

# NANYANG TECHNOLOGICAL UNIVERSITY

---

## SINGAPORE

School of Computer Science & Engineering

SC2006 Software Engineering

**Project Title: Servify**



*Connect. Serve. Sustain*

SCSY/Group 1 (Index 10307)

### **Team Members:**

Aidan Ling Zhen Yang (U2220996F)

Angie Wong Mei Chi (U2121896E)

Aryan Garg (U2220598L)

Chan Hin Wai Howell (U2221335D)

Cheng Lin (U2222079E)

## Contents

1. Product Introduction .....	3
1.1. Background.....	3
1.2. Mission Statement .....	3
1.3. Scope .....	4
1.4. UI Mockups.....	5
2. Functional Requirements .....	11
2.1. Home Page.....	11
2.2. Login Page.....	11
2.3. Service Requests Booking .....	12
2.4. Service Confirmation Notification.....	13
2.5. Service Provider Location .....	13
2.6. Service Completion.....	13
3. Non-Functional Requirements .....	14
3.1. Usability Requirements .....	14
3.2. Performance Requirements .....	14
3.3. Security Requirements .....	15
3.4. Reliability Requirements .....	15
3.5. Supportability Requirements.....	15
4. Use Case Model.....	16
4.1. Use Case Diagram.....	16
4.2. Use Case Descriptions .....	17
5. Data Dictionary.....	35
6. Diagrams.....	36
6.1. Entity Class Diagram .....	36
6.2. Conceptual Diagram .....	37
6.3. Sequence Diagrams .....	38
6.4. Dialog Map .....	40
6.5 System Architecture .....	41

## 1. Product Introduction

### 1.1. Background

---

The Smart Nation Initiative is a nation-wide project intended to transform Singapore into an intelligent and interconnected city-state. It seeks to integrate digital technology into every aspect of Singaporean life, including public services, transportation, healthcare, and more. Several flagship projects have been launched under this initiative, ranging from Smart Mobility solutions to Digital Government services.

Despite the rapid advancements in various sectors, there remains a considerable gap when it comes to the digitalisation of local services – specifically for facility services such as repair, maintenance, and cleaning works. Many service providers like plumbers, electricians, and handymen still rely on traditional methods of advertising and client acquisition, such as word-of-mouth or printed directories. This leads to inefficiencies for both service providers and consumers. Service providers miss out on a centralised platform to market their skills and manage bookings, while consumers have to go through a cumbersome process of searching, comparing, and validating available service providers.

In alignment with Singapore's vision of fostering a technologically advanced and interconnected society, we aim to develop a web platform to revolutionise the way residents secure urgent specialised services.

### 1.2. Mission Statement

---

Leveraging on a curated network of service providers, we aim to design a web platform to eliminate the friction traditionally associated with locating and engaging professional assistance. By centralising a diverse range of services within a single, user-friendly digital ecosystem, Servify aims to streamline this search process.

Overall, our mission is to provide an efficient, one-stop solution for those in urgent need of professional services, whilst elevating the standard for digital marketplaces in Singapore.

### 1.3. Scope

---

Our web platform is designed to cater to Singaporeans of all ages who require immediate professional services, encompassing everything from repair and maintenance to specialised assistance.

Users will encompass Singapore-based users seeking professional services, as well as freelance service providers exploring alternative avenues to broaden their customer reach.

## 1.4. UI Mockups

The image shows the home screen of the Servify website. At the top, there is a green header bar with the Servify logo on the left and a 'LOGIN' button on the right. Below the header, the main content area has a light blue background. On the left, there is a white rounded rectangle containing the heading 'WHAT WE PROVIDE' and several paragraphs of text. To the right of this section is a large image of a bearded handyman in a blue uniform and cap, standing in front of a wall covered with various blue line drawings of tools like hammers, wrenches, and ladders. Below this section, there is a heading 'WHY CHOOSE SERVIFY' followed by four teal-colored boxes, each containing a title and a small image: 'Effortless Booking' (image of a person booking), 'Transparent Pricing' (image of a price list), 'Trusted Professionals' (image of two professionals), and 'Doorstep Delivery' (image of a delivery). At the bottom, there is a section titled 'TESTIMONIALS' with three testimonial cards. Each card includes a name, a short quote, and a five-star rating icon.

**Servify X**

**LOGIN**

## WHAT WE PROVIDE

Welcome to Servify – Your Hassle-Free Solution for Home Repairs and Improvements!

At EzFix, we've transformed the way you handle household fixes. Skip the endless searches and waiting. With us, skilled handyman services are just a doorstep away.

No more scheduling headaches or DIY disasters. We offer effortless solutions that save you time, effort, and offer peace of mind.

## WHY CHOOSE SERVIFY

- Effortless Booking**
- Transparent Pricing**
- Trusted Professionals**
- Doorstep Delivery**

## TESTIMONIALS

**Mr. Dennys**  
"I recently used Servify to find a handyman for some home repairs, and I was pleasantly surprised by the experience. The website was easy to navigate, and I quickly found a handyman who was available to help me. The handyman arrived on time, was professional, and did a fantastic job. I'll definitely use this service again."  
★★★★★

**Ms Jenny**  
"I've used Servify multiple times, and I've been consistently impressed. The website is super convenient, and the handymen they've connected me with have all been top-notch. They were punctual, efficient, and completed my projects to my satisfaction. It's become my go-to platform for all my home repair needs."  
★★★★★

**Ms. Wong**  
"Servify is a decent platform for finding handymen, but I had a mixed experience. The website was user-friendly, and I found a handyman quickly. However, the handyman they sent was not as skilled as I had hoped. The job took longer than expected, and I had to supervise closely to ensure it was done right. It's hit or miss with this service."  
★★★★☆

Figure 1: Home Screen

Figure 1 showcases the landing page of the Servify website, which features a clean, intuitive, and user-friendly interface. This design approach will be uniformly applied across all subsequent UI mock-ups. On the landing page, users can easily gain access to a concise overview of what Servify offers, as well as testimonials and reviews of the platform. Additionally, the option to login to your user account is readily available through the "Login" button at the top right-hand corner. By clicking on this button, users will be directed to the "Login" page, as depicted in Figure 2.

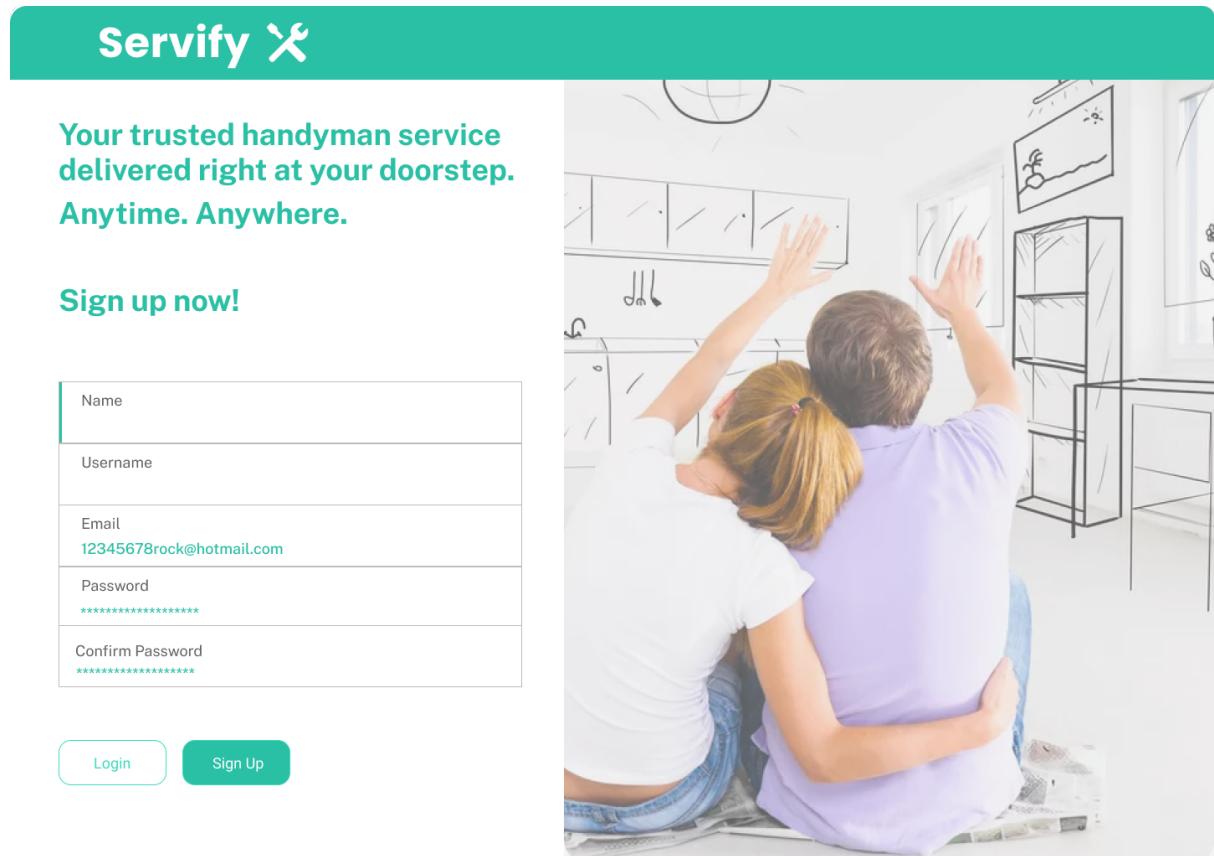


Figure 2.1: Registration Page

Figure 2.1 illustrates the registration page of the web platform. This page provides fields for users to enter their personal details necessary for account creation. Once registered, users can proceed to sign in using the “Login” button, detailed further in Figure 2.2.

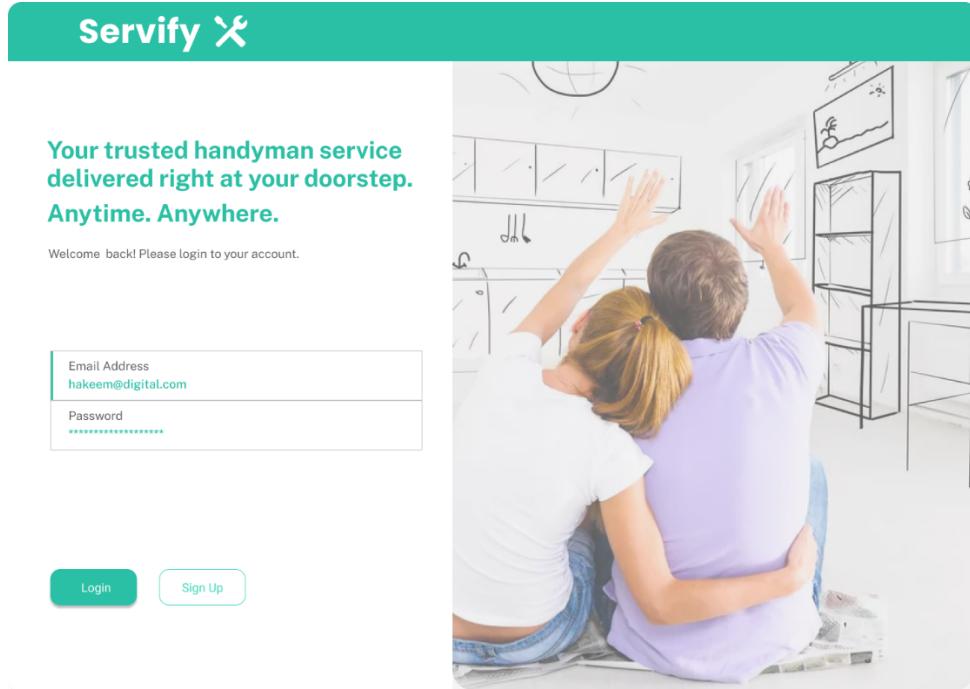


Figure 2.2: Login Page

Figure 2.2 depicts the login page of the web platform. On this page, users are prompted to enter their credentials, specifically their username and password, to gain access. Upon successful validation, they will be seamlessly redirected to the service requests booking page, as highlighted in Figure 3.

SELECT SERVICE NEEDED			
CLEANING SERVICES Lorem ipsum is simply dummy text of the printing and typesetting industry.	FLOORING INSTALLATION Lorem ipsum is simply dummy text of the printing and typesetting industry.	FIXTURE INSTALLATION Lorem ipsum is simply dummy text of the printing and typesetting industry.	WALL CONSTRUCTION Lorem ipsum is simply dummy text of the printing and typesetting industry.
SITE SURVEYS Lorem ipsum is simply dummy text of the printing and typesetting industry.	PROFESSIONAL PAINTING Lorem ipsum is simply dummy text of the printing and typesetting industry.	ELECTRICAL WORK Lorem ipsum is simply dummy text of the printing and typesetting industry.	INTERIOR BUILD-OUTS Lorem ipsum is simply dummy text of the printing and typesetting industry.
MILL WORK Lorem ipsum is simply dummy text of the printing and typesetting industry.	LIGHTING INSTALLATION Lorem ipsum is simply dummy text of the printing and typesetting industry.	ORographics Lorem ipsum is simply dummy text of the printing and typesetting industry.	COMPLETE RENOVATIONS Lorem ipsum is simply dummy text of the printing and typesetting industry.
SHELVING Lorem ipsum is simply dummy text of the printing and typesetting industry.	FINISH CARPENTRY Lorem ipsum is simply dummy text of the printing and typesetting industry.	PLUMBING ISSUES Lorem ipsum is simply dummy text of the printing and typesetting industry.	UPGRADE DECORS Lorem ipsum is simply dummy text of the printing and typesetting industry.

[NEXT](#)

30%

ATTACH PHOTO(S)  
PLEASE FURNISH CLEAR PHOTOS FOR A QUICK HANDYMAN SELECTION  
[UPLOAD](#)

WHEN DO YOU NEED OUR SERVICE?  
8 AM TO 11 AM  
12PM TO 3PM  
4PM TO 7PM

I NEED HELP

50%

Figure 3: Service Request Screen

Figure 3 outlines the initial steps for initiating a service booking. Users can start by selecting one service they require (Left Panel). To further clarify the issue, they have the option to upload a photograph illustrating specific problems that need attention (Right Panel) to visually communicate the specific issues that require servicing (Right Panel). Subsequently, users can specify the time slot that they would like the service to be performed (Right Panel).

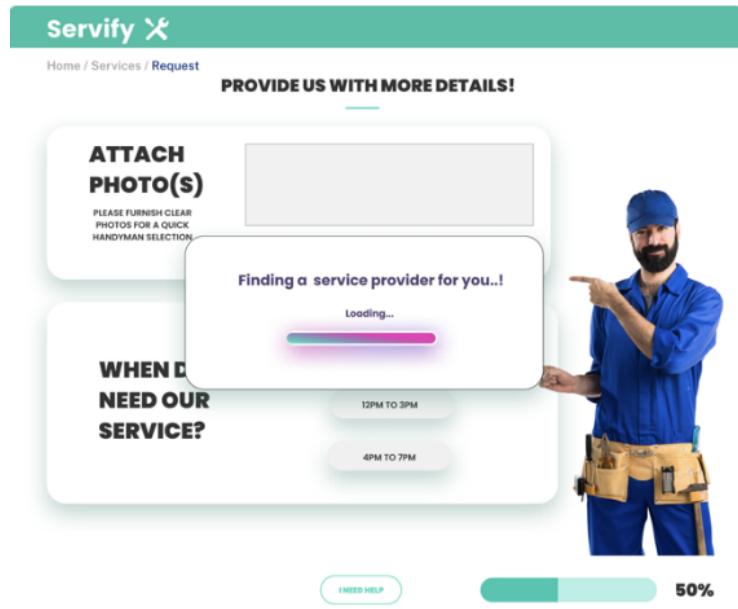


Figure 4: Booking Screen

Figure 4 shows an active loading screen that appears while the system is in the process of locating nearby service providers capable of meeting the user's specific request.

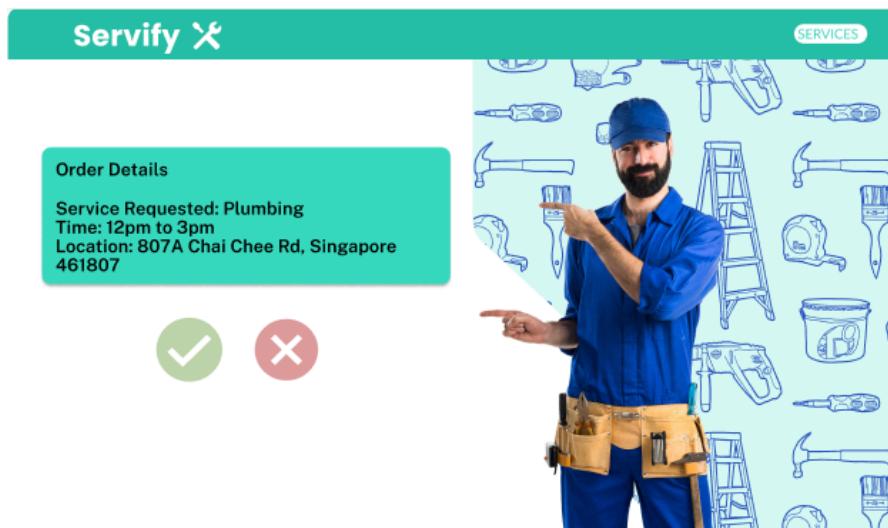
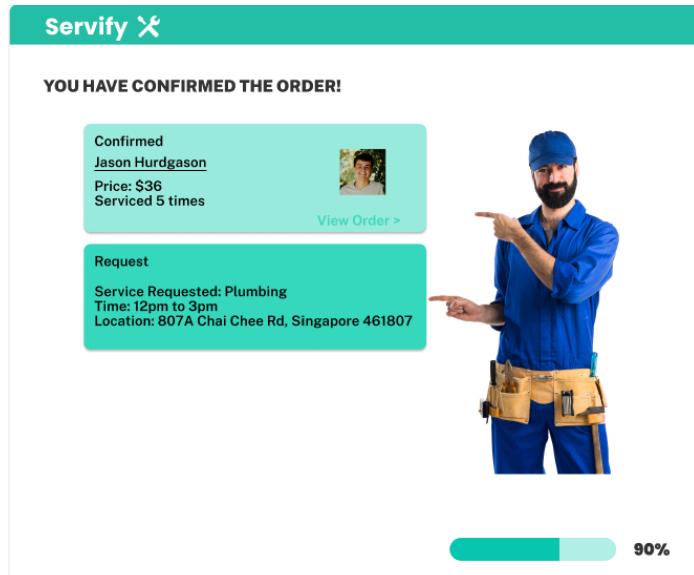


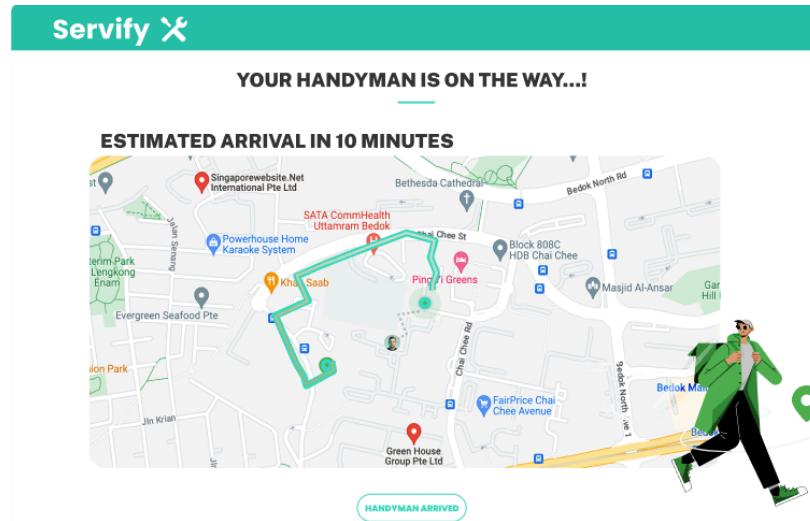
Figure 5: Service provider receives request

Figure 5 illustrates a service request notification that is sent to nearby service providers, presenting them with the details of the order. Providers have the option to either accept or decline the service request.



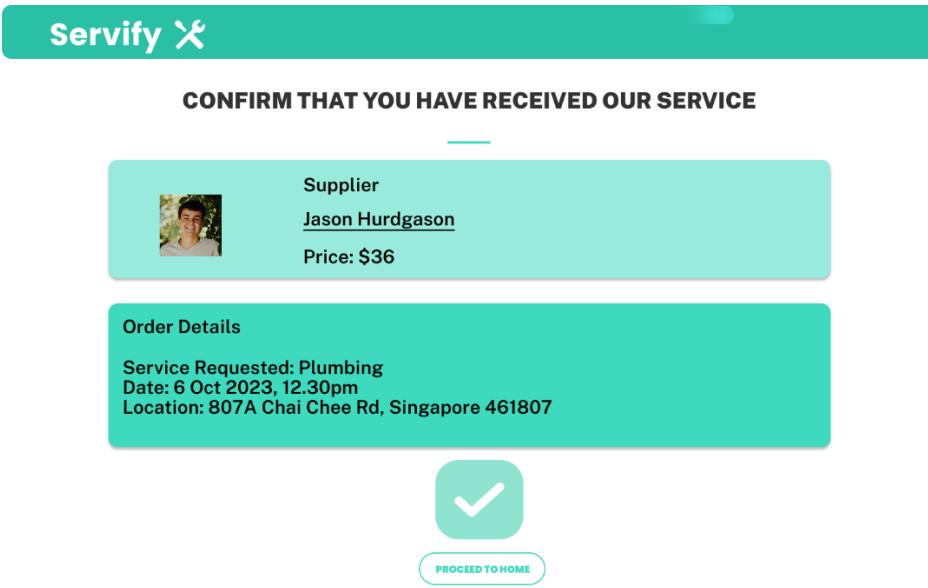
*Figure 6: Confirmation notification after the service provider accepts the request*

Figure 6 displays the confirmation notification that appears once a successful match with a service provider has been made. This notification offers a thorough breakdown of the service request, detailing elements such as the type of service, scheduled date, location, and fixed cost. It also includes information about the selected service provider, such as their name and previous experience with the requested service.



*Figure 7: Location Screen*

Figure 7 showcases the feature that allows users to track the service provider's location in real-time, ensuring that the service begins as scheduled. This screen includes a dynamic counter displaying the estimated time of arrival, which continuously updates to offer the most up-to-date information. Once the service provider reaches the user's location, the user will have to verify the service provider's arrival by pressing the button labeled "Handyman Arrived".



*Figure 8: Service Completion Confirmation Screen*

Figure 8 outlines the process by which a user can confirm the completion of a service. This is achieved by tapping on a checkbox after verifying the accuracy of the order details and the service provider's arrival.

## 2. Functional Requirements

### 2.1. Home Page

- 2.1.1 The **system** must **display** a static home page, containing details about the Servify web platform. The details should include:
  - 2.1.1.1 A summary of the services provided by the Servify web platform.
  - 2.1.1.2 User reviews and testimonials about the Servify web platform.
- 2.1.2 The **system** must **display** a button that redirects the user to the login page.

### 2.2. Login Page

- 2.2.1 The **system** must **display** a login page, containing two options to either register a new account or login to an existing account.
  - 2.2.1.1 The **registration** option must **display** fields that request for the user's personal information. This information includes:
    - 2.2.1.1.1 Full Name
    - 2.2.1.1.2 Username
    - 2.2.1.1.3 Email
    - 2.2.1.1.4 Password
  - 2.2.1.2 The **registration** option must **display** a field for users to re-enter their password to ensure its correctness.
  - 2.2.1.3 The **registration** option must **display** a button that will submit the user's personal information for registering a new user account onto the database.
  - 2.2.1.4 The **login** option must **display** fields that request for the user's credentials. This information includes:
    - 2.2.1.4.1 Username
    - 2.2.1.4.2 Password
  - 2.2.1.5 The **login** option must **display** a button that will submit the user's credentials to be validated against the stored data in the database.
- 2.2.2 The **login** option must **redirect** the user to the service requests booking page upon successful validation of the user account.

## 2.3. Service Requests Booking

- 2.3.1 The **system** must **verify** if the user has granted permission to access their Global Positioning System (GPS) data upon launching the application.
  - 2.3.1.1 If the user has not granted GPS permission, the **system** must **prompt** the user to grant access to their GPS data.
  - 2.3.1.2 If the user denies the GPS permission request, the **system** must refrain from requesting GPS access in subsequent interactions.
  - 2.3.1.3 In the absence of GPS permission, the **system** must **display** a default map view, zoomed-out to show Singapore, instead of the user's current location.
- 2.3.2 The **system** shall present an option for the **user** to **select** the type of service they require from a predefined menu of services.
- 2.3.3 The **system** must **display** an interface element for the user to attach a picture that is relevant to their service request.
  - 2.3.3.1 Once a picture is attached, the **system** shall **render** it on the web page for user confirmation, before proceeding with the service request.
- 2.3.4 The **system** must **provide** input fields for the user to specify the desired time slots.
  - 2.3.4.1 The time slots will be set at intervals of 3, 6 and 9 hours from the current time.
- 2.3.5 The **user** must be able to **submit** their service request through a dedicated "I Need Help" button.
- 2.3.6 The **system** must **process** the submitted service request in accordance with predefined rules.
  - 2.3.6.1 The **system** must **query** the database for a list of service providers for the selected service of the user.
  - 2.3.6.2 The **database** must **return** a list of the service providers, sorted by the location nearest to the user's current location.
- 2.3.7 After processing, the **system** must **disseminate** the service request to available service providers within the relevant service category and geographical area.
- 2.3.8 The **system** must **display** a loading page to the user as it searches for nearby service providers that provide the service requested by the user.
- 2.3.9 The **system** must **notify** the user once a single service provider has accepted the service request.
- 2.3.10 The **system** must **notify** the user if it was unable to find a nearby service provider and prompt the user to try again later.

## 2.4. Service Confirmation Notification

2.4.1 Upon acceptance of a service request by a service provider, the **system** must **generate** a confirmation notification targeted at the respective user.

2.4.1.1 The confirmation notification must contain the details of the service provider. These details include:

2.4.1.1.1 Name of the service provider.

2.4.1.1.2 Face of the service provider.

2.4.1.1.3 Number of times the service has been previously rendered.

2.4.1.1.4 Cost of the service provided.

## 2.5. Service Provider Location

2.5.1 The **system** must **retrieve** the geographical location of the service provider's device using the **Google Maps Geolocation API**.

2.5.1.1 The **system** must request location permissions from the service provider's device.

2.5.1.2 The **system** must handle cases where location permissions are denied.

2.5.1.3 The **system** must continuously update the location of the service provider in real-time.

2.5.2 The **system** must **display** a real-time map view, showing the service provider's route to the user's location, using **Google Maps Geolocation API**.

2.5.2.1 The map must be centered on the service provider's current location.

2.5.2.2 The **system** must **calculate** and **plot** the most efficient route between the service provider and the user's location.

2.5.2.3 The **system** must **update** the route in real-time if the service provider deviates from the planned route.

2.5.2.4 The **system** must **provide** an estimated time of arrival based on the current route.

## 2.6. Service Completion

2.6.1 The **system** must **generate** a confirmation notification checking with the user that the service provider has arrived at their location.

2.6.2 The confirmation notification should include details about the service provider as specified in 2.3.1.1.

### 3. Non-Functional Requirements

#### 3.1. Usability Requirements

- The system must have an **intuitive user interface**.
  - a. The system must allow users to complete tasks **within three or less clicks**.
- The system must search for service providers and return a confirmed request **within 5 minutes or less**.
- The system must allow users to **refine their search** based on service type.
- The system should have a **consistent sequence of actions** for web platform services.
  - a. The system should require a consistent number of steps to complete any service rendered by the Servify web platform.
  - b. Completing a booking should have a progress indicator showing the current step and the total number of steps.
- The system must maintain a **consistent design**.
  - a. A consistent visual layout must be provided in the application.
  - b. A consistent sequence of actions is required for similar situations (i.e. bookings).
  - c. Fonts, colour schemes, and buttons placements should be uniform across every page.

#### 3.2. Performance Requirements

- The system must load service provider details within **2 seconds** of a booking confirmation.
- The system must handle multiple simultaneous booking requests **without latency**.
- The system must ensure **minimal downtime** for maintenance or updates.
- The system must provide real-time location tracking with a delay of **no more than 2 seconds**.

### 3.3. Security Requirements

- The system must **validate files** during the process of uploading pictures for the service request.
  - a. The file size should be smaller than 1 megabyte.
  - b. The file should have an extension of either JPEG, JPG, or PNG only.

### 3.4. Reliability Requirements

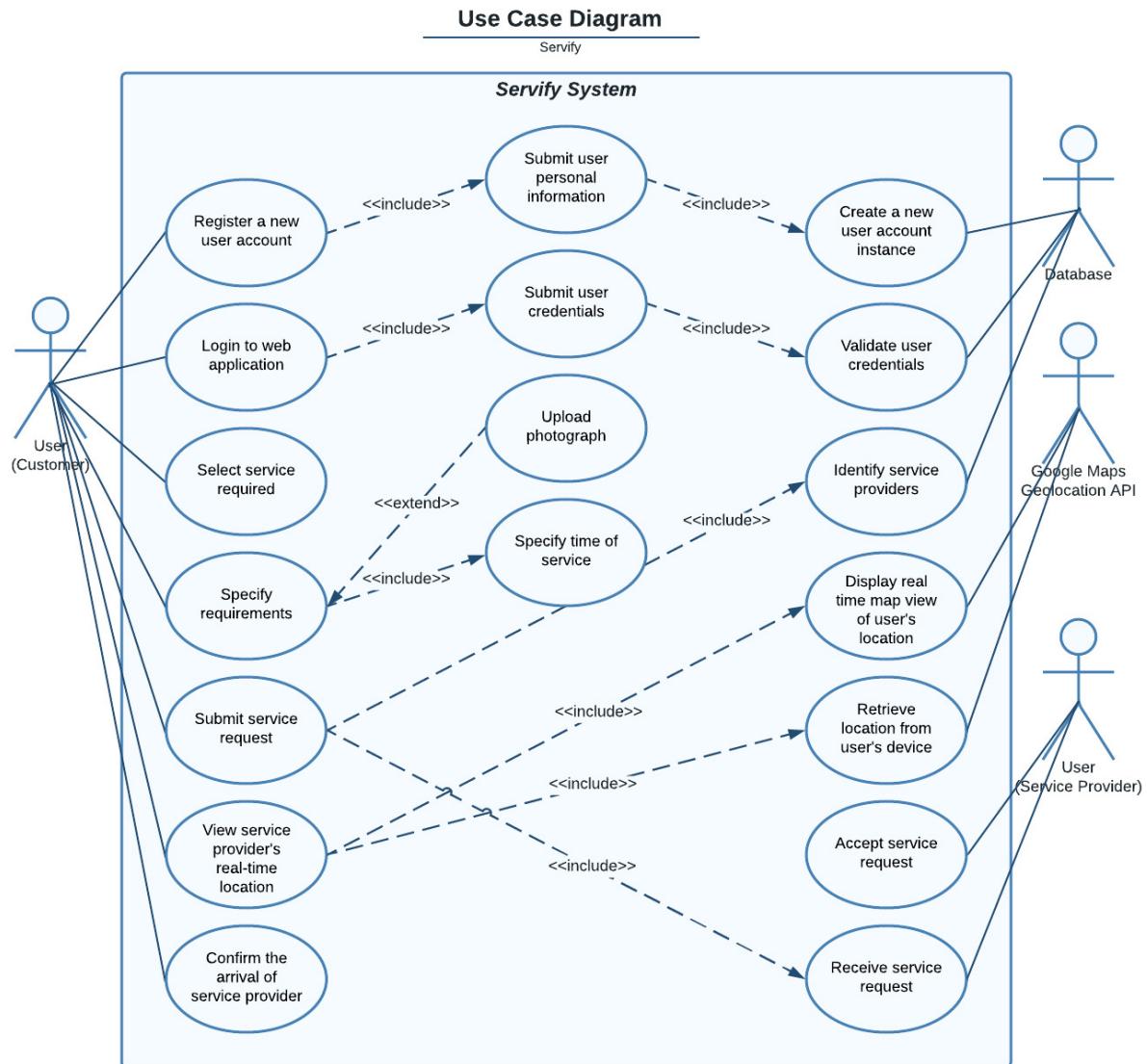
- The system must be capable of **handling** at least **10,000** concurrent users at any given time.
- The system must not have a **latency** that exceeds **2 seconds** when under peak load.
- The system must have a **backup mechanism** to prevent data loss.

### 3.5. Supportability Requirements

- The system must remain **functional** and maintain a consistent look and feel across **different web browser platforms**.
  - a. The system must render the webpages correctly across different platforms and their respective screen sizes.

## 4. Use Case Model

### 4.1. Use Case Diagram



## 4.2. Use Case Descriptions

<b>Use Case ID:</b>	UC001		
<b>Use Case Name:</b>	RegisterNewUserAccount		
<b>Created By:</b>	Aidan Ling	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	25 October 2023	<b>Date Last Updated:</b>	25 October 2023
<b>Actor:</b>	User (Customer)		
<b>Description:</b>	This use case outlines the process where a user may register a new account with the web platform		
<b>Preconditions:</b>	Customer has navigated to the Login page from the Home page.		
<b>Postconditions:</b>	A “Registration Successful!” message will be returned to the user.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>The Customer navigates to the Login page from the Home page.</li> <li>This is done by clicking the “Login” button on the top right-hand corner of the Home page, which will redirect them to the Services page.</li> <li>The Customer selects the registration option.</li> <li>The Customer may fill in their personal information to register a new user account.</li> <li>The Customer clicks the “Sign Up” button to register.</li> </ol>		
<b>Alternative Flows:</b>	<u>AF-S1: Cancellation of the registration process by closing the tab.</u> 1. No new user account is created in the database.		
<b>Exceptions:</b>	<u>EX1: User already exists in the system.</u> 1. The System will return an error message “Email already exists!”.		
<b>Includes:</b>	1. UC001.1: SubmitUserPersonalInformation 2. UC001.2: CreateNewUserAccountInstance		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC001.1		
<b>Use Case Name:</b>	SubmitUserPersonalInformation		
<b>Created By:</b>	Aidan Ling	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	25 October 2023	<b>Date Last Updated:</b>	25 October 2023
<b>Actor:</b>	User (Customer)		
<b>Description:</b>	This use case outlines the process where a user may submit personal information to the database for registration.		
<b>Preconditions:</b>	Customer has navigated to the Login page from the Home page.		
<b>Postconditions:</b>	A "Registration Successful!" message will be returned to the user.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	1. The Customer may fill in their personal information to register a new user account. 2. The Customer clicks the "Sign Up" button to register. 3. A request is sent to the database to store the personal information.		
<b>Alternative Flows:</b>	-		
<b>Exceptions:</b>	-		
<b>Includes:</b>	1. UC001.2: SubmitUserPersonalInformation		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC001.2		
<b>Use Case Name:</b>	CreateNewUserAccountInstance		
<b>Created By:</b>	Aidan Ling	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	25 October 2023	<b>Date Last Updated:</b>	25 October 2023
<b>Actor:</b>			
<b>Description:</b>	Database		
<b>Preconditions:</b>	This use case outlines the process where the database will create a new user account instance based on the personal information received.		
<b>Postconditions:</b>	A request is sent to the database to store user's personal information.		
<b>Priority:</b>	Returns a "Registration Successful!" to the User		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	High		
	1. A request is sent to the database to store the personal information. 2. The database will validate if the personal information already exists within the stored data. 3. If the personal information is new, it will create a new user account instance and store it in the database table.		
<b>Alternative Flows:</b>	-		
<b>Exceptions:</b>	<u>EX1: User already exists in the database.</u> 1. The database will inform the System to return an error message "Email already exists!".		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC002		
<b>Use Case Name:</b>	LoginToWebApplication		
<b>Created By:</b>	Aidan Ling	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	25 October 2023	<b>Date Last Updated:</b>	25 October 2023
<b>Actor:</b> User (Customer)			
<b>Description:</b>	This use case outlines the process where a user may login to an existing account on the web platform.		
<b>Preconditions:</b>	Customer has navigated to the Login page from the Home page.		
<b>Postconditions:</b>	A “Login Successful!” message will be returned to the user, and the user will be redirected to the Services Page		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. The Customer navigates to the Login page from the Home page.</li> <li>2. This is done by clicking the “Login” button on the top right-hand corner of the Home page, which will redirect them to the Login page.</li> <li>3. The Customer selects the login option.</li> <li>4. The Customer may fill in their credentials to log in to an existing user account.</li> <li>5. The Customer clicks the “Login” button to log in.</li> </ol>		
<b>Alternative Flows:</b>	<u>AF-S1: Cancellation of the login process by closing the tab.</u> 1. User will not log in		
<b>Exceptions:</b>	<u>EX1: Username/Password is wrong.</u> 1. The System will return an error message “Invalid Username or Password!”.		
<b>Includes:</b>	1. UC002.1: SubmitUserCredentials 2. UC002.2: ValidateUserCredentials		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC002.1		
<b>Use Case Name:</b>	SubmitUserCredentials		
<b>Created By:</b>	Aidan Ling	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	25 October 2023	<b>Date Last Updated:</b>	25 October 2023
<b>Actor:</b> User (Customer)			
<b>Description:</b>	This use case outlines the process where a user may submit user credentials to be validated against the stored data in the database.		
<b>Preconditions:</b>	Customer has navigated to the Login page from the Home page.		
<b>Postconditions:</b>	A "Login Successful!" message will be returned to the user.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. The Customer may fill in their credentials to log in to an existing user account.</li> <li>2. The Customer clicks the "Login" button to log in.</li> <li>3. A request is sent to the database to check the credentials against the stored data.</li> </ol>		
<b>Alternative Flows:</b>	-		
<b>Exceptions:</b>	-		
<b>Includes:</b>	UC002.2: ValidateUserCredentials		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC002.2		
<b>Use Case Name:</b>	ValidateUserCredentials		
<b>Created By:</b>	Aidan Ling	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	25 October 2023	<b>Date Last Updated:</b>	25 October 2023
<b>Actor:</b>			
<b>Description:</b>	This use case outlines the process where the database validates the submitted credentials against the existing credentials of the user stored in the database table.		
<b>Preconditions:</b>	A request is sent to the database to validate the user's credentials.		
<b>Postconditions:</b>	Returns a "Login Successful!" to the User		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	1. A request is sent to the database to validate the user's credentials. 2. The database will validate if the submitted password matches exactly with the password of the submitted username in the credentials. 3. If the password matches, the credentials are validated, and the user is allowed to log in.		
<b>Alternative Flows:</b>	-		
<b>Exceptions:</b>	<u>EX1: Username/Password is wrong.</u> 1. The database will inform the System to return an error message "Invalid Username or Password!"		
<b>Includes:</b>	Database		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC003		
<b>Use Case Name:</b>	SelectServiceRequired		
<b>Created By:</b>	Aidan Ling	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b> User (Customer)			
<b>Description:</b>	This use case outlines the process where a user may start the service request booking process by selecting a required service.		
<b>Preconditions:</b>	1. Customer has navigated to the Services page from the Home page.		
<b>Postconditions:</b>	Customer is redirected to specify the requirements of the service request booking on the Requests page.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	1. The Customer navigates to the Services page from the Home page. 2. This is done by clicking the “Services” button on the top right-hand corner of the Home page, which will redirect them to the Services page. 3. The Customer may select their required service. 4. The Customer will be redirected to the Requests page.		
<b>Alternative Flows:</b>	<u>AF-S1: Cancellation of the selection process by closing the tab.</u> 1. No booking request is made.		
<b>Exceptions:</b>	-		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC004		
<b>Use Case Name:</b>	Specify Requirements		
<b>Created By:</b>	Aidan Ling	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b> User (Customer)			
<b>Description:</b>	This use case outlines the process where a user can specify the requirements of their service request such as uploading a picture of the issue and specifying the time for the execution of the service.		
<b>Preconditions:</b>	All prior Use Cases have been done successfully.		
<b>Postconditions:</b>	A service request is created, encompassing the type of service, user's current location, and time slot preferred. This service request will be sent for further processing on the backend.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. The Customer is redirected to the Requests page from the Services page after selecting their required service.</li> <li>2. On the Requests page, the Customer may upload an image and select a timeslot.</li> <li>3. The Customer clicks on the "I Need Help" button.</li> <li>4. This creates the service request, and it will be sent to the backend for further processing.</li> </ol>		
<b>Alternative Flows:</b>	<u>AF-S1: Cancellation of the requirements specification process by closing the tab:</u> 1. No booking request is made.		
<b>Exceptions:</b>	<u>EX1: Timeslot was not specified.</u> 1. The System will return an error message "Timeslot was not specified".		
<b>Includes:</b>	2. UC004.1: UploadPhotograph 3. UC004.2: SpecifyTimeOfService		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC004.1		
<b>Use Case Name:</b>	UploadPhotograph		
<b>Created By:</b>	Howell Chan	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b> User (Customer), System			
<b>Description:</b>	This use case outlines the process where a user can upload an image of the problem that requires servicing.		
<b>Preconditions:</b>	All prior Use Cases have been done successfully.		
<b>Postconditions:</b>	An image is successfully uploaded onto the platform and rendered onto the webpage.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. The Customer will select “Upload” under the “Attach Photograph” section when specifying requirements.</li> <li>2. The Customer will browse through their file system and select an image to upload.</li> <li>3. The image is validated on the backend.</li> <li>4. Once validated, the file will be successfully uploaded and rendered onto the web page.</li> </ol>		
<b>Alternative Flows:</b>	-		
<b>Exceptions:</b>	<u>EX1: File format is invalid.</u> <ol style="list-style-type: none"> <li>1. The System will return an error message “Invalid File Format”.</li> <li>2. The Customer may reupload an image.</li> </ol>		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. File validation is required for the image.</li> <li>2. Ensure that the file has a specific extension type of either JPEG, JPG or PNG.</li> </ol>		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC004.2		
<b>Use Case Name:</b>	SpecifyTimeOfService		
<b>Created By:</b>	Aidan Ling	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b> User (Customer)			
<b>Description:</b>	This use case outlines the process where a user can select the time slot for their service request.		
<b>Preconditions:</b>	All prior Use Cases have been done successfully.		
<b>Postconditions:</b>	A timeslot between 1, 2 and 3 hours of the user's current time is rendered onto the screen. The user can select one of these timeslot's successfully.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	1. The web page will render three time slots under the section of "When do you need our service?" during the service specification process. 2. The user can only select one time slot from the three given time slots.		
<b>Alternative Flows:</b>	-		
<b>Exceptions:</b>	<u>EX1: Timeslot was not specified.</u> 1. The System will return an error message "Timeslot was not specified".		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC005		
<b>Use Case Name:</b>	SubmitServiceRequest		
<b>Created By:</b>	Howell Chan	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b> User (Customers & Service Providers)			
<b>Description:</b>	This use case outlines the process where a user can submit a service request to the web platform.		
<b>Preconditions:</b>	All prior Use Cases have been done successfully.		
<b>Postconditions:</b>	1. The service request is submitted and stored in the Servify database. 2. The user receives a confirmation notification.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	1. The Customer clicks on the “I Need Help” button. 2. This creates the service request, and it will be sent to the backend for further processing. 3. The System validates the service request. 4. The System queries the database to find Service Providers that match the desired service in the service request. 5. The database returns a list of matching Service Providers, sorted in the order of location distance from the user. 6. The System will send a notification to all Service Providers (who offer the specified service) via email. 7. The email will contain a link to the Service Provider page. 8. The Service Provider page will display the details of the service request. These details include the specified service, timeslot, and location of the Customer. 9. The Service Provider page will prompt the Service Provider to accept or decline the service request. 10. Upon first acceptance of the request, the System displays a confirmation notification to the Customer.		
<b>Alternative Flows:</b>	-		
<b>Exceptions:</b>	<u>EX1: System fails to submit the service request.</u> 1. The System will return an error message “Internal Server Error”. 2. The user may resubmit the service request.  <u>EX2: No service providers were found within the duration of search.</u> 1. The System will return a message “There are no Service Providers currently available”. 2. The user may resubmit the service request.		
<b>Includes:</b>	1. UC005.1: IdentifyServiceProviders 2. UC005.2: ReceiveServiceRequest		
<b>Special Requirements:</b>	The system must send a confirmation notification within 5 minutes of service request submission.		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC005.1		
<b>Use Case Name:</b>	IdentifyServiceProviders		
<b>Created By:</b>	Aidan Ling	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b>			
<b>Description:</b>	User (Customer), Database This use case outlines the process where the system will identify the appropriate service providers.		
<b>Preconditions:</b>	All prior Use Cases have been done successfully.		
<b>Postconditions:</b>	System will curate appropriate service providers.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	1. The System queries the database to find Service Providers that match the desired service in the service request. 2. The database returns a list of matching service providers, sorted in the order of location distance from the user.		
<b>Alternative Flows:</b>	<u>AF-S1: System query returns no Service Providers found.</u> 1. The System will return a message “There are no Service Providers currently available”.		
<b>Exceptions:</b>	<u>EX2: Database timeout or failure.</u> 1. The System will return an error message “Internal Server Error”. 2. The user may resubmit the service request.		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC005.2		
<b>Use Case Name:</b>	ReceiveServiceRequest		
<b>Created By:</b>	Howell Chan	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b>			
<b>Description:</b>	User (Customer & Service Provider), Database This use case outlines the process where a service provider receives a service request from the customer.		
<b>Preconditions:</b>	All prior Use Cases have been done successfully.		
<b>Postconditions:</b>	The Service Provider would be prompted to accept or decline the request.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>The Customer will see a pop-up stating, "Finding a service provider for you...!" until a Service Provider is found.</li> <li>Service Providers will receive an email with a link to the Service Provider page.</li> <li>The Service Provider page will display the details of the service request. These details include the specified service, timeslot, and location of the Customer.</li> <li>The Service Provider page will prompt the Service Provider to accept or decline the service request.</li> </ol>		
<b>Alternative Flows:</b>	-		
<b>Exceptions:</b>	-		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC006		
<b>Use Case Name:</b>	AcceptServiceRequest		
<b>Created By:</b>	Howell Chan	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b> User (Service Provider)			
<b>Description:</b>	This use case outlines the process where a service provider accepts the service request from a customer.		
<b>Preconditions:</b>	All prior Use Cases have been done successfully.		
<b>Postconditions:</b>	<p>The service request is accepted by the first Service Provider that responds to the service request.</p> <p>The service request is rejected if no Service Provider that accepts the service request.</p>		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. The Service Provider page will prompt the Service Provider to accept or decline the service request.</li> <li>2. The Service Provider may choose to accept or reject the service request.</li> <li>3. Upon acceptance by the first Service Provider, the Customer is notified that the service request is successful.</li> </ol>		
<b>Alternative Flows:</b>	<u>AF-S1: No service providers accepted the service request after 5 minutes.</u> <ol style="list-style-type: none"> <li>1. The System will return a message “There are no Service Providers currently available”.</li> <li>2. The user may resubmit the service request.</li> </ol>		
<b>Exceptions:</b>	-		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC007		
<b>Use Case Name:</b>	ViewServiceProviderLocation		
<b>Created By:</b>	Howell Chan	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b>			
<b>Description:</b>	User (Customer & Service Provider), Google Maps Geolocation API		
<b>Preconditions:</b>	This use case outlines the process where a user can view the real-time location of a service provider after a service booking has been confirmed.		
<b>Postconditions:</b>	All prior Use Cases have been done successfully.		
<b>Priority:</b>	The user may view the real-time location of the service provider.		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>Google Maps' Geolocation API (GeoAPI) retrieves the real-time location of the Service Provider.</li> <li>The System will call GeoAPI to display a real-time map view, showing the Service Provider's route to the Customer's location.</li> </ol>		
<b>Alternative Flows:</b>	<u>AF-S1: Service provider's device does not permit location sharing.</u> <ol style="list-style-type: none"> <li>Service Provider is prompted to allow location sharing.</li> </ol>		
<b>Exceptions:</b>	<u>EX1: Geolocation API fails to retrieve the Service Provider's location.</u> The System will return an error message "Failed to retrieve Service Provider's location".		
<b>Includes:</b>	<ol style="list-style-type: none"> <li>UC007.1: RetrieveLocationFromUserDevice</li> <li>UC007.2: DisplayServiceProviderLocation</li> </ol>		
<b>Special Requirements:</b>	System must update the location in real-time.		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	The real-time location feature's accuracy depends on the Service Provider's mobile device capabilities and settings.		

<b>Use Case ID:</b>	UC007.1		
<b>Use Case Name:</b>	RetrieveLocationFromUserDevice		
<b>Created By:</b>	Howell Chan	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b>			
<b>Actor:</b>	User (Customer), Google Maps Geolocation API (GeoAPI)		
<b>Description:</b>	This use case outlines the process where the system retrieves the geographical location of the user's device.		
<b>Preconditions:</b>	1. All prior Use Cases have been done successfully. 2. The user must grant location services permissions to the web platform. 3. The service provider must enable location services permissions on their mobile device. 4. Google Maps Geolocation API must be accessible.		
<b>Postconditions:</b>	The geographical location of the user's device is retrieved and stored temporarily for the current session.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	1. The System will verify if location services permissions have been enabled on the Customer's and Service Provider's device. 2. If permissions are enabled, the system will send a request to GeoAPI to retrieve the current location of the devices. 3. GeoAPI returns the geographical coordinates of both devices. 4. The system will store these coordinates throughout the journey from the Service Provider's location to the Customer's location. 5. The system will continuously update the coordinates by sending repeated queries to the Google Maps API.		
<b>Alternative Flows:</b>	<u>AF-S1: User's device does not permit location sharing.</u> 1. User is prompted to allow location sharing.		
<b>Exceptions:</b>	<u>EX2: Google Maps API is not accessible.</u> 1. The System will return an error message "Location Services are currently unavailable."		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

<b>Use Case ID:</b>	UC007.2		
<b>Use Case Name:</b>	DisplayServiceProviderLocation		
<b>Created By:</b>	Howell Chan	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b>			
<b>Actor:</b>	User (Service Provider), Google Maps Geolocation API (GeoAPI)		
<b>Description:</b>	This use case outlines the process where the real-time location of a service provider is displayed after a service booking has been confirmed.		
<b>Preconditions:</b>	All prior Use Cases have been done successfully.		
<b>Postconditions:</b>	The user may view the real-time location of the service provider as he moves towards customer's preferred destination.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	1. The System will use GeoAPI to display real-time map view, showing the service provider's route to the user's location.		
<b>Alternative Flows:</b>	<u>AF-S1: Geolocation API fails to retrieve the Service Provider's location midway through travel.</u> 1. The System will return an error message "Failed to retrieve Service Provider's location". 2. The map view will continue to show the last known location of the Customer and Service Provider.		
<b>Exceptions:</b>	-		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	System must update the location in real-time.		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	The real-time location feature's accuracy depends on the Service Provider's mobile device capabilities and settings.		

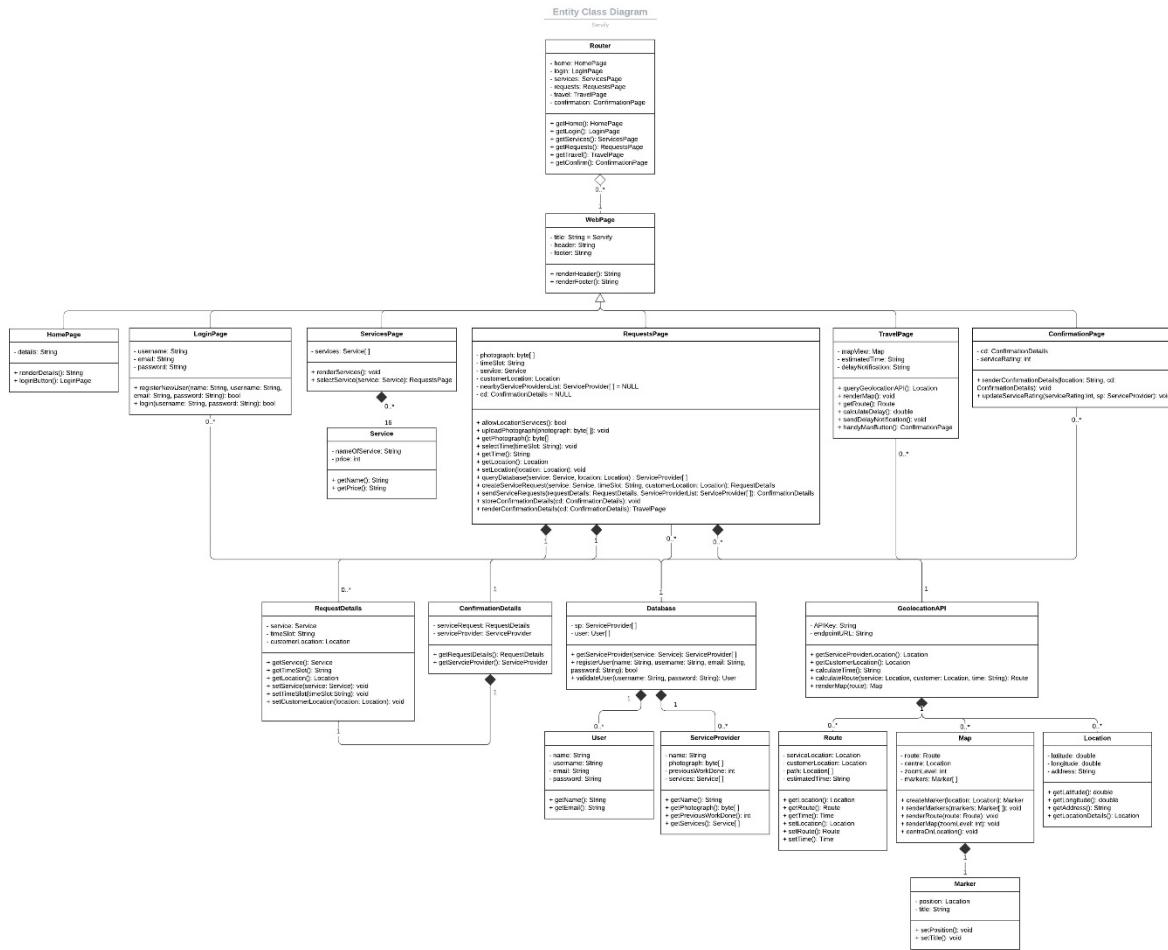
<b>Use Case ID:</b>	UC008		
<b>Use Case Name:</b>	ConfirmServiceProviderArrival		
<b>Created By:</b>	Howell Chan	<b>Last Updated By:</b>	Aidan Ling
<b>Date Created:</b>	2 September 2023	<b>Date Last Updated:</b>	12 September 2023
<b>Actor:</b> User (Customer & Service Provider)			
<b>Description:</b>	This use case outlines the process where a customer confirms the completion of a service offered by the service provider.		
<b>Preconditions:</b>	1. All prior Use Cases have been done successfully. 2. Service Provider has successfully reached the Customer's location.		
<b>Postconditions:</b>	The System will render the Confirmation page, listing a summary of the details of the service, and prompting the user to verify.		
<b>Priority:</b>	High		
<b>Frequency of Use:</b>	High		
<b>Flow of Events:</b>	1. When the Service Provider has arrived at the Customer's location, the Customer may select the "Handymen Arrived" button. 2. This will redirect the user to the Confirmation page. 3. The Confirmation page will display a summary of the details of the service. 4. The Customer is prompted to verify the confirmation details. 5. Once verified, the Customer is redirected to the Home page.		
<b>Alternative Flows:</b>	AF-S1: Customer does not verify the page within 2 minutes. The Customer is redirected to the Home page.		
<b>Exceptions:</b>	-		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

## 5. Data Dictionary

Term	Definition
User	A user refers to any entity that engages with the web platform to either access or offer services.  This includes customers who browse and book services, as well as service providers who utilise the platform to make their offerings publicly available.
System	The system refers to the Servify web platform, engineered to facilitate interactions between customers in search of services and local professionals providing those services.
Customers	Customers on are individuals who leverage on Servify to find and hire local professional service providers for a variety of services.
Service Providers	Service providers refer to individuals or businesses that are registered on the Servify platform to offer specialized services to customers.
Booking	A booking refers to a reservation made by a customer to secure a specific service from a service provider at a scheduled time and date.
Review	A review refers to a customer-generated rating that quantifies their level of satisfaction with the service received from a provider.
Payment	A payment refers to the financial transaction executed by the user to compensate the service provider for services rendered.
Notification	A notification refers to an automated alert generated by the system to inform users or service providers about specific activities or updates.
Service	A service refers to a specialized skill or work offered by service providers that users can request for using the Servify platform.

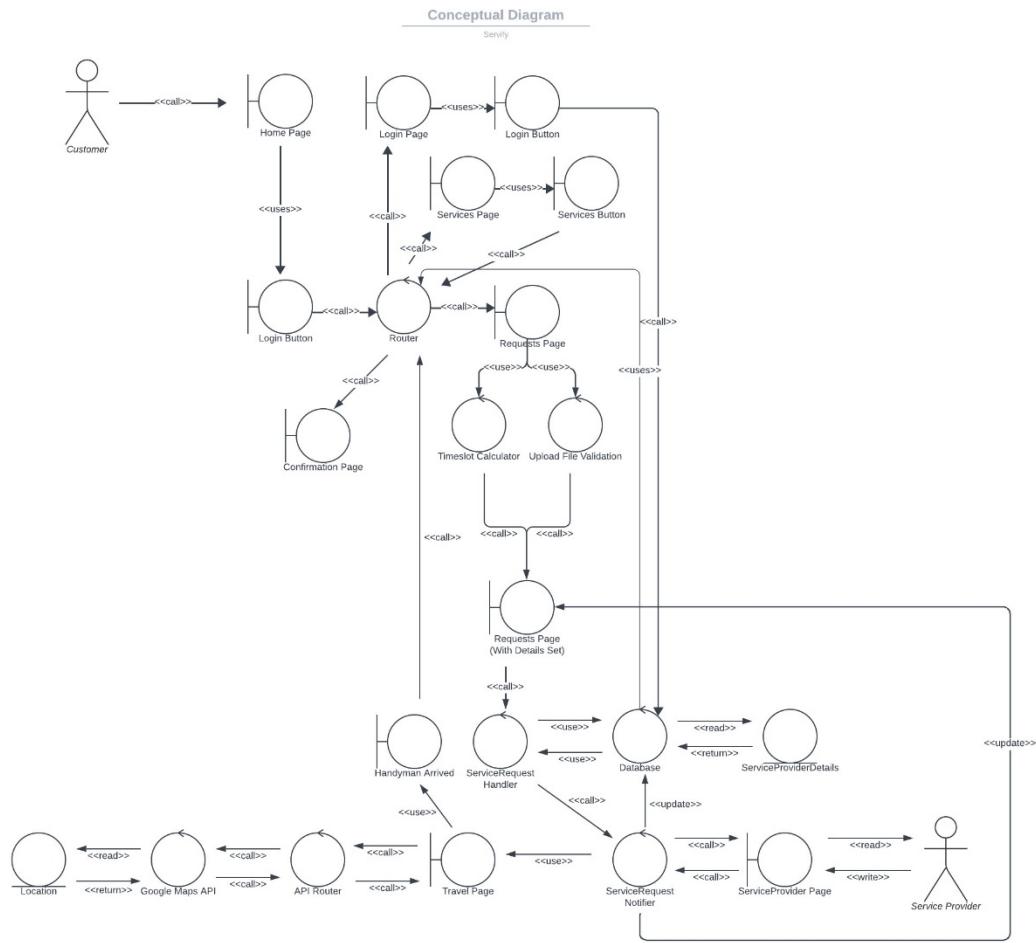
## 6. Diagrams

### 6.1. Entity Class Diagram

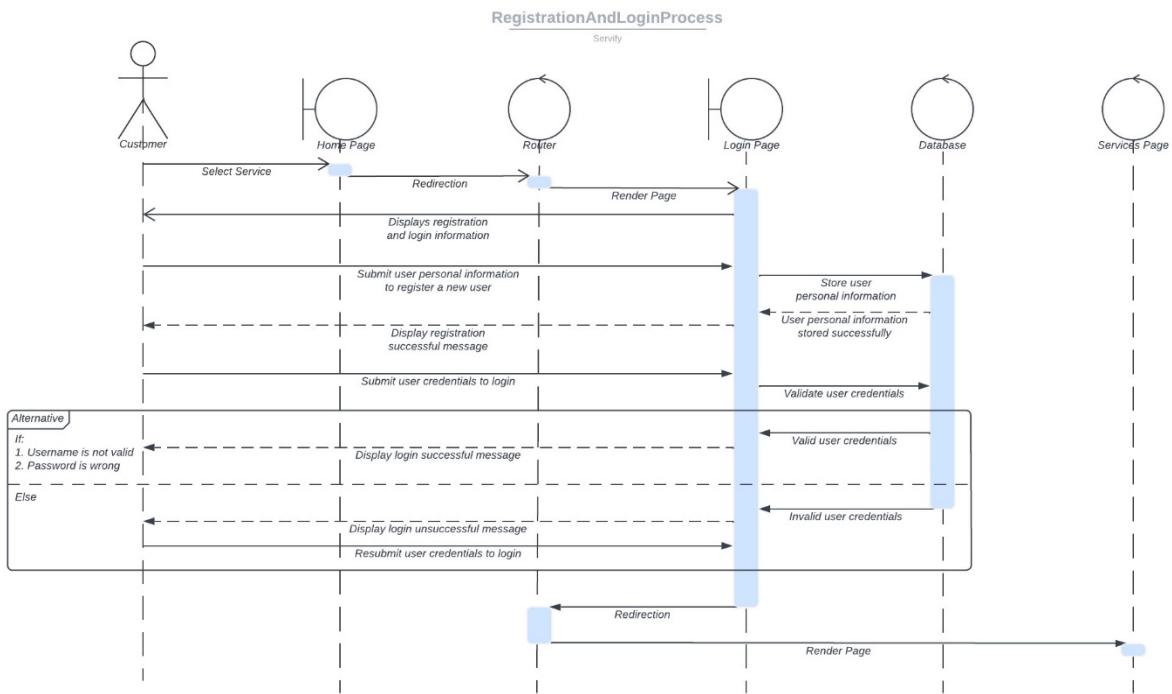


If the text clarity is an issue, you may view clearer PDF versions of all diagrams hosted on [Google Drive](#).

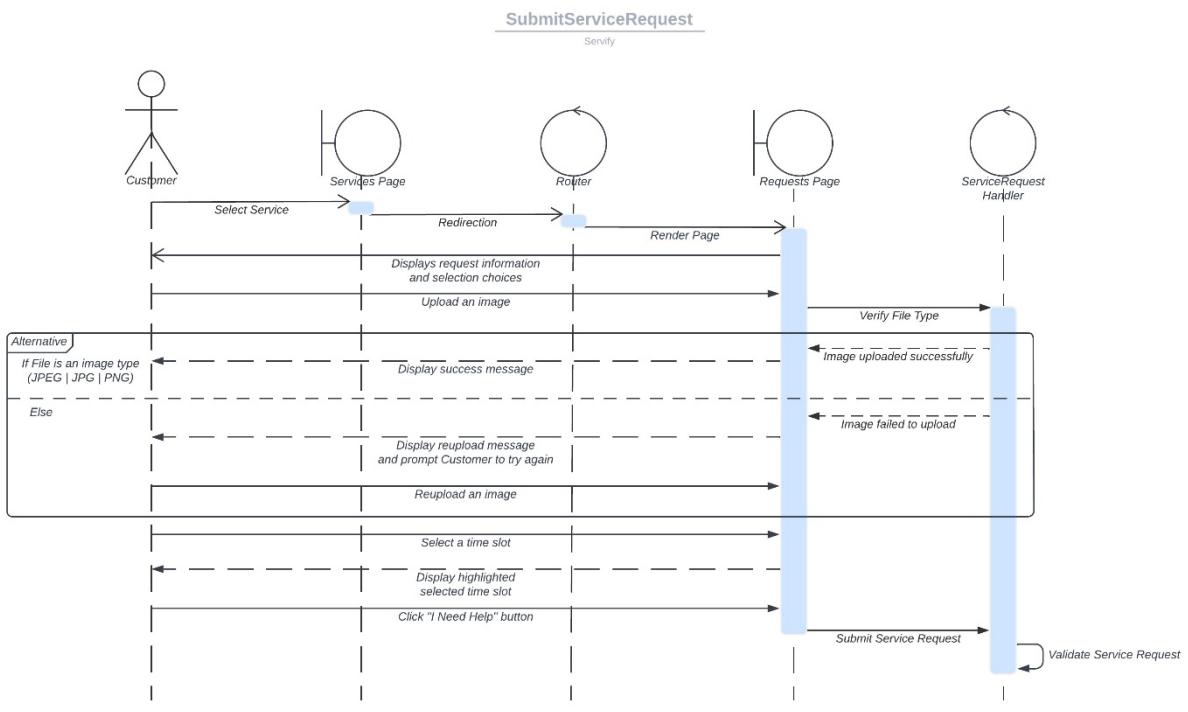
## 6.2. Conceptual Diagram



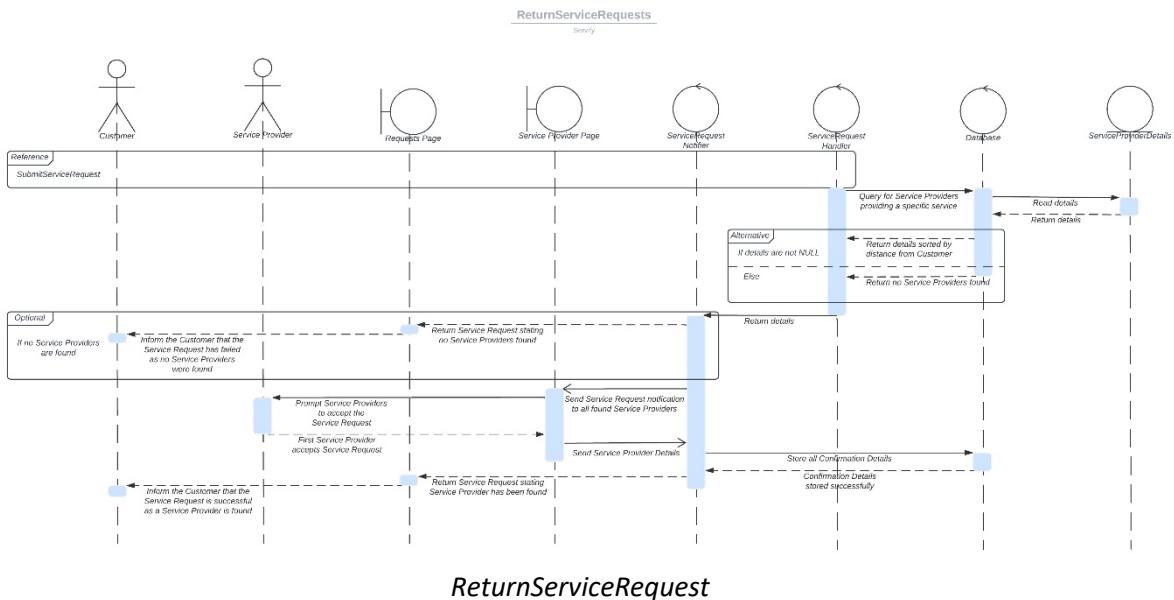
### 6.3. Sequence Diagrams



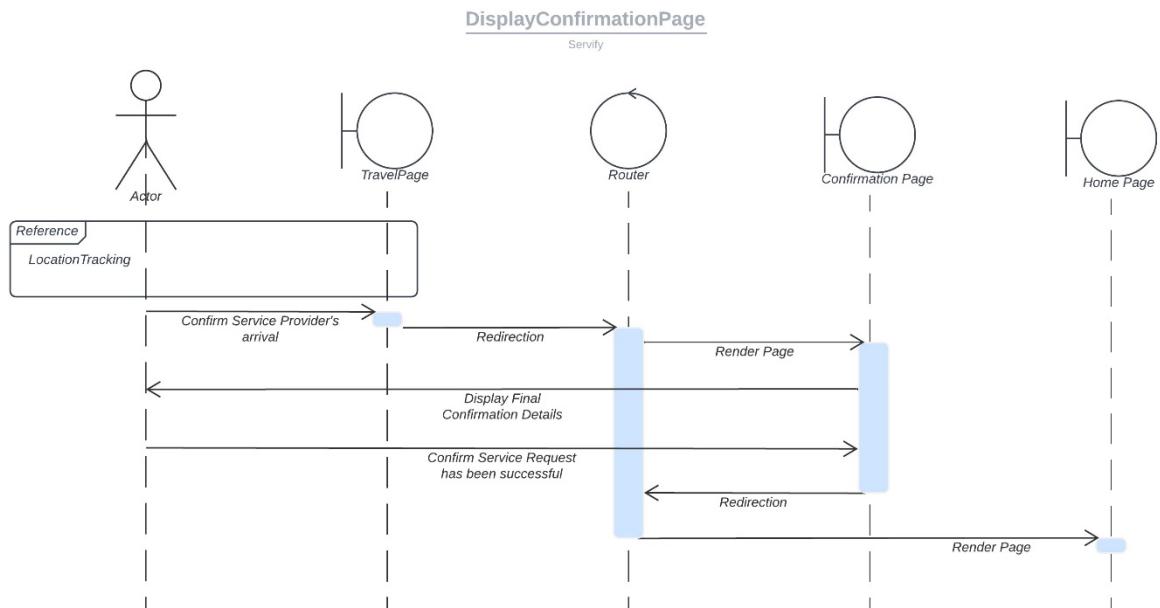
*RegistrationAndLoginProcess*



*SubmitServiceRequest*

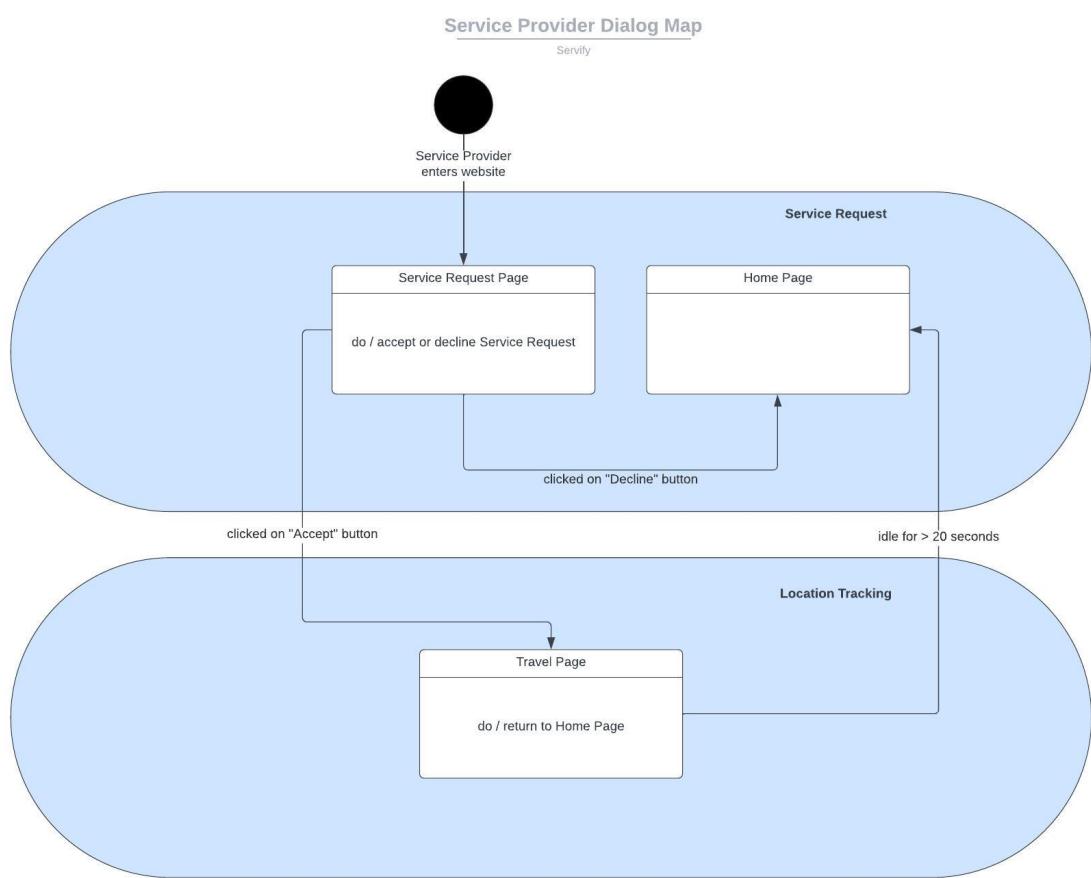
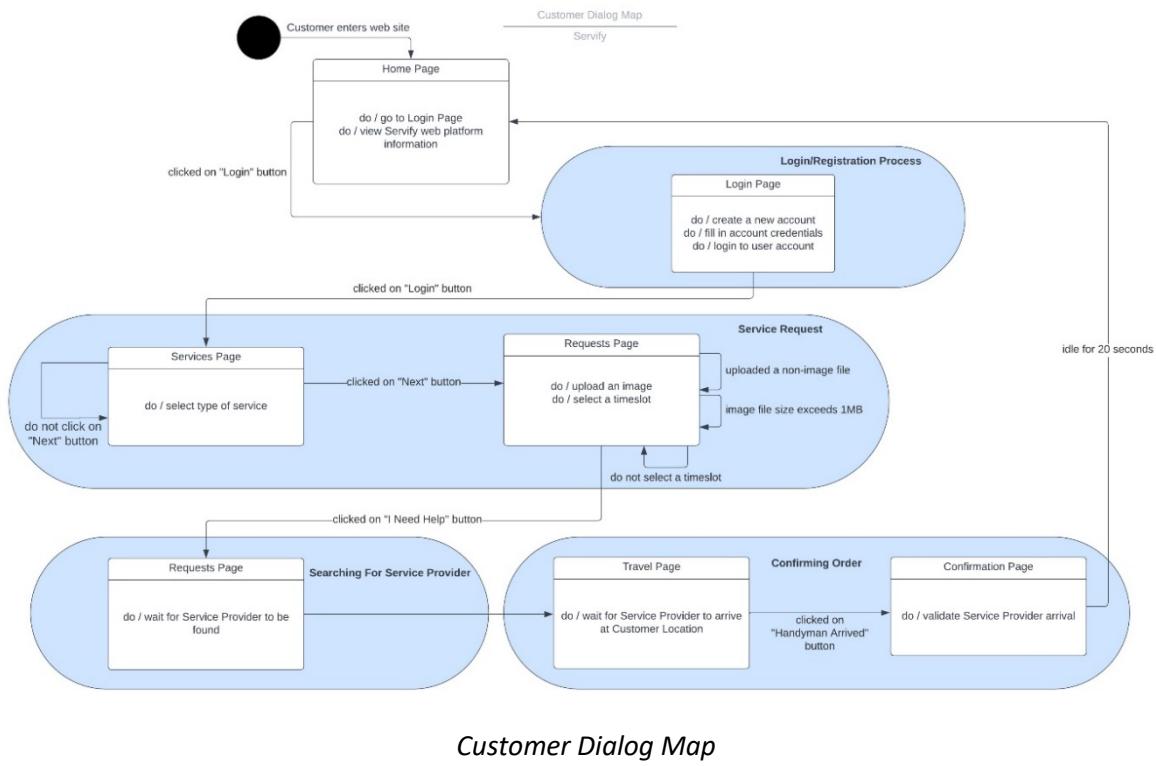


### ReturnServiceRequest



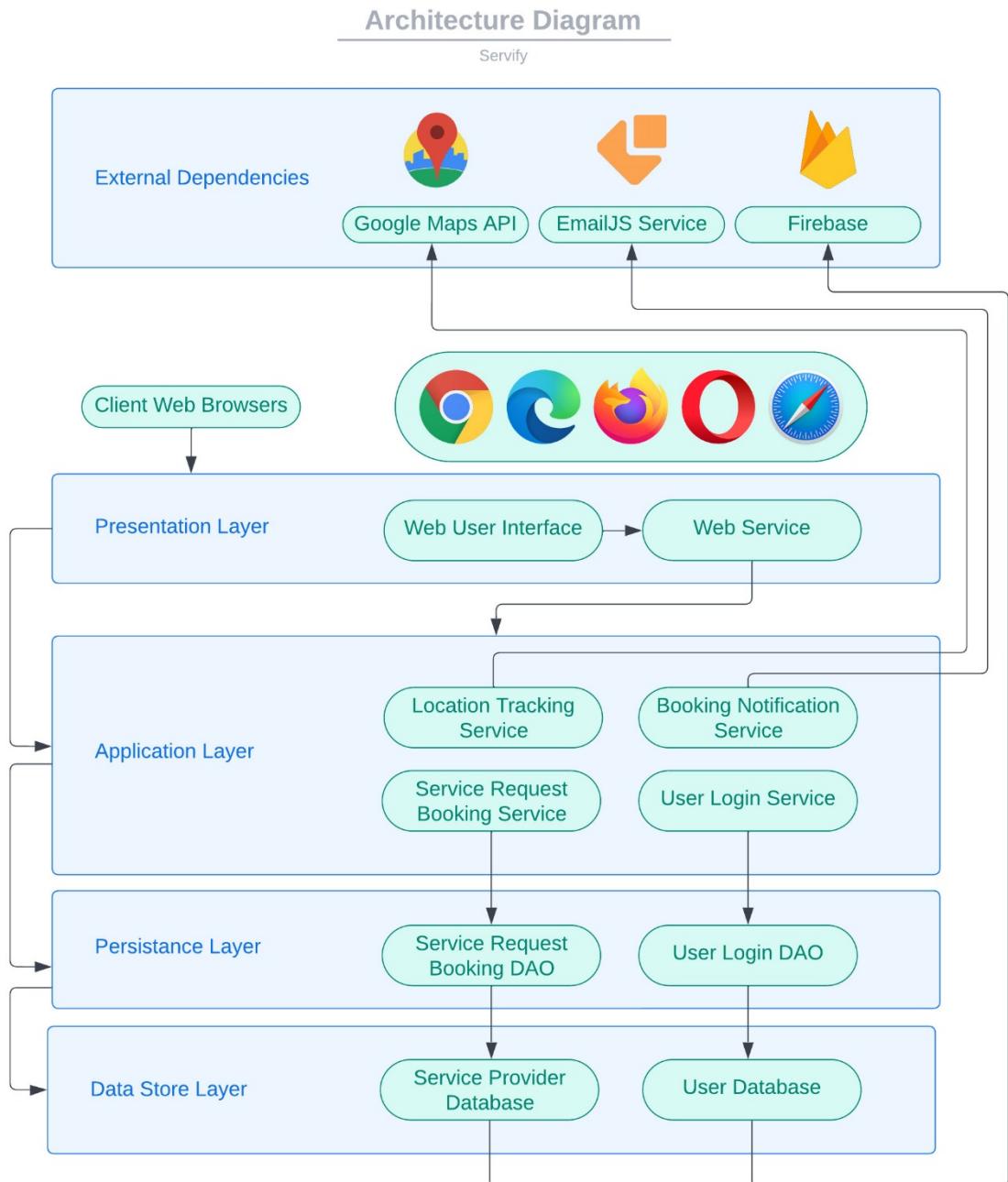
### DisplayConfirmationPage

## 6.4. Dialog Map



*Service Provider Dialog Map*

## 6.5 System Architecture



A Data Access Object (DAO) provides an abstract interface to a database or other data source, enabling CRUD operations without exposing details of data access logic.