Applied Science Private University
Faculty of Information Technology

Graduation Project (1) Report

Rabit App

Prepared by:

Asaad aljammal – 202111412

Sara okkeh – 202111353

Mahmoud Alotaibi – 201920179

Mohammad Allataifeh - 202211684

Supervised by:
DR Fadi Masalha

June 2024

Contents

Table of figure:

Table of tables:

Abstract

Rabit is a mobile application designed to bridge the gap between expats and local service providers. It accomplishes this through a unique digital currency system that benefits both expats and locals. Expats can subscribe to the app for a set number of service requests, with prices ranging from $10 to $50. Subscription tiers offer more requests at higher price points. They can also top up their app currency to request additional services. Local service providers receive this digital currency instead of cash.

This digital currency isn't just for service transactions. It can be redeemed for goods and services from local stores or for other services within the app. This system allows expats to access affordable services and connect with locals for support, ultimately facilitating their integration into the community.

Chapter 1: Introduction

1.1 Description of the current situation and opportunity

There's a gap between expats and locals in many communities. Expats often struggle to find reliable and affordable services when they move to a new place, hindering their integration and experience. Locals, on the other hand, might miss out on earning opportunities by not having a way to easily connect with expats who need help.

1.2 Related work

Provide a brief comparison with similar work. Rabit App shares similarities with various existing platforms that connect people with service providers, but caters specifically to expats. Here's a brief comparison:

TaskRabbit-style apps: These apps (e.g., Handy [1], Thumbtack [2]) connect users with local handymen, cleaners, and other service providers for various tasks. However, they might not have the expat-specific features like document translation or cultural immersion aspects that Rabit offers.

Local classifieds websites: Websites like Craigslist can help expats find services, but lack the vetting process and digital currency system that could ensure quality and security in Rabit.

Expat forums/communities: Online expat communities offer support and recommendations, but might not have a structured way to connect with service providers directly. Rabit provides a more streamlined platform for both service requests and fulfillment.

1.3 Problem statement (limitation of current systems)

Expats [3] moving to a new country often face challenges in finding reliable, affordable services and integrating into the local community Current methods have limitations :

1) **Existing task-rabbit style apps** might not cater to expat-specific needs, like document translation or finding culturally appropriate services.
2) **Local classifieds lack vetting processes and secure payment methods**, making it difficult to ensure service quality and user safety.
3) **Expat forums/communities** may not offer a structured way to connect with service providers directly, leaving expats frustrated and wasting time searching.

1.4 Problem solution

Rabit App proposes a mobile application that connects expats seeking services with local providers. Here's the approach:

- **Mobile App:** The core will be a user-friendly app with separate interfaces for expats and locals.
  - **Expats:** Search for services, browse provider profiles, request services, pay with digital currency, and rate experiences.
  - **Locals:** Register as service providers, set availability, receive requests, communicate with expats, accept payment in digital currency, and redeem earnings.

- **Digital Currency System:**
  - In-app currency eliminates the need for direct cash transactions, promoting security and convenience.

1.5 Project objectives

Rabit App aims to address the challenges faced by expats and locals by setting the following objectives:

- **Empower expats:**
    - Facilitate easy discovery and connection with reliable and affordable service providers.
    - Foster integration into the local community by enabling interaction with locals.
    - Enhance their overall experience in the new environment.
- **Benefit locals:**
    - Create opportunities to earn extra income by providing services to expats.
    - Boost local tourism by connecting with expats interested in exploring the area.
    - Encourage participation in a growing digital economy through the app's currency system.
- **Bridge the gap between expats and locals:**
    - Increase interaction and understanding between expats and local residents.
    - Promote cultural exchange and a more welcoming environment for expats.
    - Build a thriving digital marketplace for services catering to expats.

1.6 Technology and tools used

Here's a breakdown of the tools and technologies you've chosen to build Rabit App:

Mobile App Development:

- **Software:**
    - **Flutter:** Cross-platform mobile app development framework (uses Dart programming language).
    - **IDE:** Android Studio with the Flutter plugin provides a development environment for building Flutter apps.

Back-end Development:

- **Software:**
    - **Firebase**: This is a backend-as-a-service (BaaS) platform from Google that offers various services

Project Management:

- **Software:**
    - **Jira Account :** Project management tool for organizing tasks, tracking progress, and collaborating within a development team.

Diagram applications:

- Microsoft project
- Microsoft Visio
- visual paradigm

1.7 Project plan for GP2 (Gantt chart, PERT chart)

Sprint 1 :

a) **Planning (Week 1):**
   (1) Define user stories and acceptance criteria for each task (2 week).
   (2) Estimate effort required for each task (1 week).
   (3) Assign tasks to team members and create a sprint backlog (1 week).

b) **Development:**
    (1) **Mobile App UI/UX Design (1 week).**
    (2) **Mobile App Development .**
        (a) Develop user authentication system (1 week).
        (b) Complete core functionalities development (1 week).
        (c) Integrate in-app communication channels (1 week).
        (d)
    (3) **Back-end Development :**
        (a) Set up Node.js server and configure database connection (1 week).
        (b) Design and implement database schema to store user and service data(1 week).

c) **Testing :**
i) Conduct internal testing of mobile app UI, user registration, and service search functionalities. (5 days).
    (1) Conduct internal testing of new functionalities (in-app communication, digital currency top-up). (5 days).
    (2) Review database functionality and data storage.(4 days)

d) **Sprint Review :**
    (1) Showcase developed features to stakeholders and get feedback.(2 days)
    (2) Review if user stories and acceptance criteria have been met. (2 days)

e) **Retrospective :**
    (1) Analyze team performance, identify areas for improvement, and refine Agile practices (3 days).

Sprint 2 .

a) **Planning :**
    (1) Review feedback from Sprint 1 and update backlog (1 week).
    (2) Define user stories and acceptance criteria for Sprint 2 tasks. (1 week)
    (3) Estimate effort and assign tasks to team members. (1 week)

b) **Development :**

   (1) **Mobile App Development :**

      (a) Develop digital currency system with top-up functionality on the mobile app (1 week).

      (b) Implement location-based services (1 week).

   (2) **Back-end Development :**

      (a) Develop server-side APIs for user management, service listing, and digital currency transactions(1 week)

c) **Integration & Testing :**

   (1) Integrate PayPal payment gateway for secure top-up of digital currency (1 week).

   (2) Integrate mobile app with back-end server APIs for data exchange (1 week).

   (3) Conduct rigorous testing of mobile app functionalities on various Android devices (1 week).

   (4) Perform security testing of app and server to ensure data protection (1 week).

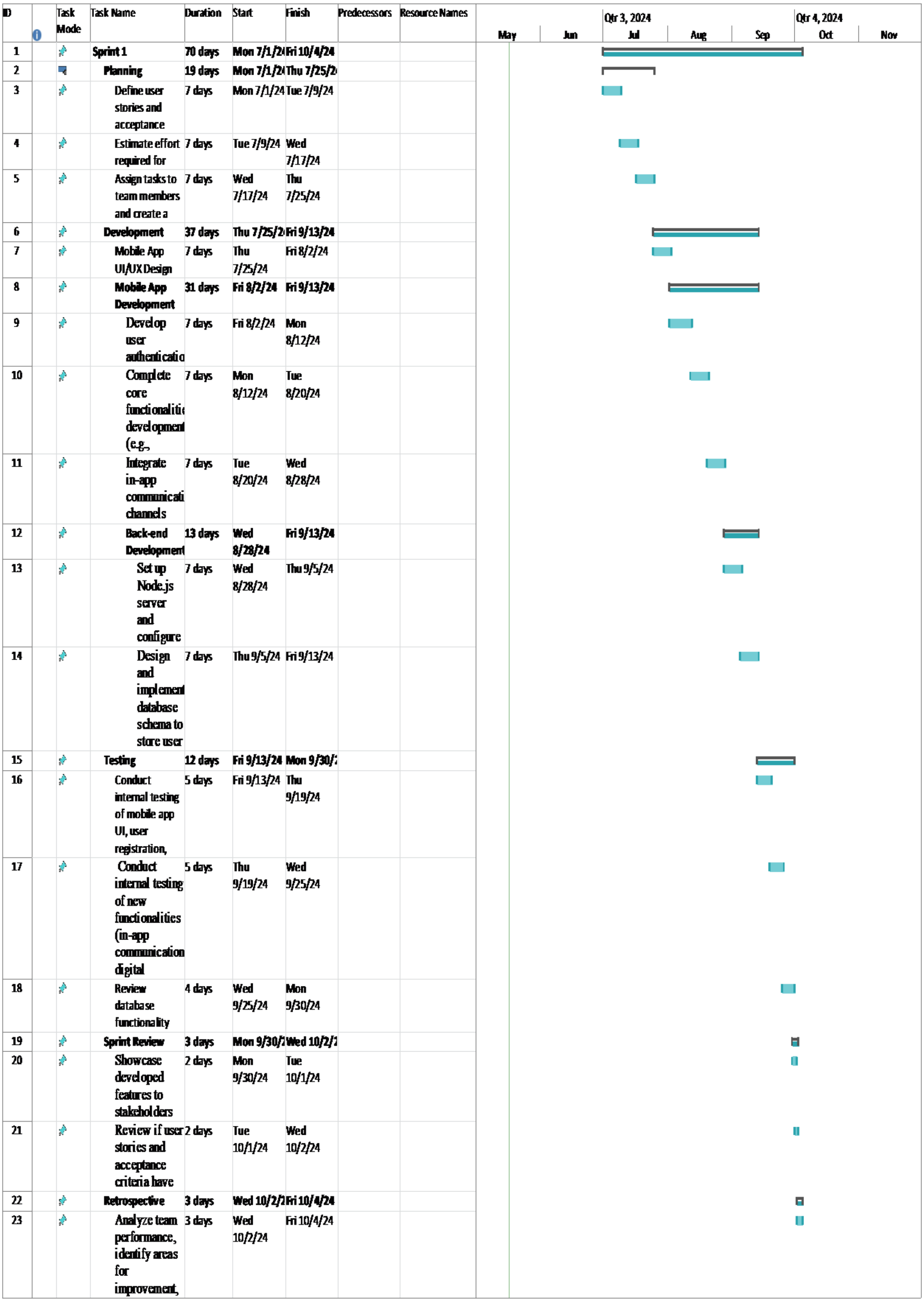
d) **Deployment & Release :**

      (a) Deploy the app to Google Play Store (1 week)

e) **Feedback:**

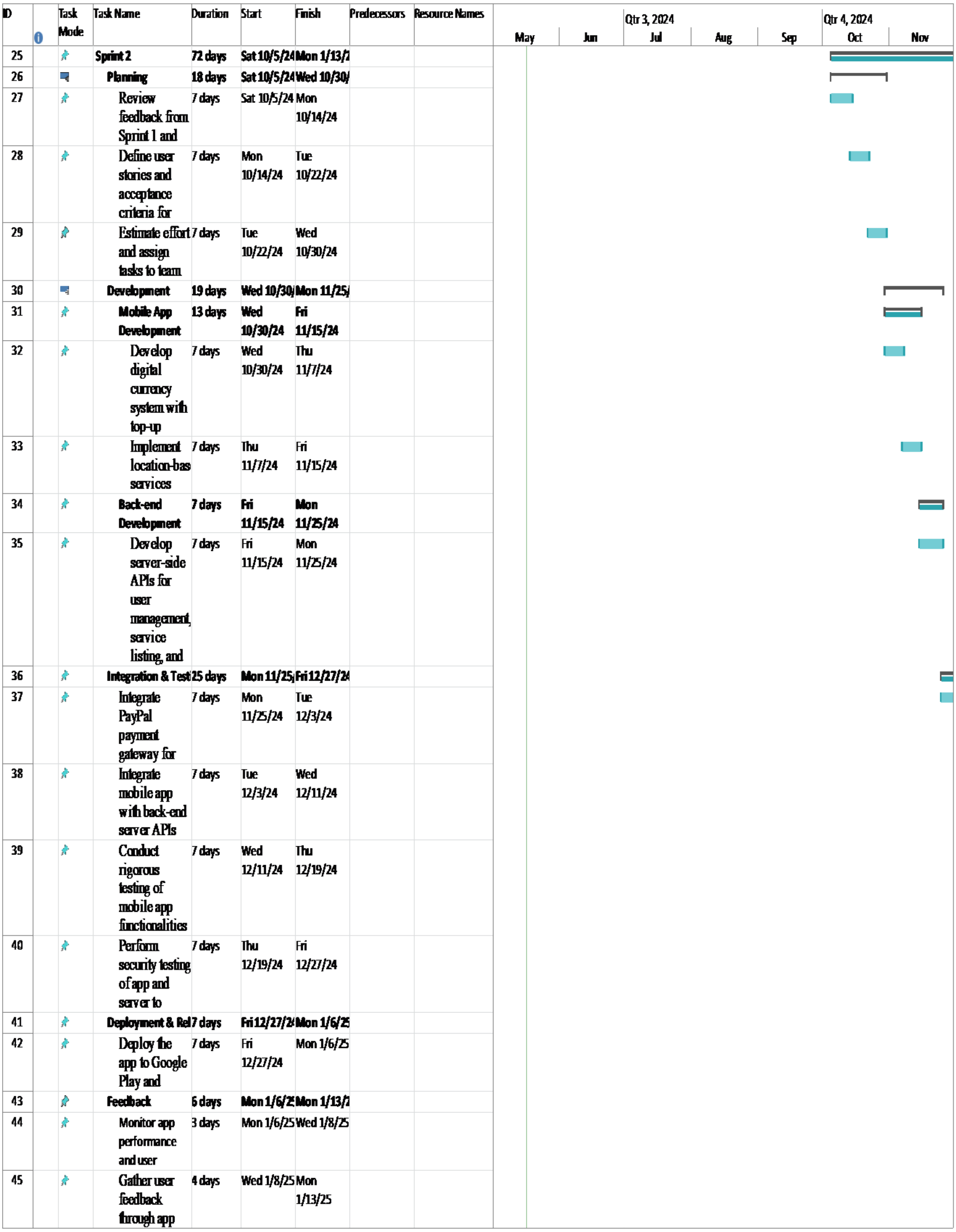
   (1) Monitor app performance and user feedback after launch (3 days).

   (2) Gather user feedback through app reviews and surveys (4 days).



ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1			**Sprint 1**	**70 days**	**Mon 7/1/2**	**Fri 10/4/24**		
2			**Planning**	**19 days**	**Mon 7/1/2**	**Thu 7/25/2**		
3			Define user stories and acceptance	7 days	Mon 7/1/24	Tue 7/9/24		
4			Estimate effort required for	7 days	Tue 7/9/24	Wed 7/17/24		
5			Assign tasks to team members and create a	7 days	Wed 7/17/24	Thu 7/25/24		
6			**Development**	**37 days**	**Thu 7/25/2**	**Fri 9/13/24**		
7			Mobile App UI/UX Design	7 days	Thu 7/25/24	Fri 8/2/24		
8			**Mobile App Development**	**31 days**	**Fri 8/2/24**	**Fri 9/13/24**		
9			Develop user authenticatio	7 days	Fri 8/2/24	Mon 8/12/24		
10			Complete core functionaliti development (e.g.,	7 days	Mon 8/12/24	Tue 8/20/24		
11			Integrate in-app communicati channels	7 days	Tue 8/20/24	Wed 8/28/24		
12			**Back-end Development**	**13 days**	**Wed 8/28/24**	**Fri 9/13/24**		
13			Set up Node.js server and configure	7 days	Wed 8/28/24	Thu 9/5/24		
14			Design and implement database schema to store user	7 days	Thu 9/5/24	Fri 9/13/24		
15			**Testing**	**12 days**	**Fri 9/13/24**	**Mon 9/30/**		
16			Conduct internal testing of mobile app UI, user registration,	5 days	Fri 9/13/24	Thu 9/19/24		
17			Conduct internal testing of new functionalities (in-app communication digital	5 days	Thu 9/19/24	Wed 9/25/24		
18			Review database functionality	4 days	Wed 9/25/24	Mon 9/30/24		
19			**Sprint Review**	**3 days**	**Mon 9/30/**	**Wed 10/2/**		
20			Showcase developed features to stakeholders	2 days	Mon 9/30/24	Tue 10/1/24		
21			Review if user stories and acceptance criteria have	2 days	Tue 10/1/24	Wed 10/2/24		
22			**Retrospective**	**3 days**	**Wed 10/2/**	**Fri 10/4/24**		
23			Analyze team performance, identify areas for improvement,	3 days	Wed 10/2/24	Fri 10/4/24		

Figure 1Gantt chart sprint 1



ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
25		**Sprint 2**	**72 days**	**Sat 10/5/24**	**Mon 1/13/2**		
26		**Planning**	**18 days**	**Sat 10/5/24**	**Wed 10/30/**		
27		Review feedback from Sprint 1 and	7 days	Sat 10/5/24	Mon 10/14/24		
28		Define user stories and acceptance criteria for	7 days	Mon 10/14/24	Tue 10/22/24		
29		Estimate effort and assign tasks to team	7 days	Tue 10/22/24	Wed 10/30/24		
30		**Development**	**19 days**	**Wed 10/30/**	**Mon 11/25/**		
31		**Mobile App Development**	**13 days**	**Wed 10/30/24**	**Fri 11/15/24**		
32		Develop digital currency system with top-up	7 days	Wed 10/30/24	Thu 11/7/24		
33		Implement location-bas services	7 days	Thu 11/7/24	Fri 11/15/24		
34		**Back-end Development**	**7 days**	**Fri 11/15/24**	**Mon 11/25/24**		
35		Develop server-side APIs for user management, service listing, and	7 days	Fri 11/15/24	Mon 11/25/24		
36		**Integration & Test**	**25 days**	**Mon 11/25/**	**Fri 12/27/24**		
37		Integrate PayPal payment gateway for	7 days	Mon 11/25/24	Tue 12/3/24		
38		Integrate mobile app with back-end server APIs	7 days	Tue 12/3/24	Wed 12/11/24		
39		Conduct rigorous testing of mobile app functionalities	7 days	Wed 12/11/24	Thu 12/19/24		
40		Perform security testing of app and server to	7 days	Thu 12/19/24	Fri 12/27/24		
41		**Deployment & Rel**	**7 days**	**Fri 12/27/2**	**Mon 1/6/25**		
42		Deploy the app to Google Play and	7 days	Fri 12/27/24	Mon 1/6/25		
43		**Feedback**	**6 days**	**Mon 1/6/2**	**Mon 1/13/2**		
44		Monitor app performance and user	3 days	Mon 1/6/25	Wed 1/8/25		
45		Gather user feedback through app	4 days	Wed 1/8/25	Mon 1/13/25		

Figure 2Gantt chart sprint 2

Chapter 2: Requirements and Analysis

2.1 Software Process Model

We will be building Rabit App from scratch and there's no existing functional software, a hybrid model is a suitable choice. This approach combines elements from different software process models to leverage their strengths and address the specific needs of your project. Here's how can tailor a hybrid model for Rabit App development:

Model Selection:

- **Waterfall Model (Early Stages):**
  - Provides a structured approach for initial planning, requirement gathering, and high-level design.
  - This initial phase helps define the core functionalities of the app for both expats and locals.
- **Agile Model (Development and Iteration):**
  - Enables iterative development with frequent releases and user feedback integration.
  - Perfect for building core functionalities, testing them with real users, and incorporating their feedback for continuous improvement.

2.2 System scope with explanation

- The proposed system aims to bridge the gap between expats and locals in the community.
- The system will provide the ability to browse service categories, search for specific services, and view provider profiles.
- The system will provide the ability to Locals can view and accept/decline service requests.
- The system will provide the ability to manage profile information and communication settings.
- The system will have features such as Secure in-app chat for expats and locals to connect and discuss service details.
- The system will have features such as subscription options.
- The project must be finished before 2/2025
- The system must be able to handle at least a thousand visitors every month.

2.3 List of Functional Requirements and Non-Functional Requirements

This document outlines the functional requirements for the Rabit App, a mobile application connecting expats with local service providers:

A. Functional Requirements:

1. The user shall be able to login to the Rabit System.

    1.1 The user shall enter his/her email.

    1.2 The user shall enter the password to login.

    1.3 The system shall open the home window, if the email and password are valid.

    1.4 The system shall show an error message, if the email or password is invalid.

    1.5 The system shall provide a registration button.

    1.6 The system shall provide a forget password button.

2. The user shall be able to register as a new user to the Rabit System.

    2.1 The user shall enter his/her first and last name.

    2.2 The user shall enter his/her email address.

        2.2.1 The email address shall be valid.

            2.2.1.1 The system shall send a verification code to the email address.

    2.3 The user shall enter his/her password.

        2.3.1 The password shall consist of at least 8 characters.

        2.3.2 The password shall contain a mix of uppercase letters, lowercase letters, and digits.

    2.4 The user shall reenter his/her password.

    2.5 The user shall enter his/her phone number.

        2.5.1 The phone number shall be valid.

            2.5.1.1 The system shall send a verification code to the phone number.

    2.6 The system shall open the home window, if the email and password are valid.

    2.7 The system shall show an error message, if the email or password is invalid.

3. The user shall be able to recover his/her password.

    3.1 The user shall enter his/her email address.

        3.3.1 The system shall send the link for recover email.

4. The user shall be able to access the home page.

   4.1 The user shall be able to search in sections.
   4.2 The user shall be able to see his/her navigation.
   4.3 The user shall be able to see his/her number of services.
   4.4 The system shall provide a cart button.
   4.5 The system shall provide Space for advertisements.
   4.6 The user shall be able to see all sections.
   4.7 The user shall be able to Navigate between application screens by bottom bar.

5. The user shall be able to access the navigation page.

   5.1 The user shall be able to search in navigation.
   5.2 The system shall provide a navigation setting button.
      5.2.1 The user shall be able to Sort navigation by time and not read.
   5.3 The system shall provide a card for navigation information.
      5.3.1 The user shall be able to see navigation information.
      5.3.2 The system shall provide a profile page button.
      5.3.3 The user shall be able to delete navigation and unfollow the page by clicking on navigation setting.

6. The user shall be able to access the services page.

   6.1 The user shall be able to see his/her number of services.
   6.2 The user shall be able to buy more services.
   6.3 The user shall be able to add offers to the shopping cart.

7. The user shall be able to access the shopping cart.

    7.1 The user shall be able to see all shopping cart contents.
    7.2 The system shall provide a field for discount coupons.
    7.3 The user shall be able to go to payment methods.

8. The user shall be able to access the payment methods page.

    8.1 The user shall be able to purchase via Visa.
    8.2 The user shall be able to purchase via master card.
    8.3 The user shall be able to purchase via PayPal.

9. The user shall be able to access the section page.

    9.1 The user shall be able to search in the section.
    9.2 The user shall be able to see his/her navigation.
    9.3 The user shall be able to sort the services by cities.
    9.4 The user shall be able to go back to the previous page.
    9.5 The user shall be able to select the section.
    9.6 The user shall be able to Navigate between application screens by bottom bar.

10. The user shall be able to access the selected section page.

10.1 The user shall be able to search in the section.

10.2 The user shall be able to see his/her navigation.

10.3 The user shall be able to sort the services by cities.

10.4 The user shall be able to go back to the previous page.

10.5 The user shall be able to select the section.

10.6 The system shall provide cards for services.

10.6.1 The user shall be able to see pictures and videos for service.

10.6.2 The user shall be able to see the name and section of the service.

10.6.3 The user shall be able to communicate with the service owner via the phone number and the application’s messaging.

10.6.4 The user shall be able to add the service to their favorite.

11. The user shall be able to access the service page.

11.1 The user shall be able to go back to the previous page.

11.2 The user shall be able to see pictures and videos for service.

11.3 The user shall be able to see the name and section of the service.

11.4 The user shall be able to see the description of the service.

11.5 The user shall be able to communicate with the service owner via the phone number and the application’s messaging.

11.6 The user shall be able to add the service to their favorite.

11.7 The user shall be able to share the service.

11.8 The user shall be able to book the service.

11.9 The user shall be able to see the service owner page.

12. The user shall be able to access the booking page.

12.1 The user shall be able to see his/her navigation.

12.2 The user shall be able to search on the booking page.

12.3 The user shall be able to go back to the previous page.

12.4 The user shall be able to see all free day and time for booking.

12.5 The user shall be able to confirm the booking.

12.6 The user shall be able to see the service owner rating.

12.7 The user shall be able to see the service owner page.

13. The user shall be able to access my account page.

13.1 The user shall be able to see his/her navigation.

13.2 The system shall provide a setting button.

13.3 The system shall provide a card for displaying information.

13.3.1 The user shall be able to see his/her information.

13.3.2 The user shall be able to see his/her rating.

13.3.3 The user shall be able to see his/her subscription.

13.3.4 The user shall be able to see and change his/her picture.

13.4 The system shall provide a card for displaying wallet information.

13.4.1 The user shall be able to see his/her coins.

13.4.2 The system shall provide charge additional services button.

13.4.3 The system shall provide a purchasing transactions button.

13.5 The user shall be able to navigate between application screens by bottom bar.

14. The user shall be able to access the setting page.

14.1 The user shall be able to go back to the previous page.
14.2 The user shall be able to change the application language.
14.3 The user shall be able to change his/her country.
14.4 The user shall be able to change his/her account settings.
14.5 The user shall be able to change his/her account password.
14.6 The user shall be able to login by face id.
14.7 The user shall be able to see all banned users.
14.8 The user shall be able to sign out.

15. The user shall be able to access the country page.

15.1 The user shall be able to go back to the previous page.
15.2 The user shall be able to search on country.
15.3 The user shall be able to see all the country.

16. The user shall be able to access the account setting page.

16.1 The user shall be able to go back to the previous page.
16.2 The user shall be able to delete his/her page.

17. The user shall be able to access the account password page.

17.1 The user shall be able to go back to the previous page.
17.2 The system shall be providing a field for the current password.
17.3 The system shall be providing a field for the new password.
17.4 The system shall be providing a field for confirmation of the new password.
17.5 The system shall be providing a button for saving the information.

18. The user shall be able to access the banded user page.

18.1 The user shall be able to see all banded users.
18.2 The user shall be able to unbanded users.

19. The user shall be able to access the subscription page.

19.1 The user shall be able to go back to the previous page.
19.2 The user shall be able to see all subscription offers.
19.3 The user shall be able to see a description for each subscription.
19.4 The user shall be able to add the subscription to the cart.
19.5 The user shall be able to navigate between application screens by bottom bar.

20. The user shall be able to access the messages page.

20.1 The user shall be able to search in messages.

20.2 The user shall be able to see his/her navigation.

20.3 The user shall be able to see his/her favorite messages.

20.4 The system shall be providing a card for messages.

20.4.1 The user shall be able to see the sender’s picture.

20.4.2 The user shall be able to see the sender's name.

20.4.3 The user shall be able to see the last message.

20.4.4 The user shall be able to see the time of the last message.

20.5 The user shall be able to navigate between application screens by bottom bar.

21. The user shall be able to access the messaging the person page.

21.1 The user shall be able to go back to the previous page.

21.2 The user shall be able to see the status of the sender.

21.3 The user shall be able to join audio call.

21.4 The user shall be able to open setting chat.

21.4.1 The user shall be able to band the user.

21.4.2 The user shall be able to delete the chat.

21.4.3 The user shall be able to add to favorite.

21.5 The user shall be able to record sound.

21.6 The user shall be able to capture pictures.

21.7 The user shall be able to send a message.

21.8 The user shall be able to send a document.

22. The user shall be able to access to providing services.

22.1 The user shall be able to go back to the previous page.

22.2 The user shall be able to select the service section.

22.3 The user shall be able to select the service sub section.

22.4 The user shall be able to add a picture for the service.

22.5 The user shall be able to add a reel for the service.

22.6 The user shall be able to add the service city.

22.7 The user shall be able to add the service location.

22.8 The user shall be able to add the service free time.

22.9 The user shall be able to add the service name.

22.10 The user shall be able to add the service description.

22.11 The user shall be able to save.

23. The user shall be able to access the publishing service.

23.1 The user shall be able to go back to the previous page.

23.2 The user shall be able to see all service information.

23.3 The user shall be able to see all communications tools details.

23.4 The user shall be able to publish the service.

Non-Functional Requirements :

- Security: This is crucial for protecting user data like passwords, payment information, and personal details.

- Performance: The app should be responsive and load quickly for a smooth user experience.

- Scalability: The app should be able to handle a growing user base without compromising performance.

- Reliability: The app should be available and functional most of the time.

- Maintainability: The codebase should be well-structured and easy to understand for future modifications and bug fixes.

2.4 Use Case Diagram with Use Cases descriptions

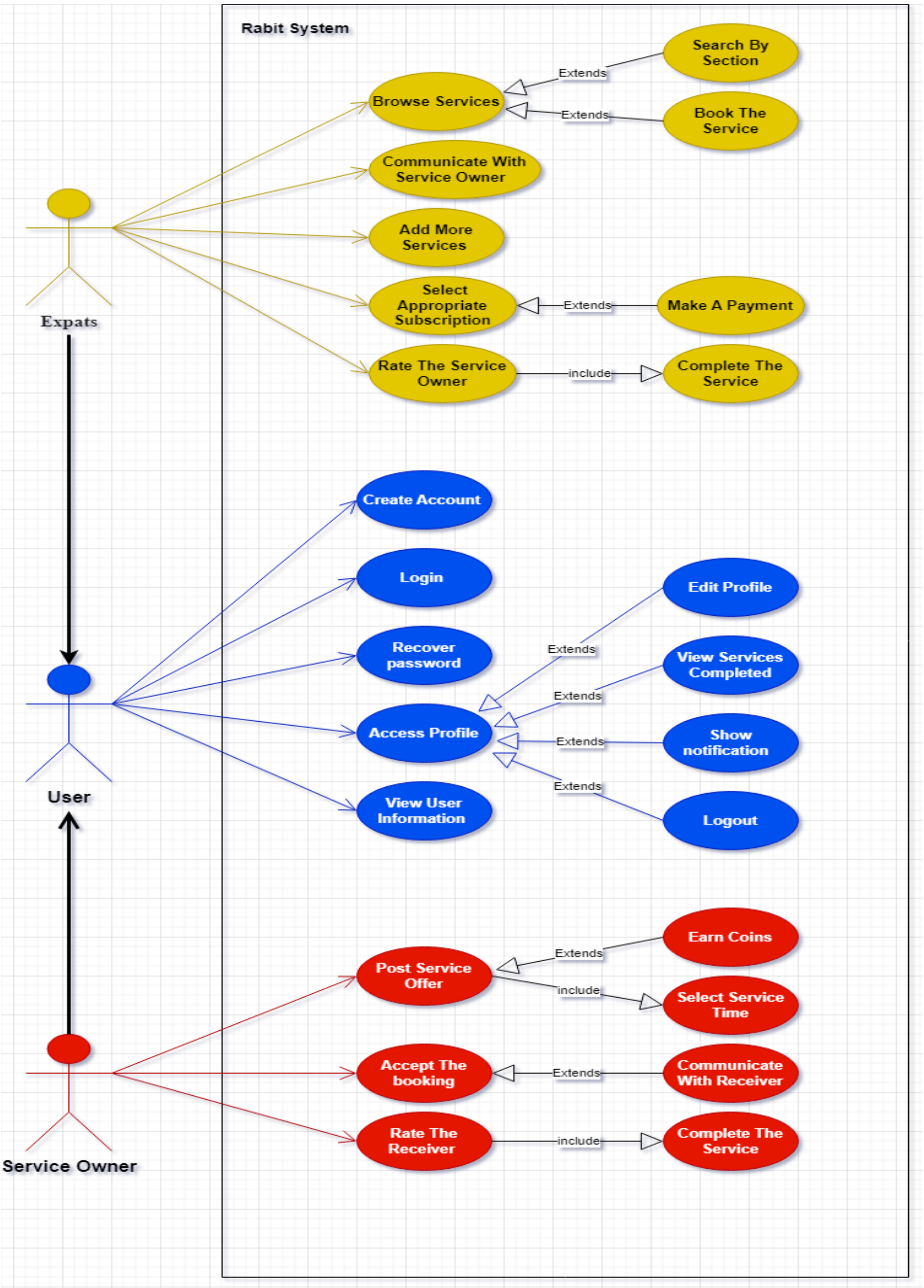


Figure 3Use Case

Table 1User Registration

ID	UC1
Name	User Registration
Description	This use case describes the process of a user (expat or service owner) creating an account in the system.
Primary Actor	User (Expats, Service Owner)
Pre-conditions	User does not have an existing account in the system.
Post-conditions	User successfully creates an account and can log in to the system.
Normal Flow (Happy path)	1. User accesses the registration page. 2. User fills out the required registration information (e.g., email, password, personal details). 3. System validates the information and creates a new user account.
Alternative Flow	❖ If credentials are invalid: • System displays an error message. • User can attempt to login again or reset password.
Include	None
Extend	None
Exceptions	• User tries to register with an existing email address. • Invalid or incomplete registration information provided.

Table 2User Login

ID	UC2
Name	User Login
Description	This use case involves a user (expat or service owner) logging into the system.
Primary Actor	User (Expats, Service Owner)
Pre-conditions	User has a registered account in the system.
Post-conditions	User successfully logs into the system and gains access to their account.
Normal Flow (Happy path)	1. User accesses the login page. 2. User enters their credentials (email and password). 3. System verifies the credentials and grants access to the user.
Alternative Flow	❖ If credentials are invalid: • System displays an error message. • User can attempt to login again or reset password.
Include	None
Extend	None
Exceptions	• User enters incorrect login credentials.

Table 3Recover Password

ID	UC3
Name	Recover Password
Description	This use case allows a user (expat or service owner) to recover their forgotten password.
Primary Actor	User (Expats, Service Owner)
Pre-conditions	User has forgotten their password.
Post-conditions	User successfully resets their password and can log in again.
Normal Flow (Happy path)	1. User accesses the password recovery page. 2. User enters their registered email address. 3. System sends a password reset link to the user's email. 4. User follows the link to reset their password.
Alternative Flow	None
Include	None
Extend	None
Exceptions	• User enters an invalid email address.

Table 4Access profile.

ID	UC4
Name	Access profile.
Description	This use case involves a user (expat or service owner) Accessing into profile page.
Primary Actor	User (Expats, Service Owner)
Pre-conditions	User is logged into the system.
Post-conditions	User successfully view their profile information.
Normal Flow (Happy path)	None.
Alternative Flow	None.
Include	None
Extend	Edit Profile (UC5) View Service Completed (UC6) Show Notification (UC7) Logout (UC8)
Exceptions	None

Table 5Edit Profile

ID	UC5
Name	Edit Profile
Description	This use case allows a user (expat or service owner) to edit their profile information.
Primary Actor	User (Expats, Service Owner)
Pre-conditions	User is logged into the system.
Post-conditions	User successfully updates their profile information.
Normal Flow (Happy path)	1. User navigates to the profile editing page. 2. User modifies the desired profile details. 3. User saves the changes.
Alternative Flow	None.
Include	None
Extend	None
Exceptions	• User encounters errors while saving changes.

Table 6View Service Completed

ID	UC6
Name	View Service Completed
Description	This use case allows a user (expat, Service Owner) to view services they have completed.
Primary Actor	User (Expats, Service Owner)
Pre-conditions	User is logged into the system.
Post-conditions	User successfully views the list of completed services.
Normal Flow (Happy path)	1. Expat navigates to the completed services section. 2. System displays a list of services previously completed by the expat.
Alternative Flow	None.
Include	None
Extend	None
Exceptions	None.

Table 7Show Notification

ID	UC7
Name	Show Notification
Description	This use case involves displaying notifications to the user (expat or service owner).
Primary Actor	User (Expats, Service Owner)
Pre-conditions	User is logged into the system.
Post-conditions	User views the notifications.
Normal Flow (Happy path)	1. User accesses the notification section. 2. System presents notifications related to the user's account activities.
Alternative Flow	None.
Include	None
Extend	None
Exceptions	None.

Table 8Logout

ID	UC8
Name	Logout
Description	This use case involves a user (expat or service owner) logging out of the system.
Primary Actor	User (Expats, Service Owner)
Pre-conditions	User is currently logged into the system.
Post-conditions	User successfully logs out of the system.
Normal Flow (Happy path)	1. User accesses the logout functionality. 2. System terminates the user's session and logs them out.
Alternative Flow	None.
Include	None.
Extend	None.
Exceptions	None.

Table 9View User Information.

ID	UC9
Name	View User Information.
Description	This use case allows a user (expat or service owner) to view their own profile information.
Primary Actor	User (Expats, Service Owner)
Pre-conditions	User is logged into the system.
Post-conditions	User successfully views their profile information.
Normal Flow (Happy path)	1. User navigates to their profile page. 2. User's profile details (e.g., name, contact information) are displayed.
Alternative Flow	None.
Include	None
Extend	None
Exceptions	None.

Table 10Browse Services

ID	UC10
Name	Browse Services
Description	This use case involves a user (expat) browsing available services in the system.
Primary Actor	User (Expats)
Pre-conditions	User is logged into the system.
Post-conditions	User successfully views the list of available services.
Normal Flow (Happy path)	1. Expat navigates to the services browsing section. 2. System displays a list of services categorized by sections.
Alternative Flow	None.
Include	None
Extend	Search by Section (UC11) Book the Service (UC12)
Exceptions	None.

Table 11Search by Section

ID	UC11
Name	Search by Section
Description	This use case allows a user (expat) to search for specific services within a section.
Primary Actor	User (Expats)
Pre-conditions	User is browsing services within a section.
Post-conditions	User views search results based on the section.
Normal Flow (Happy path)	1. Expat enters a search query within a specific service section. 2. System retrieves and displays relevant search results.
Alternative Flow	None.
Include	None
Extend	None.
Exceptions	None.

Table 12Book the Service

ID	UC12
Name	Book the Service
Description	This use case involves a user (expat) booking a selected service.
Primary Actor	User (Expats)
Pre-conditions	User has selected a service to book.
Post-conditions	User successfully completes the booking process.
Normal Flow (Happy path)	1. Expat selects a service from the browsing list. 2. Expat chooses booking options (e.g., date, time). 3. Expat confirms the booking.
Alternative Flow	None.
Include	None
Extend	None.
Exceptions	None.

Table 13Communicate with Service Owner

ID	UC13
Name	Communicate with Service Owner
Description	This use case involves a user (expat) communicating with the service owner for inquiries or arrangements.
Primary Actor	User (Expats)
Pre-conditions	User has booked a service.
Post-conditions	User successfully communicates with the service owner.
Normal Flow (Happy path)	1. Expat accesses the messaging feature related to the booked service. 2. Expat sends messages to the service owner.
Alternative Flow	None.
Include	None
Extend	None.
Exceptions	None.

Table 14Add More Services

ID	UC14
Name	Add More Services
Description	This use case involves a user (expat) adding more services to their existing bookings or requests.
Primary Actor	User (Expats)
Pre-conditions	User has one or more active bookings or service requests.
Post-conditions	User successfully adds additional services to their bookings.
Normal Flow (Happy path)	1. Expat accesses the service browsing section. 2. Expat selects additional services to add to their existing bookings.
Alternative Flow	None.
Include	None
Extend	None.
Exceptions	None.

Table 15Select Appropriate Subscription

ID	UC15
Name	Select Appropriate Subscription
Description	This use case involves a user (expat) selecting or updating their subscription plan.
Primary Actor	User (Expats)
Pre-conditions	User is logged into the system.
Post-conditions	User successfully updates their subscription plan.
Normal Flow (Happy path)	1. Expat accesses the subscription management section. 2. Expat views available subscription plans. 3. Expat selects or modifies their subscription plan.
Alternative Flow	None.
Include	None
Extend	Make a Payment (UC16)
Exceptions	None.

Table 16Make a Payment

ID	UC16
Name	Make a Payment
Description	This use case involves a user (expat) making a payment for their selected subscription.
Primary Actor	User (Expats)
Pre-conditions	User has selected a subscription plan.
Post-conditions	User successfully completes the payment for the subscription.
Normal Flow (Happy path)	1. Expat selects the payment option. 2. Expat enters payment details and confirms the transaction.
Alternative Flow	None.
Include	None
Extend	None.
Exceptions	None.

Table 17Rate the Service Owner

ID	UC17
Name	Rate the Service Owner
Description	This use case involves a user (expat) rating the service owner after receiving a service.
Primary Actor	User (Expats)
Pre-conditions	User has received and completed a service.
Post-conditions	User successfully rates the service owner.
Normal Flow (Happy path)	1. Expat accesses the service completion section. 2. Expat rates the service owner based on their experience.
Alternative Flow	None.
Include	None
Extend	Complete the Service (UC18)
Exceptions	None.

Table 18Complete the Service

ID	UC18
Name	Complete the Service
Description	This use case involves the completion of a service transaction between the user (expat) and the service owner.
Primary Actor	User (Expats)
Pre-conditions	Service has been provided and received.
Post-conditions	Service transaction is marked as completed.
Normal Flow (Happy path)	1. Expat confirms the completion of the service. 2. System updates the service status accordingly.
Alternative Flow	None.
Include	None
Extend	None.
Exceptions	None.

Table 19Post Service Offer

ID	UC19
Name	Post Service Offer
Description	This use case involves a service owner posting a new service offer in the system.
Primary Actor	User (Service Owner)
Pre-conditions	Service owner is logged into the system.
Post-conditions	New service offer is listed in the system.
Normal Flow (Happy path)	1. Service owner accesses the service posting section. 2. Service owner enters details of the new service offer. 3. Service offer is submitted and listed for users to book.
Alternative Flow	None.
Include	Select Service Time (UC20).
Extend	Earn Coins (UC21).
Exceptions	None.

Table 20Select Service Time

ID	UC20
Name	Select Service Time
Description	This use case involves a service owner selecting the availability or time slots for their service.
Primary Actor	User (Service Owner)
Pre-conditions	Service owner is posting a new service offer.
Post-conditions	Service availability is set for users to book.
Normal Flow (Happy path)	1. Service owner specifies available time slots or scheduling details for the service.
Alternative Flow	None.
Include	None
Extend	None.
Exceptions	None.

Table 21Earn Coins

ID	UC21
Name	Earn Coins
Description	This use case involves a service owner earning coins upon posting a new service offer.
Primary Actor	User (Service Owner)
Pre-conditions	Service owner has posted a new service offer.
Post-conditions	Service owner earns coins based on the service offer.
Normal Flow (Happy path)	1. System rewards the service owner with coins upon successful service posting.
Alternative Flow	None.
Include	None
Extend	None.
Exceptions	None.

Table 22Accept the Booking

ID	UC22
Name	Accept the Booking
Description	This use case involves a service owner accepting a booking request from a user (expat).
Primary Actor	User (Service Owner)
Pre-conditions	Service owner receives a booking request.
Post-conditions	Booking request is confirmed and scheduled.
Normal Flow (Happy path)	1. Service owner reviews the booking request. 2. Service owner accepts the booking.
Alternative Flow	None.
Include	None
Extend	Communicate with Receiver (UC23)
Exceptions	None.

Table 23Communicate with Receiver

ID	UC23
Name	Communicate with Receiver
Description	This use case involves a service owner communicating with the receiver (expat) of a booked service.
Primary Actor	User (Service Owner)
Pre-conditions	Service owner has accepted a booking request.
Post-conditions	Service owner communicates with the receiver regarding service details.
Normal Flow (Happy path)	1. Service owner accesses the messaging feature related to the booked service. 2. Service owner exchanges messages with the receiver to coordinate service arrangements.
Alternative Flow	None.
Include	None.
Extend	None.
Exceptions	None.

Table 24Rate the Receiver

ID	UC24
Name	Rate the Receiver
Description	This use case involves a service owner rating the receiver (expat) after completing a service.
Primary Actor	User (Service Owner)
Pre-conditions	Service transaction with the receiver has been completed.
Post-conditions	Service owner successfully rates the receiver.
Normal Flow (Happy path)	1. Service owner accesses the service completion section. 2. Service owner rates the receiver based on the service experience.
Alternative Flow	None.
Include	None
Extend	Complete the Service (UC18)
Exceptions	None.

Chapter 3: Design

3.1 Conceptual Class Diagram

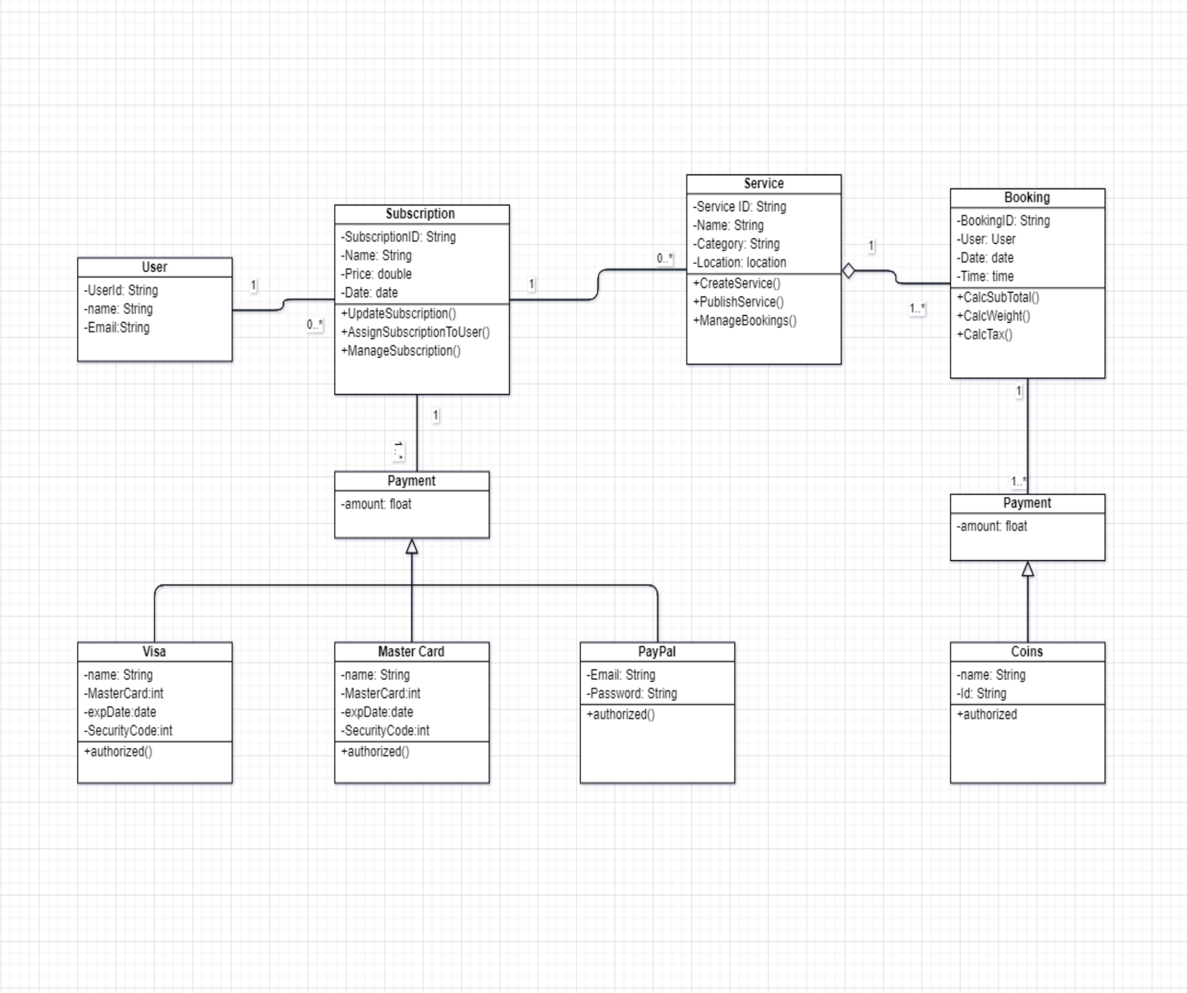


Figure 4Conceptual Class Diagram

References

[1] Handy, "Handy," Handy, 21 4 2024. [Online]. Available: https://www.handy.com/.

[2] Thumbtack, "Thumbtack," Thumbtack, 21 4 2024. [Online]. Available: https://www.thumbtack.com/.

[3] Expats, "Expats," Expats, 23 4 2024. [Online]. Available: https://journals.aom.org/doi/abs/10.5465/AME.2005.15841966.

Appendix

Any additional information or data that supports or extends the main document or report.