|  | **MINISTRY OF EDUCATION AND TRAINING** |
| --- | --- |

| **FPT UNIVERSITY** |
| --- |
| Capstone Project Document |
| [MoveMate Application] |

| **GFA24SE06** | |
| --- | --- |
| **Group Members** | Cổ Phước Vinh – Team Leader – SE161891  Lê Hàn Anh – Team Member – SE161941  Đặng Nguyễn Hoài Phương – Team Member – SE161958  Nguyễn Thành Vinh – Team Member – SE161950  Phan Văn Tuấn – Team Member – SE162029 |
| **Supervisor** | Mrs. Nguyễn Thị Cẩm Hương |
| **Capstone Project code** | FA24SE085 |

- Ho Chi Minh, Dec 2024 -

**Table of Contents**

Acknowledgment [4](#_heading=h.30j0zll)

[Definition and Acronyms 4](#_heading=h.1fob9te)

[I. Project Introduction 5](#_heading=h.3znysh7)

[1. Overview 5](#_heading=h.2et92p0)

[2. Product Background 5](#_heading=h.1t3h5sf)

[3. Existing Systems 5](#_heading=h.4d34og8)

[4. Business Opportunity 5](#_heading=h.2s8eyo1)

[5. Software Product Vision 5](#_heading=h.17dp8vu)

[6. Project Scope & Limitations 5](#_heading=h.3rdcrjn)

[II. Project Management Plan 6](#_heading=h.lnxbz9)

[1. Overview 6](#_heading=h.35nkun2)

[2. Management Approach 6](#_heading=h.z337ya)

[3. Master Schedule 6](#_heading=h.4i7ojhp)

[4. Project Organization 6](#_heading=h.1rvwp1q)

[5. Project Communication 6](#_heading=h.2xcytpi)

[6. Configuration Management 6](#_heading=h.1ci93xb)

[III. Software Requirement Specification 7](#_heading=h.3as4poj)

[1. Overall Description 7](#_heading=h.4bvk7pj)

[2. User Requirements 7](#_heading=h.49x2ik5)

[3. Functional Requirements 7](#_heading=h.2p2csry)

[4. Non-Functional Requirements 7](#_heading=h.3o7alnk)

[5. Other Requirements 7](#_heading=h.32hioqz)

[IV. Software Design Description 8](#_heading=h.41mghml)

[1. Overall Description 8](#_heading=h.2grqrue)

[2. System Architecture Design 8](#_heading=h.2r0uhxc)

[3. System Detailed Design 8](#_heading=h.4f1mdlm)

[4. Class Specification 8](#_heading=h.1664s55)

[5. Data & Database Design 8](#_heading=h.3q5sasy)

[V. Software Testing Documentation 9](#_heading=h.19c6y18)

[1. Overall Description 9](#_heading=h.25b2l0r)

[2. Test Plan 9](#_heading=h.nmf14n)

[3. Test Cases 9](#_heading=h.2lwamvv)

[4. Test Reports 9](#_heading=h.111kx3o)

[VI. Release Package & User Guides 10](#_heading=h.3l18frh)

[1. Deliverable Package 10](#_heading=h.kgcv8k)

[2. Installation Guides 10](#_heading=h.34g0dwd)

[3. User Manual 10](#_heading=h.1jlao46)

[VII. Appendix 10](#_heading=h.43ky6rz)

[1. Glossary [Optional] 10](#_heading=h.2iq8gzs)

[2. References [Optional] 10](#_heading=h.xvir7l)

[3. Others [Optional] 10](#_heading=h.3hv69ve)

# Acknowledgment

*[Fill team’s acknowledgment here…]*

# Definition and Acronyms

*[Fill all the definitions, acronyms,… used within the document] in the table format as below]*

| **Acronym** | **Definition** |
| --- | --- |
| PWM | Psychology website |
| AWS | Amazon Web Services |
| BA | Business Analysis |
| BR | Business Rule |
| ERD | Entity Relationship Diagram |
| GUI | Graphical User Interface |
| PM | Project Manager |
| SDD | Software Design Description |
| SPMP | Software Project Management Plan |
| SRS | Software Requirement Specification |
| UAT | User Acceptance Test |
| UC | Use Case |
| API | Application Program Interface |

# I. Project Introduction

## 1. Overview

### 1.1 Project Information

* Project name: MoveMate Application
* Project code: GFA24SE06
* Group name: FA24SE085
* Software type: Web application and Mobile application

### 1.2 Project Team

| **Full Name** | **Role** | **Email** | **Mobile** |
| --- | --- | --- | --- |
| Nguyễn Thị Cẩm Hương | Lecturer | HuongNTC2@fe.edu.vn | 0909912347 |
| Cổ Phước Vinh | Leader | vinhcpse161891@fpt.edu.vn | 0374027409 |
| Lê Hàn Anh | Member | anhlhse161941@fpt.edu.vn | 0915714533 |
| Đặng Nguyễn Hoài Phương | Member | phuongdnhse161958@fpt.edu.vn | 0386751935 |
| Nguyễn Thành Vinh | Member | vinhntse161950@fpt.edu.vn | 0382703625 |
| Phan Văn Tuấn | Member | tuanpvse162029@fpt.edu.vn | 0394581438 |

## 2. Product Background

Moving houses or offices can be a stressful and time-consuming process for individuals, families, and small businesses. Traditionally, people spend hours coordinating with moving services, trying to determine the right vehicle size, porter selection for loading and unloading, and dealing with costs that are often unclear. This process typically involves multiple calls, in-person meetings, and unforeseen issues that add to the stress.

Users often face challenges such as miscommunication with service providers, and time-consuming to find an appropriate service provider. These inefficiencies lead to wasted time, frustration, and sometimes even increased costs. Additionally, with no real-time tracking available, customers are left in the dark about the status of their service, increasing anxiety about the entire process.

Recognizing these pain points, the MoveMate app was created to streamline and simplify the entire moving experience. By integrating essential features like vehicle booking, porter service selection, transparent pricing, and real-time tracking. The idea was raised by customers who wanted a more efficient, modernized approach to managing moving houses without the typical challenges of dealing with multiple service providers and unclear processes.

## 3. Existing Systems

### 3.1 Lalamove

Lalamove is a logistics service that helps individuals and businesses move goods by booking trucks on demand. The platform focuses on providing fast and flexible delivery options through a large fleet of vehicles, including vans and trucks.

#### Pros:

▪ Fast and flexible delivery options.

▪ Real-time tracking for transparency.

▪ Wide range of vehicle options for different delivery needs.

▪ Convenient mobile app.

#### Cons:

▪ Doesn’t offer specific house-moving services.

▪ Limited scope in terms of customized pricing for larger homes and offices.

## 

### 3.2 Ahamove

▪ Ahamove is a logistics service designed for both individuals and businesses to book trucks and vehicles for moving goods on demand. It focuses on fast delivery and real-time tracking, similar to Lalamove

#### Pros:

▪ Easy-to-use app for booking deliveries.

▪ Real-time tracking and pricing transparency.

#### Cons:

▪ Lacks specific house-moving services.

▪ Limited coverage for larger, customized moves like office relocations.

▪ Pricing options may not be suitable for complex moving needs.

## 4. Business Opportunity

## The demand for moving services is on the rise, particularly in urban areas where more people are relocating for work and lifestyle changes. However, the moving process remains stressful due to unclear pricing, and unreliable labor. Traditional moving companies often lack transparency, leading to inefficiencies and customer dissatisfaction.

## Market Trends: As urban relocations and demand for convenient, digital services rise, MoveMate is perfectly positioned to offer a mobile-first, on-demand solution, aligning with modern consumer preferences for transparency and efficiency.

## 5. Software Product Vision

## For individuals and families looking to move who need a simple and efficient way to schedule moving services. The MoveMate is a comprehensive moving service platform that streamlines the booking process, connects users with reliable movers, and offers real-time tracking, unlike traditional moving services that often involve lengthy phone calls and fees not clear. Our Product provides a transparent, user-friendly interface with integrated features for scheduling, communication, and payment.

## 6. Project Scope & Limitations

### 6.1 Major Features

### 

| **Feature ID** | **FEATURE DESCRIPTION** |
| --- | --- |
| FE1: Booking Management | This feature enables customers to create, manage, and modify booking requests by specifying details such as date, time, truck type, and pick-up/drop-off locations. Once a booking request is made, drivers and porters receive notifications and can review the details. Reviewers assess the booking information to recommend suitable trucks based on customer requirements. Additionally, customers have the option to approve or reject any proposed changes to their booking, including adjustments to services, fees, or truck type. Drivers and porters can request real-time updates to the booking details, ensuring flexibility and responsiveness throughout the booking process. |
| FE2: Service Management | This feature allows managers to define and manage the various services offered, including standard truck rentals, specialty services (like packing or loading). Customers can browse and select from available services when making a booking, ensuring they receive the right support for their booking needs. Reviewers can assess the demand for specific services and suggest updates or changes based on customer feedback and usage data. Drivers are informed about the specific services included in each rental request, allowing them to prepare accordingly for each job. |
| FE3: Truck Management | This feature allows managers to manage the fleet of trucks, including adding new trucks and updating truck information. Reviewers can view truck availability and assign trucks to specific rental requests based on customer needs and truck suitability. Drivers can update the status of their assigned trucks(PENDING, AVAILABLE, IN\_USE, MAINTENANCE). |
| FE4: User Management | This feature enables administrators to create, update, and manage user accounts, including setting roles and permissions for users, reviewers, drivers, and porters. Admins can update their personal information. Additionally, managers can review, approve, and reject driver and porter registrations |
| FE5: Fee Management | This feature enables managers to set and adjust rental fees based on factors like truck type, rental duration, distance, and additional services. Customers can view estimated costs for their rental requests, receive detailed invoices, and choose from multiple payment options such as credit cards or digital wallets. Reviewers can verify fee calculations for each rental request to ensure accuracy before confirming the booking. |
| FE6: Discount Management | This feature enables customers to collect and apply vouchers to their bookings. Managers are responsible for managing discount codes, ensuring customers benefit from special offers. Customers can view available discounts and apply them to their rental requests at checkout. |
| FE7: Payment Management | This feature enables customers to pay for their moving service either online or via cash. Drivers are responsible for updating the payment status when cash is received. Managers can monitor and oversee financial reports, ensuring payment accuracy and proper accounting. |
| FE8: Real-Tracking Booking | This feature allows customers to track the progress of their booking in real-time, including the current location of the truck, estimated arrival time, and status updates throughout the move. Drivers, reviewers and porters can update their status during each phase of the move, such as when they are en route, arriving, loading, or unloading. They can also notify customers and administrators of any delays or issues. Managers can monitor the overall progress of all active bookings, ensuring that services are delivered on time and addressing any potential issues promptly. |
| FE9: Incident Management | This feature allows customers to report issues during the booking process. Drivers and porters can also report incidents like vehicle breakdowns or furniture damage. Managers then review and handle compensation requests |
| FE9: Communication Management | This feature allows seamless communication between customers, drivers, porters, reviewers, and managers through an in-app chat. |
| FE10: Notifications Management | All roles receive notifications for updates, incidents, and other booking-related information, allowing for swift response and coordination. |
| FE11: Schedule management | This feature allows managers to create and manage the work schedules of reviewers, drivers, and porters, ensuring optimal coverage for bookings and operations. Reviewers can view their assigned shifts, manage their availability, and request time off or shift changes as needed. Drivers can access their schedules, receive notifications for upcoming assignments, and report their availability for new bookings. Porters can manage their shifts and communicate with drivers regarding loading and unloading schedules to ensure efficient operations. |

### 

### *6.2 Limitations & Exclusions*

#### 6.2.1. Limitations

*This system is not the best and most optimal solution.*

*The mobile application only supports Android OS*

*The system lacks integration with certain third-party services such as Lalamove or Ahamove specialized logistics providers and additional transport systems.*

*MoveMate's integration with payment services is currently limited to basic features, and advanced financial services like installment payments are not available.*

#### 6.2.2. Exclusions

*The system does not handle damage insurance claims for goods in transit. Customers will receive process claims directly after finishing the payment.*

*MoveMate does not offer a service for storing goods temporarily if the move-in location is unavailable.*

*It does not support direct communication between customers and drivers before a booking is confirmed.*

# II. Project Management Plan

## 1. Overview

### 1.1. Scope & Estimation

| **No** | **WBS Item** | **Complexity** | **Est. Effort**  **(man-days)** |
| --- | --- | --- | --- |
| **1** | **Initiating** |  | **10** |
| 1.1 | Create a project management plan | Complex | 5 |
| 1.2 | Rent a server, buy a domain | Medium | 2 |
| 1.3 | Define required functionality | Medium | 1 |
| 1.4 | Define required screens | Medium | 1 |
| 1.5 | Define tables, relationships, and data schemas | Medium | 5 |
| **2** | **Planning** |  | **40** |
| 2.1 | Technology Training | Medium | 5 |
| 2.2 | Requirement Analysis | Complex | 4 |
| 2.3 | BRS – Business requirement specification | Complex | 4 |
| 2.4 | ERD Design | Complex | 4 |
| 2.5 | UI Design (Figma Design) | Medium | 10 |
| 2.6 | Database Design | Medium | 2 |
| 2.7 | Create test cases | Medium | 3 |
| 2.8 | Team meeting | Simple | 1 |
| 2.9 | Supervisor meeting | Medium | 1 |
| 2.9 | Create Source Base | Complex | 5 |
| **3** | **Executing** |  | **100** |
| **3.1** | **Iteration 1** |  | **20** |
| 3.1.1 | Write software requirement specification | Complex | 7 |
| 3.1.2 | Write software design document | Complex | 5 |
| 3.1.3 | Create test cases | **Medium** | 4 |
| 3.1.4 | Login and logout | Simple | 4 |
| 3.1.5 | Forgot password | Simple | 3 |
| 3.1.6 | Re-generate tokens | Simple | 2 |
| **3.2** | **Iteration 2** |  | **35** |
| 3.2.1 | Create a new booking | Medium | 2 |
| 3.2.2 | Update an existed booking information | Medium | 2 |
| 3.2.3 | View list of booking and view booking details | Simple | 2 |
| 3.2.4 | Update a booking status | Medium | 3 |
| 3.2.5 | Delete an existed booking | Medium | 1 |
| 3.2.6 | Track the moving process | Medium | 5 |
| 3.2.7 | View moving schedule | Simple | 1 |
| 3.2.8 | Update moving schedule | Medium | 1 |
| 3.2.9 | Delete moving schedule | Medium | 2 |
| 3.2.10 | Create a new promotion | Medium | 1 |
| 3.2.11 | View list of promotion | Simple | 2 |
| 3.2.12 | Delete a promotion | Medium | 1 |
| 3.2.13 | Review staff quality | Medium | 1 |
| 3.2.14 | Report problems during moving | Medium | 2 |
| 3.2.15 | Report damage to items during transportation | Medium | 3 |
| 3.2.16 | Resolve issues | Complex | 3 |
| 3.2.17 | Recharge money into the wallet in the system | Medium | 2 |
| 3.2.18 | View wallet information | Simple | 4 |
| 3.2.19 | Create payment for customer | Medium | 5 |
| 3.2.20 | View transaction history | Simple | 5 |
| **3.3** | **Iteration 3** |  | **35** |
| 3.3.1 | Create a new driver, porter, reviewer and manager | Medium | 2 |
| 3.3.2 | Update an existed driver, porter, reviewer and manager information | Medium | 2 |
| 3.3.3 | View list of driver, porter, reviewer and manager and view driver, porter, reviewer and manager details | Simple | 1 |
| 3.3.4 | Updatedriver, porter, reviewer and manager status | Medium | 1 |
| 3.3.5 | Delete an existed driver, porter, reviewer and manager | Medium | 1 |
| 3.3.6 | Refund to customer's wallet | Complex | 2 |
| 3.3.7 | View list of truck category and view truck details | Complex | 1 |
| 3.3.8 | Update a truck status | Medium | 2 |
| 3.3.9 | Create a new service | Medium | 1 |
| 3.3.10 | View list of services and view service details | Medium | 3 |
| 3.3.11 | Update an existed service | Simple | 5 |
| 3.3.12 | Delete an existed service | Complex | 2 |
| 3.3.13 | Chat between customers and reviewer, driver, porter and manager | Complex | 5 |
| 3.3.14 | Create a new fee | Complex | 3 |
| 3.3.15 | View list of fees and view fee details | Simple | 2 |
| 3.3.16 | Update an existed fee | Medium | 4 |
| 3.3.17 | Delete an existed fee | Simple | 1 |
| **3.4** | **Iteration 4** |  | **7** |
| 3.4.1 | Unit testing | Simple | 7 |
| 3.4.2 | Integration testing | Medium | 10 |
| 3.4.3 | System testing | Complex | 12 |
| 3.4.4 | Optimizing Performance and Security | Medium | 3 |
| 3.4.5 | Testing in staging environment | Medium | 12 |
| 3.4.6 | Deploy code in product environment | Medium | 3 |
| 3.4.7 | Testing in product environment | Medium | 14 |
| 4 | **Monitoring and Controlling** |  | **7** |
| 4.1 | Monitoring and Controlling | Complex | 4 |
| 4.2 | Scope of control | Medium | 1 |
| 4.3 | Tracking progress | Medium | 1 |
| 4.4 | Monitoring and controlling risks | Complex | 2 |
| **5** | **Closing** |  | **2** |
| 5.1 | Lesson learn | Simple | 1 |
| 5.2 | Final project report | Simple | 1 |
| 5.3 | Project archive | Simple | 1 |
| ***Total Estimated Effort (man-days)*** | | | ***77*** |

### 1.2. Project Objectives

| **No** | **Testing Stage** | **Test Coverage** | **No. of Defects** | **% of Defect** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| 1 | Integration Test | 300 | 25 | 8,3% |  |
| 2 | System Test | 300 | 25 | 8,3% |  |

### 

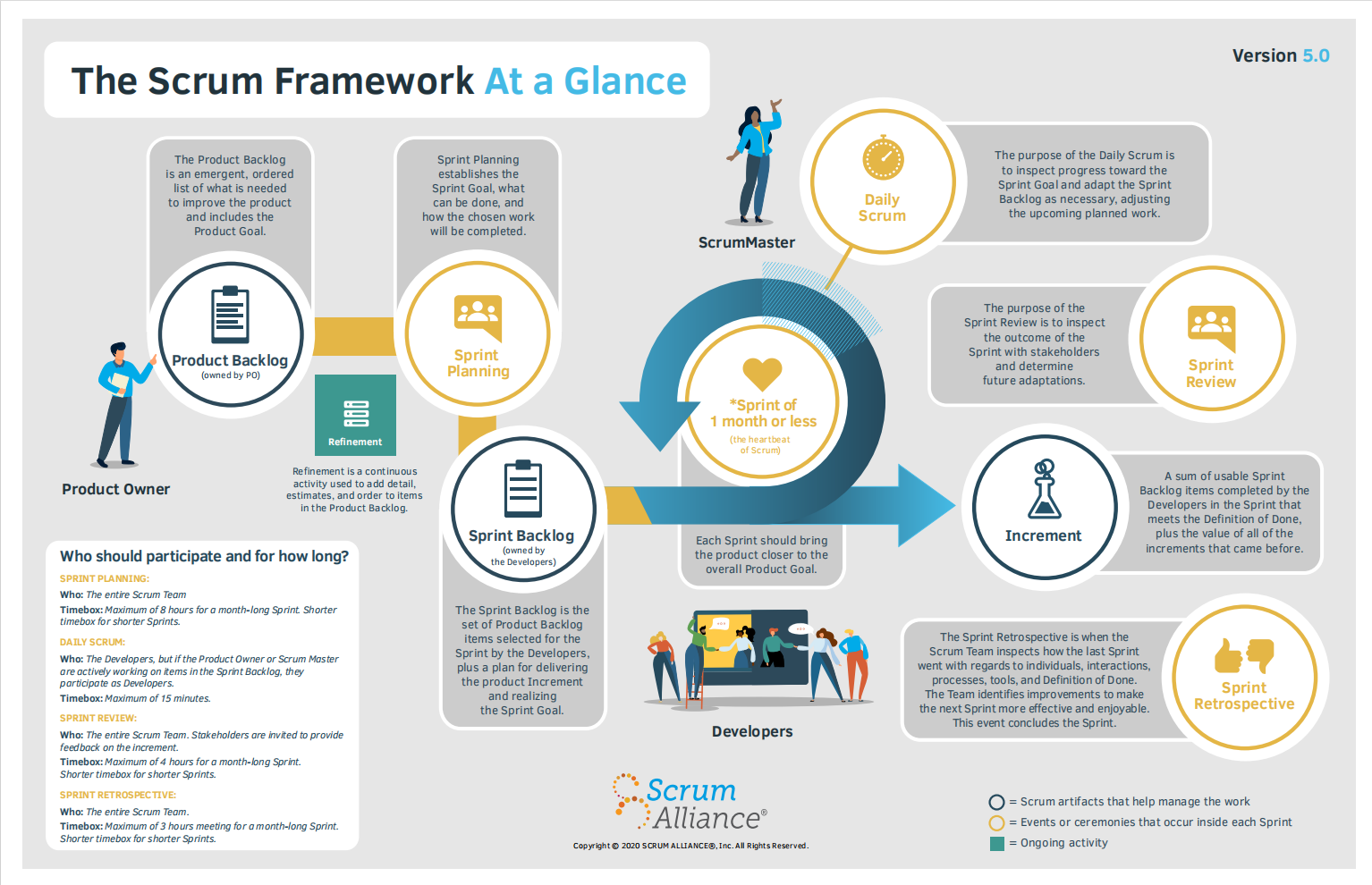
### 1.3. Project Risks

| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| --- | --- | --- | --- | --- |
| 01 | Driver/Porter/Reviewer No-shows or Unavailability | High | Medium | Create an automated backup assignment system or notify managers for manual assignment; incentivize backup staff. |
| 02 | Technical challenges in integrating real-time tracking and map services into the app. | High | Medium | Allocate more resources for technical research, and work closely with third-party service providers to resolve integration issues. |
| 03 | Server instability or downtime affecting user access to the MoveMate platform. | High | High | Implement backup servers and monitor server performance closely |

## 

## 2. Management Approach

### 2.1 Project Process

**

*Source:* [*https://www.scrumalliance.org/about-scrum*](https://www.scrumalliance.org/about-scrum)

We chose to build this project using the Scrum framework from Agile because:

It allows us to adapt to the unpredictable requirements of customers. By breaking the development down into smaller, manageable parts, we can refine the process step by step, ensuring that the final system truly meets the customer’s vision.

### 2.2 Quality Management

In this project, we employ the following strategy to raise project quality:

· API convention:

o Name the API Route that corresponds to the system entity.

o The response of APIs must be returned following:

* 200, 400, 401, 403, 404, 500 and 520 are the status codes used in the system.
* Message body must be returned in JSON format.

· Coding convention:

o Set the meaningful names for variables.

o Add comments to code can be confusing.

o Use Pascal case for naming Class, Interface, Method, Property, Constant.

o Use Camel case for naming Variable.

o Use Screaming snake case for naming Enum.

· CI/CD:

o Use GitHub Action to continue deploying, testing on server site

· Source code:

o Create new branch/pull request for every new feature/fix/enhance

o Branches: main, development, and features

o New branch is merged to develop after reviewed by leader or assigned member

· System testing:

o All team members must run all test cases and free tests again after completing the code for the entire system to make sure the system is operating smoothly.

### 2.3. Training Plan

| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| --- | --- | --- | --- |
| ASP.NET Web API Core 8 with Entity Framework | Cổ Phước Vinh  Lê Hàn Anh | Week 1, 7 days | Mandatory |
| Flutter | Nguyễn Thành Vinh  Phan Văn Tuấn  Đặng Nguyễn Hoài Phương | Week 1, 7 days | Mandatory |
| SQL Server | All members | Week 1, 7 days | Mandatory |
| GitHub | All members | Week 1, 7 days | Mandatory |
| Trello | All members | Week 1, 7 days | Mandatory |
| Figma | All members | Week 1, 7 days | Mandatory |
| Firebase | All members | Week 1, 7 days | Mandatory |

## 3. Project Deliverables

| **Iteration** | **Sprint objective** | **Duration** | **Deliverable** |
| --- | --- | --- | --- |
| 1 | Project Initialization, User Stories, SRS Document | 14 days | Project planning, Software Requirements and defined user stories |
| 2 | Basic authentication and initial booking module | 14 days | Working authentication module, basic booking system, demo |
| 3 | Service management module and booking enhancements | 14 days | Service management module, expanded booking features, functional testing |
| 3 | Schedule management, wallet integration and transaction module | 14 days | Schedule management, wallet and transaction integration, testing |
| 3 | Staff Management and Truck Management Modules | 14 days | Working staff management and truck management modules, integration tests |
| 3 | Discount and Fee Management Module | 14 days | Discount and fee management module, system test cases, and feedback |
| 4 | End-to-End System Testing, Bug Fixes | 14 days | System testing, bug fixes, usability improvements, ready for demo |
| 5 | User Manual and Final Documentation | 14 days | User guides, final system testing, final report, and deployment |

## 

## 4. Responsibility Assignments

| **Fullname** | **Role** | **Responsibility** |
| --- | --- | --- |
| Nguyễn Thị Cẩm Hương | Supervisor | · Instruct project team  · Supervise project status  · Review deliverables  · Answer questions about the project |
| Cổ Phước Vinh | Leader | · Managing process  · Track backlog  · Clarifying requirements  · Prepare documents  · Quality management  · Support other team members  · Arrange meeting |
| Lê Hàn Anh  Đặng Nguyễn Hoài Phương  Phan Văn Tuấn  Nguyễn Thành Vinh | Team members | · Designing database  · Clarifying requirements  · Prepare documents  · Coding  · Create test plan  · Testing  · GUI design  · Support other team members |

## 

## 5. Project Communications

| Communication Item | Who/ Target | Purpose | When, Frequency | Type, Tool, Method(s) |
| --- | --- | --- | --- | --- |
| Supervisor  communication | Nguyen Thi Cam Huong and team members | · Review documentation  · Demonstrate features  · Evaluate progress and result | 2 times per week | Face to face, Google Meet, Zalo |
| Daily meeting | Team members | · Raise opinions, problems  · Ask for help from other members  · Report working status to leader | Always | Face to Face, Messenger, Google Meet,  Zalo |

## 

## 6. Configuration Management

### 

### 6.1. Document Management

We use Google Drive, Office 365, Trello to manage documents since they supply real-time data synchronization. Team members can easily collaborate in real-time thanks to Google Drive. Each participant can at once view the outcomes of the others. We can also evaluate document updates in Office 365 so that we can do so whenever necessary.

### 

### 6.2. Source Code Management

We decided to manage our source code on GitHub. The version control system makes it possible for team members to work together on source code efficiently, conveniently, and easily to resolve conflict. It is especially useful for looking back in time and at once recognizing changes a collaborator made.

### 

### 6.3. Tools & Infrastructures

#### 6.3.1. Tools

| Type | Tools |
| --- | --- |
| IDEs/Editors | Visual Studio Code, Visual Studio, Figma, |
| UML tools | lucid.app |
| Source Version Control | GitHub |
| Project Management Tool | Trello |

#### 

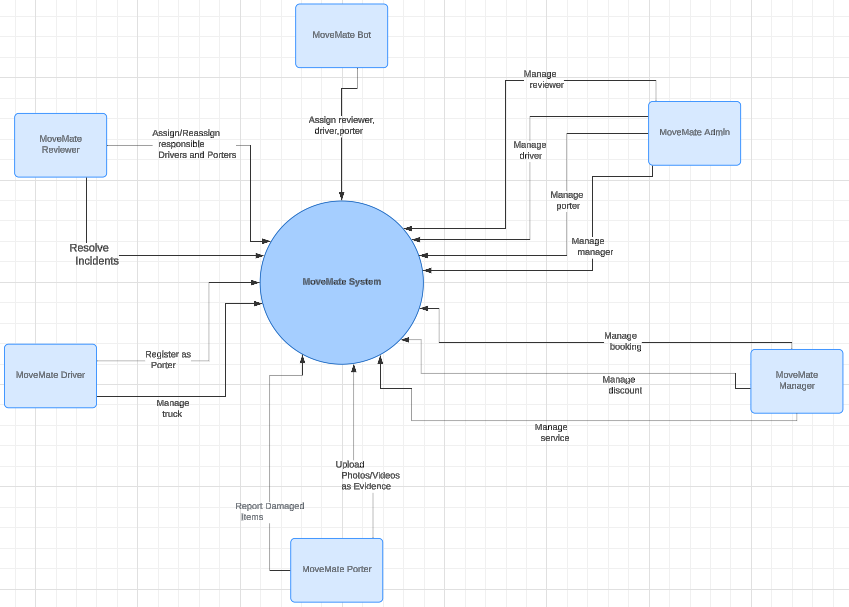
#### 6.3.2. Infrastructures

| Type | Infrastructure |
| --- | --- |
| Framework | Entity Framework Core 8 (Backend),  Flutter (Mobile),,  Nest, |
| Programming Languages | Typescript, C#, Dart |
| DBMS | SQL Server |
| Deployment Server | Github Action, VPS - Ubuntu server - NGINX |
| Storage | Firebase Database, Cloudiary |

# III. Software Requirement Specification

## 1. Product Overview

MoveMate is a web and mobile app created to make moving houses easier for individuals, families, and small businesses. It simplifies everything, from planning to execution, by offering features like vehicle booking, loading services, and real-time tracking. MoveMate takes care of the tough parts of moving, helping users figure out the right truck size, manage labor, and get clear, upfront pricing. It’s designed to remove the hassle and uncertainty, making the moving process smoother and more efficient.



## 2. User Requirements

### 2.1 Actor

| **No** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | Admin | The admin is responsible for managing the system related to managing accounts in the system including accounts of Managers, Drivers, Porters, Reviewers, and Customers. |
| 2 | Manager | Manages bookings, services, and discounts. Reviews and approves driver and their truck registrations. Reviews and approves compensation claims, assigns driver, porter and reviewer in case of incidents |
| 3 | Driver | Receives moving jobs and may register as a porter. Updates the status of the booking and handles cash payments if required. |
| 4 | Porter | Receives jobs for loading/unloading services. Updates booking statuses, reports any issues or damages, and provides proof (photos/videos) of completed tasks. Reports incidents during the move. |
| 5 | Reviewer | Visits the customer’s house to review and assess the moving requirements. Updates booking details based on customer information, recommend services (e.g., vehicle type), and assigns lead drivers and lead porters for the job. |
| 6 | Customer | The customer books moving services, provides details about their house, and tracks their booking in real-time. They can modify their booking or submit claims if occur incident. |

### 2.2. Use Case Diagram

#### 2.2.1. Admin and Manager Use Case Diagram

#### 2.2.2. Driver, Porter, Reviewer Use Case Diagram

#### 2.2.2. Customer Use Case Diagram

### 2.3. List of Use Case Descriptions

#### 2.3.1. Customer Use Case Descriptions

| **No** | **Actor** | **Use case** | **Use case description** |
| --- | --- | --- | --- |
| 01 | Customer | Log in | The customer logs into the MoveMate system to access the booking and service features. |
| 02 | Customer | View profile | The customer views their personal profile |
| 03 | Customer | Update profile | The customer updates their personal information such as address, contact details |
| 04 | Customer | Forgot password | The customer resets their password by following a recovery process if they have forgotten it |
| 05 | Customer | Feedback | The customer provides feedback on services, including rating the drivers, porters, or overall service |
| 06 | Customer | Create booking | The customer creates a new booking by selecting type truck and service details such as date, time, and type of service |
| 07 | Customer | Confirm booking update | The customer approver or reject any updates made to their booking (e.g., changes to time, truck type, …) |
| 08 | Customer | Update time booking | The customer updates the time of their booking once time |
| 09 | Customer | Upload image and video | The customer uploads images or videos of their household items for review |
| 10 | Customer | View booking details | The customer views the detailed information about their booking, including driver details, service status,... |
| 11 | Customer | Book round-trip | The customer books a round-trip moving service if they need to transport more items |
| 12 | Customer | Cancel booking | The customer cancels an existing booking |
| 13 | Customer | View transaction history | The customer views their transaction history, including past payments and deposits |
| 14 | Customer | Chat with reviewer, driver, or porter | The customer communicates with the reviewer, driver, or porter via chat |
| 15 | Customer | Withdraw money | The customer withdraws money from their MoveMate wallet |
| 16 | Customer | Recharge money | The customer recharges or adds funds to their MoveMate wallet for payments or services. |
| 17 | Customer | Payment | The customer makes payments for services, either using their wallet such as VNPay, Momo, PayOS |
| 18 | Customer | Report issues | The customer reports any issues or incidents during the move (e.g., damages, delays) to seek resolution |
| 19 | Customer | Give bonus | The customer can give bonuses or tips to drivers or porters for exceptional service |
| 20 | Customer | View services | The customer views all services provided |

### 

#### 2.3.2. Driver, Porter, Reviewer Use Case Descriptions

| **No** | **Actor** | **Use case** | **Use case description** |
| --- | --- | --- | --- |
| 01 | Driver, Porter, Reviewer | Log in |  |
| 02 | Driver, Porter, Reviewer | View profile |  |
| 03 | Driver, Porter, Reviewer | Forgot password |  |
| 04 | Driver | Register as porter |  |
| 05 | Driver | Confirm driver team arrival |  |
| 06 |  | Update status of truck |  |

## 

#### 2.3.1. Admin and Manager Use Case Descriptions

| **No** | **Actor** | **Use case** | **Use case description** |
| --- | --- | --- | --- |
| 01 | Admin | View customer | The admin views customer profiles, contact information, their booking history |
| 02 | Admin | Ban customer | The admin bans a customer from accessing the system |
| 03 | Admin | Remove customer | The admin removes a customer's account and all associated data from the system |
| 04 | Admin | Create driver | The admin creates a new driver account, including personal details, driving license, and vehicle information |
| 05 | Admin | Update driver | The admin updates a driver’s profile, modifying details like contact information, assigned vehicle, or work status |
| 06 | Admin | Remove driver | The admin removes a driver from the system |
| 07 | Admin | View driver | The admin views driver profiles |
| 08 | Admin | Create porter | The admin creates a new porter account with details porter’s profile, contact information |
| 09 | Admin | Update porter | The admin updates a porter’s information, contact details, or work status |
| 10 | Admin | Remove porter | The admin removes a porter from the system, ending their access to porter-related tasks |
| 11 | Admin | View porter | The admin views porter profiles |
| 12 | Admin | Create reviewer | The admin creates a new reviewer account with details reviewer’s profile, contact information |
| 13 | Admin | Update reviewer | The admin updates a reviewer’s information, such as changing their contact details, work status |
| 14 | Admin | Remove reviewer | The admin removes a reviewer from the system |
| 15 | Admin | View reviewer | The admin views reviewer profiles |
| 16 | Admin | Create manager | The admin creates a manager account, assigning permissions |
| 17 | Admin | Update manager | The admin updates a manager’s profile, modifying details such as their permissions |
| 18 | Admin | Remove manager | The admin removes a manager from the system, terminating their administrative access and control. |
| 19 | Admin | View manager | The admin views manager profiles |
| 20 | Manager | Track booking status | The manager monitors the status of ongoing bookings |
| 21 | Manager | Resolve issue | The manager resolves any incidents or issues (e.g., delays, vehicle breakdowns) that arise during a booking. |
| 22 | Manager | View booking details | The manager views detailed information about specific bookings, including customer details and assigned staff |
| 23 | Manager | Remove discount codes and promotions | The manager deletes outdated or invalid discount codes and promotional offers from the system |
| 24 | Manager | Create discount codes and promotions | The manager generates new discount codes and promotional offers for customers |
| 25 | Manager | Reply customer feedback | The manager responds to customer feedback |
| 26 | Manager | Remove services offered | The manager removes services that are no longer offered from the system |
| 27 | Manager | Modified services offered | The manager updates the details of existing services (e.g., price, description) to reflect current offerings |
| 28 | Manager | Add services offered | The manager adds new services to the list of available options for customers |
| 29 | Manager | Reject application register for driver | The manager declines a driver's application to join the system, if it does not meet requirements |
| 30 | Manager | Approve application register for driver | The manager approves a driver's application to join the system, including approving the driver's truck |
| 31 | Manager | Approve compensation claim | The manager approves customer claims for compensation related to service issues |
| 32 | Manager | Reject compensation claim | The manager rejects customer claims for compensation if they do not meet the necessary criteria |
| 33 | Manager | Chat with drivers, porters, and reviewers | The manager communicates with drivers, porters, and reviewers regarding bookings, incidents, and updates |

## 

## 3. Functional Requirements

### 3.1 System Functional Overview

*[Provide functionality overview of software system: screen flow, screen descriptions, system user roles, screen authorization, non-screen functions, ERD]*

### 3.2 <Feature Name 1>

*[Profile functional specification for the feature, with the details on each function]*

#### 3.2.1 <Function Name 1.1>

…

### 3.3 <Feature Name 2>

#### 3.3.1 <Function Name 2.1>

…

## 4. Non-Functional Requirements

### 4.1 External Interfaces

*[This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements.]*

### 4.2 Quality Attributes

*[List all the required system characteristics (quality attributes) specification]*

## 5. Requirement Appendix

*[List out other requirements, appendix information etc. in this part]*

### 5.1 Business Rules

*[Provide common business rules that you must follow. The information can be provided in the table format as the sample below]*

### 5.2 Common Requirements

*[Fill all the common requirements here..]*

### 5.3 Application Messages List

### 5.4 Other Requirements…

# IV. Software Design Description

*[Provide final software design information follow the template as part II in the Report #4]*

## 1. System Design

### 1.1 System Architecture

*[The content of this section includes the overall diagram which includes the sub-systems, the external systems, and the relationship/connection among them. You need also provide the explanation for each of the diagram components (modules, sub-systems, external systems, etc.)].*

### 1.2 Package Diagram

*[Provide the package diagram for each sub-system. The content of this section includes overall package diagram(s) and the explanation for each package (or namespace)]*

## 2. Database Design

*[Provide the files description, database table relationship & table descriptions]*

## 3. Detailed Design

### 3.1 <Feature/Function Name1>

*[Provide the detailed design for the feature <Feature Name1>. It includes Class Diagram, Class Specifications, and Sequence Diagram(s);* ***For the features/functions with the same structure of class & sequence diagrams, you need to provide the diagrams once for one feature/function and refer to those diagrams from other features/functions****]*

#### 3.1.1 Class Diagram

*[This part presents the class diagram for the relevant feature]*

***3.1.2 <Sequence Diagram Name1>***

*[Provide the sequence diagram(s) for the feature]*

***3.1.2 <Sequence Diagram Name2>***

***3.1.3 …***

### 3.2 <Feature/Function Name2>

…

# V. Software Testing Documentation

*[Provide final software testing information follow the template as part II in the Report #5]*

## 1. Scope of Testing

*[Describe the scopes of the test. Those include the target-of-test’s features, functions, and non-functional requirements that will or will not be tested.*

*Describe the stages/levels of testing that would be applied to your project - Unit, Integration, or System test. Each includes the in-charge, inputs/time, focuses, acceptance criteria.*

*List any constraints or assumptions made during the development of this document that may impact the design, development or implementation of testing]*

## 2. Test Strategy

*[List out and describe all testing types (you can refer the test types listed below or any other test types to selected the suitable ones for the project; for each selected test types you need to provide the following information: test objective, technique, completion criteria, etc.), test levels that those test types would be performed, & the details of test supporting tools would be used in the project]*

### 2.1 Testing Types

*[List out and describe here the testing types which you would apply in your project. You need to mention following information for each type of testing: objective, technique, completion criteria]*

### 2.2 Test Levels

*<List out and describe here the testing levels which you would execute in your project. Besides, clearly state the test types which are performed in each test level that you plan for this project>*

### 2.3 Supporting Tools

*<List of the test supporting tools which will be employed for this project>*

## 3. Test Plan

### 3.1 Human Resources

*[List and provide the details on roles and responsibilities of the project members who would involve in testing works]*

### 3.2 Test Environment

*[List and provide the details about the tools (software, hardware, infrastructure) which the project would use for testing]*

### 3.3 Test Milestones

*[Separate test milestones, which should be identified to communicate project status accomplishments]*

## 4. Test Cases

*[Prepare the details on the test cases following the provided template*

* *Unit Test Cases: Report5\_Unit Test.xls*
* *Other Test Cases (IT, ST, AT): Report5\_Test Report.xls]*

## 5. Test Reports

*[Provide the test result, statistics and the relevant test analysis for your testing in the project]*

# VI. Release Package & User Guides

*[Provide final software testing information follow the template as part II in the Report #6]*

## 1. Deliverable Package

*[The section will list all source programs, scripts, documents with version number in this release. You can see the example following table for reference, can customize or delete if not using belong to each project characteristics]*

| **No.** | **Deliverable Item** | **Description** |
| --- | --- | --- |
| 1 | Schedule/Task Tracking |  |
| 2 | Project Backlog |  |
| 3 | Source Codes |  |
| 4 | Database Script(s) |  |
| 5 | Final Report Document |  |
| 6 | Test Cases Document |  |
| 7 | Defects List |  |
| 8 | Issues List |  |
| 9 | Slide |  |

## 2. Installation Guides

### 2.1 System Requirements

*[Define any system requirements necessary to support the application, including the software and relevant configurations]*

### 2.2 Installation Instruction

*[Includes installation instructions and configuration guidelines]*

## 3. User Manual

### 3.1 Overview

*[Descript the overview of the application and if could, insert the features workflow to help user has the overview of all the features in this application]*

### 3.2 Workflow 1

*[Describe the purpose of this workflow, draw workflow diagram and other relevant diagrams]*

*[Describe the detailed guides for the workflow by providing the brief description, step by step guides (attached with user interface) of how to use that function]*

### 3.3 Workflow 2

…