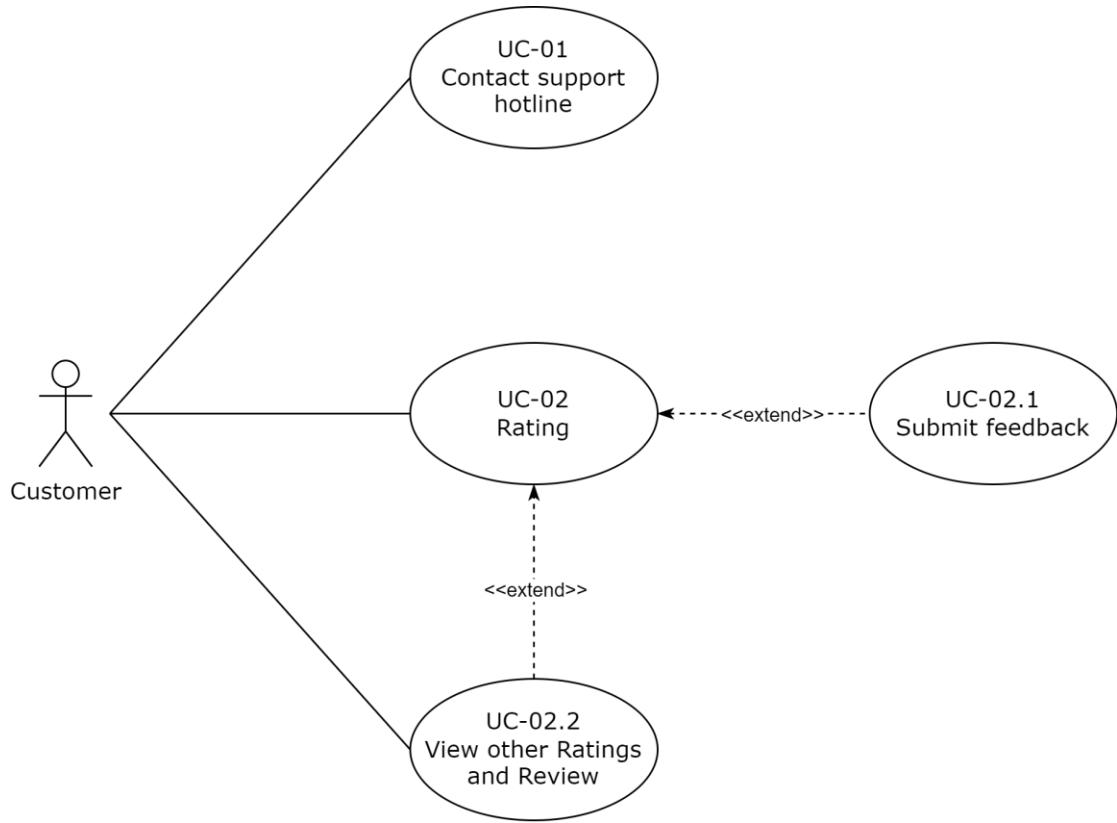


Figure 5.15. Use case of Support and Feedback

(Sources: Author's Sources)

Customers can contact the support hotline (**UC-01**) for assistance or engage with the rating and feedback system (**UC-02**). The rating use case includes submitting feedback (**UC-02.1**) and viewing other customer ratings and reviews (**UC-02.2**), creating a loop of feedback and transparency. This flow ensures customers have access to support and can share their experiences to improve service quality.

Use case: Contact support hotline

- Overall:

Table 5.46. Use Case Description: Contact support hotline

Name of Use case	Contact support hotline
Actors	Customer
Description	This use case outlines the procedure a customer follows when contacting the hotline for assistance .
Precondition(s)	The user's device establishes an Internet connection.
Trigger	The user clicks on the "Call Hotline" button.
Postcondition(s)	The customer is connected to the hotline support.

- **Flow of Events:**

Step	Actor	Action	Note/ References
Basic flow			
BF-1	Customer	Go to the "Account" tab.	
BF-2	System	Display the "Account" screen.	
BF-3	Customer	The user clicks on the "Call Hotline" button.	
BF-4	System	Initiate the call to the hotline for customer support.	

Use case: Rating

- **Overall:**

Table 5.47. Use Case Description: Rating

Name of Use case	Rating
Actors	Customer
Precondition(s)	<ul style="list-style-type: none"> - The user's device connects to the Internet. - The user account must have at least one complete ticket linked to it.
Trigger	The user clicks on the “Rating” button.
Postcondition(s)	The rating is recorded.

- **Flow of Events:**

Table. Flow of Events: Rating

Step	Actors	Action	Note/ References
Basic flow			
BF-1	Customer	Navigate to the “History Ticket” tab.	
BF-2	System	Show the “History ticket” screen.	
BF-3	Customer	Tap the “Paid ticket” button.	
BF-4	System	Show the “Paid ticket” screen.	
BF-5	Customer	Select the specific paid ticket with a completed status.	

BF-6	Customer	Tap the “Rating” button.	
BF-7	System	Show the “Rating” screen.	
BF-8	Customer	Choose the rating star for the selected ticket.	

Use case: Submit feedback

- Overall:

Table 5.48. Use Case Description: Submit feedback

Name of Use case	Submit feedback
Actors	Customer
Description	This use case outlines the process of a customer submitting reviews after giving a rating.
Precondition(s)	<ul style="list-style-type: none"> - The user’s device connects to the Internet. - Rating has been provided.
Trigger	Submitting feedback after providing a rating.
Postcondition(s)	The reviews are successfully submitted.

- Flow of Events:

Step	Actors	Action	Note/ References
Basic flow			

BF-1	System	Asked to provide reviews after giving a rating.	
BF-2	System	Show the review input screen.	
BF-3	Customer	Complete the review input form.	
BF-4	System	Display the successful rating and review message.	

Use case: View other Ratings and Review

- Overall:

Table 5.49. Use Case Description: View other Ratings and Review

Name of Use case	View other Ratings and Review
Actors	Customer
Description	This use case outlines how a customer views ratings and feedback submitted by other users.
Precondition(s)	The user's device connects to the Internet.
Trigger	Go to the section to view ratings and feedback from other users.
Postcondition(s)	Ratings and feedback from other users are shown for the customer to see.

- Flow of Events:

Table. Flow of Events: View other Ratings and Review

Step	Actors	Action	Note/ References

Basic flow			
BF-1	Customer	Go to the “Homepage” screen.	
BF-2	System	Show the "Homepage" screen, which includes a list of other ratings and feedback.	
Alternative Flow			
AF-1.1	System	Go to the Result List.	
AF-1.2	Customer	Click on the "Details" button for a specific trip.	
AF-1.3	System	Show the details of the bus trip.	
AF-1.4	Customer	Select the Ratings and Reviews tab.	
AF-1.5	System	Show the Ratings and Reviews tab.	

5.5.7. Notifications

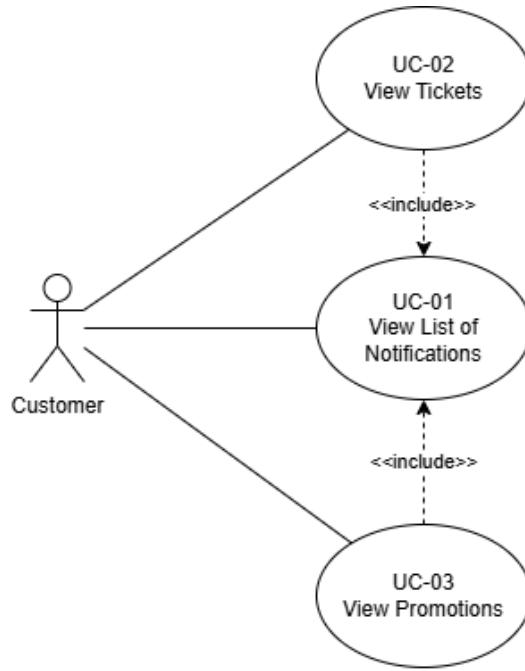


Figure 5.16. Use case of Notifications

(Sources: Author's Sources)

Customers can view their tickets (**UC-02**), check a list of notifications for updates or alerts (**UC-01**), and explore current promotions (**UC-03**). The inclusion relationships indicate that notifications and promotions are supplementary to the ticket viewing process, enhancing the user experience with additional relevant information.

UC-01: View List of Notifications

- Overall:

Table 5.50. Use Case Detail - Description: Module View List of Notifications

Module	View List of Notifications.
Description	This use case describes how a user views a list of notifications on the website.

Actors	Customer
Precondition(s)	The user's device connects to the Internet.
Trigger	Click on the "Notifications" icon on the taskbar.
Postcondition(s)	The website shows a list of notifications.

- **Flow of Events:**

Table 5.51. Flow of Events: Module View List of Notifications

Step	Actor	Action	Note/References
Flow of Events			
BF-1	System	Send notifications to the customer.	
BF-2	Customer	Click on the "Notifications" icon on the taskbar.	
BF-3	System	Display the "Notification" Screen.	
BF-4	Customer	View the notifications.	

UC-02: View Tickets

- **Overall:**

Table 5.52. Use Case Detail - Description: Module View Tickets

Module	View tickets in the list of Notifications.
Description	This use case describes how a user views their ticket in a list of notifications on the website to stay informed of the latest ticket information.

Actors	Customer
Precondition(s)	The user's device connects to the Internet.
Trigger	Click on the "Notifications" icon on the taskbar.
Postcondition(s)	The website shows the details of the booked ticket to the passenger.

- **Flow of Events:**

Table 5.53. Flow of Events: Module View Tickets

Step	Actor	Action	Note/References
Flow of Events			
BF-1	Customer	Click on the "Notifications" icon on the taskbar.	
BF-2	System	Display the "Notification" Screen.	
BF-3	Customer	View the notifications of booked-tickets.	

UC-03: View Promotions

- **Overall:**

Table 5.54. Use Case Detail - Description: Module View promotions

Module	View promotions in the list of Notifications
Description	This use case describes how a user views their promotions in a list of notifications on the website to stay updated with the latest information about the received promotions.
Actors	Customer

Precondition(s)	The user's device connects to the Internet.
Trigger	Click on the “Notifications” icon on the taskbar.
Postcondition(s)	The website shows the details of the promotions to the passenger.

- **Flow of Events:**

Table 5.55. Flow of Events: Module View promotions

Step	Actor	Action	Note/ References
Flow of Events			
BF-1	Customer	Click on the “Notifications” icon on the taskbar.	
BF-2	System	Display the “Notification” Screen.	
BF-3	Customer	View the notifications of promotion.	

5.6. Sequence Diagram

5.6.1. Deliverables

Table 5.56. Deliverables

Section	Process Name	Description	Purpose
1	Account Management	Managing user accounts including login, registration, and updates	Maintain user information securely
2	Booking	Booking a bus, choosing seats, and processing payments	Facilitate bus reservation process

3	Transactions	Payment processing with third-party services	Handle transactions securely
4	Support and Feedback	User support and feedback interactions	Enhance user experience with support services
5	Notifications	Sending notifications to users	Keep users informed about bookings and updates

5.6.2. Account management

This sequence diagram illustrates the login and registration processes, essential for user security and experience. The login starts when a user clicks the login button, enters their username and password, and the system checks these credentials against its database. If correct, the user is logged in successfully; if not, an error is shown. For registration, the user fills out a form, and the system checks if the username is available. If so, an OTP (One-Time Password) is sent to the user for verification. Upon entering the valid OTP, the registration is completed, and the user can log in.

Both processes protect user data and enhance security, providing a smooth and secure user experience.

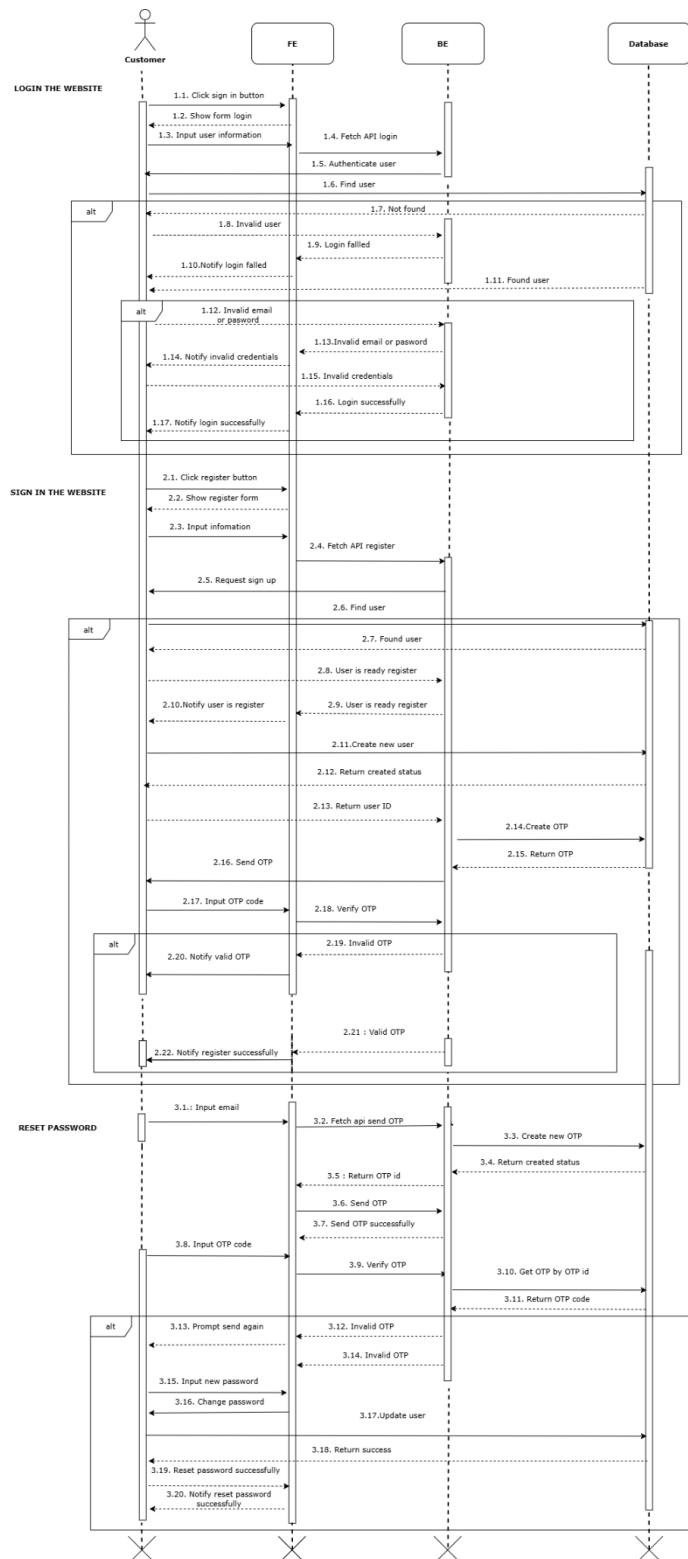


Figure 5.17. Sequence diagram of Account management

(Sources: Authors' Source)

5.6.3. Booking

This sequence diagram shows the booking and registration flow for a customer on a platform, detailing steps for finding locations, scheduling, registration, and finalizing a booking. The customer searches for a location, selects services, and checks schedule availability. The system displays the location list, service types, and schedules. After selecting a date and time, the system confirms availability. If the customer is not registered, they fill out a registration form. The system verifies the information and confirms registration, allowing the customer to proceed. The customer selects their preferred time slot, inputs details, and reviews their information. The system verifies all data, confirms the final selection, and completes the booking.

Each step ensures accuracy and guides the customer from browsing services to successfully making a booking, enhancing the user experience.

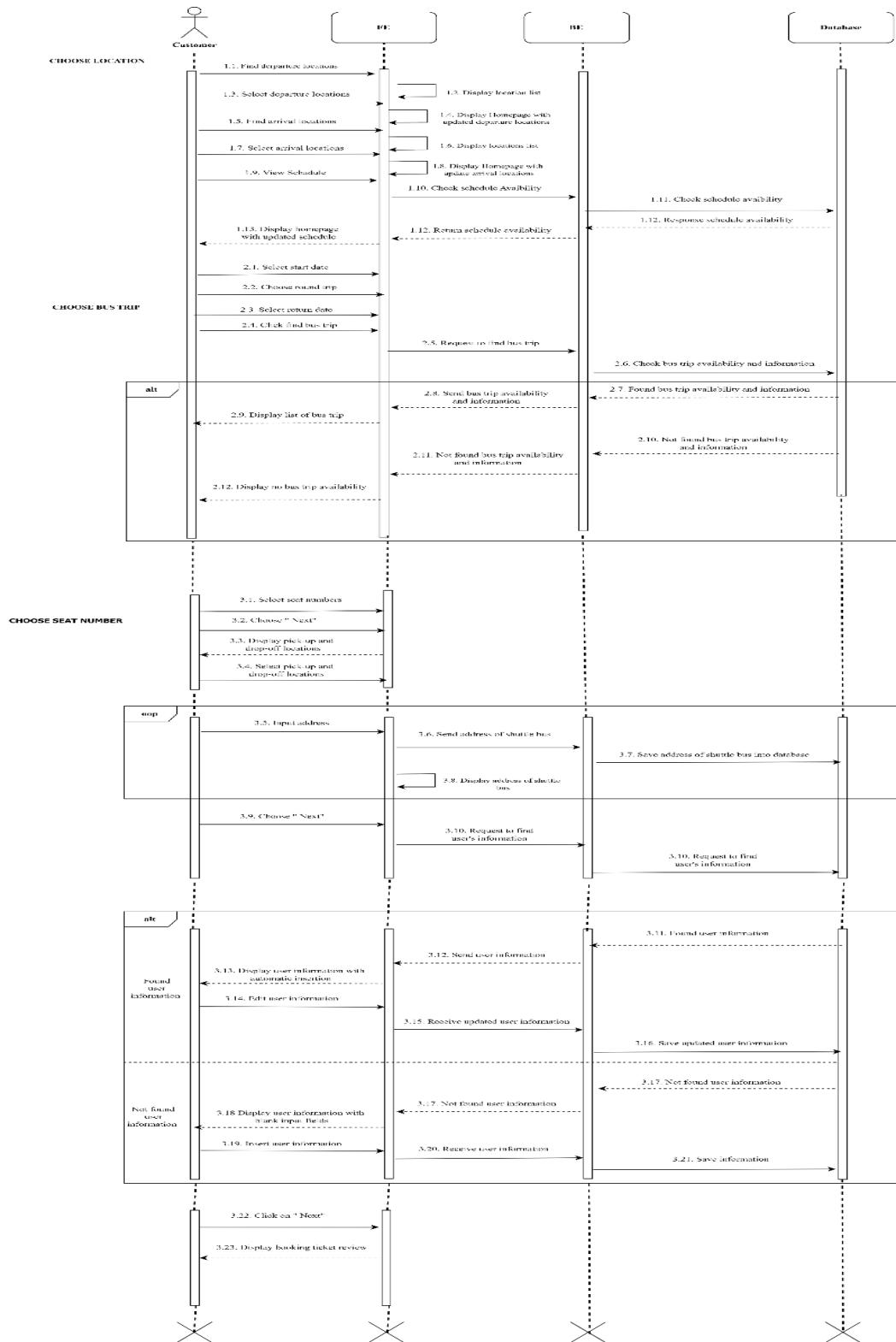


Figure 5.18. Sequence diagram of Booking

(Sources: Authors' Source)

5.6.4. Transactions

The diagram illustrates the sequence of interactions involved in processing transactions within the bus booking system. It shows the flow from the customer selecting a payment method to the validation of vouchers, and ultimately to the successful or unsuccessful payment status. Key elements include interactions between the front end (FE), back end (BE), database (DB), and external systems like banking or e-wallet services. The diagram ensures that each step in the transaction process, including selection, verification, and payment completion, is clearly represented.

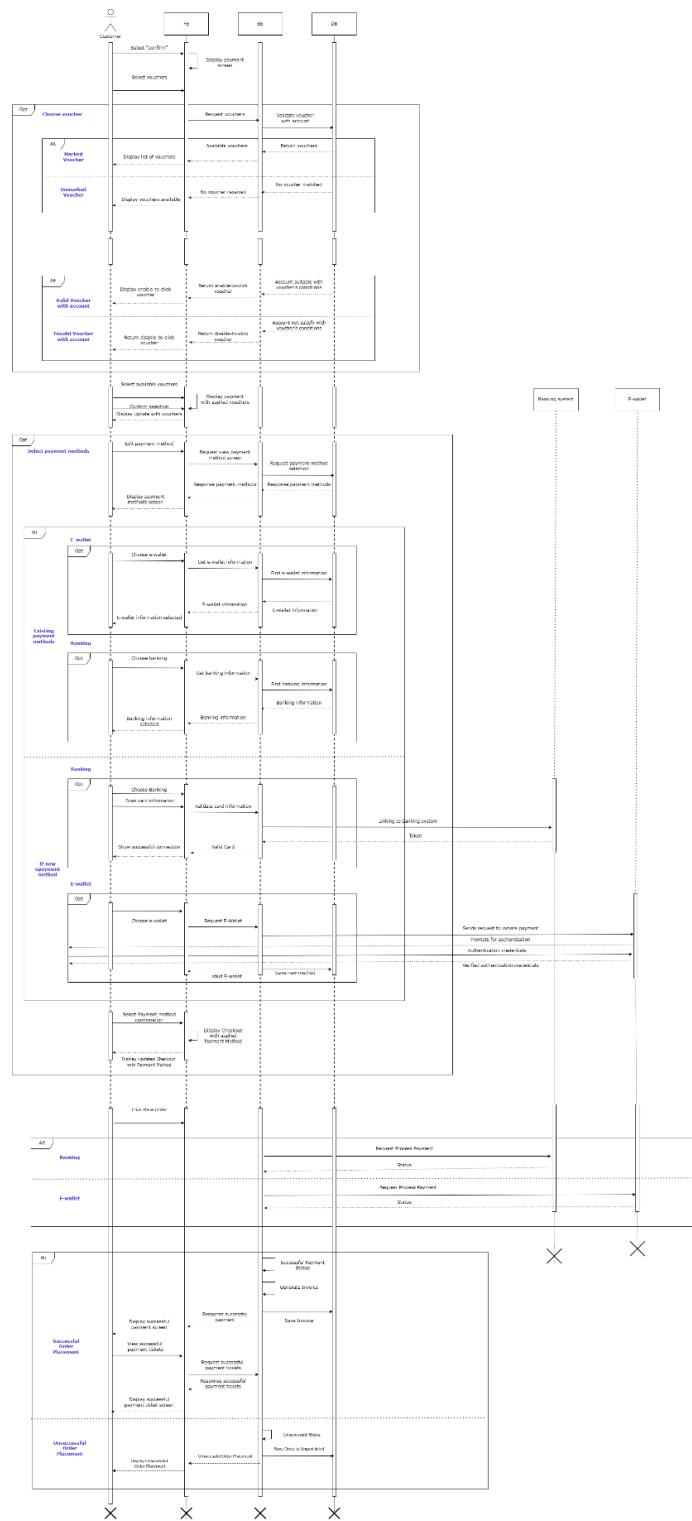


Figure 5.19. Sequence diagram of Transactions

(Sources: Authors' Source)

5.6.5. Booking History

The diagram depicts the user flow for managing ticket history in a bus booking system, covering Unpaid, Paid, and Canceled Tickets.

- **Unpaid Tickets:** Users can view unpaid tickets, which remain accessible for 15 minutes before automatic cancellation. While unpaid, users can modify details like pick-up, drop-off, and transit locations, with changes updated in the database.
- **Paid Tickets:** Organized into Upcoming, Ongoing, and Completed categories, tickets update in real-time as trips progress—Upcoming for future travel, Ongoing for active trips, and Completed for concluded journeys.
- **Cancellation Process:** Users initiate cancellation by reviewing and accepting terms, providing a reason, and confirming the action. The back end updates the ticket status to "Canceled," and users receive a confirmation.

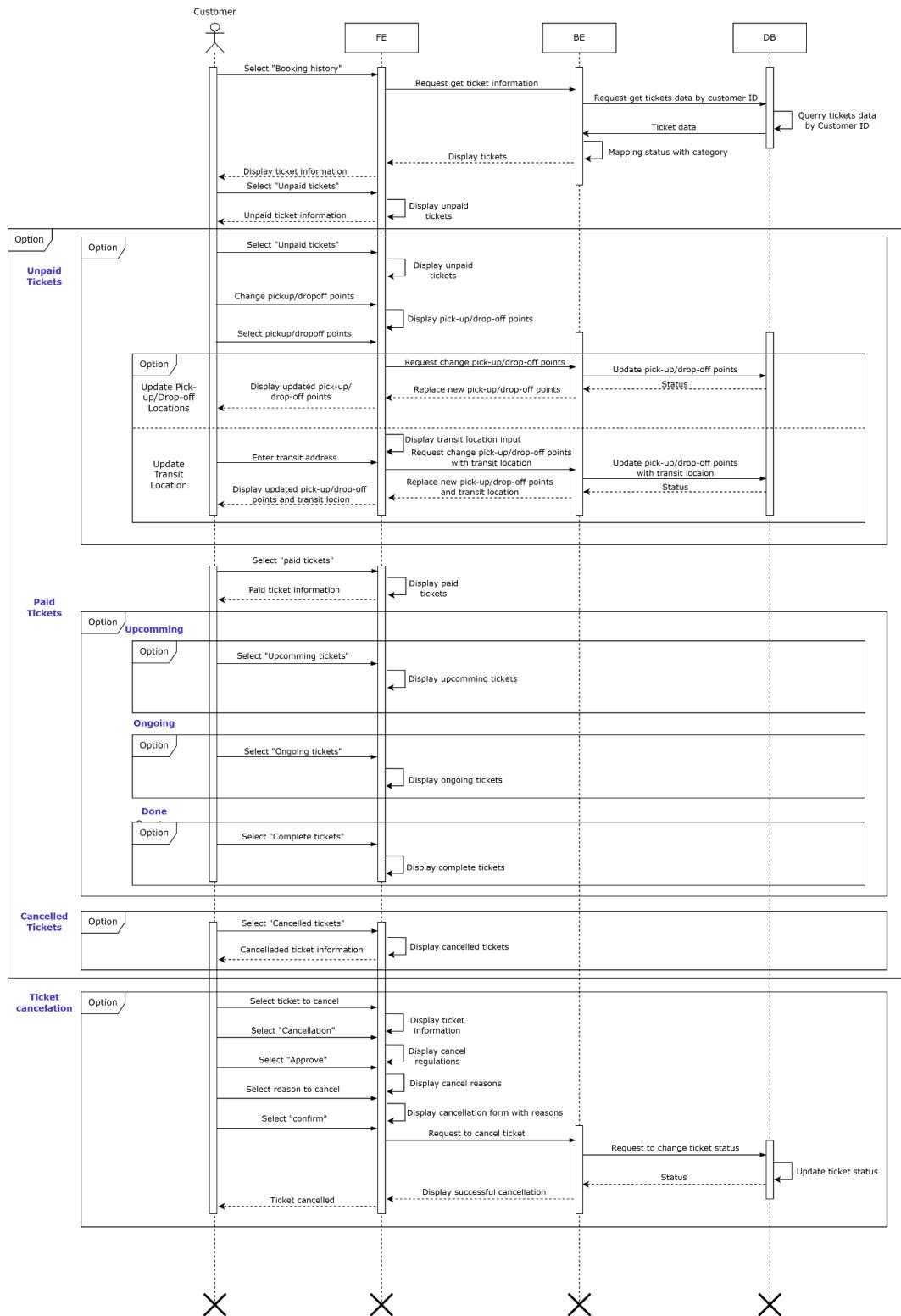


Figure 5.20. Sequence diagram of Booking History

(Sources: Authors' Source)

5.6.6. Support and Feedback

After a successful trip, customers can share their experience by providing ratings and feedback through the Ticket History section. The system will first check for any existing tickets. If a completed ticket is found in the database, the system displays it, allowing customers to select "View Completed Tickets" to submit their feedback and ratings for the trip. Once submitted, this information is updated in the database, ensuring accurate record-keeping of customer experiences.

For support, customers can easily reach the hotline by accessing the Account tab and clicking on the hotline number. If they confirm to make the call, they can select the Help Center's phone number on the Account Management screen, and the system will connect them to their telecom provider, displaying the call interface. If they decide against calling, they can choose the cancel option, which will redirect them back to the Account Management screen.

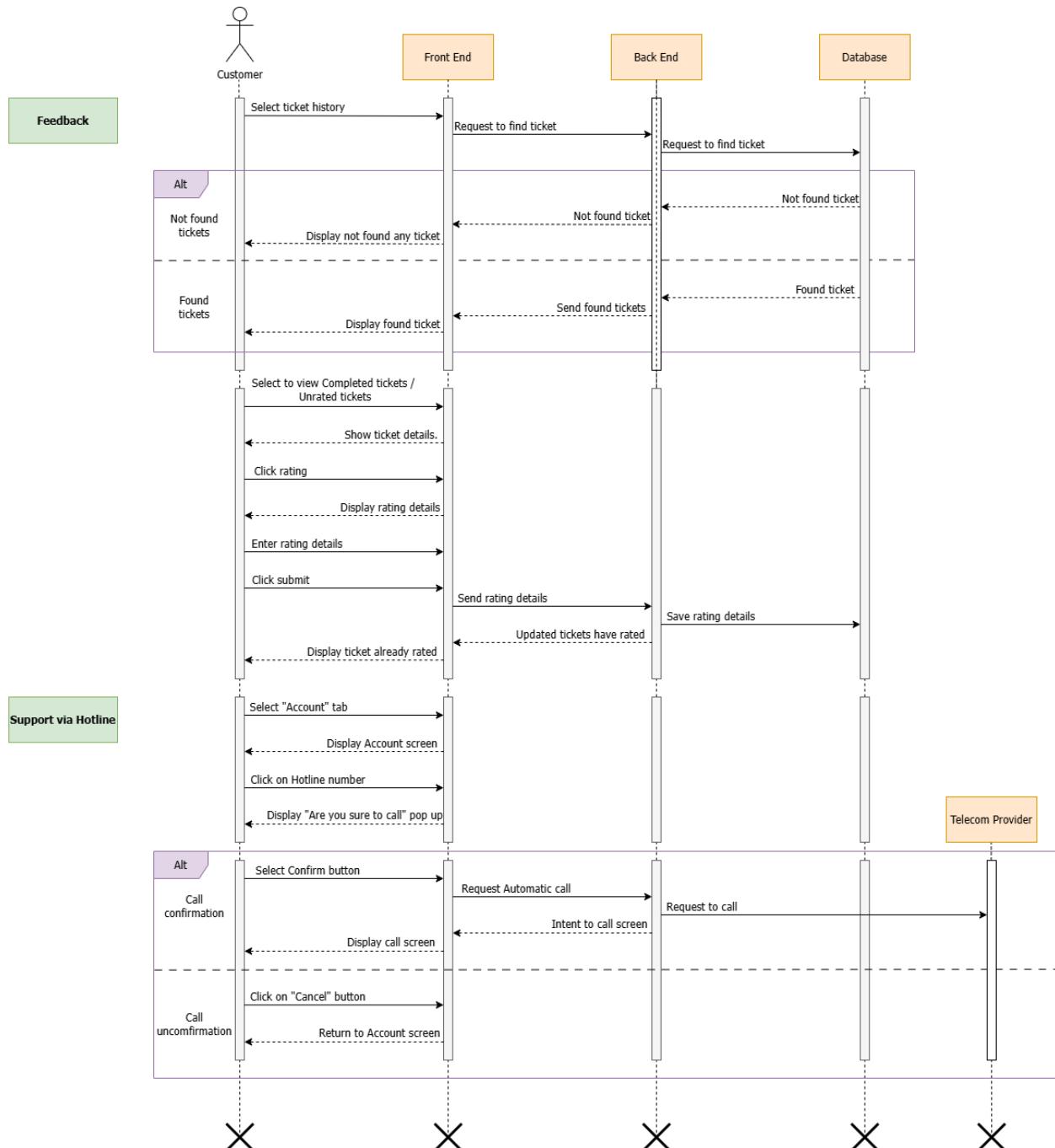


Figure 5.21. Sequence diagram of Support & Feedback

(Sources: Authors' Source)

5.6.7. Notifications

After logging in, customers can access Notifications to view updates, with the system retrieving and displaying them from the database.

View List of Notifications: The *Customer* selects the "Notification" tab, prompting the *Front End* (FE) to request notifications from the *Back End* (BE), which then queries the *Database* (DB).

Alternative Scenarios:

- **If no notifications are found**, the *DB* responds to the *BE* and *FE*, which shows "Not found notifications" to the *Customer*.
- **If notifications are found**, the *DB* sends them to the *BE* and *FE*, which displays the list to the *Customer*.

View Details of Notifications: The *Customer* selects a notification to view details, and the *FE* shows the specific notification information.

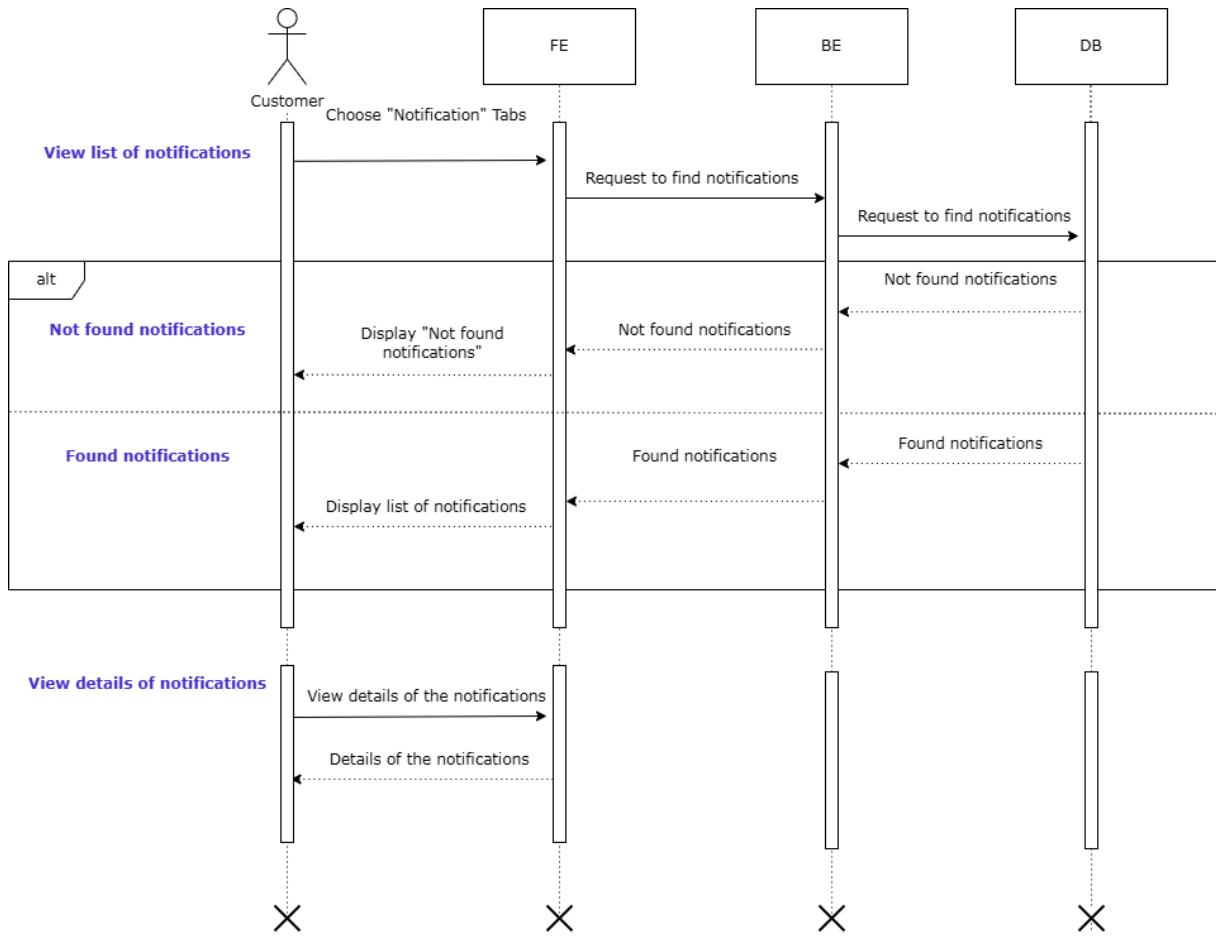


Figure 5.22. Sequence diagram of Notifications

(Sources: Authors' Sources)

5.7. Business Rule

Table 5.57. Business Rule

Rule ID	Rule Title	Mandatory	Rule Description
Account Management			
BR-01	Phone Number Format	x	The phone number must be unique, consist of exactly 10 digits, and begin with a 0.
BR-02	OTP Code	x	The OTP code should be a 4-digit number, entered from the SMS provided by the telecom provider, and is valid for only 60 seconds.
BR-03	Verification Code	x	The verification code must be a 4-digit number, entered from the email sent to the registered email address by the email provider.
BR-04	Password	x	The password must be a minimum of 8 characters and should not match the registered phone number or email address.
BR-05	Confirm Password	x	The password must be a minimum of 8 characters and the confirm password field must match the password exactly.
BR-06	Personal Information	x	Personal information should include: <ul style="list-style-type: none"> • Name • Email • Phone Number
BR-07	Name	x	The name must have at least 1 character.

BR-08	Email Address		<p>The general email format should accommodate two types: educational and conventional emails. For educational emails, the format is local-part@edu.vn, while for conventional emails, it is local-part@domain (e.g., duyhd@example.com).</p>
BR-09	Address		House Number, Street Name, Ward, City and District are all placed.
Ticket Booking			
BR-10	Bus Searching	x	<p>The Bus Search section should contain:</p> <ul style="list-style-type: none"> ● Departure Location ● Arrival Location ● Departure/Start Date ● Round Trip Option ● Return Date ● CTA Button to search for bus results <p>Users must proceed step-by-step in the above order before moving to the next field.</p> <p>The Round Trip Option must be enabled before selecting a Return Date, at which point the Return Trip Section will appear.</p>
BR-11	Result List	x	<p>The Result List should display bus routes according to:</p> <ul style="list-style-type: none"> ● Ordered from the earliest to the latest departure times. ● Sorted from the lowest to the highest prices.

BR-12	Filter	x	<p>The Filter section should include:</p> <ul style="list-style-type: none"> • Departure Time Range • Price Range • Ratings • Seat Type • Seat Selection • General/Common Filters
BR-13	Sort	x	<p>The Sort section should include the following options, with only one selection allowed at a time:</p> <ul style="list-style-type: none"> • Default • Earliest Departure Time • Latest Departure Time • Most Reviews • Price Decreased • Price Increased
BR-14	Booking Regulations	x	<p>Users are required to accept the Booking Regulations before proceeding with the booking process. This acceptance is only needed once.</p>
BR-15	Departure Time Filter	x	<p>The Departure Time filter should include:</p> <ul style="list-style-type: none"> • Departure Time Range • Radio selection with a single selectable option: Default, Earliest Departure Time, or Latest Departure Time
BR-16	Departure Location	x	<p>City/Province is required.</p>
BR-17	Arrival Location	x	<p>City/Province is required.</p>
BR-18	Start/Departure Date	x	<p>The fields Year, Month, and Date are required and</p>

			should follow the general format “DD/MM/YYYY.”
BR-19	Return Date	x	The fields Year, Month, and Date are required and should follow the general format “DD/MM/YYYY.”
BR-20	Number of Seats	x	Selected seats must be empty and unique. At least one seat must be chosen before proceeding to the next step. Occupied seats should be disabled and unavailable for selection.
BR-21	Pick-up Location	x	<p>The following fields are required:</p> <ul style="list-style-type: none"> ● Departure/Start Time ● Street Name and Number ● Ward ● City ● District <p>Only one location can be selected. If the chosen pick-up location provides a shuttle service, users must enter the transit address.</p>
BR-22	Drop-off Location	x	<p>The following fields are mandatory:</p> <ul style="list-style-type: none"> ● Departure/Start Time ● Street Name ● Street Number ● Ward ● City ● District <p>Only one location may be selected. If the selected drop-off location provides a shuttle service, users must enter the transit address.</p>

BR-23	Passenger Information	x	<p>All registered user information will be automatically populated in the fields.</p> <p>If the fields do not auto-fill and require manual input, users must enter the information manually, after which the "Continue" button will become active.</p>
BR-24	Booking Review	x	<p>The Booking Review section should include the following:</p> <ul style="list-style-type: none"> • Departure Trip details • Return Trip details • Passenger Information <p>Users are permitted to edit the pick-up, drop-off, and transit locations, as well as passenger information.</p>
Payment Transaction			
BR-25	Payment Method	x	<p>Two payment options will be offered to customers:</p> <ul style="list-style-type: none"> • Banking Systems • E-Wallets <p>Customers must select one of these options.</p>
BR-26	Banking Information		<p>The Banking Information section should include the following fields:</p> <p>Card Number: Must be 8 to 16 digits.</p> <p>Issue Date: Must be a future date in the format MM/YY.</p> <p>Cardholder Name: Must consist of alphabetic characters, starting with the issuing bank's name followed by the specific card name.</p> <p>CVV: Must be exactly 3 digits.</p> <p>List of Accepted Banks</p>

BR-27	E-Ticket	x	<p>The E-Ticket should include:</p> <p>Ticket Booking ID Customer's Name and Phone Number Number of Seats and Selected Seat Numbers Departure Location Arrival Location Start/Departure Date Return Date (if applicable) Pick-up Location Drop-off Location</p>
BR-28	Invoice	x	<p>The invoice should include:</p> <ul style="list-style-type: none"> ● Invoice ID ● Ticket Booking ID ● Customer's Name and Phone Number ● Number of Seats and Selected Seat Numbers ● Departure Location ● Arrival Location ● Start/Departure Date ● Return Date (if applicable) ● Pick-up Location ● Drop-off Location ● Payment Method Used ● Price
BR-29	Payment Transaction	x	<p>The payment transaction is limited to a 15-minute duration.</p> <p>If payment is not completed within this timeframe, the system will automatically cancel the booked tickets and update the ticket status from “Unpaid” to</p>

			“Cancellation”
BR-30	Refund		<p>The refund methods for canceled bookings include the following cases:</p> <p>Banking Payment: The refund will be automatically issued to the card used for the original payment.</p> <p>E-Wallet Payment: The refund will be automatically issued to the electronic wallet used for the original payment.</p>
BR-31	Booking Cancellation		<p>Regulations for Ticket Changes or Cancellations:</p> <ol style="list-style-type: none"> 1. Users must agree to the Cancellation Regulations before proceeding with ticket cancellation. 2. Tickets are eligible for only one exchange. 3. Cancellation fees are applied based on the timing of the cancellation: <ul style="list-style-type: none"> • 100% Refund: If canceled more than 24 hours before departure. • 50% Refund: If canceled between 12 and 24 hours before departure. • 30% Refund: If canceled between 6 and 12 hours before departure. • No Refund: If canceled within 6 hours of departure. 4. Customers wishing to change or cancel paid tickets must contact the support center via hotline, email, or text message for further assistance.
Booking History			
BR-32	Booking History	x	The Booking History section should include:

			<ul style="list-style-type: none"> ● Unpaid Tickets ● Paid Tickets ● Cancellations
BR-33	Unpaid Tickets	x	<p>The time remaining to complete payment for unpaid tickets must be displayed.</p> <p>Users are permitted to update any details related to unpaid tickets.</p>
BR-34	Paid Tickets	x	<p>The Paid Tickets section includes the following categories:</p> <ul style="list-style-type: none"> ● Upcoming ● Ongoing ● Complete <p>Users can provide ratings and reviews for tickets in the Complete status.</p> <p>The statuses of Paid Tickets are determined by the following timeframes:</p> <ul style="list-style-type: none"> ● Upcoming: Before the first departure time on the start date. ● Ongoing: <ul style="list-style-type: none"> - Departure Trips: From the first to the last departure time on the departure date. - Return Trips: From the first to the last departure time on the return date. ● Complete: After the entire departure time range has concluded. <p>For tickets in the Upcoming and Ongoing statuses, customers are permitted to view the license plate number, the driver's phone number, and the</p>

			countdown time.
BR-35	Ratings & Review		<p>Only the selection of the number of rating stars is required.</p> <p>The additional fields—add images, add videos, and add comments—are optional.</p>

5.8. Database design

5.8.1 Overview and purpose

In this project, database design provides a foundational structure to support the essential operations of a bus ticket booking application. While the data used is simulated, it accurately represents the type of information that a real-world system would handle, including user details, bookings, payment transactions, and trip schedules. The primary objective of this database is to facilitate a realistic testing environment, allowing developers to evaluate and refine the application's functionality.

By implementing this mock dataset, we can ensure that critical components like user access, booking processes, and data integrity are effectively managed. This approach aids in simulating end-to-end scenarios, providing insights into data interactions and system responses that will inform the development and optimization of a scalable, efficient bus ticketing platform.

5.8.2. Deliverables

The database design deliverables for the bus ticket booking application encompass detailed documentation, diagrams, and implementation resources to ensure that all functional and structural requirements are met. These deliverables include:

Table 5.58. Deliverables

Deliverable	Description
Entity-Relationship Diagram (ERD)	A comprehensive ERD illustrating the relationships among all entities within the database. This diagram will serve as a visual guide to understand data interactions and dependencies.
Data Dictionary	A detailed documentation of each entity, attribute, and relationship in the database, specifying data types, constraints, and descriptions for consistent data handling

	and integration.
Database schema	NoSQL structure representing the finalized design of the database, ready for deployment or migration, covering tables, fields, indexes, and relationships.
Sample data	A set of simulated data entries for each table or collection, allowing for initial testing and validation of system functionalities, such as bookings, payments, and seat management.
Database Queries	A collection of predefined queries for common operations, including ticket booking, cancellations, user profile retrieval, and payment summaries, designed to streamline common transactions and data retrieval tasks.
Backup and Recovery Plan	A strategy outlining procedures for data backup, restoration, and recovery to maintain data integrity and prevent data loss during testing and deployment.
Testing and Validation Reports	Documentation of tests conducted to validate data accuracy, query performance, and integrity constraints, ensuring the database performs reliably under simulated operational conditions.

5.8.3 Database

5.8.3.1. Identify and describe entities

Table 5.59. Identify and describe entities

No .	Entity	Description
1	Account	Acts as the backbone for user profiles, handling access control and personal data management within the system
2	Permission	Regulates system access by assigning unique identifiers and descriptive labels to different authorization levels.
3	Customer	Connects users to their accounts and records customers' specific information, enabling seamless transactions.

4	ViewInvoice	Records financial information related to transactions, offering a clear overview of billing details.
5	Payment Method	Classifies and defines multiple payment methods, enhancing flexibility in financial dealings.
6	Employee	Organizes employee information, aiding in efficient personnel management within the system.
7	Department	Defines organizational units, providing a framework for managing different segments of the system.
8	Manager	Supervises managerial roles, ensuring proper coordination and oversight within the organizational hierarchy.
9	Staff	Tracks skills using unique identifiers for effective evaluation within the system.
10	Driver	Manages transport personnel, linking them to their accounts and associated operational details.
11	Shift	Manages shift details for efficient scheduling and organization, encompassing unique identifiers and corresponding time information.
12	Administrator	Supervises administrative roles, a crucial part in maintaining system security and operational functionality.
13	BookedTicket	Facilitates user bookings, ensuring a smooth and organized process.
14	Order	Records financial details related to ticket bookings, providing a transparent view of transaction data.
15	Voucher	Represents vouchers used for discounts or promotional purposes within the system.
16	Cancellation	Handles cancellation requests and processes efficiently within the system.

17	Bus	Represents the fleet, serving as the core of the transportation network within the system.
18	Seat	Details of the spatial arrangements within buses, contributing to an efficient passenger management system.
19	Bus type	Classifies different seat types, assisting in the spatial arrangement of the transportation system.
20	RouteWithPoints	Outlines routes, providing a structured plan for journeys within the system.
21	Seat Booking	Manages seat reservations and allocations, ensuring systematic seat assignment and occupancy tracking.
22	Pick Drop Point	Represents locations where passengers can be picked up and dropped off along the journey.
23	Pickup Point	Specifies exact locations for passenger pick-ups.
24	Dropoff Point	Specifies exact locations for passenger drop-offs.
25	Amenity	Represents additional amenities or services, such as Wi-Fi or refreshments, enhancing passenger comfort.
26	Feedback	Captures user feedback on their experiences, aiding in customer satisfaction assessment and identifying areas for improvement.
27	Notification	Facilitates communication by sending users important notifications, updates, or alerts regarding bookings, transactions, or system-wide announcements.

5.8.3.2. Define data type

Table 5.60. Data type

Attribute	Data type	Allow Null	Note

Table: Account

AccountID	varchar(10)	<input type="checkbox"/>	PK
AccountUsername	varchar(50)	<input type="checkbox"/>	PhoneNumber, Email
AccountPassword	varchar(50)	<input type="checkbox"/>	
FName	Nvarchar(50)	<input type="checkbox"/>	
LName	Nvarchar(50)	<input checked="" type="checkbox"/>	
PhoneNumber	bigint	<input type="checkbox"/>	
Email	varchar(50)	<input type="checkbox"/>	
AccountType	varchar(1)	<input type="checkbox"/>	C: Customer E: Employee
PermissionID	varchar(10)	<input type="checkbox"/>	FK

Table: Permission

PermissionID	varchar(10)	<input type="checkbox"/>	PK
PermissionDescription	Nvarchar(200)	<input type="checkbox"/>	

Table: Customer

CustomerID	varchar(10)	<input type="checkbox"/>	PK
MembershipStatus	varchar(10)	<input type="checkbox"/>	

Table: ViewInvoice

InvoiceID	varchar(10)	<input type="checkbox"/>	PK
Category	Nvarchar(10)	<input type="checkbox"/>	
DriverName	Nvarchar(50)	<input type="checkbox"/>	
DropoffDescription	Nvarchar(200)	<input type="checkbox"/>	
DropoffTimeDrop	int	<input type="checkbox"/>	
DropoffTitle	Nvarchar(200)	<input type="checkbox"/>	
License	varchar(10)	<input type="checkbox"/>	
CustomerEmail	Nvarchar(200)	<input type="checkbox"/>	
CustomerName	Nvarchar(200)	<input type="checkbox"/>	
CustomerPhoneNumber	bigint	<input type="checkbox"/>	
PaymentDiscount	float	<input type="checkbox"/>	
PaymentSubtotal	float	<input type="checkbox"/>	
PaymentTotal	float	<input type="checkbox"/>	
PickupDescription	Nvarchar(200)	<input type="checkbox"/>	
PickupTimePick	int	<input type="checkbox"/>	
PickupTitle	Nvarchar(200)	<input type="checkbox"/>	
TicketDetailBusType	Nvarchar(50)	<input type="checkbox"/>	

TicketDetailQuantity	int	<input type="checkbox"/>	
TicketDetailSeatNum	varchar(50)	<input type="checkbox"/>	
TicketDetailTime	int	<input type="checkbox"/>	
TicketDetailTransactionNumber	varchar(10)	<input type="checkbox"/>	
TicketDetailTrip	Nvarchar(50)	<input type="checkbox"/>	

Table: Payment Method

PaymentID	varchar(50)	<input type="checkbox"/>	PK
Method	Nvarchar(200)	<input type="checkbox"/>	
Image	image	<input type="checkbox"/>	

Table: Employee

EmployeeID	varchar(10)	<input type="checkbox"/>	PK
DepartmentID	varchar(10)	<input type="checkbox"/>	FK
Salary	float	<input type="checkbox"/>	
HireDate	datetime	<input type="checkbox"/>	
EmployeeType	varchar(1)	<input type="checkbox"/>	Manager, Staff, Driver, Administrator

Table: Department

DepartmentID	varchar(10)	<input type="checkbox"/>	PK
--------------	-------------	--------------------------	----

DepartmentName	Nvarchar(50)	<input type="checkbox"/>	
DepartmentDescription	Nvarchar(100)	<input checked="" type="checkbox"/>	

Table: Manager

MEmployeeID	varchar(10)	<input type="checkbox"/>	PK FK (EmployeeID)
Level	varchar(50)	<input type="checkbox"/>	
Manager	varchar(10)	<input type="checkbox"/>	FK

Table: Staff

SEmployeeID	varchar(10)	<input type="checkbox"/>	PK FK (EmployeeID)
MEmployeeID	varchar(10)	<input type="checkbox"/>	FK
TrainingCourseID	varchar(10)	<input type="checkbox"/>	FK

Table: Driver

DEmployeeID	varchar(10)	<input type="checkbox"/>	PK FK (EmployeeID)
Avatar	image	<input type="checkbox"/>	
Hours	int	<input type="checkbox"/>	
License	varchar(1)	<input type="checkbox"/>	
DriverName	Nvarchar(50)	<input type="checkbox"/>	

PhoneNumber	bigint	<input type="checkbox"/>	
Rating	int	<input type="checkbox"/>	
TravelTrip	int	<input type="checkbox"/>	

Table: Administrator

AEmployeeID	varchar(10)	<input type="checkbox"/>	PK FK (EmployeeID)
Description	Nvarchar(100)	<input checked="" type="checkbox"/>	

Table: Shift

ShiftID	varchar(10)	<input type="checkbox"/>	PK
ShiftTime	datetime	<input type="checkbox"/>	
ShiftDescription	Nvarchar(100)	<input checked="" type="checkbox"/>	
DEmployeeID	varchar(10)	<input type="checkbox"/>	FK
BusID	varchar(10)	<input type="checkbox"/>	FK

Table: BookedTicket

TransactionNumber	varchar(10)	<input type="checkbox"/>	PK
AccountID	varchar(10)	<input type="checkbox"/>	FK
BookedTime	datetime	<input type="checkbox"/>	
DepartureALocation	varchar(50)	<input type="checkbox"/>	

DepartureBus	varchar(50)	<input type="checkbox"/>	
DepartureDLocation	varchar(50)	<input type="checkbox"/>	
DropOffPointAddress	Nvarchar(200)	<input type="checkbox"/>	
DropOffPointChecked	bool	<input type="checkbox"/>	
DropOffPointDate	date	<input type="checkbox"/>	
DropOffPoint	Nvarchar(200)	<input type="checkbox"/>	
DropOffPointShuttleBus	bool	<input type="checkbox"/>	
DropOffPointTime	time	<input type="checkbox"/>	
PickupPointAddress	Nvarchar(200)	<input type="checkbox"/>	
PickupPointChecked	bool	<input type="checkbox"/>	
PickupPointDate	date	<input type="checkbox"/>	
PickupPoint	Nvarchar(200)	<input type="checkbox"/>	
PickupPointShuttleBus	bool	<input type="checkbox"/>	
PickupPointTime	time	<input type="checkbox"/>	
Seat	varchar(50)	<input type="checkbox"/>	
TicketID	varchar(10)	<input type="checkbox"/>	
TicketAmentiites	Nvarchar(200)	<input type="checkbox"/>	
TicketAOFFice	Nvarchar(200)	<input type="checkbox"/>	

TicketATime	time	<input type="checkbox"/>	
TicketBus	Nvarchar(200)	<input type="checkbox"/>	
TicketDate	date	<input type="checkbox"/>	
TicketDOffice	Nvarchar(200)	<input type="checkbox"/>	
TicketDrive	Nvarchar(200)	<input type="checkbox"/>	
TicketDTime	time	<input type="checkbox"/>	
TicketImage	image	<input type="checkbox"/>	
TicketPrice	float	<input type="checkbox"/>	
TicketReviews	Nvarchar(200)	<input type="checkbox"/>	
TicketRoute	varchar(50)	<input type="checkbox"/>	
TicketSeat	varchar(200)	<input type="checkbox"/>	
TicketTotalPrice	float	<input type="checkbox"/>	
DiscountPrice	float	<input type="checkbox"/>	
CustomerEmail	Nvarchar(200)	<input type="checkbox"/>	
CustomerName	Nvarchar(200)	<input type="checkbox"/>	
CustomerPhoneNumber	bigint	<input type="checkbox"/>	
Status	varchar(10)	<input type="checkbox"/>	
Table: Order			

OrderID	varchar(10)	<input type="checkbox"/>	PK
OrderAmount	float	<input type="checkbox"/>	
OrderStatus	varchar(1)	<input type="checkbox"/>	Paid Cancel
PaymentID	varchar(10)	<input type="checkbox"/>	FK

Table: Voucher

VoucherID	varchar(10)	<input type="checkbox"/>	PK
VoucherDescription	Nvarchar(100)	<input type="checkbox"/>	
VoucherAmount	float	<input type="checkbox"/>	
VoucherCondition	Nvarchar(100)	<input type="checkbox"/>	
VoucherConditionValue	int	<input type="checkbox"/>	
VoucherExpired	datetime	<input type="checkbox"/>	
VoucherImage	image	<input type="checkbox"/>	
VoucherPercentage	int	<input type="checkbox"/>	

Table: Cancellation

CancelID	varchar(10)	<input type="checkbox"/>	PK
CancelDescription	Nvarchar(100)	<input type="checkbox"/>	

Table: Bus

BusID	varchar(10)	<input type="checkbox"/>	PK
BusName	varchar(10)	<input type="checkbox"/>	
BusDescription	Nvarchar(100)	<input type="checkbox"/>	
BusImage	image	<input type="checkbox"/>	
BusTag	Nvarchar(100)	<input type="checkbox"/>	

Table: Bus Type

BusTypeID	varchar(10)	<input type="checkbox"/>	PK
TypeName	Nvarchar(100)	<input type="checkbox"/>	
PriceRate	float	<input type="checkbox"/>	
TypeDescription	Nvarchar(100)	<input checked="" type="checkbox"/>	

Table: RouteWithPoints

BusID	varchar(10)	<input type="checkbox"/>	PK
RouteID	varchar(10)	<input type="checkbox"/>	FK
ALocation	Nvarchar(100)	<input type="checkbox"/>	
DLocation	Nvarchar(100)	<input type="checkbox"/>	

Table: Seat

SeatID	varchar(10)	<input type="checkbox"/>	PK
BusID	varchar(10)	<input type="checkbox"/>	PK

SeatType	varchar(50)	<input type="checkbox"/>	
SeatDescription	Nvarchar(100)	<input checked="" type="checkbox"/>	

Table: SeatBooking

TicketBookingID	varchar(10)	<input type="checkbox"/>	PK
SeatID	varchar(10)	<input type="checkbox"/>	PK
Date	datetime	<input type="checkbox"/>	

Table: Pick Drop Point

RouteID	varchar(10)	<input type="checkbox"/>	PK
PickupID	varchar(10)	<input type="checkbox"/>	FK
DropoffID	varchar(10)	<input type="checkbox"/>	FK

Table: Pickup Point

PickupID	varchar(10)	<input type="checkbox"/>	PK
Pickup Location	Nvarchar(100)	<input type="checkbox"/>	
ShuttleService	bool	<input checked="" type="checkbox"/>	

Table: Dropoff Point

DropoffID	varchar(10)	<input type="checkbox"/>	PK
Dropoff Location	Nvarchar(100)	<input type="checkbox"/>	
ShuttleService	bool	<input checked="" type="checkbox"/>	

Table: Amenity

AmenityID	varchar(10)	<input type="checkbox"/>	PK
Title	Nvarchar(100)	<input type="checkbox"/>	
Description	Nvarchar(100)	<input type="checkbox"/>	
BusID	varchar(10)	<input type="checkbox"/>	FK
Image	image	<input type="checkbox"/>	

Table: Feedback

FeedbackID	varchar(10)	<input type="checkbox"/>	PK
Avatar	image	<input type="checkbox"/>	
CustomerName	Nvarchar(100)	<input type="checkbox"/>	
Date	datetime	<input type="checkbox"/>	
Destination	Nvarchar(10)	<input type="checkbox"/>	
Feedback	Nvarchar(100)	<input type="checkbox"/>	
Image	image	<input type="checkbox"/>	
Rating	tinyint	<input type="checkbox"/>	
Tag	varchar(10)	<input type="checkbox"/>	
Total Trip	int	<input type="checkbox"/>	

Table: Notification

NotificationID	varchar(10)	<input type="checkbox"/>	PK
Title	Nvarchar(100)	<input type="checkbox"/>	
Description	Nvarchar(100)	<input type="checkbox"/>	
Timestamp	datetime	<input type="checkbox"/>	
Image	image	<input type="checkbox"/>	

5.8.4. ERD Model

5.8.4.1. ERD

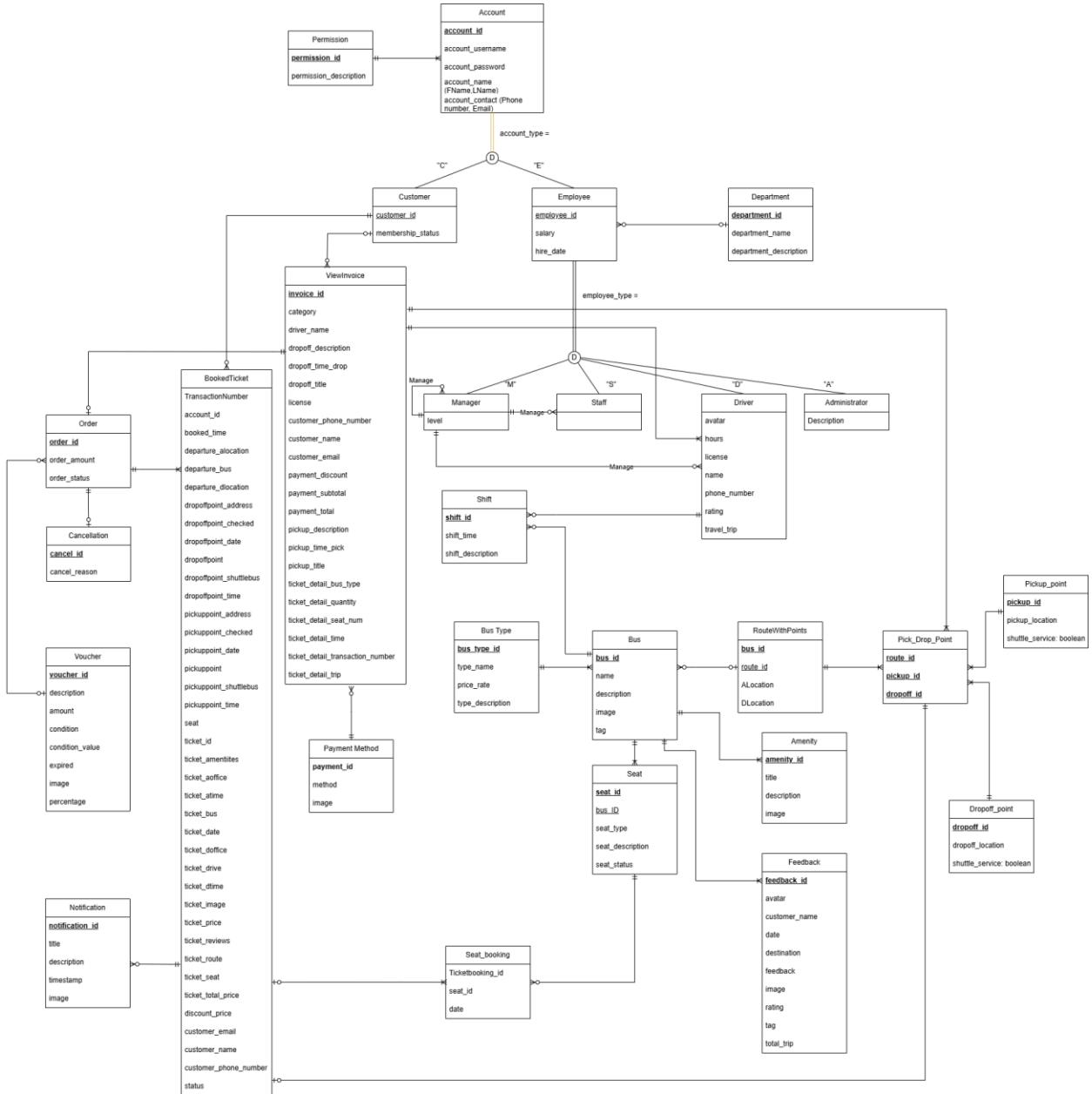


Figure 5.23. ERD Model

(Sources: Author's Sources)

This Entity-Relationship Diagram (ERD) models a comprehensive bus booking system, linking core entities like **Account** (representing either **Customers** or **Employees**) with booking, payment, and operational management. The **BookedTicket** table connects to **Order**, **Cancellation**, and **Invoice** for tracking transactions and status changes. Route and bus details are managed through **Bus**, **RouteWithPoints**, **Pick/Drop Points**, **Seat**, and **Bus Types**, while features like **Voucher**, **Feedback**, and **Notification** enhance user experience. Administrative functions, including **Shift** and **Department**, support employee roles such as drivers, staff, and managers. This structure ensures efficient handling of bookings, payments, and user interactions.

5.8.4.2. Relationship description

Table 5.61. Relationship description table

Name	Relationship	Type	Description
R-1	Permission - Account	1-n	Each permission can apply to many accounts. Each account has only one permission.
R-2	Department - Employees	1-n	Each department can have many employees. Each employee belongs to only one department.
R-3	Manager - Staff	1-n	Each manager can manage many staff. Each staff member has only one manager.
R-4	Manager - Driver	1-n	Each manager can manage many drivers. A driver can be managed by one manager.
R-5	Manager - Manager	1-n	Each high-level manager can manage many low-level managers. Each low-level manager can be managed by a high-level manager.
R-6	Driver - Shift	1-n	Each driver has many shifts. Each shift has only one driver.

R-7	Bus - Shift	n-n	Each bus has many shifts. Each shift has many buses.
R-8	RouteWithPoints - Bus	1-n	Each route can have many buses. Each bus can only be driven on one route.
R-9	RouteWithPoints - Pick Drop point	1-n	Each route can have many pick-up/drop-off points. Each pick-up/drop-off point belongs to only one route.
R-10	Pickup point - Pick Drop point	1 - n	Each pickup point can belong to many pick-up/drop-off points. Each pick-up/drop-off point has only one pickup point.
R-11	Dropoff point - Pick Drop point	1 - n	Each drop-off point can belong to many pick-up/drop-off points. Each pick-up/drop-off point has only one drop-off point.
R-12	Bus type - Bus	1-n	Each bus type can have many buses. Each bus belongs to only one bus type.
R-13	Bus - Seat	1-n	Each bus can have several seats. Each seat belongs to only one bus.
R-15	Seat - Seat booking	1-n	Each seat can have many seat bookings. Each seat booking just belongs to one seat.
R-16	BookedTicket - Seat booking	1-n	Each ticket booking can have many seat bookings. Each seat booking has only one ticket booking.
R-17	Customer - ViewInvoice	1-n	Each customer can have many invoices. Each invoice belongs to only one customer.
R-18	Payment Method - ViewInvoice	1-n	Each payment type can be used in many invoices. Each invoice has only one payment type.
R-19	ViewInvoice - Order	1-1	Each invoice must have only one order. Each order may or may not have an invoice.

R-20	Customer - BookedTicket	1-n	Each customer can perform many ticket bookings. Each ticket booking can only be performed by one customer.
R-21	Order - ViewInvoice	1-1	Each order is associated with exactly one invoice. Each invoice may or may not have an order.
R-22	Order - Ticket Booking	1-1	Each order can be associated with exactly one ticket booking. Each ticket booking is included in only one order.
R-23	BookedTicket - Pick Drop point	1-1	Each ticket booking is associated with exactly one pick-drop point. Each pick-drop point may or may not be associated with a ticket.
R-24	Voucher - Order	1-n	Each order may have zero or more vouchers applied to it. Each voucher may or may not be associated with an order.
R-25	Order - Cancellation	1-1	Each order may or may not have a corresponding cancellation. Each cancellation is associated with one order.
R-27	Bus - Amenity	1-n	Each bus can have many associated amenities. Each amenity is linked to only one bus.
R-28	Bus - Feedback	1-n	Each bus can contain a lot of feedback from passengers. Each feedback is related to one bus.
R-29	BookedTicket - Notification	1-n	Each ticket booking can be associated with multiple notifications. Each notification is linked to only one ticket booking.

5.8.5 Data Normalization



Figure 5.24. Data Normalization

(Source: Author's sources)

The provided image illustrates a detailed Entity-Relationship Diagram (ERD) that represents a normalized database structure. The model is designed to organize data efficiently by dividing it into multiple entities, each representing a specific aspect of the system. Key entities such as Account, User, Products, Order, and Feedback are defined with distinct attributes like username, password, name, order_id, and rating. This structure ensures that each entity focuses on a single responsibility, which is a fundamental principle of database normalization.

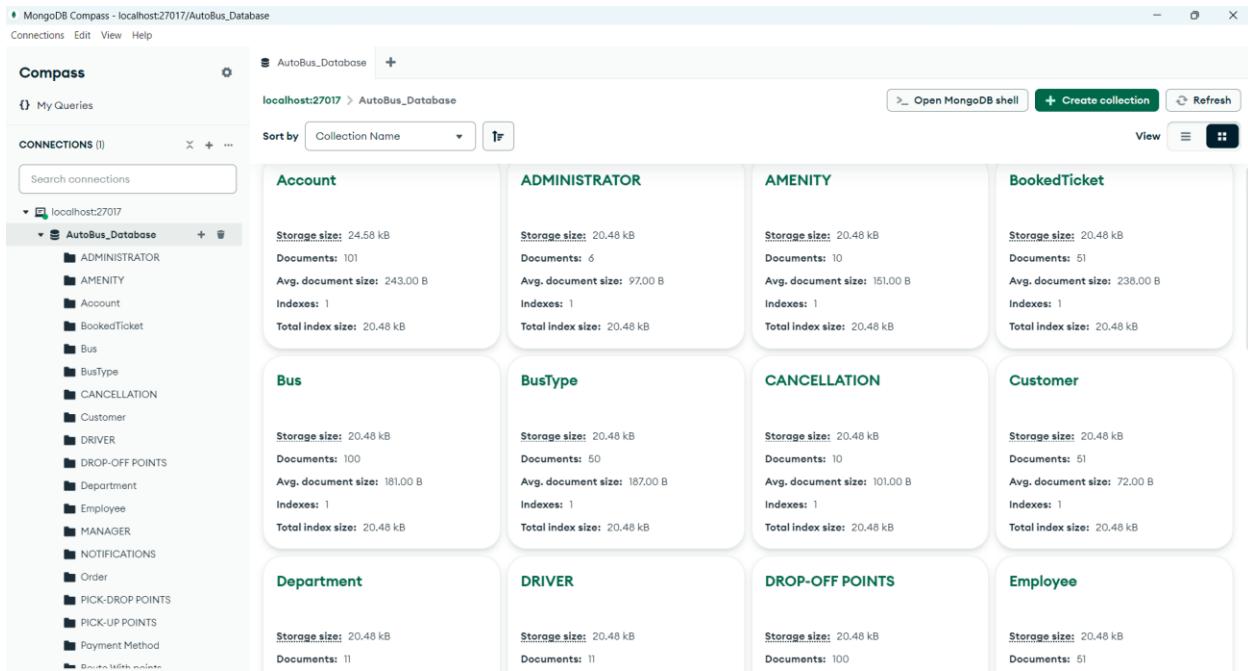
The relationships between these entities are visually represented through connecting lines, demonstrating how they interact with each other. Primary keys and foreign keys establish these relationships, ensuring data integrity and enabling efficient querying. For instance, the Account entity is linked to the User entity to store login credentials and user-specific

details, while the Order entity is associated with subentities like Order_Items to manage order details such as product quantities and prices. These connections reduce redundancy by avoiding repetitive storage of the same data in multiple places.

Normalization, as implemented in this model, plays a critical role in improving the efficiency of the database. By dividing larger tables into smaller, related ones, it minimizes data duplication, reduces storage requirements, and ensures data consistency. For example, instead of embedding product information directly into the Order table, the diagram separates it into the Products entity, which can be referenced multiple times across various orders. This approach not only simplifies updates but also makes the database more scalable for future expansion.

Overall, this ERD represents a well-structured blueprint for a relational database, adhering to the principles of normalization. It balances the need for efficient data storage with ease of access, ensuring that the system can handle complex queries and maintain data consistency across all entities. This design is essential for building a robust and scalable database system that can support the requirements of a modern application.

5.8.6 Physical database in MongoDB



The screenshot shows the MongoDB Compass interface connected to the 'AutoBus_Database' on 'localhost:27017'. The left sidebar lists collections: ADMINISTRATOR, AMENITY, BookedTicket, Bus, BusType, CANCELLATION, Customer, DRIVER, DROP-OFF POINTS, Department, Employee, MANAGER, NOTIFICATIONS, Order, PICK-DROP POINTS, PICK-UP POINTS, Payment Method, and RouteDetails. The main area displays 12 collection cards in a grid:

Collection	Storage size	Documents	Avg. document size	Indexes	Total index size
ADMINISTRATOR	20.48 kB	6	97.00 B	1	20.48 kB
AMENITY	20.48 kB	10	161.00 B	1	20.48 kB
BookedTicket	20.48 kB	51	238.00 B	1	20.48 kB
Bus	20.48 kB	50	187.00 B	1	20.48 kB
BusType	20.48 kB	10	101.00 B	1	20.48 kB
CANCELLATION	20.48 kB	100	72.00 B	1	20.48 kB
Customer	20.48 kB	61	20.48 kB	1	20.48 kB
Department	20.48 kB	11	20.48 kB	1	20.48 kB
DRIVER	20.48 kB	11	20.48 kB	1	20.48 kB
DROP-OFF POINTS	20.48 kB	100	20.48 kB	1	20.48 kB
Employee	20.48 kB	51	20.48 kB	1	20.48 kB

Figure 5.25. Physical database in MongoDB

(Source: Authors' Sources)

In this system, the physical database is designed using **MongoDB**, a NoSQL database that provides excellent flexibility and scalability for handling semi-structured data. MongoDB is chosen to accommodate the system's need for high-performance data retrieval and storage, supporting complex queries and large-scale operations efficiently.

More specifically, in order to store data well in MongoDB, several configurations and optimizations have been implemented to ensure the database is robust, scalable, and performant. These include carefully structured collections, indexing strategies, and features like sharding and replica sets for scalability and reliability. Below is a detailed overview of the database configuration and optimization:

Table 5.62. Database Configuration and Optimization

Aspect	Details	Description
Database Name	Autobus_Database	The name of the MongoDB database used to store all collections related to the bus ticket booking system.
Management Tool	MongoDB Compass	A graphical user interface tool used to interact with the MongoDB database for visualization, queries, and data management.
Connection String	mongodb://127.0.0.1:27017	The local connection string used to connect to the MongoDB instance during development. This can be configured for cloud or production environments later.
Sharding Support	Designed for future scalability.	MongoDB's sharding mechanism allows horizontal scaling by distributing data across multiple servers, ensuring the system can handle large datasets easily.
Replica Sets	Optional for production.	MongoDB supports replica sets for fault tolerance and data redundancy, ensuring data is consistently backed up and the system remains operational.
Backup Strategy	Regular data backups using mongodump.	The database is configured to perform regular backups, ensuring data integrity and availability in case of unexpected failures or data corruption.
Authentication & Security	Configured to use	User roles are defined to

	authentication with roles and permissions.	restrict access to sensitive data (e.g., admin, read-only, and write access) and ensure system security during multi-user access.
Performance Optimization	Compound indexes for frequent queries.	Indexing multiple fields together (e.g., travel_date and bus_id) in BookedTicket allows for faster lookups and optimized database queries.

5.9. Data Dictionary

5.9.1. Data type definition

Table 5.63. Data Type Definition

Single Line of Text	aA - Zz 0 - 9 including special characters !@#\$%^&*():"
Multiple Lines of Text	aA - Zz 0 - 9 including special characters !@#\$%^&*():"
Two Options	Apply to True / False; Yes / No
Option Set	Apply to a list of predefined static values with single selection
Lookup	Look up values from another objects (Look up from Customer Object to reference to Delivery Order)
Multi-Selection	Apply to a list of predefined static values with multi-selection
Whole Number	0 - 9
Decimal Number	Apply to a number with decimal places. Default is 2
Currency	Apply to currency value with currency-formatted display (E.g 2.000.000VND)
Date	Date without time (Format: DD/MM/YYYY)

Datetime string	Format: DayOfWeek Mon DD HH:mm:ss Timezone YYYY
Time	Time without date (Format: h:mm:ss)
Countdown	Minute & Second (Format: mm:ss)
Date & Time	Date and Time (same as above)
Month & Date	Month & Date (Format: MM/DD)
Month & Year	Month & Year (Format: MM/YY)

5.9.2. Data dictionary

Table 5.64. Data dictionary

Rule ID	Rule Title	Mandatory (Yes - Y/No - N)	Data Type	Value	Rule Description	System/User
Account Management						
DD-01	Phone Number Format	Y	Whole Number		<ul style="list-style-type: none"> Start with 0. Maximum number length: 10. 	User
DD-02	OTP Code	Y	Whole Number		Maximum length: 4.	User
DD-03	Verification Code	Y	Whole Number		Maximum length: 4.	
DD-04	Password Format	Y	Single Line of Text		Minimum length: 8.	User

DD-05	Name	Y	Single Line of Text		<ul style="list-style-type: none"> - Only allow character input (aA - zZ). - Must have at least 1 character. 	User
DD-06	Email	N	Single Line of Text		Must contain @.	User
DD-07	Phone Number	Y	Whole Number		<ul style="list-style-type: none"> • Start with 0. • Maximum number length: 10. 	User
DD-08	Email Address	N	Single Line of Text		Must contain @.	User
DD-09	City	N	Option Set		Refer to City List.	User
DD-10	District	Y	Option Set		Refer to District List, based on chosen City.	User
DD-11	Ward	Y	Option Set		Refer to Ward List, based on chosen District.	User

DD-12	Street Name	Y	Single Line of Text			User
DD-13	House Number	Y	Single Line of Text			User
Ticket Booking						
DD-14	Departure Location	Y	Option Set		Refer to the company's predetermined Departure Location list.	User
DD-15	Arrival Location	Y	Option Set		Refer to the company's predetermined Arrival Location list.	User
DD-16	Start/Departure Date	Y	Option Set		The date of the request for Departure must be a future date.	User
DD-17	Round Trip Option	N	Two Options	Default: One Way	Choose between Round Trip and One Way.	User
DD-18	Return Date	Y	Option Set		The date of the request for Return	User

					must be a future date.	
DD-19	Number of Seats	Y	Whole Number		Only allow to choose equally to the existing number of bus seats.	User
DD-20	Pick-up Location	Y	Option Set		Refer to the company's predetermined Pick-up Location list.	User
DD-21	Drop-off Location	Y	Option Set		Refer to the company's predetermined Drop-off Location list.	User
DD-22	Name	Y	Single Line of Text		<ul style="list-style-type: none"> - Only allow character input (aA - zZ). - Must have at least 1 character. 	User
DD-23	Email	N	Single Line of Text		Must contain @.	User
DD-24	Phone Number	Y	Whole Number		<ul style="list-style-type: none"> ● Start with 0. ● Maximum number 	User

					length: 10.	
Payment Transaction						
DD-25	Payment Method	Y	Option Set	Payment List: <ul style="list-style-type: none"> ● VISA ● MasterCard ● Internet Banking ● MOMO ● ZaloPay ● VNPay ● PayPal 	Choose from predefined payment methods.	User
DD-26	Card Number	Y	Whole Number		Length: 8 to 16 digits.	User
DD-27	Issue Date	Y	Month & Date		Issue Date on Card.	User
DD-26	Cardholder Name	Y	Single Line of Text		Only allow character input (aA - zZ).	User
DD-27	CVV	Y	Whole Number		Length: 3 digits. All 3 digits must be fulfilled.	User
DD-28	Ticket-booking ID	Y	Single Line of Text		The Ticket-booking ID is formatted	System

					by numbers and texts.	
DD-29	Customer's Name	Y	Single Line of Text		<ul style="list-style-type: none"> - Only allow character input (aA - zZ). - Must have at least 1 character. 	System
DD-30	Phone Number	Y	Whole Number		<ul style="list-style-type: none"> • Start with 0. • Maximum number length: 10. 	System
DD-31	Number of Seats	Y	Whole Number		Only allow to choose equally to the existing number of bus seats.	System
DD-32	Departure Location	Y	Option Set		Refer to the company's predetermined Departure Location list.	System
DD-33	Arrival Location	Y	Option Set		Refer to the company's predetermined Arrival Location list.	System

DD-34	Start/Departure Date	Y	Date & Time		The date of the request for Departure must be a future date.	System
DD-35	Return Date	Y	Date		The date of the request for Return must be a future date.	System
DD-36	Pick-up Location	Y	Option Set		Refer to the company's predetermined Pick-up Location list.	System
DD-37	Drop-off Location	Y	Option Set		Refer to the company's predetermined Drop-off Location list.	System
DD-38	Invoice ID	Y	Single Line of Text		The Invoice ID is formatted by texts and numbers.	System
DD-39	Used Payment Method	Y	Option Set	Payment List: <ul style="list-style-type: none">• VISA• MasterCard• Internet Banking	Choose from predefined payment methods.	System

				<ul style="list-style-type: none"> • MOMO • ZaloPay • PayPal 		
DD-40	Price	Y	Currency		Minimum amount: 1.000VND	System
DD-41	Booking Cancellation	Y	Single Line of Text		Users must accept the Cancellation Regulations before canceling the booked tickets.	System
Ratings and Feedback						
DD-42	Ratings	N	Option Set		Require to select the number of rating stars.	User
DD-43	Image	N	Multi-Selection		Upload images related to the booking.	User
DD-44	Video	N	Multi-Selection		Upload videos related to the booking.	User

DD-45	Comment	N	Multiple Lines of Text		Add comments or feedback related to the booking.	User
Booking History						
DD-46	Unpaid Tickets	N	Multiple Lines of Text		Automatically cancel after 15-minute non-payment.	System
DD-47	Time Left	Y	Countdown		<ul style="list-style-type: none"> • Start from: 15:00 • End by: 00:00 	System
DD-48	Payment Duration	Y	Datetime String		Calculate from the time of booking tickets.	System
View Invoice						
DD-49	Arrival Time	Y	Time		Start to countdown from 2 hours before the trip arrives.	System

DD-50	License Plate Number	Y	Single Lines of Text		Include numbers, texts, and a hyphen.	System
DD-51	Driver's Phone Number	Y	Whole Number		<ul style="list-style-type: none"> • Start with 0. • Maximum number length: 10. 	System
DD-52	Departure Time	Y	Time		Start to countdown from 2 hours before the trip departs.	System
DD-53	Complete Time	Y	Date		Update the time the trip departs.	System

CHAPTER 6: SYSTEM DESIGN

6.1. Develop Wireframe for the app

6.1.1. Account Management

This wireframe describes every step of a mobile application's account management process, from registration and login to account recovery, profile settings, and transaction history. It offers a methodical process to make it easy for users to observe activities, manage account information, and change settings.

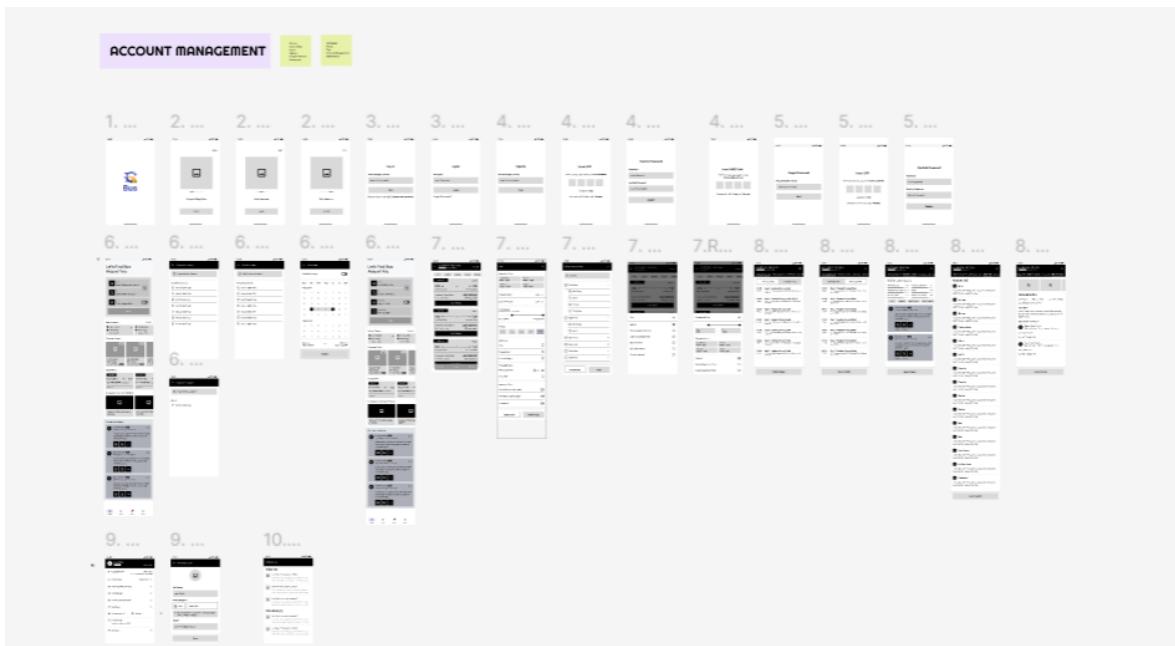


Figure 6.1. Account Management Process in AutoBus App

(Sources: Authors' Sources)

6.1.1.1. Log in /Sign up/ Forgot Password

This wireframe describes a mobile application's registration, login, and password recovery procedures. Screens for checking in, creating an account using an OTP (One-Time Password), email verification, and changing a forgotten password are all included. Every stage helps the user provide important data, like their email address or phone number, get

and validate one-time passwords, and create a new one. Users can easily access or restore their accounts thanks to the layout's simple and obvious flow.

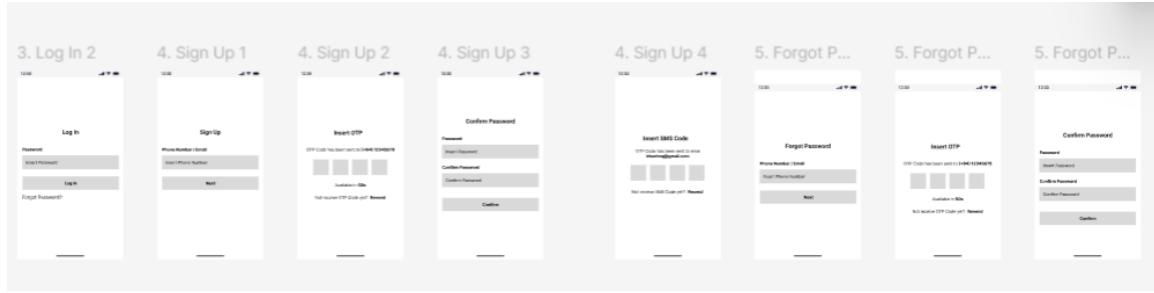


Figure 6.2. Log in /Sign up/ Forgot Password

(Sources: Authors' Sources)

6.1.1.2. Travel Booking and Management App UI Overview

The account management procedure is continued in this wireframe, which includes areas for transaction history, notifications, and account settings. The pages display options to check notification history, manage app preferences, update personal information, and get transaction data. Additionally, there are interfaces for dealing with notifications and seeing particular transaction data. This user-friendly interface makes it easy for users to modify their account settings and effectively examine previous actions.



Figure 6.3. Travel Booking and Management App UI Overview

(Sources: Authors' Sources)

6.1.1.3. Account Management

The personal profile page shown in this wireframe is intended to be user-friendly. User information (name, rewards, promotions, and payment options), personal information (editable contact details), reward points (accumulated points and advancement to the next level), promotions (available discounts and coupons), payment management (edit payment methods and payment history), trip ratings (user reviews), invitations (promotional event invites), feedback (user suggestions), language settings (language switch), help center (support access), and settings (customizable profile preferences) are some of its main sections. A thorough framework for handling user accounts and enhancing user interaction is offered by this wireframe.

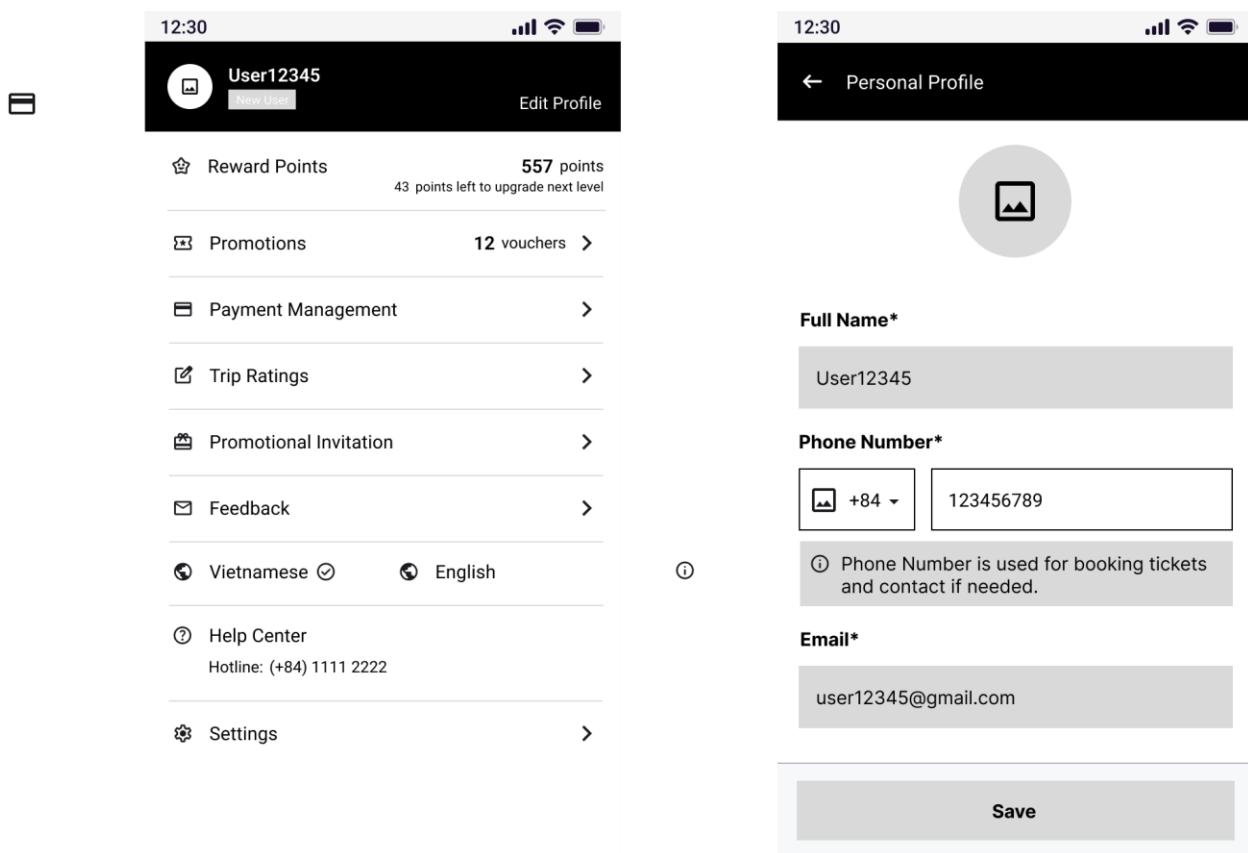


Figure 6.4. Account Management

(Sources: Authors' Sources)

6.1.2. Booking Ticket

This wireframe outlines the entire booking process for a bus ticketing system, from seat selection to pick-up/drop-off points, passenger information input, and final booking preview. It provides a step-by-step flow to guide users through completing their booking efficiently.

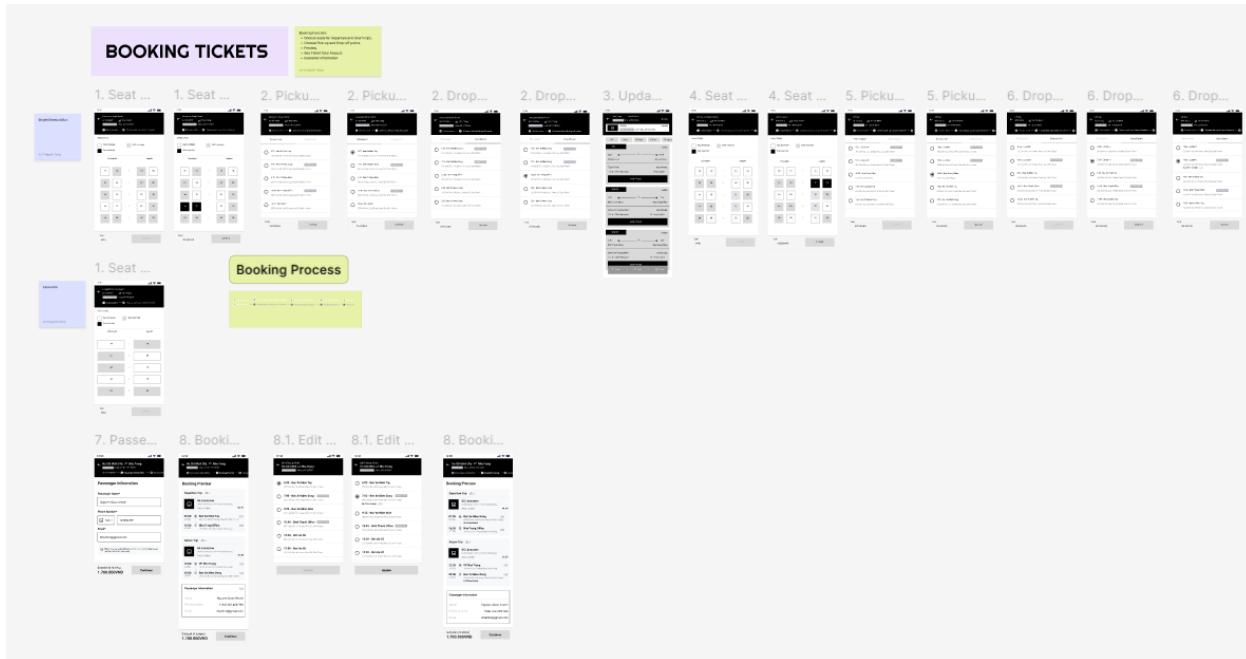


Figure 6.5. Booking Tickets Process in AutoBus App

(Sources: Authors' Sources)

6.1.2.1. Select pick up/drop off points (departure trip)

This wireframe shows the seat selection and pick-up/drop-off selection process for a bus booking app. Users choose seats, and the total updates with an active "Continue" button. Next, they select pick-up and drop-off points, with details on location and time. The total and "Continue" button remain at the bottom, guiding users through each booking step.

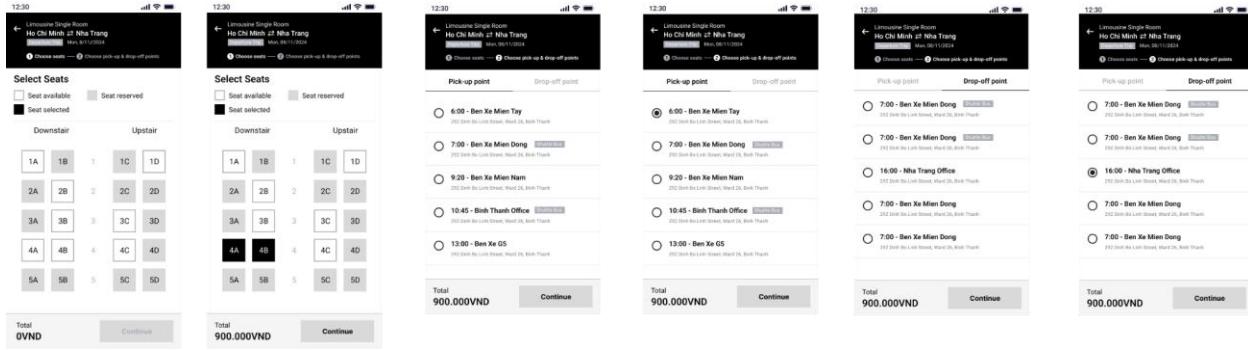


Figure 6.6. Select pick up/drop off points for Single/Standard Bus

(Sources: Authors' Sources)

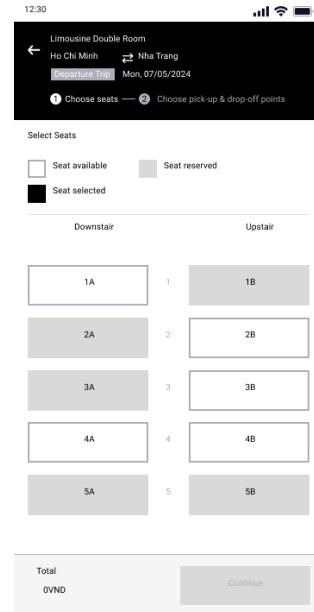


Figure 6.7. Select pick up/drop off points for Limousine Bus

(Sources: Authors' Sources)

6.1.2.1. Select pick up/drop off points (return trip)

This wireframe outlines the return trip booking process in the bus booking app, covering seat selection and pick-up/drop-off point selection. Users choose seats, updating the total cost, and then select pick-up and drop-off points with location and time details. The total

and "Continue" button remain at the bottom to guide users through the return trip booking steps.

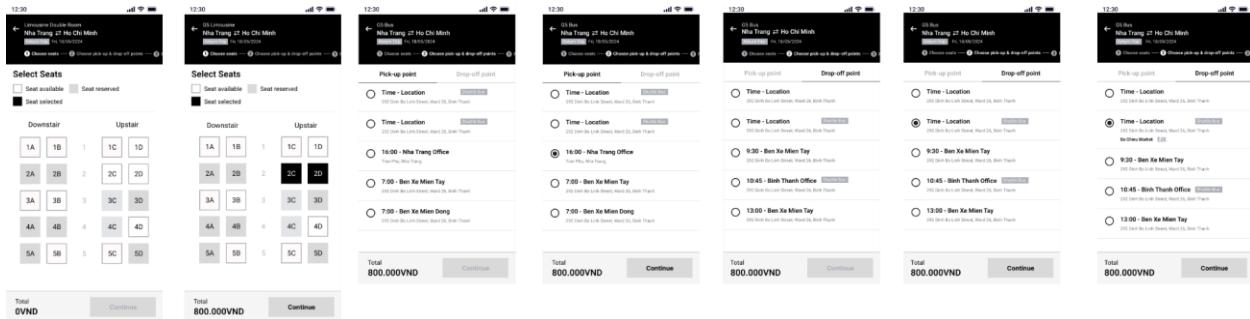


Figure 6.8. Select pick up/drop off points

(Sources: Authors' Sources)

6.1.3. Transaction

This wireframe shows the transaction flow in a bus booking app, from selecting vouchers and payment methods to handling successful or failed payments. Users can choose discounts, confirm payments, and address any issues with options for retrying or accessing "Unpaid Tickets" for later resolution.

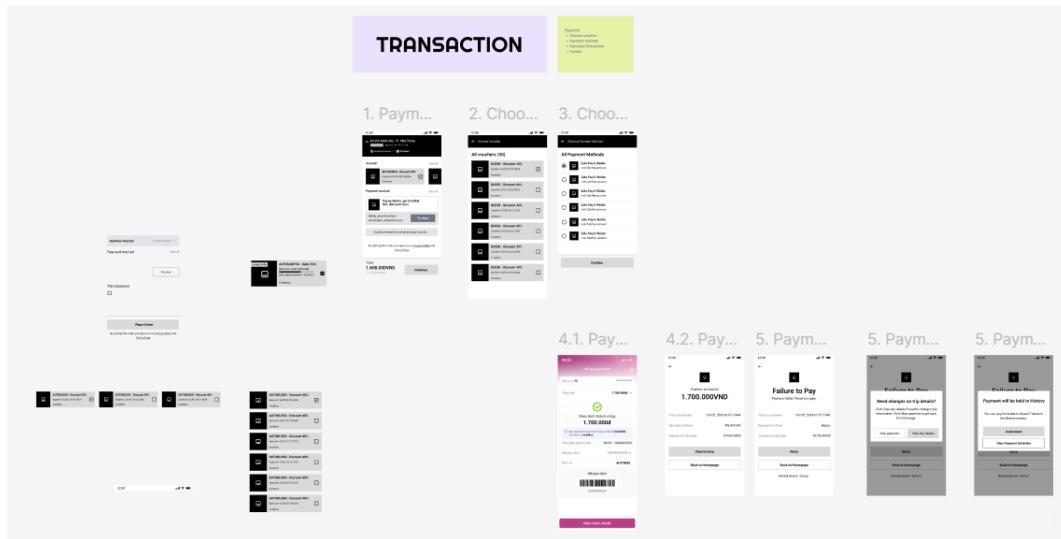


Figure 6.9. Transaction

(Sources: Authors' Sources)

6.1.3.1. Voucher

This wireframe displays the voucher selection process in a bus booking app, where users can choose from various discounts with details like percentage, expiration date, and conditions. Some vouchers are marked as "Limited offer" for special deals with specific requirements. Once selected, the discount applies to the booking total, and a "Place Order" button lets users proceed, agreeing to the Privacy Policy and Terms of Use. This layout integrates voucher selection smoothly into the booking process.

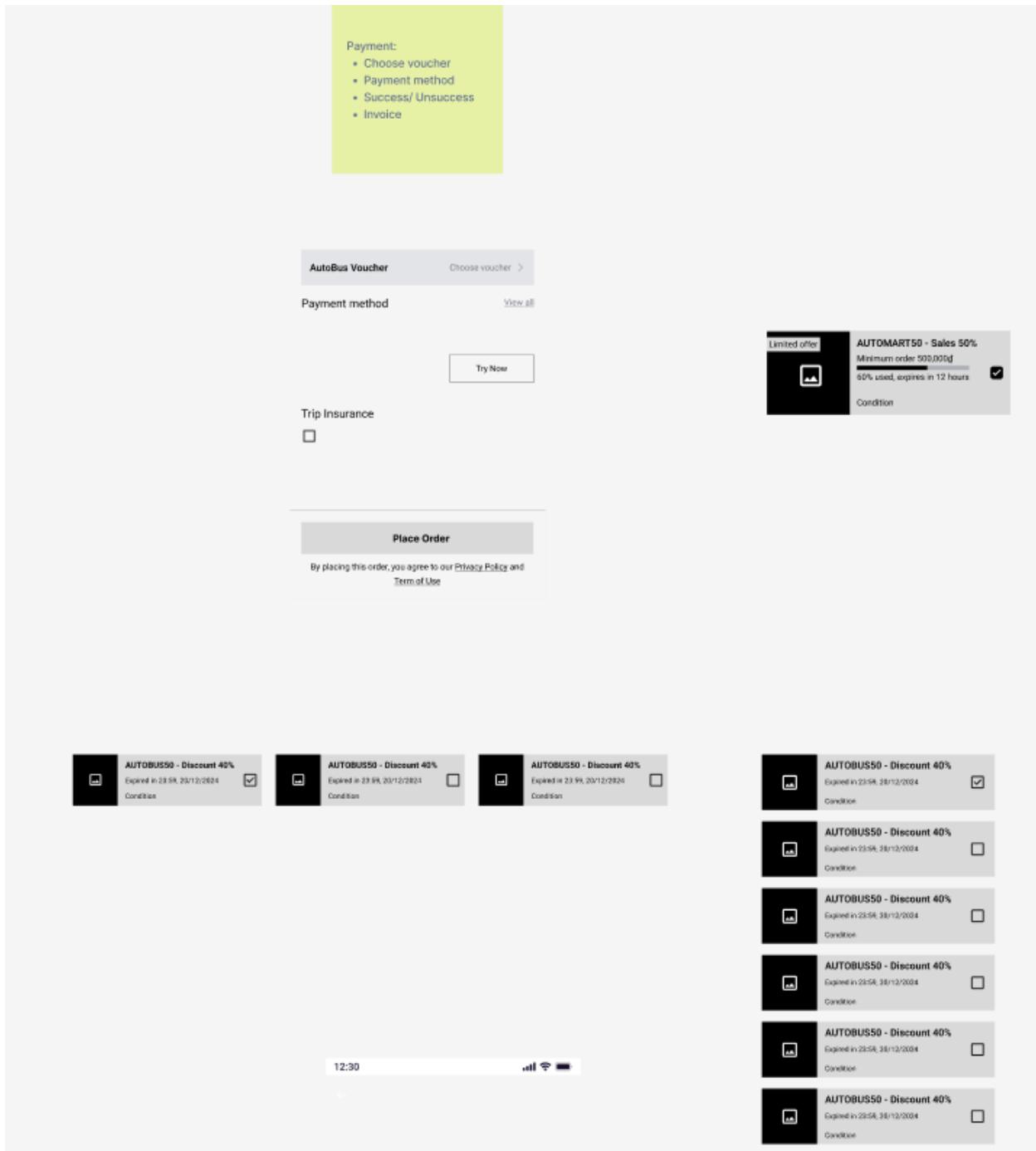


Figure 6.10. Voucher

(Sources: Authors' Sources)

6.1.3.2. Choose Voucher and Payment Method

This image shows the voucher and payment selection steps in a bus booking app. Users view trip details, select a voucher and payment method, and see the discounted total in "Payment (1)." In "Choose Voucher," they browse discounts, and in "Choose Payment Method," they pick a payment option. A "Confirm" button finalizes choices, making the process easy and clear.

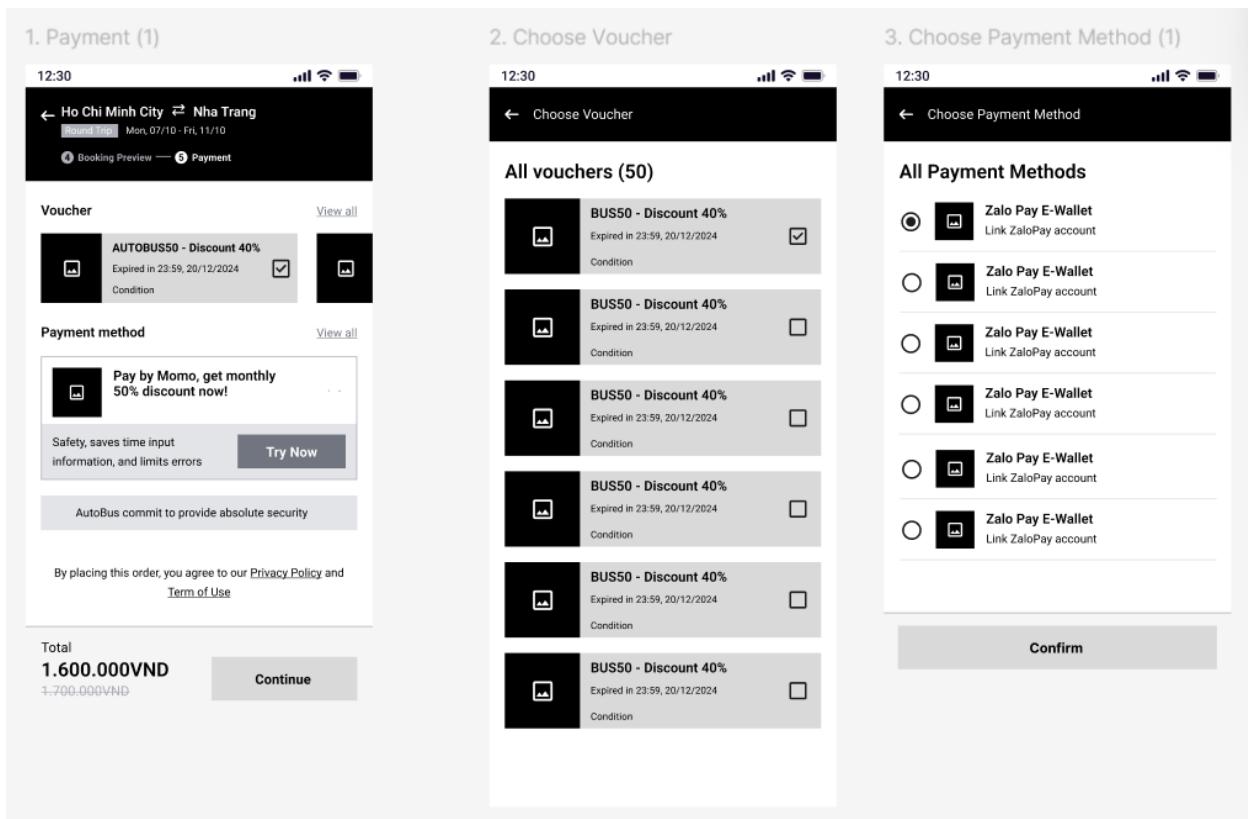


Figure 6.11. Choose Voucher and Payment Method

(Sources: Authors' Sources)

6.1.3.3. Payment

This image shows different payment outcomes in a bus booking app for successful and unsuccessful scenarios. For successful payments, "Pay E-Wallet" and "Pay Credit Card" screens confirm the transaction with details like amount and transaction ID, offering

options to view the invoice or return home. In case of failure, the "Failure to Pay" screen lets users retry or go back. There's also an option to modify trip details if needed. Unresolved payments are saved in "Unpaid Tickets," with further guidance provided. These screens ensure a user-friendly experience for handling both successful and failed payments.

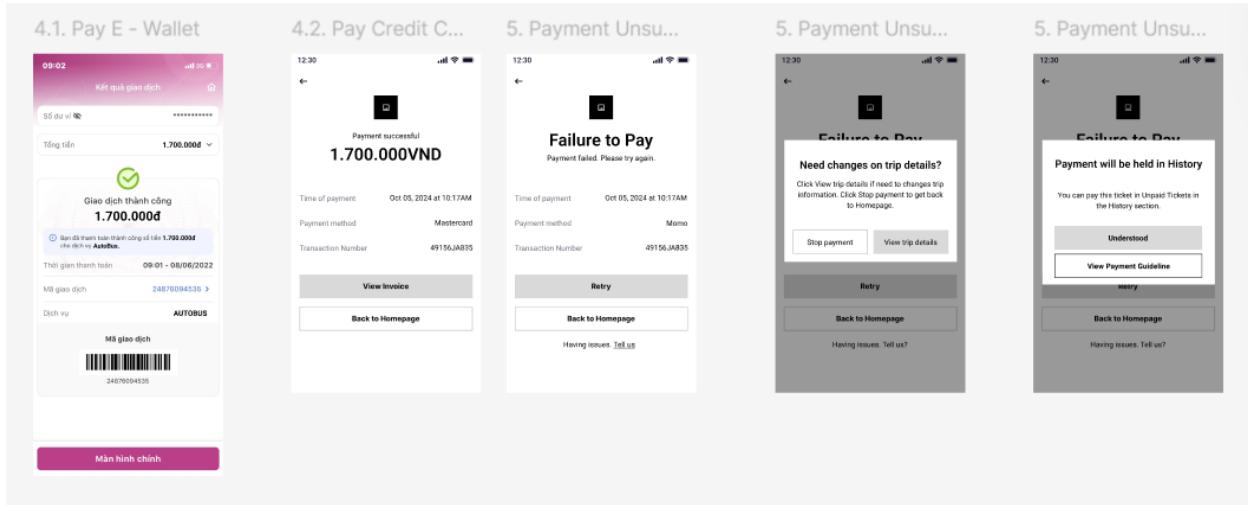


Figure 6.12. Payment

(Sources: Authors' Sources)

6.1.4. Booking History

This wireframe displays the "Booking History" section of a bus ticket booking app. The interface includes key screens like booking history, ticket details, cancellation, and service rating. Users can view trip details, make cancellations or edits, track bus status, and rate the service after each trip. The design is minimalist, allowing easy navigation and quick access to essential features.

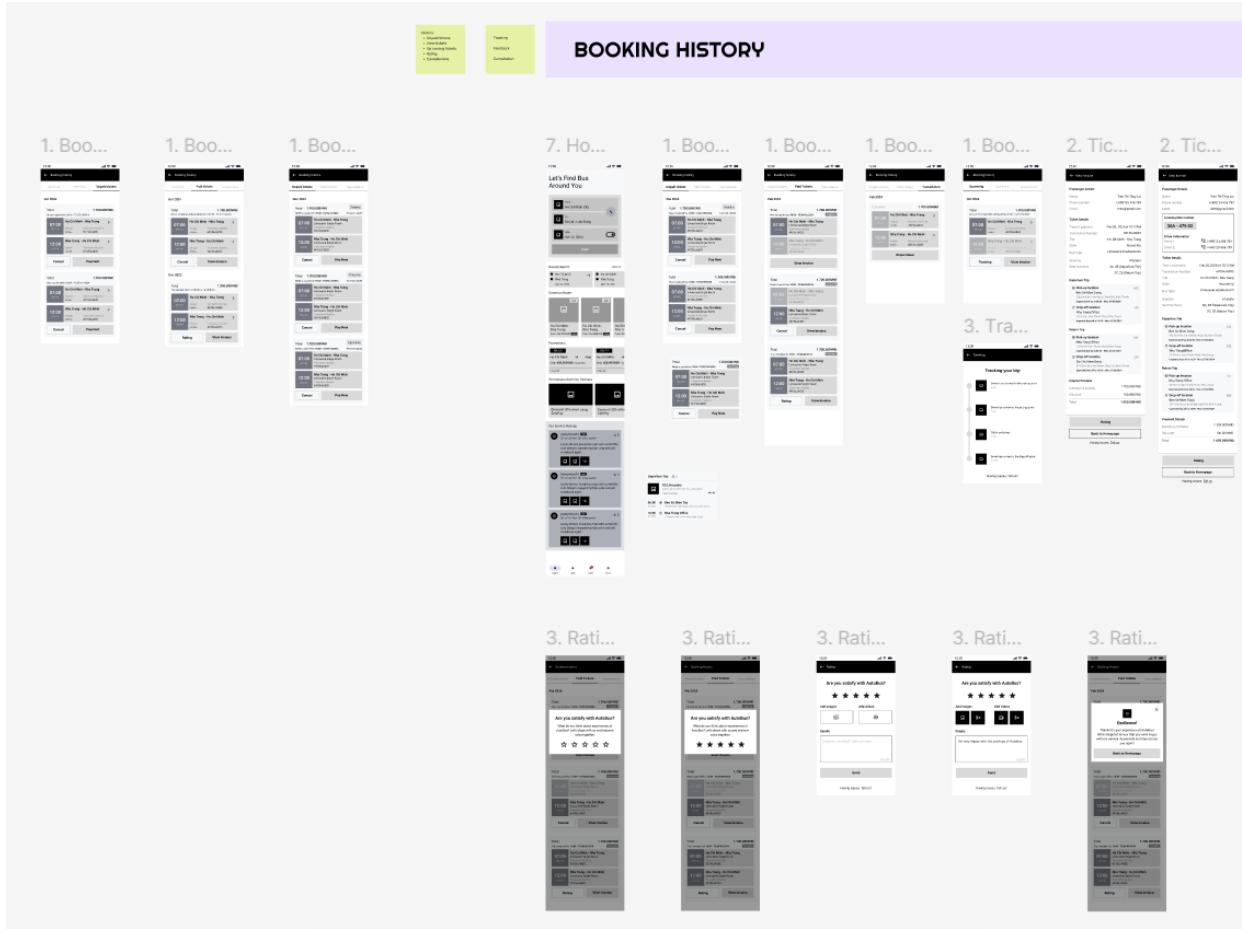


Figure 6.13. Booking History

(Sources: Authors' Sources)

6.1.3.3. View Unpaid/Paid/Upcoming Ticket/Invoice and Cancellation

This wireframe presents various screens for a bus ticket booking app, primarily focused on "Booking History" and related functions. It includes views for booking history, ticket details, and cancellation options, along with a feature to locate nearby buses. Users can view and manage their bookings, check ticket details, track buses, and provide feedback. The interface is structured for easy navigation, with each screen designed to give users quick access to essential information and actions related to their trips.

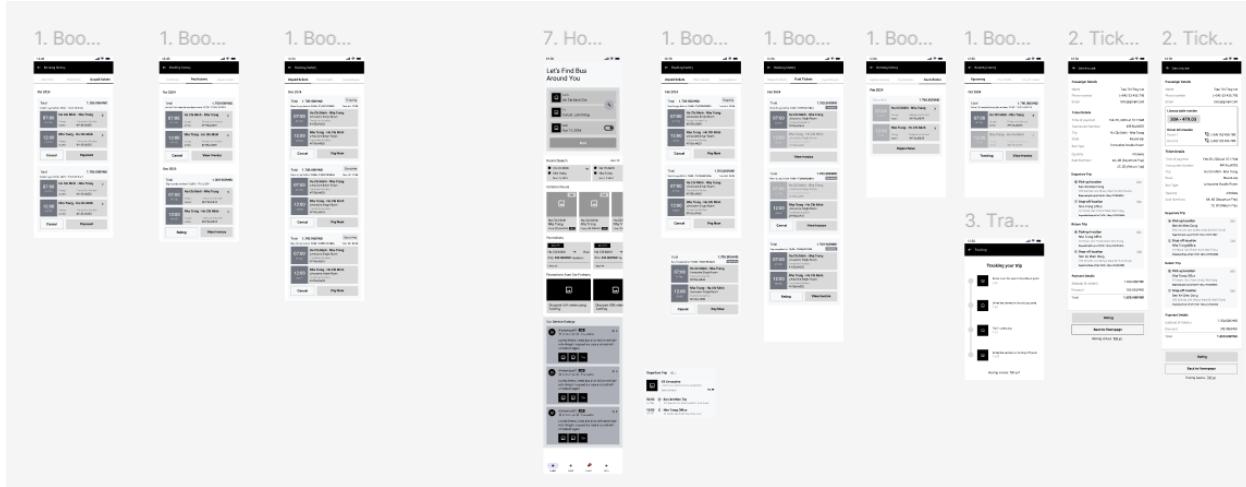


Figure 6.14. View Unpaid/Paid/Upcoming Ticket and Invoice

(Sources: Authors' Sources)

6.1.3.3. Rating

This wireframe displays the rating and feedback process within a bus ticket booking app. The flow starts with a prompt asking the user to rate their experience with the service using a star rating system. Users can add images or videos to their feedback and provide additional comments. After submitting, they receive a confirmation message thanking them for their feedback and offering a "Back to Homepage" button. This design encourages user engagement by making it easy to provide detailed feedback on their experience.

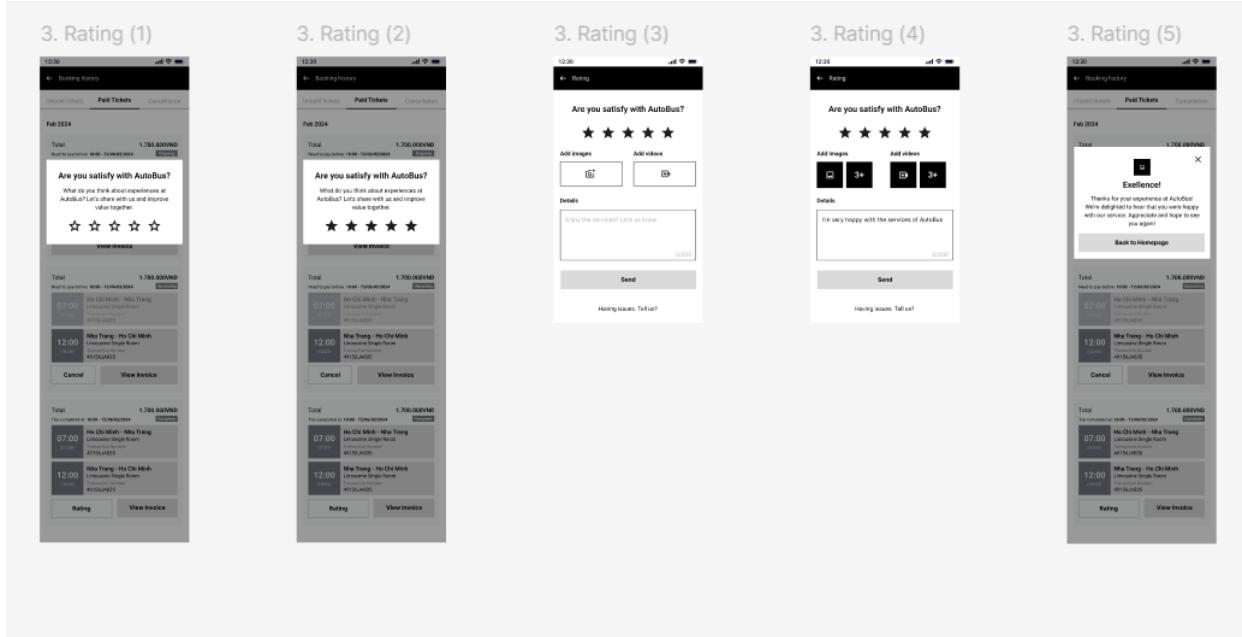


Figure 6.15. Rating and Feedback

(Sources: Authors' Sources)

6.1.5. Cancellation

The process begins on the "Booking History" screen, where the user can select a trip for cancellation. Upon confirming cancellation, the app displays a page with "Cancel Regulations" and then prompts the user to choose a reason for canceling, such as selecting the wrong route or time. After confirming, the "Details Cancellation" page shows cancellation information, including the reason, transaction number, cancellation fee, refundable amount, and trip details (departure and return trips with times and locations). The final step displays a confirmation message, informing the user that a customer service representative will contact them shortly to complete the refund process. The interface maintains a consistent, clean design throughout the flow.

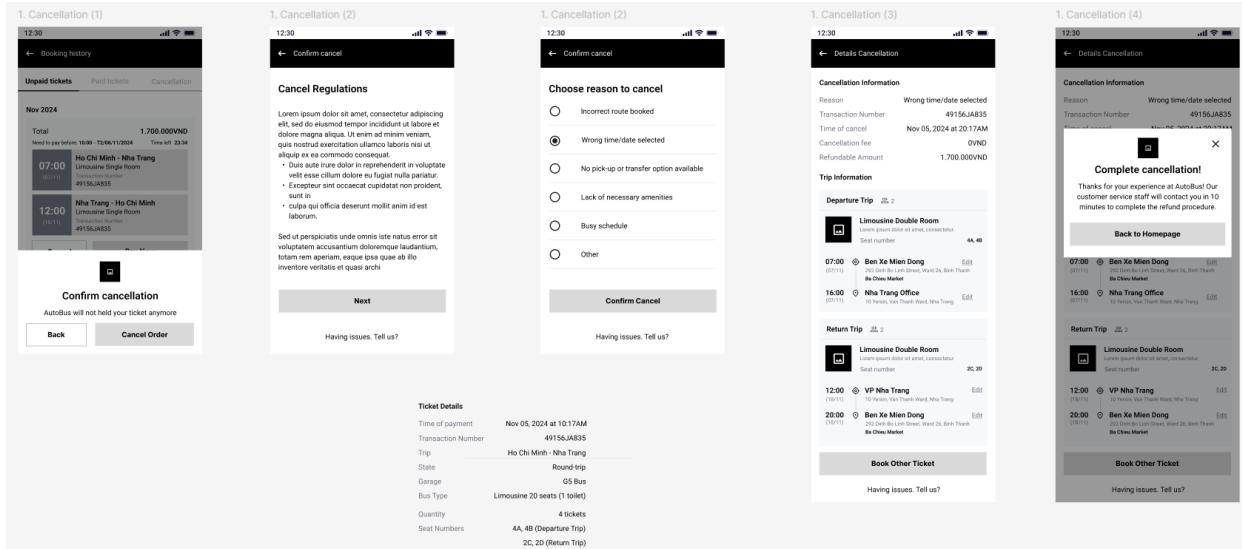


Figure 6.16. Cancellation

(Sources: Authors' Sources)

6.2. Design UI Mockup screen

6.2.1. Design System

6.2.1.1. Company Logo



Figure 6.17. AutoBus' logo

(Source: Authors' source)

The logo features the brand name "AutoBus" with a stylized letter "A" inside a circular shape on the left. The "A" is designed with a split-tone effect, using a darker shade to create a sense of depth. The circle surrounding the "A" is a soft blue, providing a modern, tech-inspired look. To the right of the text is a minimalist outline of a bus with lines indicating speed or motion, suggesting efficiency and quick service. The color palette includes different shades of blue and black, conveying a professional, reliable, and approachable brand image suitable for a travel or transportation business.

6.2.1.2 Color Palette



Figure 6.18. AutoBus' color palette

(Source: Authors' source)

The color palette of AutoBus combines light, inviting tones with deeper, stable shades. It includes a soft light blue (#95B1EE), which brings a calming and modern feel, and a gentle, warm base white (#FFFDF5) that adds a clean and open look to the design. A light green (#E7F1A8) offers a refreshing and eco-friendly touch, while a dark blue (#364C84) provides depth and a sense of reliability, ideal for buttons or headers. Together, these colors

create a balanced, user-friendly palette suitable for a professional and approachable website.

6.2.2. UI Screens of AutoBus

6.2.2.1. Account Management

This wireframe showcases the account management process for a bus ticket booking app, illustrating the flow and layout of various account-related functions in a digital interface. The design starts with basic account options like login and registration, progresses to detailed settings, and covers user preferences, privacy controls, and notification settings. Each row represents different stages in the user journey, offering a clear, structured approach to managing personal account features in a streamlined interface.

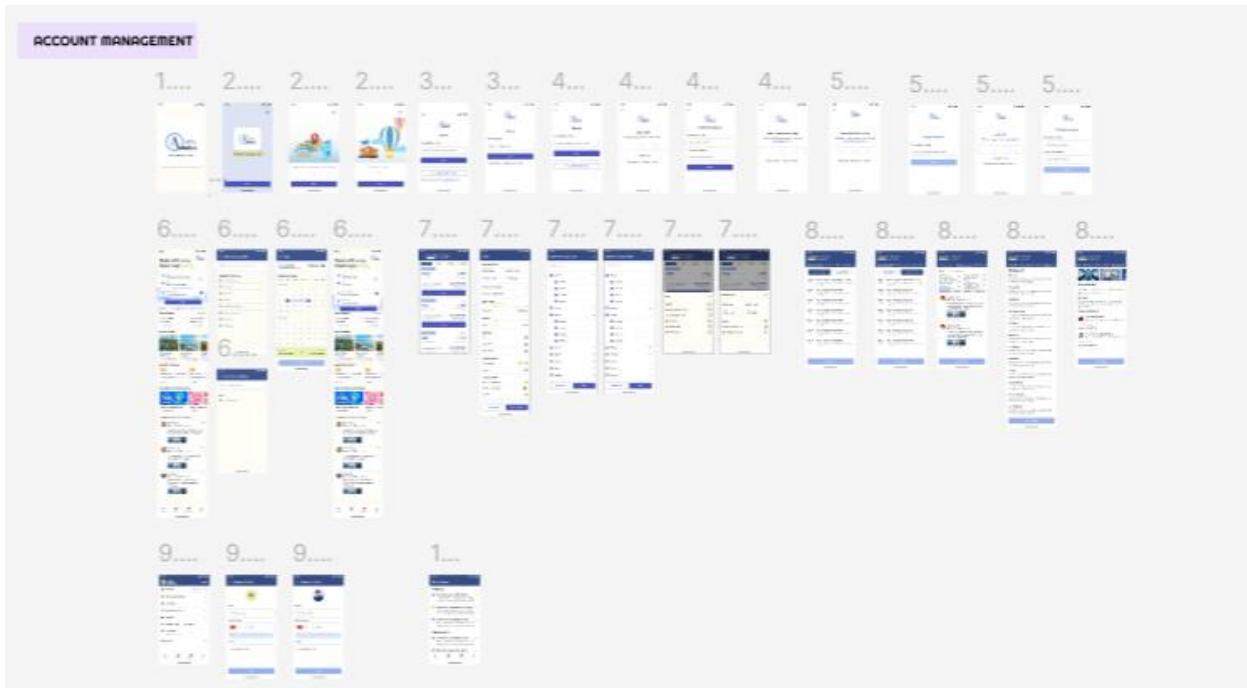


Figure 6.19. Account Management in AutoBus App

(Sources: Authors' Sources)

6.2.2.1.1. Log in /Sign up/ Forgot Password

This wireframe showcases the login, sign-up, and password recovery processes. The login screens allow users to access their accounts or create a new one. The sign-up process includes account creation and verification for security. Password recovery guides users through identity verification and resetting their password. Together, these screens ensure secure and accessible account management.

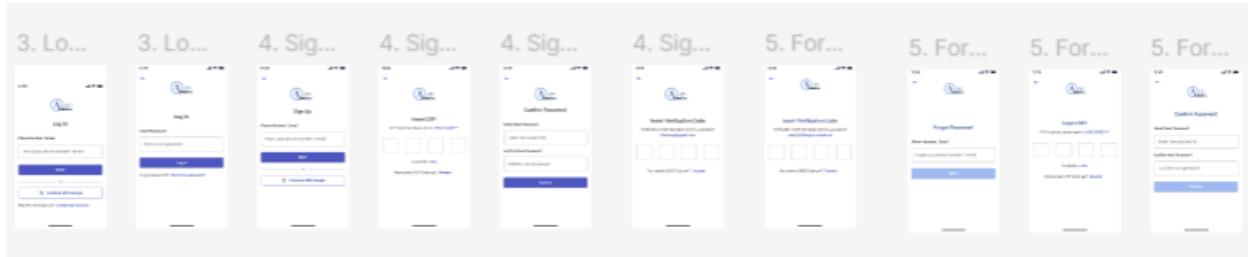


Figure 6.20. Log in /Sign up/ Forgot Password

(Sources: Authors' Sources)

6.2.2.1.2. Travel Booking and Management App UI Overview

This wireframe presents a series of mobile app UI screens for travel booking and management. The first section highlights booking options, allowing users to choose destinations, set dates, and browse deals. The second section enables users to refine their selections with filters like location and amenities. The final section provides travel tips or user reviews to aid in planning. Overall, the app's clean, organized layout makes it simple for users to book, customize, and manage their travel plans efficiently.

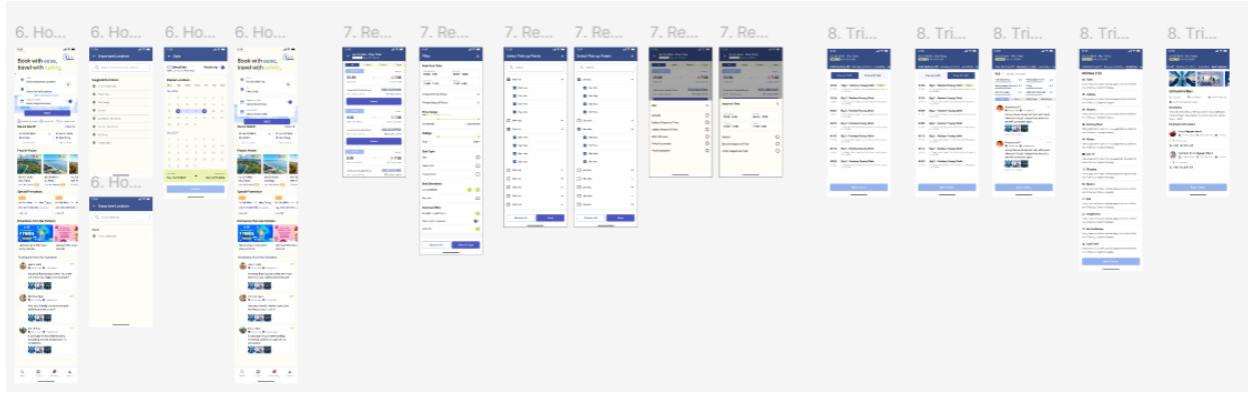


Figure 6.21. Travel Booking and Management App UI Overview

(Sources: Authors' Sources)

6.2.2.1.3. Account Management

This set of wireframes illustrates the "Account" section of a mobile app UI. The first screen displays an account overview, featuring options for profile management, notification settings, language selection, and support. The second and third screens show the profile editing interface, where users can update their profile picture, name, phone number, and email. The layout is clean and user-friendly, making it easy for users to view and modify their account information efficiently.

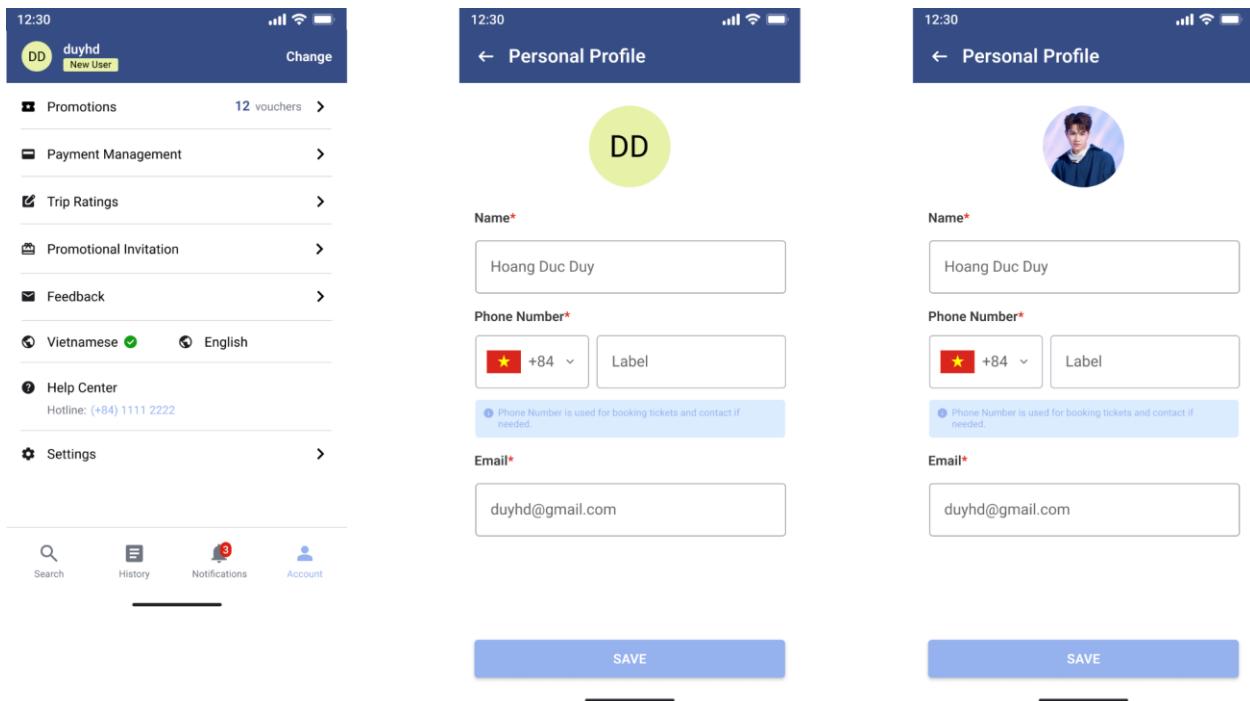


Figure 6.22. Account Management

(Sources: Authors' Sources)

6.2.2.2. Booking

This wireframe showcases the booking process for a bus ticket booking app, covering multiple stages from selecting seats to finalizing payment. The flow begins with seat selection and progresses through various steps, including passenger details, choosing amenities, and applying promo codes. The interface also includes screens for payment method selection, adding payment details, and a confirmation step. The layout maintains consistency with a clean, user-friendly design, making it easy for users to follow each step of the booking process, input necessary information, and confirm their ticket booking.

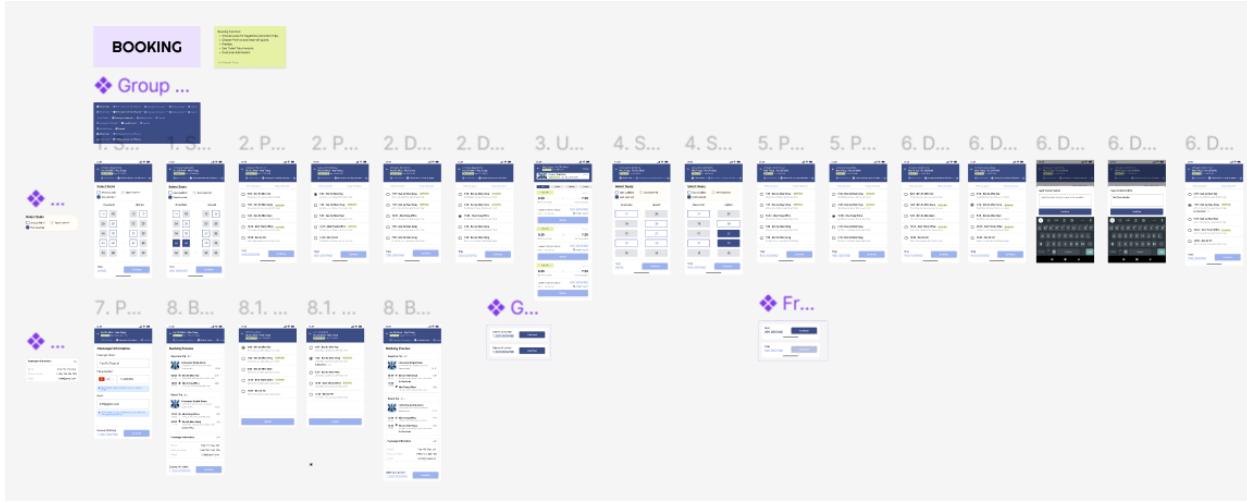


Figure 6.23. Booking Tickets Process in AutoBus App

(Sources: Authors' Sources)

6.2.2.2.1. Select pick up/drop off points (departure trip)

This wireframe shows the seat selection and pickup/drop-off point selection for a bus booking app. Users choose seats from a bus layout and select pickup/drop-off points from a list with times and availability. The design is clear, guiding users through each step with updated total pricing displayed.

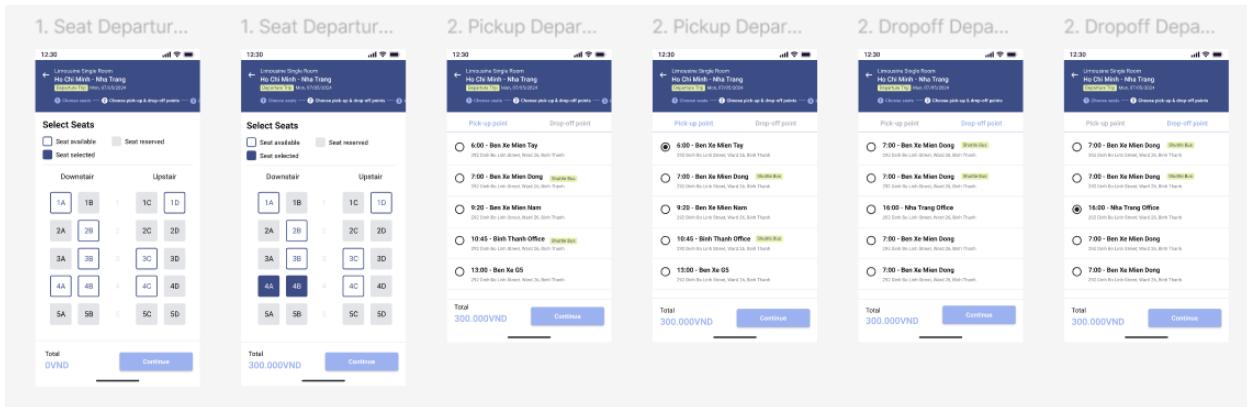


Figure 6.24. Select pick up/drop off points for Limousine Single/Standard Bus

(Sources: Authors' Sources)

6.2.2.2.2. Select pick up/drop off points (return trip)

This wireframe shows the seat selection and pickup/drop-off points for a return trip in a bus booking app. Users select seats on a bus layout, then choose pickup and drop-off points from a list with times and availability. The interface updates the total cost at each step, guiding users through the booking process efficiently.

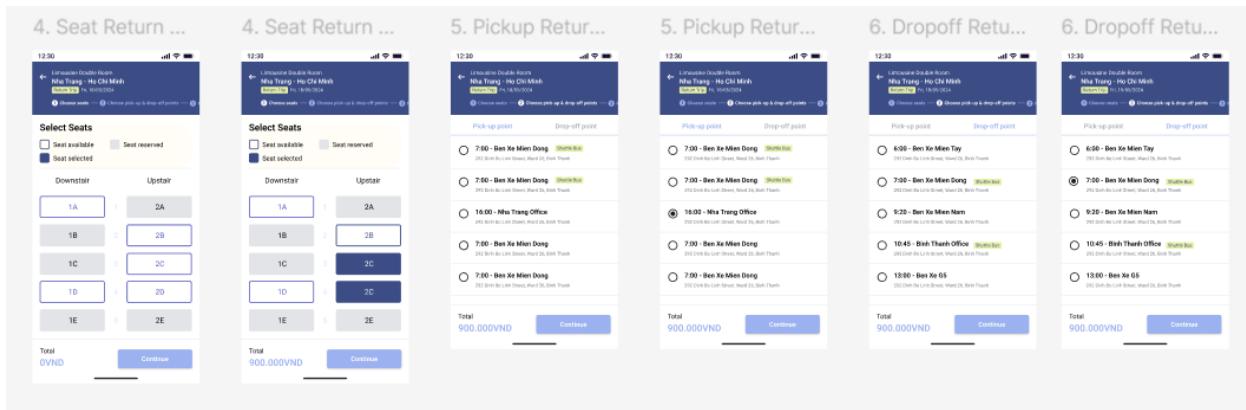


Figure 6.25. Select pick up/drop off points

(Sources: Authors' Sources)

6.2.2.3. Transaction

This wireframe displays the transaction process in a bus booking app, covering voucher selection, payment options, and outcomes for both successful and unsuccessful payments. Users can choose a voucher, select from various payment methods, and proceed with the transaction. Successful payments show confirmation details, while failed payments provide options to retry, adjust trip details, or access "Unpaid Tickets" in history for later resolution. This layout ensures a structured and user-friendly payment experience.

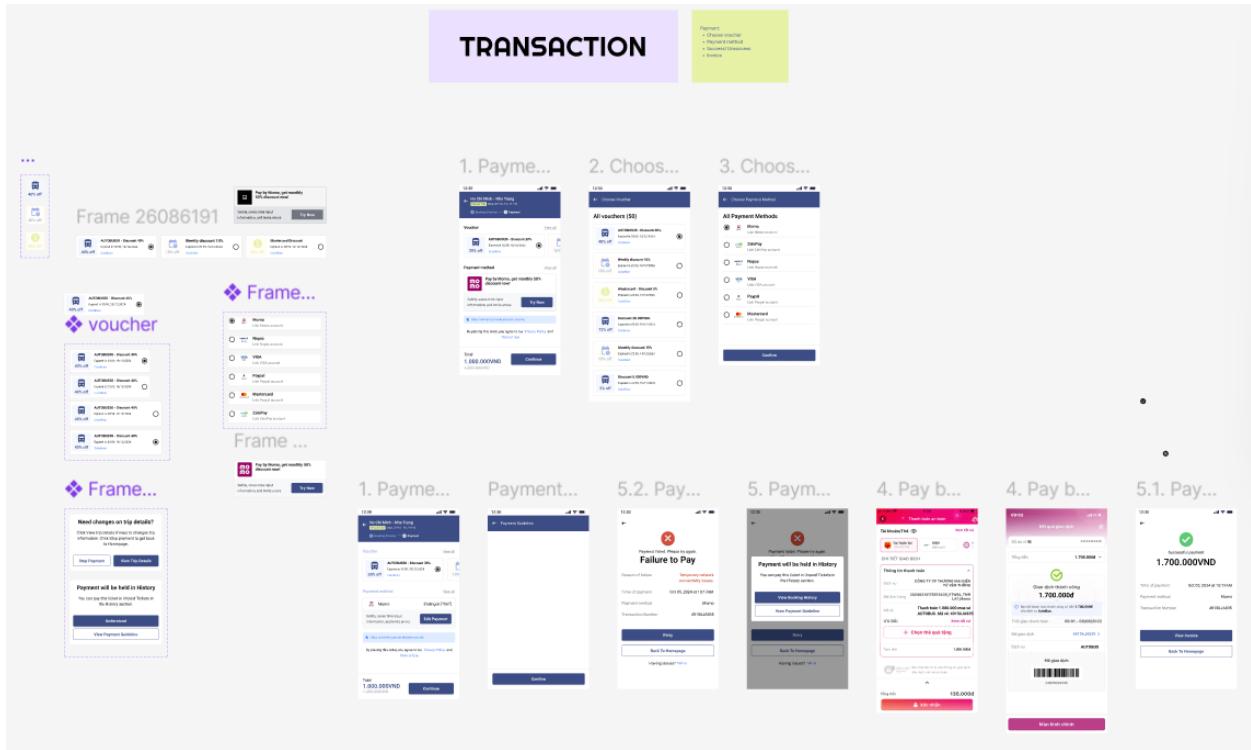


Figure 6.26. Transaction

(Sources: Authors' Sources)

6.2.2.3.1. Voucher

This wireframe outlines the voucher selection and payment options in a bus booking app. Users can choose from various discounts, each showing expiration and conditions, and select from payment methods like "Momo," "Napas," "VISA," and "PayPal," with some offering promotions (e.g., Momo's 50% monthly discount). If payment issues occur, users can adjust trip details or access payment guidelines. Unresolved payments are saved as "Unpaid Tickets," with further help available. This layout provides a smooth experience, integrating discounts, payment options, and guidance for payment issues.

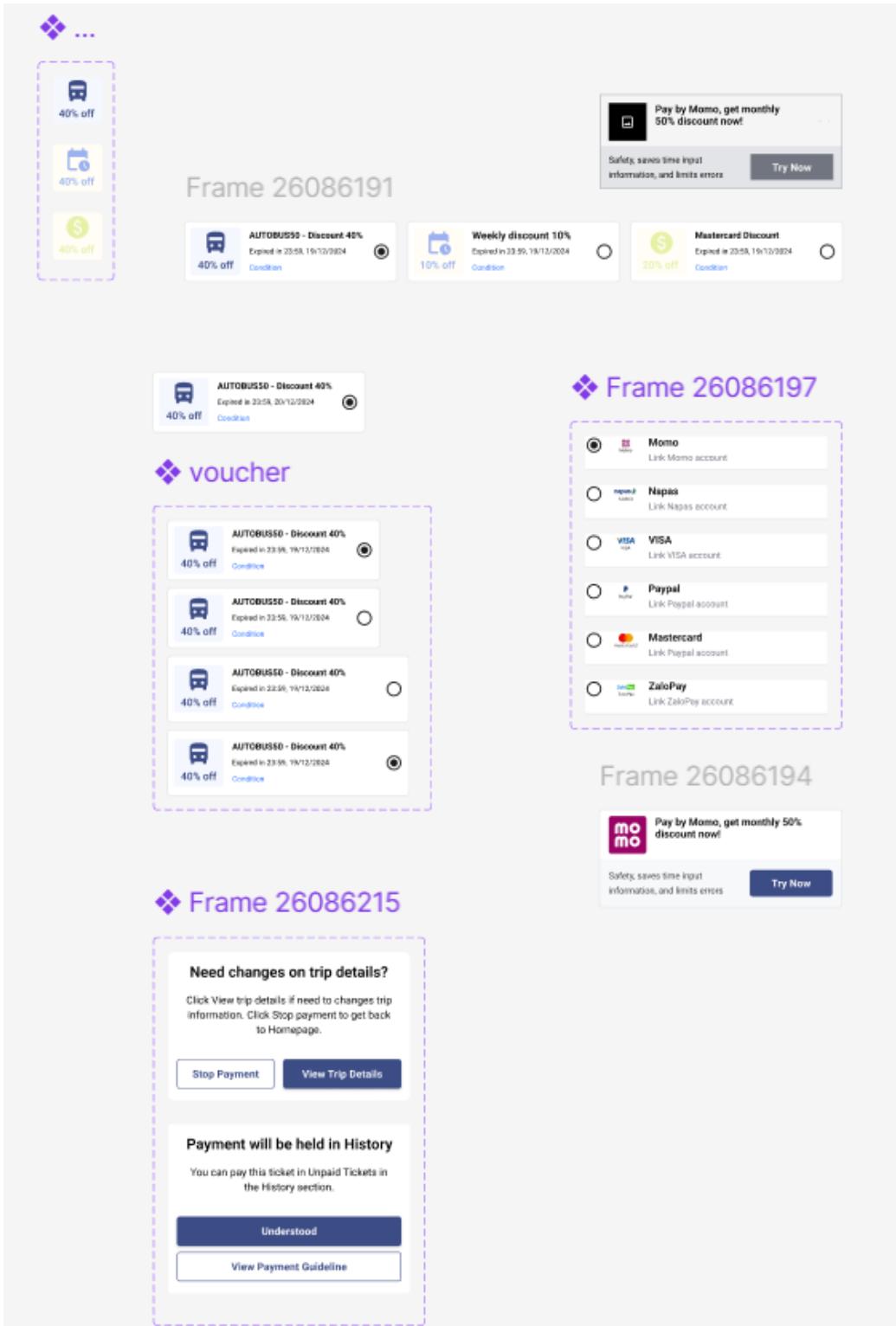


Figure 6.27. Voucher

(Sources: Authors' Sources)

6.2.2.3.2. Choose Voucher and Payment Method

The "Payment (1)" screen shows trip details, a selected voucher (e.g., "AUTOBUS30 - Discount 30%"), and a payment option with discounts, like "Pay by Momo." Users see the discounted total and a "Continue" button.

In "Choose Voucher," users select from various discounts, and in "Choose Payment Method," they pick options like "Momo" or "VISA," finalizing with "Confirm."

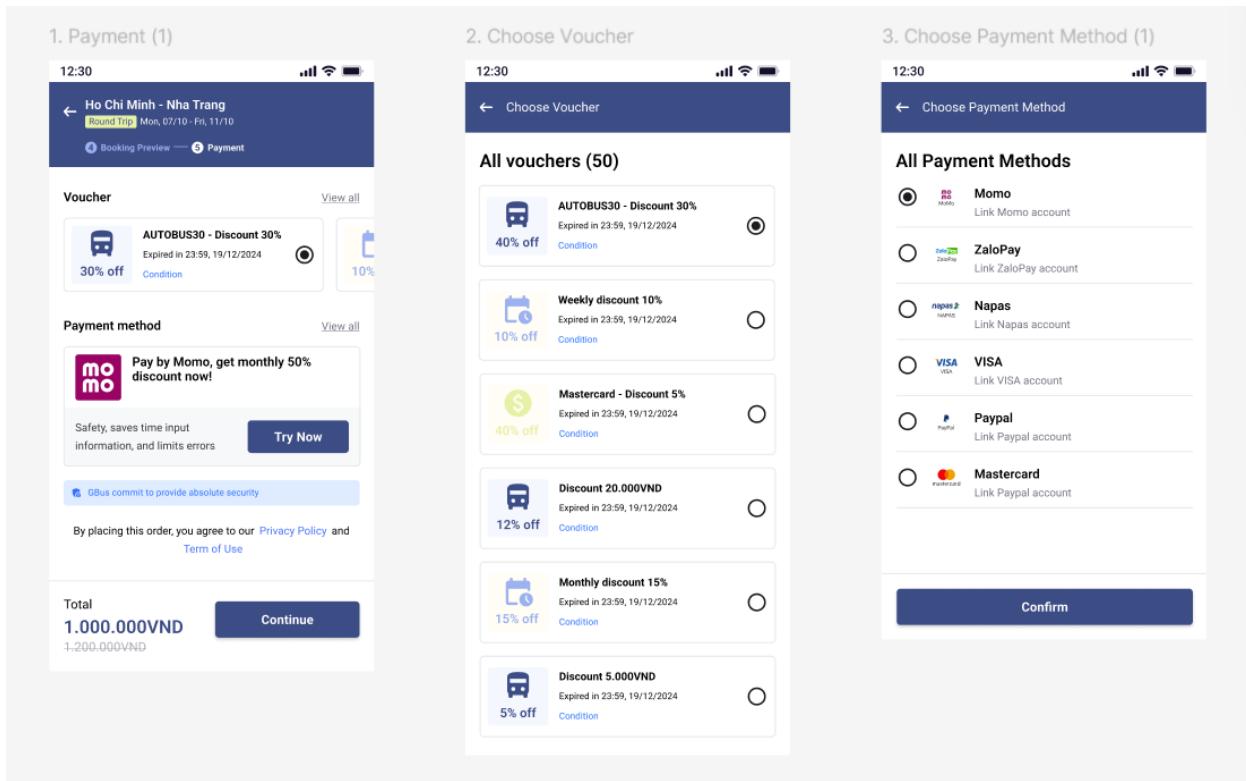


Figure 6.28. Choose Voucher and Payment Method

(Sources: Authors' Sources)

6.2.2.3.3. Payment

The "Payment (2)" screen shows an overview of selected payment options, including a voucher and method (e.g., "Momo"), with a "Continue" button to proceed. The "Payment Guideline" screen offers instructions, while "Failure to Pay" notifies users of an

unsuccessful attempt, with options to retry or go back. "Payment Held in History" saves unresolved payments under "Unpaid Tickets" with additional guidance. The "Pay by E-Wallet" screen displays transaction details, and the next screen confirms a successful e-wallet payment. The final screen confirms successful payment, allowing users to view the invoice or return home.

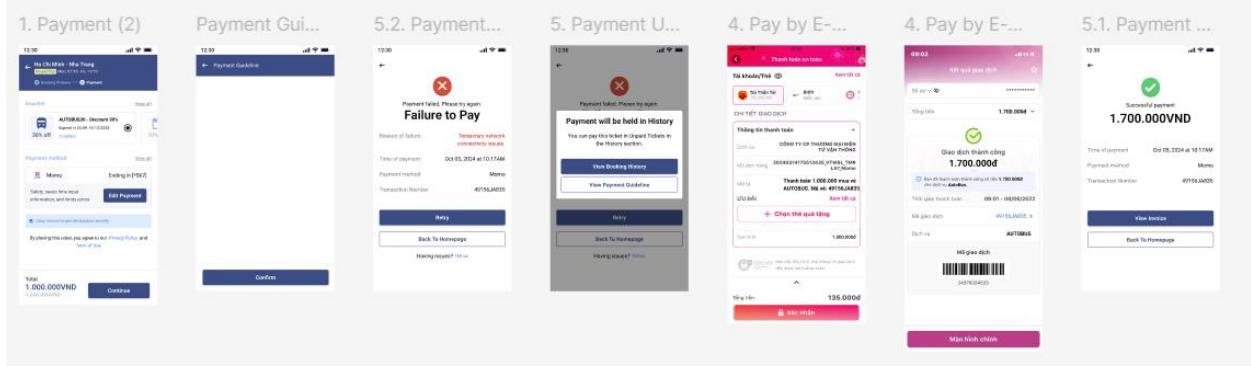


Figure 6.29. Payment

(Sources: Authors' Sources)

6.2.2.4. Booking History

This wireframe showcases the Booking History section of the app, where users can view unpaid, paid, and canceled tickets, access detailed invoices, and provide trip ratings. It offers a comprehensive flow for tracking booking statuses and submitting feedback.

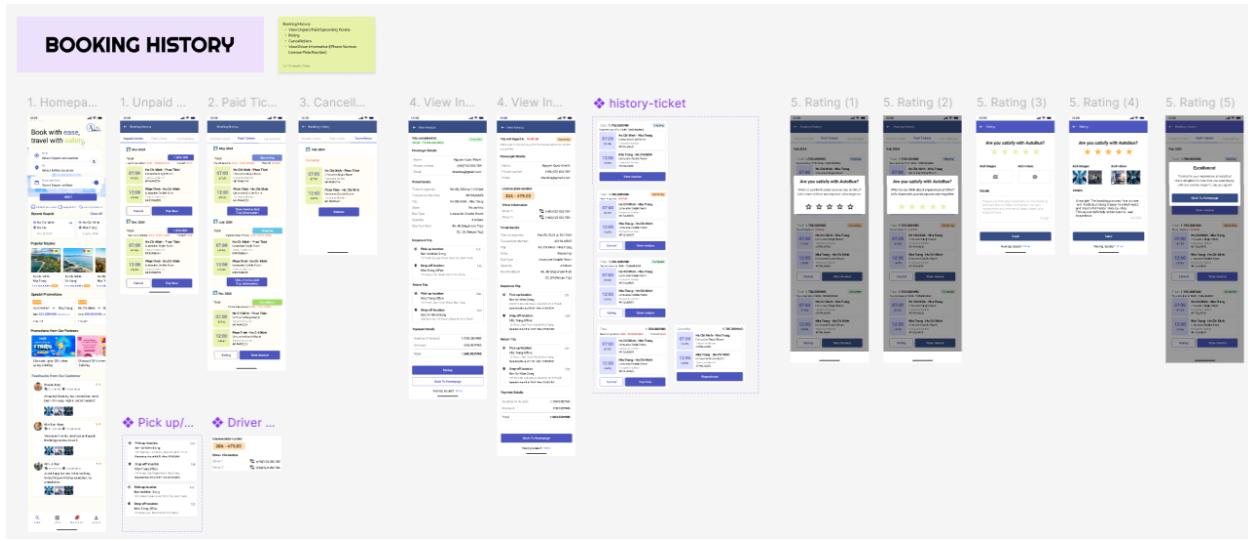


Figure 6.30. Booking History

(Sources: Authors' Sources)

6.2.2.4.1. View Unpaid/Paid/Upcoming Ticket/Invoice and Cancellation

This wireframe showcases the key screens for a bus booking app, including the homepage for route selection, ticket management (unpaid, paid, and cancellation options), and detailed invoice views.

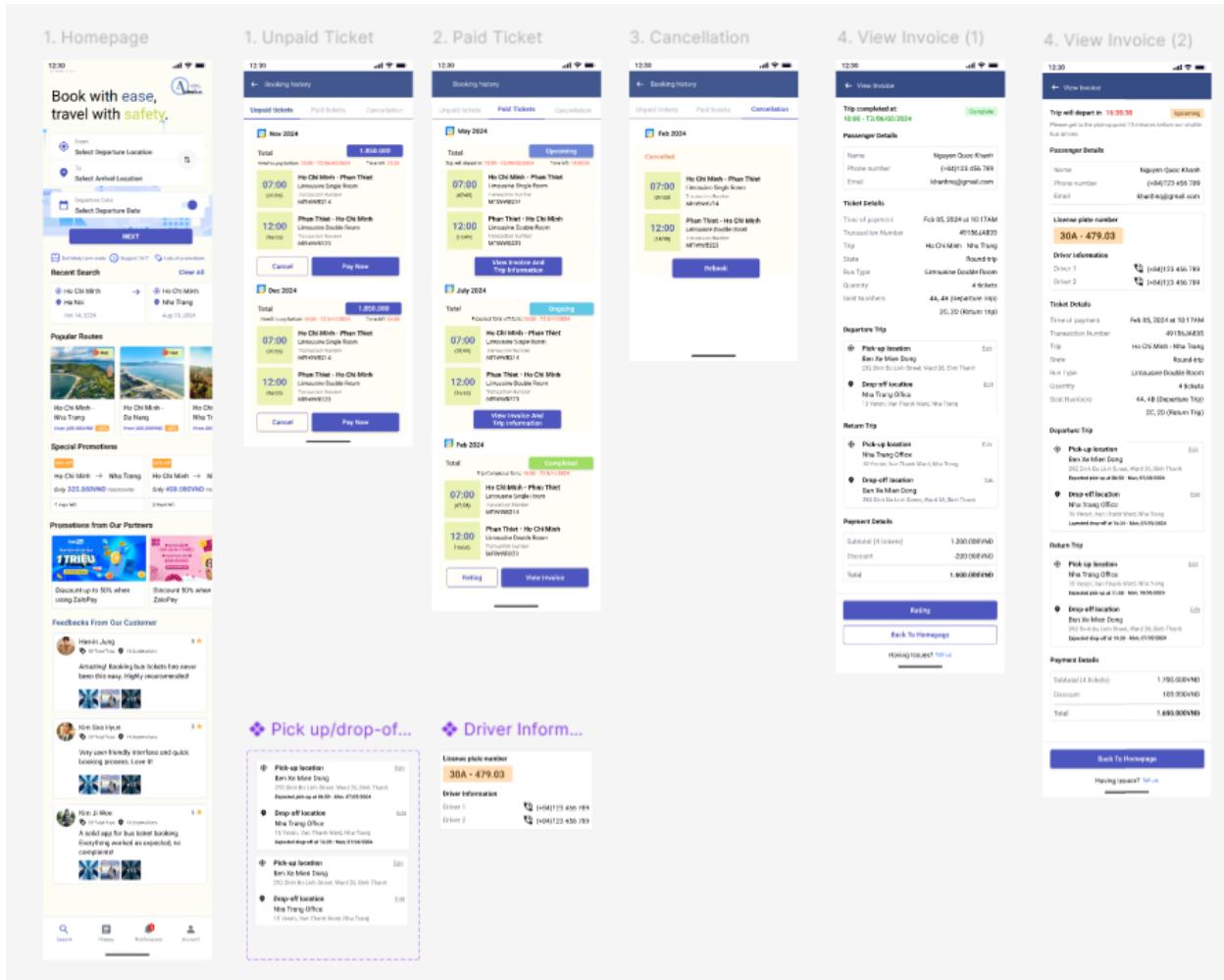


Figure 6.31. View Unpaid/Paid/Upcoming Ticket and Invoice

(Sources: Authors' Sources)

6.2.2.4.2. Rating

This wireframe shows the rating and feedback process for a completed bus trip. Users can rate their experience, add images or videos, leave detailed feedback, and view a thank-you message upon submission.

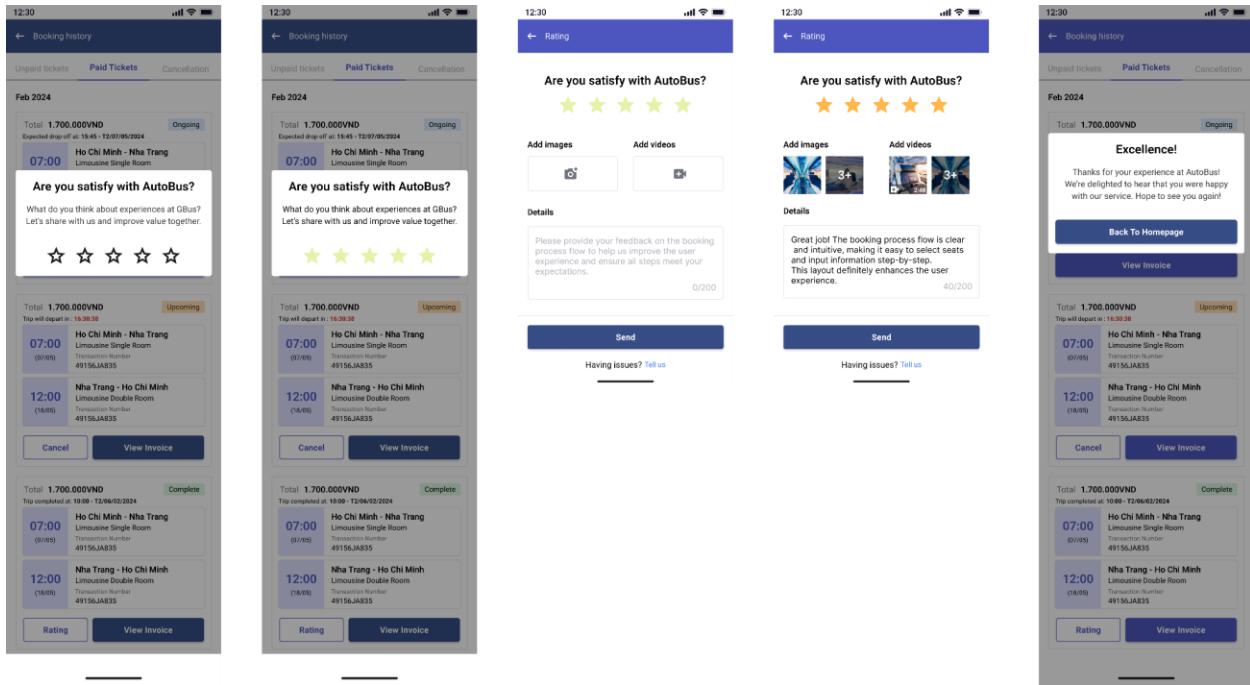


Figure 6.32. Rating and Feedback

(Sources: Authors' Sources)

6.2.2.5. Cancellation

It begins with users viewing their booking history, which includes paid tickets with options to view invoices or cancel. Upon selecting to cancel, users are presented with cancellation regulations, followed by a form to choose the reason for cancellation, such as incorrect route or busy schedule. After confirming their cancellation, they see detailed cancellation information, including trip details and refund amounts. Finally, users are shown a completion screen that thanks them for using the service and informs them about the next steps in the refund process.

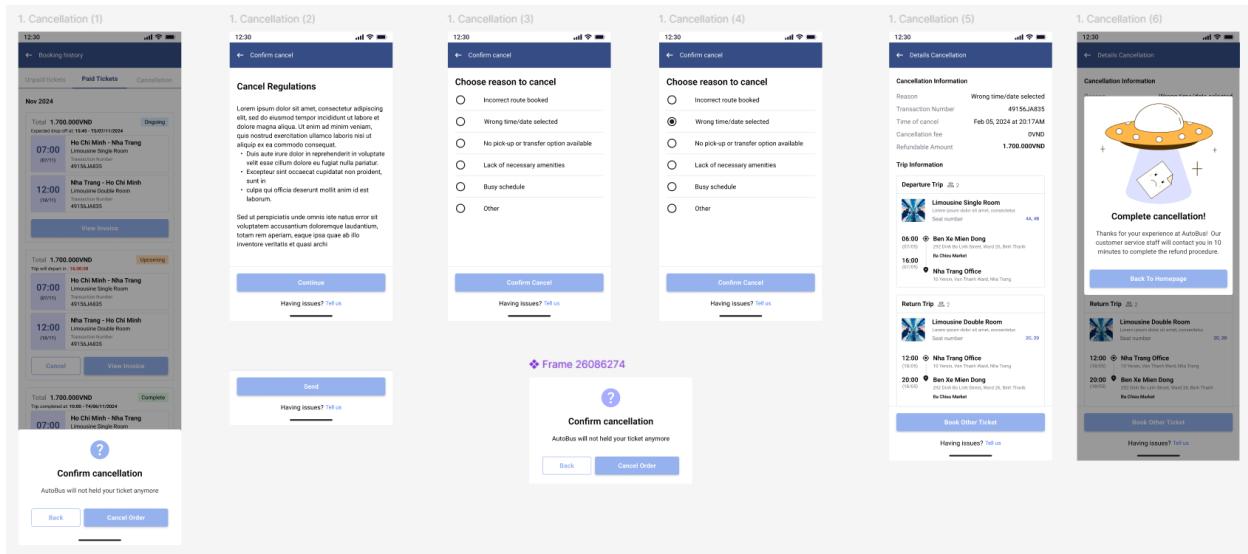


Figure 6.33. Rating and Feedback

(Sources: Authors' Sources)

CHAPTER 7: SYSTEM IMPLEMENTATION AND TESTING

7.1. Prototype

7.1.1. Account Management

7.1.1.1. Login, register and forgot password process

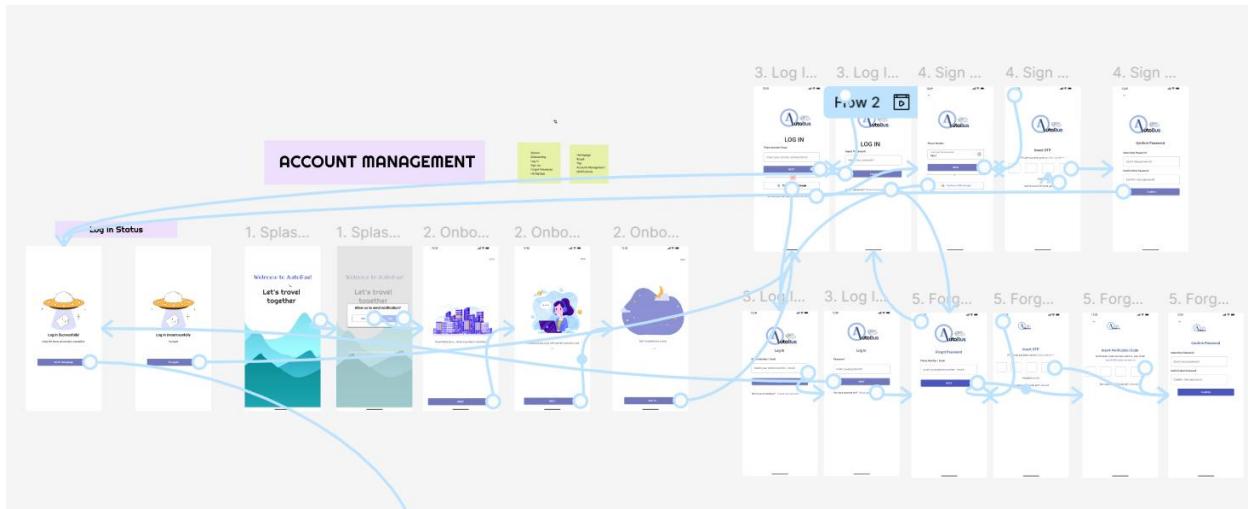


Figure 7.1. Prototype of login, registration and forgotten password process

(Source: Authors' sources)

Splash Screens: Provides an engaging and visually appealing introduction to the app. Welcomes users with key messages such as “Let’s travel together,” helping establish the app’s purpose and user experience.

Onboarding: A step-by-step walkthrough designed to guide users through the app's main features and functionalities. Helps new users familiarize themselves with the app's interface and benefits.

Login Flow: Contains a straightforward login screen where users can enter their credentials (email and password). Integrates alternative methods such as a “Forgot

Password” option, allowing users to recover access easily. Ensures secure transitions to the next stage if credentials are entered successfully.

Registration: Dedicated screens for new users to sign up by entering essential details like email, password, and other personal information. Streamlined registration flow ensures quick and hassle-free account creation.

Forgot Password: Includes a multi-step recovery process for users who have lost access to their accounts. Users can initiate recovery by entering their registered email or phone number. Verification is achieved via **one-time codes** or other secure methods, ensuring account safety.

Prototype Highlights:

- **Seamless Transitions:** The flow emphasizes smooth navigation between screens, reducing friction for users during account management tasks.
- **User-Friendly Design:** Each screen is designed to minimize confusion and guide users effectively through the process.
- **Accessibility:** Thoughtfully accommodates different user scenarios, such as first-time login, forgotten passwords, and new user registration.

This prototype demonstrates a well-structured user journey, prioritizing usability, security, and a positive onboarding experience for all users. The clear visual connections between steps ensure intuitive and efficient account management.

7.1.1.2. Searching process

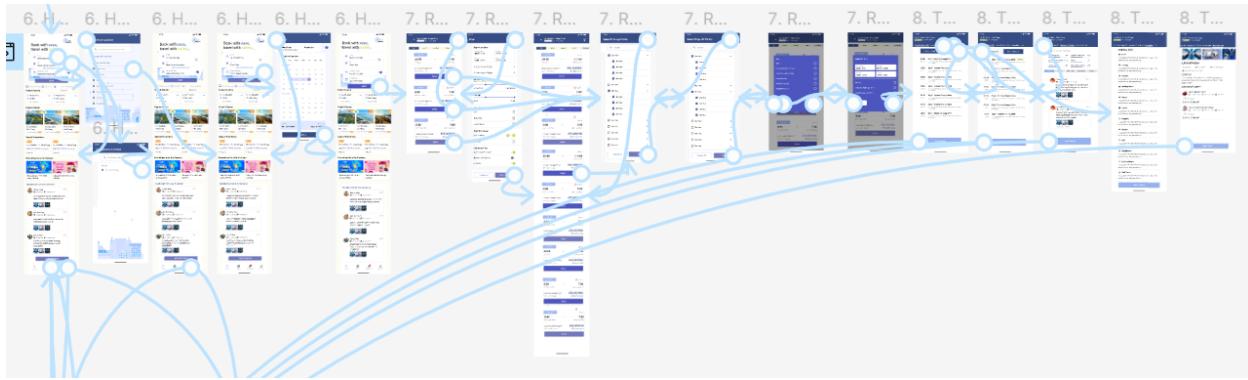


Figure 7.2 Prototype of searching process

(Source: Authors' sources)

The searching process begins with the user initiating a query and progresses through a series of steps to refine, execute, and view results. Below is a breakdown of the flow based on the prototype:

Entry Point (Search Input Interface): Users start by accessing a **search bar** or a **search page**. Besides, options are provided to input keywords or criteria for the search (e.g., destination, date, or other relevant filters).

Filter and Criteria Selection:

- Users can refine their search using filters such as:
 - Location or route selection.
 - Travel date and time ranges.
 - Additional preferences (e.g., amenities, bus types, or price range).
- Dynamic filtering ensures the options are updated based on available data.

Search Execution: After inputting keywords and selecting filters, the user executes the search by clicking a button (e.g., "Search" or "Find"). The system processes the query and fetches relevant results.

Search Results Page:

- Results are displayed in a **list view** or **grid view**, showing options such as:
 - Bus names, routes, departure/arrival times, and prices.
 - Availability status and other key details (e.g., seat availability, user reviews).
- Sorting and additional filtering options (e.g., sort by price, time, or popularity) may be provided.

Result Exploration and Selection:

- Users can click on a specific result to view more details.
- Detailed pages may include:
 - Bus details.
 - Map routes or pickup/drop-off locations.
 - Pricing breakdown.

Interactive Features: Interactive elements, such as a **map view** to visualize routes, are included. Booking or reserving a specific result can be initiated directly from this stage.

Error Handling (Optional): If no results match the query, the system provides:

- Suggestions to modify the search query.
- Prompts for broader criteria or alternative routes.

Navigation and Confirmation: Users finalize their choice and proceed to booking or saving the search result. Navigation flows back to the main system for further actions like payment or user account updates.

7.1.1.2. Account management process

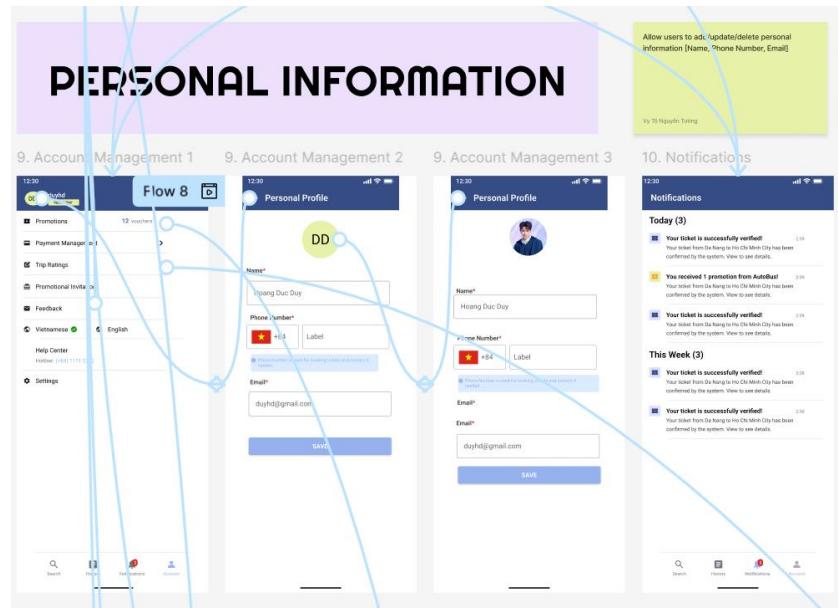


Figure 7.3. Prototype of account management process

(Source: Authors' sources)

The account management process begins with the user accessing their personal information and provides a series of steps to view, update, and manage their account details effectively. Below is a breakdown of the flow based on the prototype:

Accessing the Personal Information Page: Users start by navigating to the **Personal Information** page. Options to edit profile details or access additional account settings are prominently displayed. This page displays key details, including:

- The user's profile picture or initials.
- Basic information such as full name, email, and contact number.

Navigating Account Management Options: Users can choose from the following management options:

- **Edit Profile:** Allows updating of basic personal information, including name, email, phone number, or profile picture.
- **Change Password:** A dedicated option for updating the login password to enhance security.
- Each option leads to a specific screen, simplifying the interaction process.

Editing Personal Information: On the **Edit Profile** screen, users can update their personal details:

- Input fields are pre-filled with current data for easy editing.
- After making changes, users confirm updates by clicking a "Save" button, ensuring the modifications are stored in the system.

Changing Password: Built-in validations ensure that the new password meets security standards, and users are notified in case of mismatches or weak passwords. The **Change Password** screen prompts users to:

- Enter their current password.
- Input and confirm a new password.

Viewing Notifications: A separate **Notifications** section provides updates related to account activity:

- Notifications include login attempts, successful profile changes, or alerts about account security.
- Users can configure their notification preferences, opting for email, SMS, or in-app notifications.

Navigation and User-Friendly Design: The system ensures easy navigation between sections using:

- A menu or tab system for quick access to profile, settings, and notifications.

- Clearly marked "Back" and "Save" buttons to guide users through the process.

This account management process provides a clean and effective interface for users to maintain and update their information while ensuring security and ease of use.

7.1.2. Booking

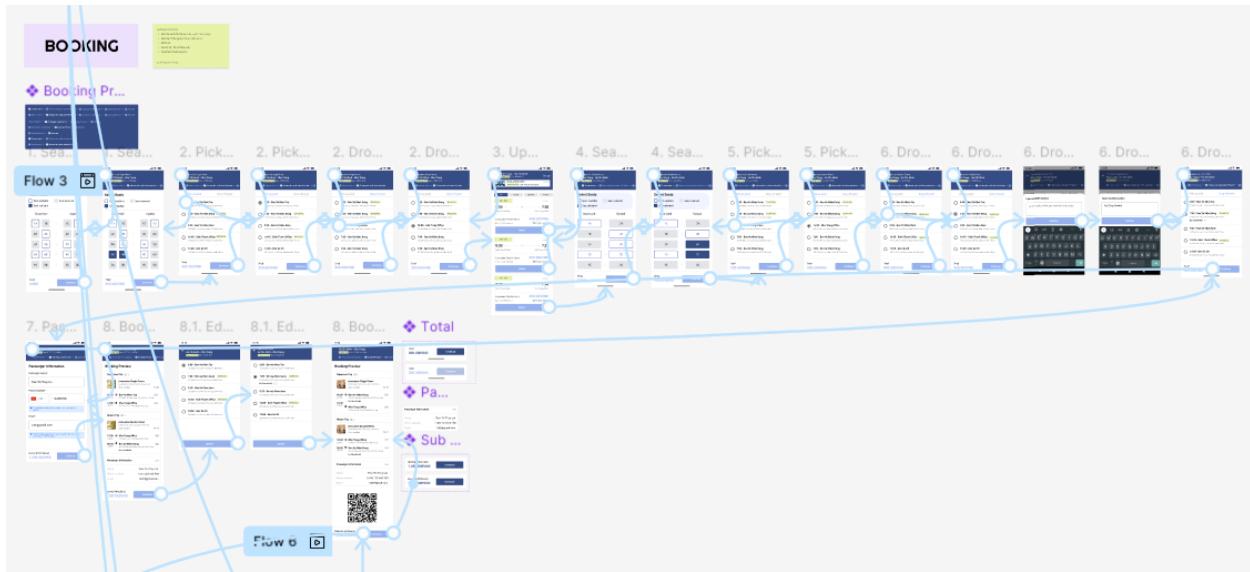


Figure 7.4. Prototype of Booking Process

(Sources: Authors' Sources)

The flow begins with selecting a seat, where users can pick options related to seats, pickup points, and drop-off locations. The prototype progresses through several steps, covering different stages in the booking journey:

Seat Selection: Users choose their preferred seat from the available options.

Pickup Point: Users select a pickup point based on their convenience.

Drop-off Point: Users choose a drop-off location.

Passenger Information: Users input details such as passenger name, contact information, etc.

Payment and Summary: Users are shown a summary of their booking details, including fare and any additional charges. This section also includes options for payment processing.

7.1.3. *Transaction*

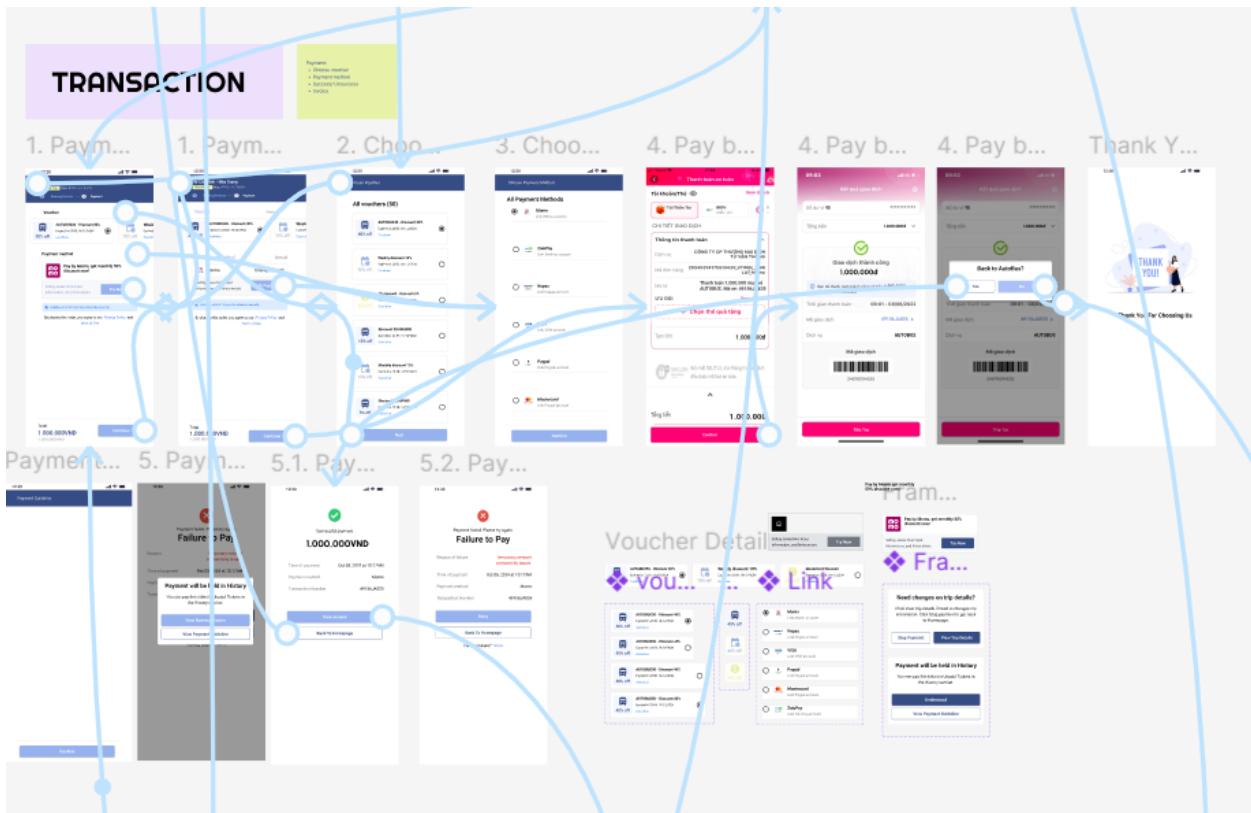


Figure 7.5. Prototype of transaction process

(Source: Authors' sources)

The transaction process begins with the user selecting a payment method and progresses through a series of steps to confirm the payment, apply discounts, and complete the transaction. Below is a breakdown of the flow based on the prototype:

Initiating the Payment: Users navigate to the **Transaction** section from their booking or order page. A summary of the booking/order details is displayed, including: Total amount due, and items or services being paid for.

Choosing a Payment Method: Users are directed to specific screens depending on the selected method. The user selects a preferred payment method, such as::

- Credit/Debit Card.
- Digital Wallets (e.g., PayPal, Google Pay).
- Bank Transfer or other supported methods.

Entering Payment Details: The user provides the necessary payment details, which may include:

- Card number, expiry date, and CVV for card payments.
- Login credentials for digital wallets.
- Bank account details for direct transfers.

Applying Discounts or Vouchers: Users can apply promotional vouchers or discount codes to reduce the transaction amount. A validation process ensures that the voucher is applicable and adjusts the total accordingly.

Payment Confirmation: Users confirm the payment by clicking on a “Pay Now” or equivalent button. A summary page is displayed for users to review:

- Payment method.
- Final amount after discounts.
- Booking or order details.

Processing Payment: The system processes the payment, displaying a loading screen or progress bar.

- Upon successful payment, a confirmation message or receipt is generated.
- In case of payment failure, users are prompted to retry or select another payment method.

Receiving the Receipt and Thank-You Message: Once the payment is completed, users are directed to a **Thank You** page. A receipt is provided, including:

- Transaction ID.
- Booking or order details.
- Payment method and amount paid.

Additional Options: They can also navigate back to the main menu or explore further actions, such as viewing their bookings or account information.

7.1.4. Booking History

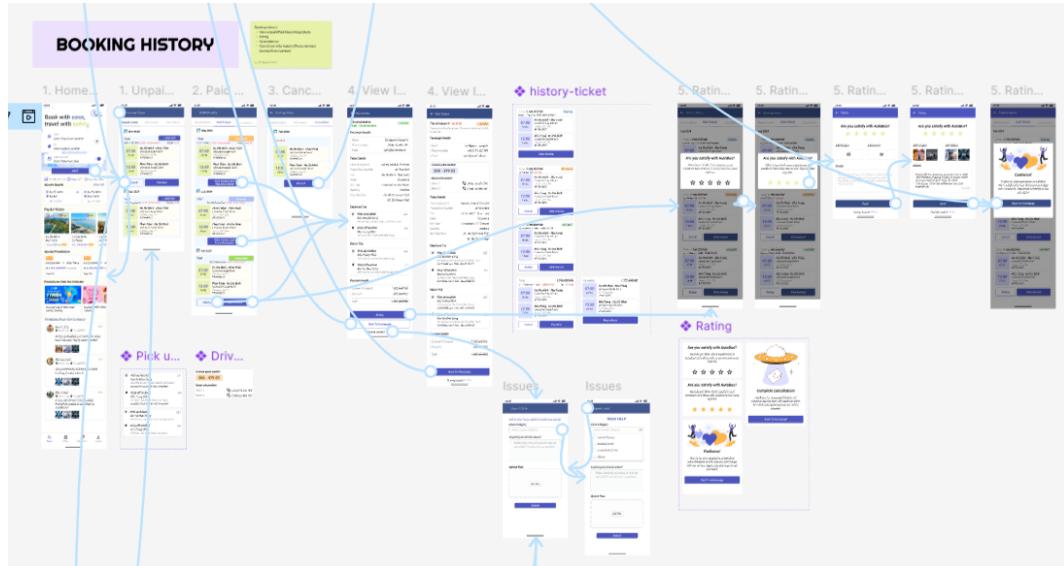


Figure 7.6. Prototype of Booking History Process

(Sources: Authors' Sources)

Home Screen Access: Users access the booking history from the home screen, where they can review past trips.

Trip Status:

- **Unpaid Trips:** Displays a list of bookings awaiting payment, allowing users to complete payment.
- **Paid Trips:** Shows successfully paid trips, enabling users to view trip details.
- **Canceled Trips:** Lists bookings that were canceled, with reasons or status updates.

Detailed View:

- Users can click on any trip to view detailed information about the booking, including trip dates, seat information, pickup and drop-off points, and driver details.
- A “history-ticket” screen appears to provide a ticket-like view, enhancing the sense of a completed trip.

Trip Rating:

- After viewing a completed trip, users can rate their experience. The flow includes a rating screen with options to give stars, leave comments, and possibly upload images related to the experience.
- This feedback process is detailed with multiple screens, offering a seamless, user-friendly experience for providing feedback.

Issue Resolution:

- If users encounter issues with a booking, they can access an "Issues" screen. This screen allows them to report problems or seek help, potentially providing options to choose a specific type of issue and describe the problem.

7.1.5. *Cancellation*

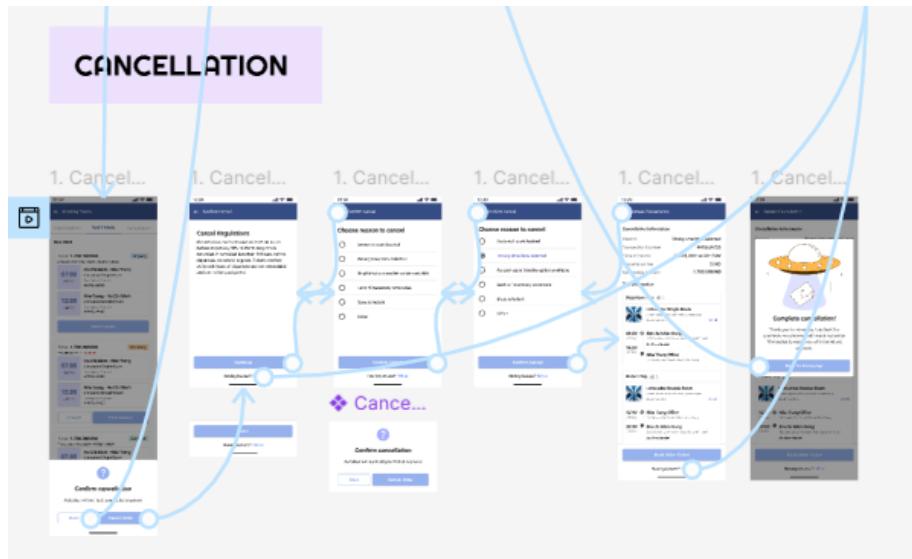


Figure 7.7. Prototype of Cancellation Process

(Sources: Authors' Sources)

Initiate Cancellation: The process begins with the user selecting the booking they wish to cancel. The first screen displays details of the booking, allowing the user to confirm that this is the trip they want to cancel.

Cancellation Confirmation: The next screen prompts the user to confirm their intent to cancel. This acts as a safeguard to ensure that the user really wants to proceed with the cancellation.

Choose Reason for Cancellation: After confirming the cancellation, the user is asked to provide a reason for the cancellation. Multiple options are presented, such as schedule changes, personal reasons, or issues with the service. This step helps the service provider gather feedback on why users are canceling.

Additional Details: If necessary, users may be asked for additional details to clarify their reason for cancellation. This could involve a text box for more specific feedback, which helps the company understand user needs and improve services.

Cancellation Fee or Policy Information: Users are shown information about any applicable cancellation fees or policies. This ensures they are aware of any financial or policy-related implications of canceling the booking.

Final Confirmation and Success Screen: After agreeing to any terms or conditions, the user can proceed to finalize the cancellation. The last screen confirms that the cancellation is complete with a success message or graphic, providing a sense of closure to the process.

7.2. Test Scenarios and Test Case

7.2.1. *Summary testing*

Table 7.1. Summary testing

Count of Result	Expected Result					
	Scenario	Blocked	Failed	Passed	Pending	Total
1. Login		1	1	2		4
2. Sign Up			1	1		2
3. Forgot Password			2	1		3
4. Update Personal Profile			2	2		4
5. Check Notifications				1		1
6. Bus Searching				4		4
7. Result List			3			3
8. Trip Details				7		7
9. Booking			2	9		11

10. Transaction		1	7		8
11. Booking History		4	7		11
Cancellation			3		3
Total	1	16	44	0	61

7.2.2. *Testing*

Table 7.2. Testing

ID	FUNCTION	SCENARIO	TEST CASE NAME	TEST PROCEDURE	EXPECTED RESULT	RESULT
TC001	Account Management	1. Login	Login with valid phone number/ email	1. Navigate to Log In screen. 2. Enter Phone Number/ Email as " anhnq22406c@st.uel.edu.vn ". 3. Hit "Next" button.	Passed	Passed
TC002	Account Management		Login with invalid phone number/ email	1. Navigate to Log In screen. 2. Enter Phone Number/ Email as " anhnq22406c@st.uel.edu.vn ". 3. Hit "Next" button.	Blocked	Failed
TC003	Account Management		Input valid password	1. Enter password as "autobus123". 2. Hit "Log In" button.	Passed	Passed
TC004	Account Management		Input invalid password	1. Enter password as "autobus123". 2. Hit "Log In" button.	Failed	Failed
TC005	Account Management	2. Sign Up	Sign Up with non-registered account	1. Click "Create an account" on the Log In screen. 2. Navigate to the "Sign Up" screen. 3. Enter Phone Number/ Email. 4. Click "Next". 5. Enter Verification Code/ OTP Code. 6. Enter password and confirm password. 7. Hit "Sign Up" button.	Passed	Passed
TC006	Account Management		Sign Up with Invalid	1. Click "Create an account" on the Log In screen.	Failed	Failed

			Verification Code/ OTP Code	<ol style="list-style-type: none"> 2. Navigate to the “Sign Up” screen. 3. Enter Phone Number/ Email. 4. Enter Verification Code/ OTP Code. 5. Enter password. 6. Hit “Sign Up” button. 		
TC007	Account Management	3. Forgot Password	Forgot Password with registered account	<ol style="list-style-type: none"> 1. Click “Reset Password” on the Login with Password screen. 2. Navigate to the Forgot Password screen. 3. Enter Phone Number. 4. Hit “Next” button. 5. Enter Verification Code/ OTP Code. 6. Enter password. 7. Hit “Verify” button. 	Passed	Passed
TC008	Account Management		Forgot Password with non-registered account	<ol style="list-style-type: none"> 1. Click “Reset Password” on the Login with Password screen. 2. Navigate to the Forgot Password screen. 3. Enter Phone Number. 4. Hit “Next” button. 5. Enter Verification Code/ OTP Code. 6. Enter password. 7. Hit “Verify” button. 	Failed	Failed
TC009	Account Management		Forgot Password with Invalid Verification Code/ OTP	<ol style="list-style-type: none"> 1. Click “Reset Password” on the Login with Password screen. 2. Navigate to the Forgot Password screen. 3. Enter Phone Number/ Email. 	Failed	Failed

			Code.	4. Enter Verification Code/ OTP Code. 5. Enter password. 6. Hit “Confirm” button.		
TC010	Account Management	4. Update Personal Profile	Provide valid Personal Profile	1. Click “Account” on the bottom bar (navigation bar) from the Homepage screen. 2. Navigate to the Account Management screen. 3. Click on “Change” button. 4. Navigate to the Personal Profile screen. 5. Change Personal Information. 6. Hit “Save” button.	Passed	Passed
TC011	Account Management		Provide invalid email	1. Click “Account” on the bottom bar (navigation bar) from the Homepage screen. 2. Navigate to the Account Management screen. 3. Click on “Change” button. 4. Navigate to the Personal Profile screen. 5. Change Personal Information. 6. Hit “Save” button.	Failed	Passed
TC012	Account Management		Provide Invalid Phone Number	1. Click “Account” on the bottom bar (navigation bar) from the Homepage screen. 2. Navigate to the Account Management screen.	Failed	Failed

				3. Click on “Change” button. 4. Navigate to the Personal Profile screen. 5. Change Personal Information. 6. Hit “Save” button.		
TC013	Account Management	Provide Duplicate Phone Number		1. Click “Account” on the bottom bar (navigation bar) from the Homepage screen. 2. Navigate to the Account Management screen. 3. Click on “Change” button. 4. Navigate to the Personal Profile screen. 5. Change Personal Information. 6. Hit “Save” button.	Failed	Failed
TC014	Account Management	Provide Photo of User Profile		1. Click “Account” on the bottom bar (navigation bar) from the Homepage screen. 2. Navigate to the Account Management screen. 3. Click on “Change” button. 4. Navigate to the Personal Profile screen. 5. Change Personal Information. 6. Hit “Save” button.	Passed	Passed
TC015	Account Management	5. Check Notification	Check latest notification	1. Click “Notifications” on the bottom bar from the Homepage Screen 2. Navigate to the Notification Screen	Passed	Passed

				3. Click on 1 notification item		
TC16	Booking	6. Bus Searching	Provide Departure Location	<ol style="list-style-type: none"> 1. Click “Select Departure Location” on the Bus Searching Section from the Homepage Screen 2. Navigate to the List of Departure Locations Screen 3. Input a location keyword 4. Select a location 5. Navigate to the Homepage with updated Departure Location 	Passed	Passed
TC17	Booking		Provide Arrival Location	<ol style="list-style-type: none"> 1. Click “Select Arrival Location” on the Bus Searching Section from the Homepage Screen 2. Navigate to the List of Arrival Locations Screen 3. Input a location keyword 4. Select a location 5. Navigate to the Homepage with updated Arrival Location 	Passed	Passed
TC18	Booking		Provide One-way Ticket Trip	<ol style="list-style-type: none"> 1. Click “Select Departure Date” on the Bus Searching Section from the Homepage Screen 2. Display Schedule pop-up 3. Select a date 4. Navigate to the Homepage with updated Schedule 5. Hit “Find” button 	Passed	Passed

TC19	Booking		Provide Round Trip	<ol style="list-style-type: none"> 1. Switch the trip type on the Bus Searching Section from the Homepage Screen into the Round Trip 2. Click “Select Departure Date” on the Bus Searching Section from the Homepage Screen 3. Display Schedule pop-up 4. Select departure date and arrival date 5. Navigate to the Homepage with updated Schedule 6. Hit “Find” button 	Passed	Passed
TC20	Booking	7. Result List	Select Quick Filter	<ol style="list-style-type: none"> 1. Navigate to the Result List Screen 2. Select 1 option on the Quick Filter 	Passed	Failed
TC21	Booking		Provide Filter Details	<ol style="list-style-type: none"> 1. Navigate to the Result List Screen 2. Click on the Filter 3. Display the Filter Detail 	Passed	Failed
TC22	Booking		Provide Sort	<ol style="list-style-type: none"> 1. Navigate to the Result List Screen 2. Click on the Sort 3. Display the Sort Details 4. Select options from the Sort Details 	Passed	Failed
TC23	Booking	8.Trip Details	View Pick-up Points	<ol style="list-style-type: none"> 1. Select 1 bus trip on the Result List 2. Navigate to the Pick-up/Drop-off points 	Passed	Passed
TC24	Booking		View Drop-off Points	<ol style="list-style-type: none"> 1. Select 1 bus trip on the Result List 2. Navigate to the Pick-up/Drop-off points 	Passed	Passed

				3. Change to the List off Pick-up/Drop-off Points		
TC25	Booking		View Reviews	1. Select 1 bus trip on the Result List 2. Navigate to the Pick-up/Drop-off points 3. Change to the Reviews tabs	Passed	Passed
TC26	Booking		View Details of Photo Reviews	1. Select 1 bus trip on the Result List 2. Navigate to the Pick-up/Drop-off points 3. Change to the Reviews tabs 4. Click on 1 photo to view details	Passed	Passed
TC27	Booking		View Amenities	1. Select 1 bus trip on the Result List 2. Navigate to the Pick-up/Drop-off points 3. Change to the Amenities tab	Passed	Passed
TC28	Booking		View Bus Features	1. Select 1 bus trip on the Result List 2. Navigate to the Pick-up/Drop-off points 3. Change to the Bus Features tab	Passed	Passed
TC29	Booking		View Details of Photos in Bus Features	1. Select 1 bus trip on the Result List 2. Navigate to the Pick-up/Drop-off points 3. Change to the Amenities tab 4. Click on 1 photo to view details	Passed	Passed
TC30	Booking	9. Booking	Provide Seat Departures	1. Hit “Book Tickets” button to navigate to the Seat Departures Screen	Passed	Passed

			2. Choose Depart Trip 3. Select Seat Position and the Seat Number 4. Hit “Continue” to navigate to the next booking stages		
TC31	Booking	Provide Pick-up/Drop-off Point	1. Navigate to the Pick-up/Drop-off Point Screen 2. Select 1 Pick-up/Drop-off with Shuttle Bus Service 3. Open the Shuttle Service Input 4. Select 1 Drop-off Point with Shuttle Bus Service 5. Open the Shuttle Service Input 6. Provide the location of the Shuttle Service 7. Hit “Continue” button to navigate to the next booking stages	Passed	Passed
TC32	Booking	Estimate the Price after selecting Drop-off Points	1. Select Drop-off Points 2. Estimate the Total Prices	Passed	Passed
TC33	Booking	Display Return Trip	1. Hit “Continue”button on the Pick-up/Drop-off Points Screen from the Departure Selection Stage 2. Display Result List of Return Bus Trip 3. Select Return Bus Trip	Passed	Passed

TC34	Booking	Select Return Seats	1. Hit “Book Tickets”button to navigate to the Seat Departures Screen 2. Select Seat Positions and the Seat Numbers 3. Hit “Continue” button to navigate to the next booking stages	Passed	Passed
TC35	Booking	Provide Pick-up/Drop-off Points	1. Navigate to the Pick-up/Drop-off Point Screen 2. Select 1 Pick-up/Drop-off with Shuttle Bus Service 3. Open the Shuttle Service Input 4. Select 1 Drop-off Point with Shuttle Bus Service 5. Open the Shuttle Service Input 6. Provide the location of the Shuttle Service 7. Hit “Continue” button to navigate to the nest booking stages	Passed	Passed
TC36	Booking	Estimate the total prices	1. Select Drop-off Point 2. Estimate the Total Price	Passed	Passed
TC37	Booking	Passenger Information with empty required values	1. Navigate to the Passenger Information Screen 2. Leave Input Field empty 3. Hit “Continue” button	Passed	Failed
TC38	Booking	Passenger Information	1. Navigate to the Passenger Information Screen	Pending	Failed

			with invalid values	2. Leave Input Field with invalid values 3. Hit “Continue” button		
TC39	Booking		Passenger Information with valid values	1. Navigate to the Passenger Information Screen 2. Input valid value to the Field 3. Hit “Continue” button	Passed	Passed
TC40	Booking		Update Booking Preview with Pick-up/Drop-off Points	1. Navigate to the Passenger Information Screen 2. Hit “Edit” button of the Pick-up Points on the ticket 3. Open Edit Pick-up Point Screen 4. Update Pick-up Point Screen with Shuttle Service 5. Open Shuttle Service Locations 6. Input Shuttle Service Locations 7. Hit “Update” button 8. Return to the Booking Review Screen with updated Pick-up Point	Passed	Passed
TC43	Transactions	10. Transactions	Select Voucher	1. Navigate to the Payment Screen. 2. Automatically select the last used voucher. 3. Hit "View All" button on the Select Voucher section. 4. Open All Vouchers Screen. 5. Select one available to select voucher. 6. Return to the Payment Screen with selected voucher.	Passed	Passed

TC44	Transactions	Make Payment	1. Navigate to the Payment Screen. 2. Display All Payment Methods Screen. 3. Select one Payment Method. 4. Hit "Pay Now" button.	Passed	Passed
TC45	Transactions	Select new Payment Method	1. Navigate to the Payment Screen. 2. Display All Payment Methods Screen. 3. Select one Payment Method. 4. Display Payment Details Screen. 5. Input Payment Information. 6. Hit "Pay Now" button.	Passed	Passed
TC46	Transactions	Open Card Information tab	1. Hit "View All" button on the Payment Screen. 2. Display All Payment Methods Screen. 3. Select one Payment Method. 4. Open Card Information Tab. 5. Input mandatory fields. 6. Hit the "Pay Now" button.	Passed	Passed
TC47	Transactions	Display Successful Payment	1. Process Payment. 2. Display Successful Payment.	Passed	Passed
TC48	Transactions	Display Unsuccessful Payment	1. Process Payment. 2. Display Unsuccessful Payment.	Passed	Passed
TC49	Transactions	Display Payment Guideline	1. Display Unsuccessful Payment Screen. 2. Hit "Retry" button. 3. Open Dialog.	Passed	Passed

				4. Hit "View Payment Guideline" button. 5. Display Payment Guideline Screen.		
TC50	Transactions		Update Unpaid Ticket in the History	1. Click on "Booking History" on the side bar from the Homepage Screen. 2. Display Unpaid Ticket Screen. 3. Hit "Pay Now" button. 4. Refer to the Payment Screen.	Passed	Passed
TC52	Booking History		Update real-time range of tickets	1. Click on "Booking History" on the side bar from the Homepage Screen. 2. Display Unpaid Ticket Screen. 3. Countdown until the Calculated Time is over to check the status.	Passed	Passed
TC53	Booking History	11.Booking History	Display Status of the Paid Tickets	1. Click on "Booking History" on the side bar from the Homepage Screen. 2. Display Unpaid Ticket Screen. 3. Change to the Paid Ticket Tab.	Passed	Passed
TC54	Booking History		Display Cancellation Tickets	1. Click on "Booking History" on the side bar from the Homepage Screen. 2. Display Unpaid Ticket Screen. 3. Change to the Cancellation Ticket Tab.	Passed	Passed
TC55	Booking History		View Complete Invoice	1. Click on "Booking History" on the side bar from the Homepage Screen. 2. Display Unpaid Ticket Screen. 3. Change to the Paid Ticket Tab. 4. Select one Complete Ticket. 5. Display Complete Ticket Invoice.	Passed	Passed

TC56	Booking History		View Ongoing Invoice	<ol style="list-style-type: none"> Click on "Booking History" on the side bar from the Homepage Screen. Display Unpaid Ticket Screen. Change to the Paid Ticket Tab. Select one Ongoing Ticket. Display Ongoing Ticket Invoice. 	Passed	Passed
TC57	Booking History		View Upcoming Invoice	<ol style="list-style-type: none"> Click on "Booking History" on the side bar from the Homepage Screen. Display Unpaid Ticket Screen. Change to the Paid Ticket Tab. Select one Upcoming Ticket. Display Upcoming Ticket Invoice. 	Passed	Passed
TC58	Booking History		Enable to Edit Pick-up/Drop-off Points on Unpaid Tickets	<ol style="list-style-type: none"> Click on "Booking History" on the side bar from the Homepage Screen. Display Unpaid Ticket Screen. Select one Unpaid Ticket. Refer to the Booking Review Screen. Select to edit Pick-up/Drop-off Points and Shuttle Bus. 	Passed	Passed
TC59	Booking History		Provide Quick Ratings	<ol style="list-style-type: none"> Click on "Booking History" on the side bar from the Homepage Screen. Display Unpaid Ticket Screen. Change to the Paid Ticket Tab. Click on "Rating" button on one Complete Ticket. Display Rating Dialog. Provide Rating Status. 	Passed	Failed

TC60	Booking History	Provide Detailed Ratings	<ol style="list-style-type: none"> Click on "Booking History" on the side bar from the Homepage Screen. Display Unpaid Ticket Screen. Change to the Paid Ticket Tab. Click on "Rating" button on one Complete Ticket. Display Rating Dialog. Provide Rating Status. Refer to the Detailed Ratings Screen. Provide Images and Videos. Provide Details. Click "Save" button. Display Successful Rating Dialog. Hit "Back to Homepage" button. 	Passed	Failed
TC61	Booking History	Pass Detailed Ratings	<ol style="list-style-type: none"> Click on "Booking History" on the side bar from the Homepage Screen. Display Unpaid Ticket Screen. Change to the Paid Ticket Tab. Click on "Rating" button on one Complete Ticket. Display Rating Dialog. Provide Rating Status. Refer to the Detailed Ratings Screen. Click "Save" button. Display Successful Rating Dialog. Hit "Back to Homepage" button. 	Passed	Failed
TC62	Booking History	Upload Photos/Videos to Update	<ol style="list-style-type: none"> Click on "Booking History" on the side bar from the Homepage Screen. Display Unpaid Ticket Screen. 	Passed	Failed

			Detailed Ratings	3. Change to the Paid Ticket Tab. 4. Click on "Rating" button on one Complete Ticket. 5. Display Rating Dialog. 6. Provide Rating Status. 7. Refer to the Detailed Ratings Screen. 8. Provide Images and Videos.		
TC64	Cancellation		View Cancellation Confirmation	1. Click on "Booking History" on the side bar from the Homepage Screen. 2. Display Unpaid Ticket Screen. 3. Hit "Cancel" button on one ticket. 4. Display Cancel Regulations.	Passed	Passed
TC65	Cancellation	12.Cancellation	View Cancel Regulations with Estimated Fees	1.Click on "Booking History" on the side bar from the Homepage Screen. 2. Display Unpaid Ticket Screen. 3. Hit "Cancel" button on one ticket. 4. Display Cancel Regulations. 5. Hit "Continue" button. 6. Display Confirmation Dialog. 7. Hit "Confirm" button.	Passed	Passed
TC66	Cancellation		Provide Reasons of Cancellation	1. Click on "Booking History" on the side bar from the Homepage Screen. 2. Display Unpaid Ticket Screen. 3. Hit "Cancel" button on one ticket. 4. Display Cancel Regulations. 5. Hit "Continue" button. 6. Display Confirmation Dialog. 7. Hit "Continue" button.	Passed	Passed

				8. Display Cancellation Reasons Screen. 9. Provide Reasons. 10. Hit "Continue" button. 11. Display Confirmation Details with Successful Cancellation Dialog 12. Hit "Back to Homepage" button.		
--	--	--	--	--	--	--

CHAPTER 8: PROJECT PLAN

8.1. Work Breakdown Structure (WBS)

The Bus Booking System Mobile Application project is divided into 4 main epics, scheduled from September 27 to November 13, with each epic lasting approximately 1 to 1.5 weeks. Specifically:

1. **Epic 1 (1 week: 27/09/2024 - 03/10/2024): Project Overview and System Planning** - Covers initial project setup, objectives, and planning activities to establish the foundation.
2. **Epic 2 (04/10/2024 - 10/10/2024): Theoretical Background and Market Analysis**
- Involves research and analysis of market trends and theoretical aspects, setting the context for the application's development.
3. **Epic 3 (11/10/2024 - 27/10/2024) - Analysis:** The Analysis epic in the project plan includes Detailed Analysis and Overview Analysis. The Overview Analysis provides a broad understanding of system requirements, scope, and primary features, setting the groundwork for the Detailed Analysis. This phase comprehensively examines the current and future process flows, main functions, business processes, data flow, use cases, sequence diagrams, business rules, data model, data dictionary, and API specifications, ensuring a thorough understanding of system requirements and specifications.
4. **Epic 4 (28/10/2024 - 13/11/2024) - System Design and Implementation** - Encompasses designing and implementing the system architecture, features, and components. This phase is scheduled for November.

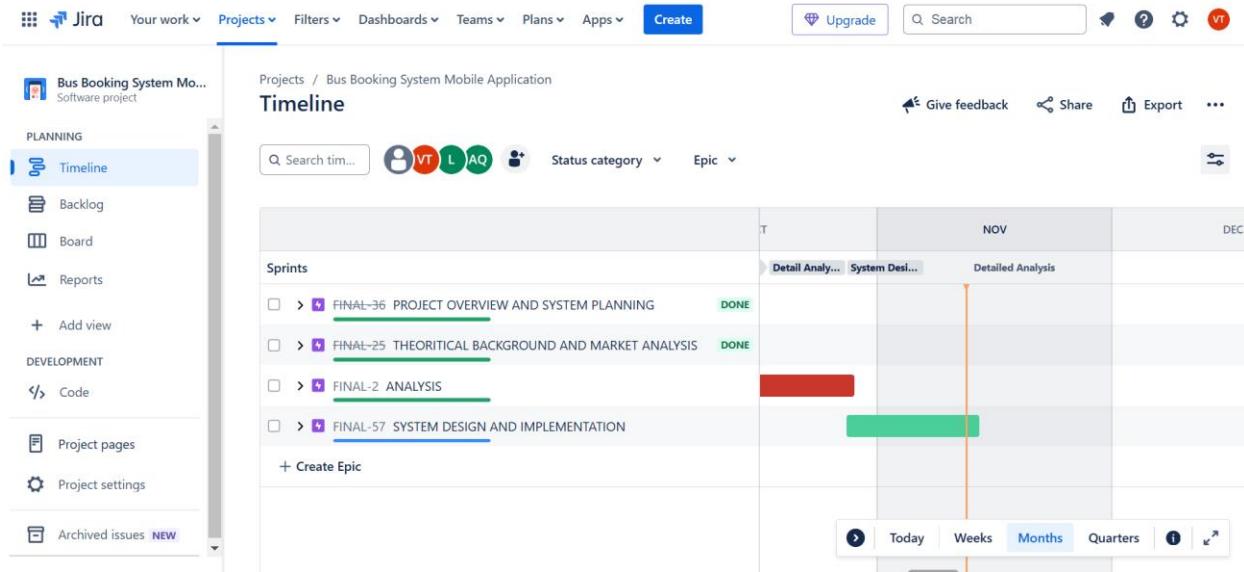


Figure 8.1. AutoBus's Booking System Project with 4 epics

(Sources: Author's Sources)

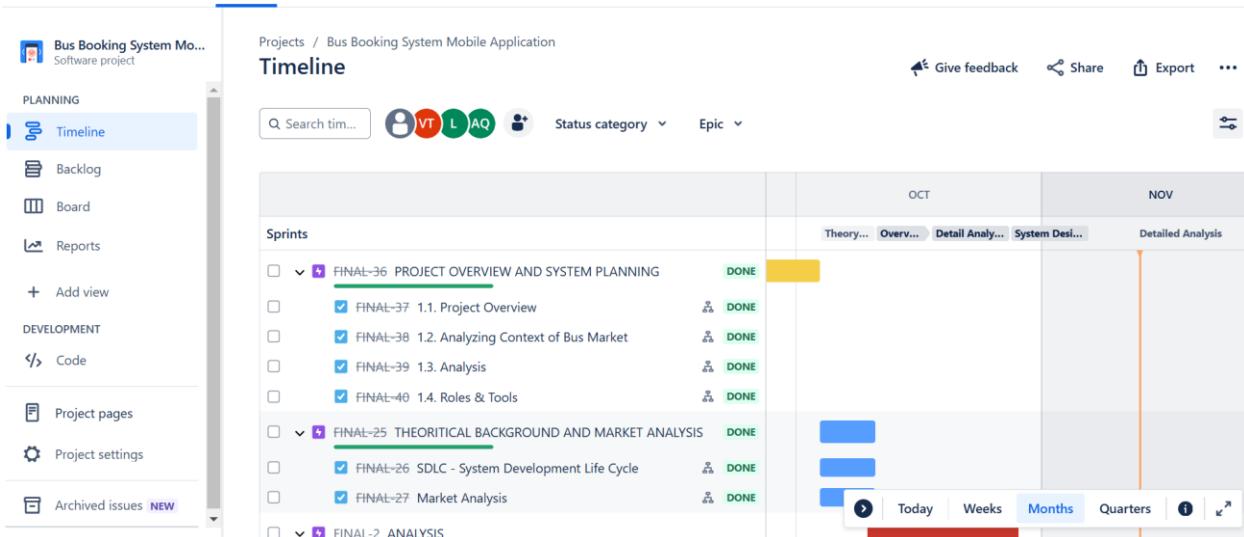


Figure 8.2. Sprints, Tasks & SubTasks related to each epic

(Sources: Author's Sources)

8.2. Project Milestone

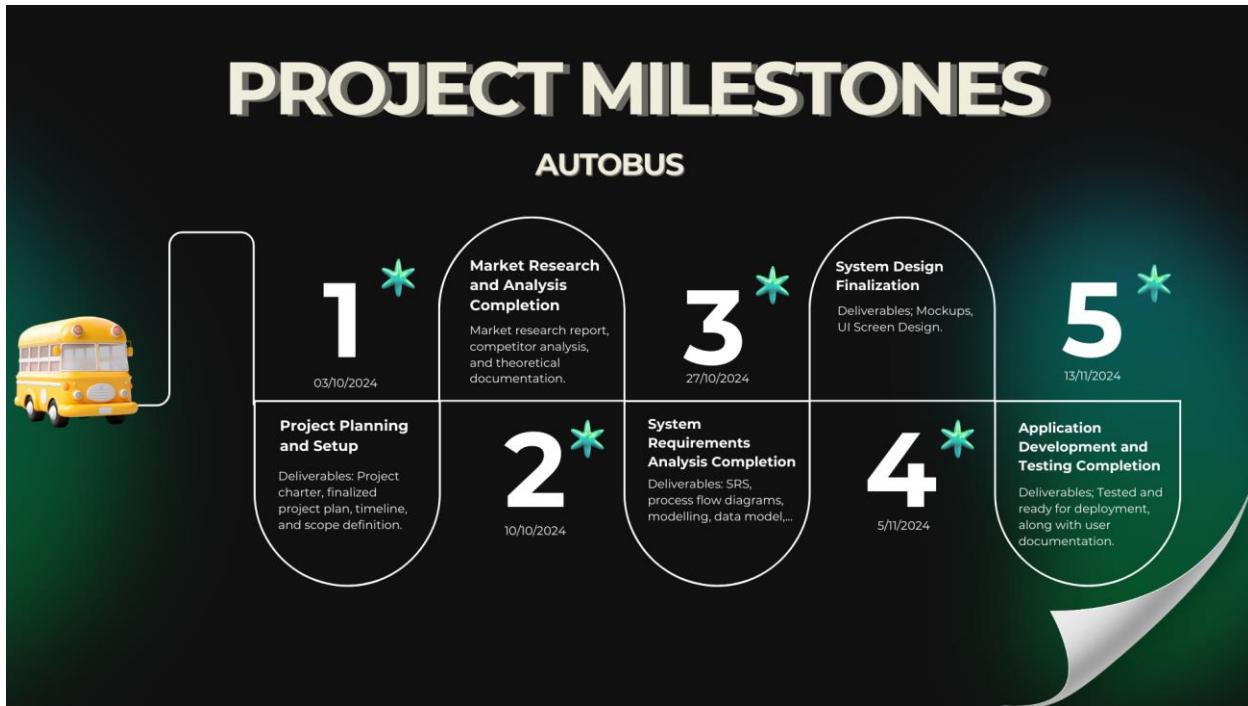


Figure 8.3. Project Milestone

(Sources: Author's Sources)

Autobus project's five milestones:

1. **Project Planning (03/10/2024):** Deliverables include the project charter, timeline, and scope.
2. **Market Research (10/10/2024):** Completion of market analysis and competitor research.
3. **Requirements Analysis (27/10/2024):** Includes SRS, process diagrams, and data models.
4. **System Design (05/11/2024):** Mockups and UI screen designs.
5. **Development and Testing (13/11/2024):** Application testing, deployment readiness, and user documentation.

These steps ensure efficient project execution.

8.3. Pert Chart

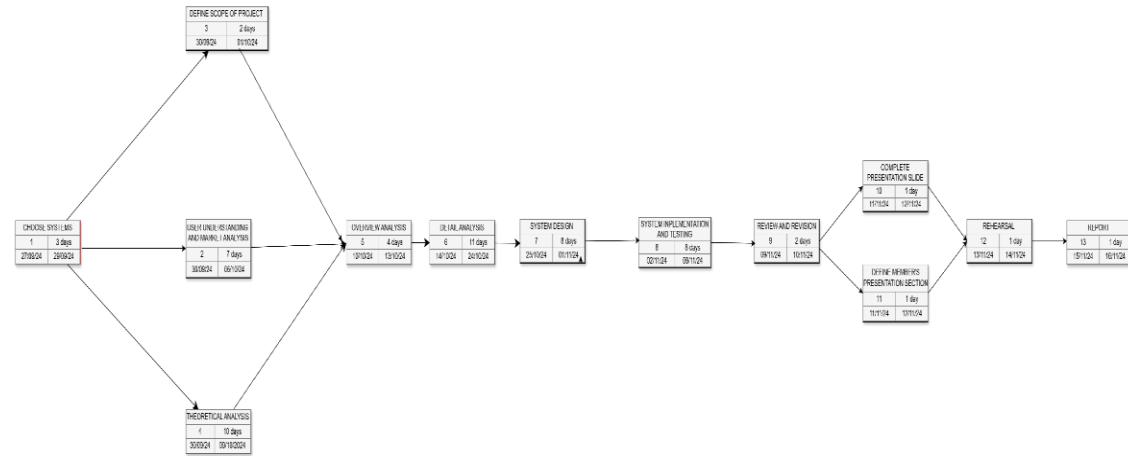


Figure 8.4. Pert chart

(Source: Authors' source)

This PERT chart outlines the project timeline, breaking down each phase and its duration. The project begins with Choose Systems (3 days, Sep 27-29, 2024) to select the right tools, followed by User Understanding and Market Analysis (7 days, Sep 30-Oct 6) to identify target audience needs. Concurrently, Define Scope of Project (2 days, Sep 30-Oct 1) and Theoretical Analysis (10 days, Sep 30-Oct 9) establish the project's focus and theoretical foundation. Overview Analysis (4 days, Oct 10-13) and Detail Analysis (11 days, Oct 14-24) dive into specifics. System Design (8 days, Oct 25-Nov 1) creates the project structure, leading to System Implementation and Testing (8 days, Nov 2-8) to ensure functionality. Finally, Review and Revision (2 days, Nov 9-10) allow for adjustments, and Complete Presentation Slide (1 day, Nov 11) prepares materials for presentation. This chart ensures a systematic, step-by-step approach to project completion.

8.4. Gantt Chart

Our group also utilized the Gantt chart to illustrate the project schedule, define essential routes, milestones, and resource allocation, as well as track project progress and identify any concerns that may affect the timeline.



Figure 8.5. Gantt chart

(Source: Authors' source)

CHAPTER 9: CONCLUSION

9.1. Practical significance

9.1.1. For user

No.	Issue / What Did Not Work Well	What Should the Team Have Done Instead?	Lessons Learned
1	Lack of diverse and secure payment options.	Integrate multiple popular payment methods, such as credit cards, e-wallets, and QR codes, while ensuring secure transactions through data encryption. Provide an offline payment option if possible.	Offering a variety of payment methods makes transactions easier, while enhanced security builds customer trust.
2	App not compatible with various mobile screen sizes.	Design the app to be fully responsive, with the interface adjusting automatically to different screen sizes (smartphones, tablets). Perform thorough testing on multiple device types and operating systems to ensure a stable user experience.	Optimizing for multiple platforms and screen sizes increases user reach and prevents negative feedback about the interface.
3	Lack of real-time bus tracking feature.	Integrate live map tracking so passengers can monitor the bus location, along with the estimated time of arrival. This helps passengers track the bus in real-time and wait at the correct stop.	Real-time tracking offers reassurance and allows passengers to plan better, improving satisfaction and reducing issues related to wait times.
4	No multilingual	Add a multilingual feature,	Multilingual

	support, causing difficulties for foreign users.	especially English, to better accommodate tourists and international users.	support expands the customer base, making the app more welcoming and user-friendly for international users.
5	Slow app loading and response times, frustrating users.	Optimize code and remove unnecessary features to improve app loading and response times.	Faster response times help retain users, reduce bounce rates, and provide a smoother experience.

9.1.2. For team

No.	Issue / What Did Not Work Well	What Should the Team Have Done Instead?	Lessons Learned
1	Limited Monitoring of Individual Progress, Leading to Delayed Deadlines	The team should have implemented more frequent check-ins and progress tracking to catch potential delays sooner.	Consistent and closer monitoring of each member's progress allows early identification of issues, enabling timely adjustments and helping meet deadlines.
2	Lack of Contingency Planning for Unexpected Issues	The team should have built contingency time into the schedule to handle unforeseen challenges.	Having a buffer time in project plans can help manage unexpected delays without impacting the final deadline.

3	Did not anticipate unforeseen issues during implementation	Allocate extra time for potential issues or unexpected adjustments	Always prepare a buffer time to avoid delays
4	Insufficient time to implement all desired features	Plan more thoroughly and prioritize essential features	Prioritizing key features ensures timely project delivery

9.2. Conclusion

The completion of the AutoBus ticket booking system marks a significant achievement in addressing the common challenges associated with bus travel in Vietnam. By integrating user-friendly features like real-time seat selection, secure payment gateways, and comprehensive customer support, AutoBus has streamlined the ticketing process for both users and operators. Through the meticulous application of the System Development Life Cycle (SDLC) framework, the project exemplifies the synergy between modern technology and thoughtful system design.

This project has not only enhanced technical and analytical skills but also underscored the importance of teamwork, effective communication, and adaptability in solving complex problems. Although challenges such as time constraints and resource limitations were encountered, they provided invaluable lessons in project management and problem-solving.

AutoBus represents a scalable and efficient solution, with the potential for further development to include new features and expand its user base. Moving forward, the insights gained from this endeavor will serve as a foundation for future initiatives aimed at enhancing the digitalization of transportation systems. We look forward to contributing to the evolution of smart mobility in Vietnam and beyond.

REFERENCES

- [1] Visual Paradigm. (n.d.). What is BPMN? Visual Paradigm.
<https://www.visual-paradigm.com/guide/bpmn/what-is-bpmn/>
- [2] Lucidchart. (n.d.). Data flow diagram: Examples, parts, and tips. Lucidchart.
<https://www.lucidchart.com/pages/data-flow-diagram>
- [3] Bizfly Cloud. (2022, March 31). Use case là gì? Tìm hiểu về use case trong phát triển phần mềm. Bizfly Cloud. <https://bizflycloud.vn/tin-tuc/use-case-la-gi-2022033112157017.htm>
- [4] Moriarty, P. Global Passenger Transport. Encyclopedia 2021, 1, 189–197.
<https://doi.org/10.3390/encyclopedia1010018>
- [5] Ammirato, S., Felicetti, A. M., Linzalone, R., Volpentesta, A. P., & Schiuma, G. (2020). A systematic literature review of revenue management in passenger transportation. *Measuring Business Excellence*.
- [6] Wrike. (n.d.). What is a milestone in project management? Retrieved from <https://www.wrike.com/project-management-guide/faq/what-is-a-milestone-in-project-management/>
- [7] Hoang, D.A.; Lam, Y.Y.; Amos, P.; Reddel, P.; Phuong, P.T.; Hien, N.T.P. Sustainable Development of Inland Waterways Transport in Vietnam: Strengthening the Regulatory, Institutional and Funding Frameworks; World Bank working paper; World Bank Group: Washington, DC, USA, 2019.
- [8] Lilac Infotech. (n.d.). Revolutionizing the way we travel and the rise of bus booking app development. Retrieved from <https://lilacinfotech.com/blog/232/Revolutionizing-the-Way-We-Travel-and-The-Rise-of-Bus-Booking-App-Development>
- [9] Wrike. (n.d.). What is a use case? Definition and examples. Wrike.
<https://www.wrike.com/blog/what-is-a-use-case/>
- [10] Angular official website: [Angular - What is Angular?](https://angular.io/guide/what-is-angular)
- [11] CloudDefense. (n.d.). *System development life cycle*. CloudDefense.
<https://www.clouddefense.ai/system-development-life-cycle/>

- [12] GeeksforGeeks. (2023, August 2). *System development life cycle (SDLC)*. GeeksforGeeks. <https://www.geeksforgeeks.org/system-development-life-cycle/>
- [13] TheProductManager. (2023, July 15). *What is SDLC? Understand the software development life cycle*. TheProductManager. <https://theproductmanager.com/topics/software-development-life-cycle/>
- [14] Intellectsoft. (2023, May 10). *What is system development life cycle?* Intellectsoft. <https://www.intellectsoft.net/blog/what-is-system-development-life-cycle/>
- [15] Frontegg. (n.d.). *User roles and permissions: Guide to securing your app*. Frontegg. <https://frontegg.com/guides/user-role-and-permission>
- [16] FPT Shop. (2021, July 29). *Visual Studio Code là gì? Các tính năng nổi bật của Visual Studio Code*. FPT Shop. <https://fptshop.com.vn/tin-tuc/danh-gia/visual-studio-code-la-gi-cac-tinh-nang-noi-bat-cua-visual-studio-code-146213>
- [17] Dung, P. V. (2019, December 17). *Tổng quan về MongoDB*. Viblo. <https://viblo.asia/p/tong-quan-ve-mongodb-EoDkQoxqGbV>
- [18] FPT Shop. (2022, May 11). *Figma là gì? Những điểm nổi bật*. FPT Shop. <https://fptshop.com.vn/tin-tuc/danh-gia/figma-la-gi-nhung-diem-noi-bat-150734>
- [19] Dung, P. V. (2019, October 16). *Git - Tổng quan về Git*. Viblo. <https://viblo.asia/p/git-tong-quan-ve-git-1VgZvJ2YZAw>
- [20] GeeksforGeeks. (2024, July 4). *User interface design*. GeeksforGeeks. [User Interface Design - Software Engineering \(geeksforgeeks.org\)](https://geeksforgeeks.org/user-interface-design/)
- [21] GeeksforGeeks. (2022, August 22). Qualities of good user interface design. GeeksforGeeks. [Qualities of Good User Interface Design - GeeksforGeeks](https://geeksforgeeks.org/qualities-of-good-user-interface-design/)
- [22] GeeksforGeeks. (2024b, August 8). What is User Experience (UX) Design? GeeksforGeeks. [What is User Experience \(UX\) Design? \(geeksforgeeks.org\)](https://geeksforgeeks.org/what-is-user-experience-ux-design/)
- [23] Wagner, R. (2015). "The Software Development Life Cycle: A Complete Guide." International Journal of Computer Science and Information Technology Research, 3(2), 45-56.

- [24] FUTA Software Development. (n.d.). *Vé xe Phương Trang | Công ty cổ phần xe khách Phương Trang | FUTA Bus Lines | Tổng Đài đặt vé và Chăm Sóc Khách Hàng 19006067*. FUTA Bus Lines. <https://futabus.vn/ve-chung-toi>
- [25] VeXeRe. (n.d.). *VeXeRe - Đặt vé xe khách chất lượng cao*. Retrieved November 16, 2024, from <https://vexere.com>
- [26] RedBus. (n.d.). *RedBus Việt Nam - Đặt vé xe khách trực tuyến*. Retrieved from <https://www.redbus.vn/>