**Project: Massage Booking System**

**Prepared by:**

Tyler Gettle, Saihaj Mann, Vinicius Melo,

Victor Odhiambo, Andrew Wahlers

Southern Alberta Institute of Technology

**Project Statement**

**Project:**

Massage Booking System

**Project Sponsor:**

Aisha Carrazo, Kaan Massage

**Sponsors Current System:**

Our client’s current website with a booking system is very simple and straightforward; it allows users to view available times for both the client as well as her mother. The system starts as soon as the user selects an available time, then it takes the user to a separate page where the user is prompted to enter their personal information. Once the user has entered their information, they will be booked into the system.

**Proposed Solution:**

Our proposed solution is to transition our client from her previous website to a redesigned website with a booking system, that allows any user to create an account and log in and have their information be retained within the systems database. This will allow the user to be able to access their booking and rebook if needed, as well as change their information more easily. Also, as requested by the client, we will be redesigning the front-end of the website, giving it more detail and a modern design that gives off a more user-friendly experience.

**Use Case Diagram**

A diagram of a system

Description automatically generated

**Use Case Diagram Descriptions Extended Format**

**Use Case: Create Account**

**Description:** Users, including admins and clients, should be able to create accounts to access the application. They must provide their information, which will be securely stored in a database.

**Preconditions:** Users have a valid email or phone number.

**Postconditions:** A user account is created and stored securely in the database.

**Flow 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| User enters their information | Information is verified |
|  | Information is not duplicate |
|  | Send confirmation though email or text |
|  | User account is created |

**Flow 2**

|  |  |
| --- | --- |
| Actor action | System Response |
| User enters their information | Information is verified |
|  | Information is not duplicate |
|  | User account is created |

**Exceptions 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| User enter | Information is verified |
|  | Information is duplicate, displayed error |

**Exceptions 2**

|  |  |
| --- | --- |
| Actor action | System Response |
| User enter | Information is verified |
|  | Information is duplicate |
|  | Display error, sign in or reset password to sign in |

**Exceptions 3**

|  |  |
| --- | --- |
| Actor action | System Response |
| User enters their information | Information is verified |
|  | Information is not duplicate |
|  | User account is created |
|  | Sometime went wrong, please try signing up again or try later |

**Assumptions:** Users have a valid phone number or email address.

**Notes:**

**Use Case: Reset Password**

**Description:** Users, both admins and clients, can reset their password if they forget it or suspect a security issue with their current password.

**Preconditions:** User has an existing account.

**Postconditions:** Password is successfully reset.

**Flow 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| User click forgot password, and enter phone or email to sent the code to | System verifies |
|  | System sent a reset code to email or phone |
|  | User enters code and resets password |

**Exceptions:** None

**Assumptions:** Users have a valid phone number or email address.

**Notes:**

**Use Case: Log In**

**Description:** Once users have accounts, they need to log in to access the application's resources, including creating, viewing, and editing bookings.

**Preconditions:** User has an existing account.

**Postconditions:** User is logged into the application.

**Flow 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| User enter username and password | Computer check if it is correct |
|  | It is correct, log the person in |
|  | Display logged in status, send the session tracker for it |

**Exceptions 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| User enter username and password | Computer check if it is correct |
|  | Invalid combination, display error to user, wrong username or password |

**Assumptions:** Users have an existing account.

**Notes:**

**Use Case: Allow Booking Time**

**Description:** Administrators should be able to select available booking times, allowing them to configure recurring time slots that are not bookable.

**Preconditions:** Admin has a valid account and is signed in.

**Postconditions:** Booking times are configured.

**Flow 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| Admin enter the change time page | Verify |
|  | Display time page |
| User makes changes | Make changes in the server |

**Exceptions 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| Admin enter the change time page | Verify |
|  | Display error, user is not logged in or does not have access |

**Assumptions:** Users have logged in.

**Notes:**

**Use Case: Create Booking**

**Description:** Users, both admins and clients, can create bookings, ensuring they are within the allowable time and do not conflict with existing bookings.

**Preconditions:** Booking time is available, user has an account.

**Postconditions:** Booking is successfully created, or the booking process is aborted.

**Flow 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| Users enter the time select screen | Verify the user can book |
|  | Display the screen |
| User select the time | Verify time |
|  | Create booking |
|  | Display booking create to the user |

**Exceptions 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| Users enter the time select screen | Verify the user can book |
|  | Display the screen |
| User select the time | Verify time |
|  | Time is not available |
|  | Display error and ask user to select another time |

**Exceptions 2**

|  |  |
| --- | --- |
| Actor action | System Response |
| Users enter the time select screen | Verify the user can book |
|  | User is not logged in, display error and ask user to log in |

**Assumptions:** None

**Notes:**

**Use Case: Edit Bookings**

**Description:** Admins and users can make changes to existing bookings, including rebooking and adding notes.

**Preconditions:** A booking exists, and the user or admin has access to it.

**Postconditions:** User is redirected to the booking view page.

**Flow 1**

|  |  |
| --- | --- |
|  |  |
| User pulls up booking | Is user allowed to view booking |
|  | Display booking |
| User changes information | Verifies the data changes |
| Click save button | Changes save |

**Exceptions 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| User pulls up booking | Is user allowed to view booking |
|  | User is not allowed to access book, display error and send to previous page |

**Exceptions 2**

|  |  |
| --- | --- |
| Actor action | System Response |
| User pulls up booking | Is user allowed to view booking |
|  | User is not signed, ask user to sign in the view booking |

**Assumptions:** None

**Notes:**

**Use Case: Cancel Bookings**

**Description:** Users, both admins and clients, should be able to cancel their reservations.

**Preconditions:** A booking exists, and the user or admin has access to it.

**Postconditions:** User is taken to the view booking time section of the app.

**Flow 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| User pulls up booking | Is user allowed to view booking |
|  | Display booking |
| User click cancel button and confirm | Booking is cancelled |
|  | Booking is removed and time is opened up |

**Exceptions 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| User pulls up booking | Is user allowed to view booking |
|  | User is not allowed to access book, display error and send to previous page |

**Exceptions 2**

|  |  |
| --- | --- |
| Actor action | System Response |
| User pulls up booking | Is user allowed to view booking |
|  | User is not signed, ask user to sign in the view booking |

**Assumptions:** None

**Notes:**

**Use Case: View Booking**

**Description:** Users and administrators can view information about their bookings. Only users on a booking can view its details.

**Preconditions:** A booking exists, and the user is signed in.

**Postconditions:** User is taken to the view booking time section of the app.

**Flow 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| User pulls up booking | Is user allowed to view booking |
|  | Display booking |

**Exceptions 1**

|  |  |
| --- | --- |
| Actor action | System Response |
| User attempt to pull up a booking | The system verify the user identity |
|  | Display error the user is not signed in |
|  | Send user to sign in page, and ask user to sign in |

**Exceptions 2**

|  |  |
| --- | --- |
| Actor action | System Response |
| User attempt to view booking | The system verify the user identity |
|  | User is not on the book, take user to home screen, and display, you do not have access to this booking |

**Assumptions:** None

**Notes:**

**Use Case: Receive Notifications**

**Description:** Admins and clients should receive notifications for upcoming bookings and confirmation of new bookings.

**Preconditions:** A booking exists, the user has added their phone number, and they can receive text messages.

**Postconditions:** Notifications are sent as required.

**Flow 1**

* Reminder notifications are sent to clients as appointments approach.

|  |  |
| --- | --- |
| Actor action | System Response |
|  | Booking start time is coming up |
|  | Notification is sent to phone |

**Exceptions 1**

* A notification cannot be sent, and the user is informed on the page.

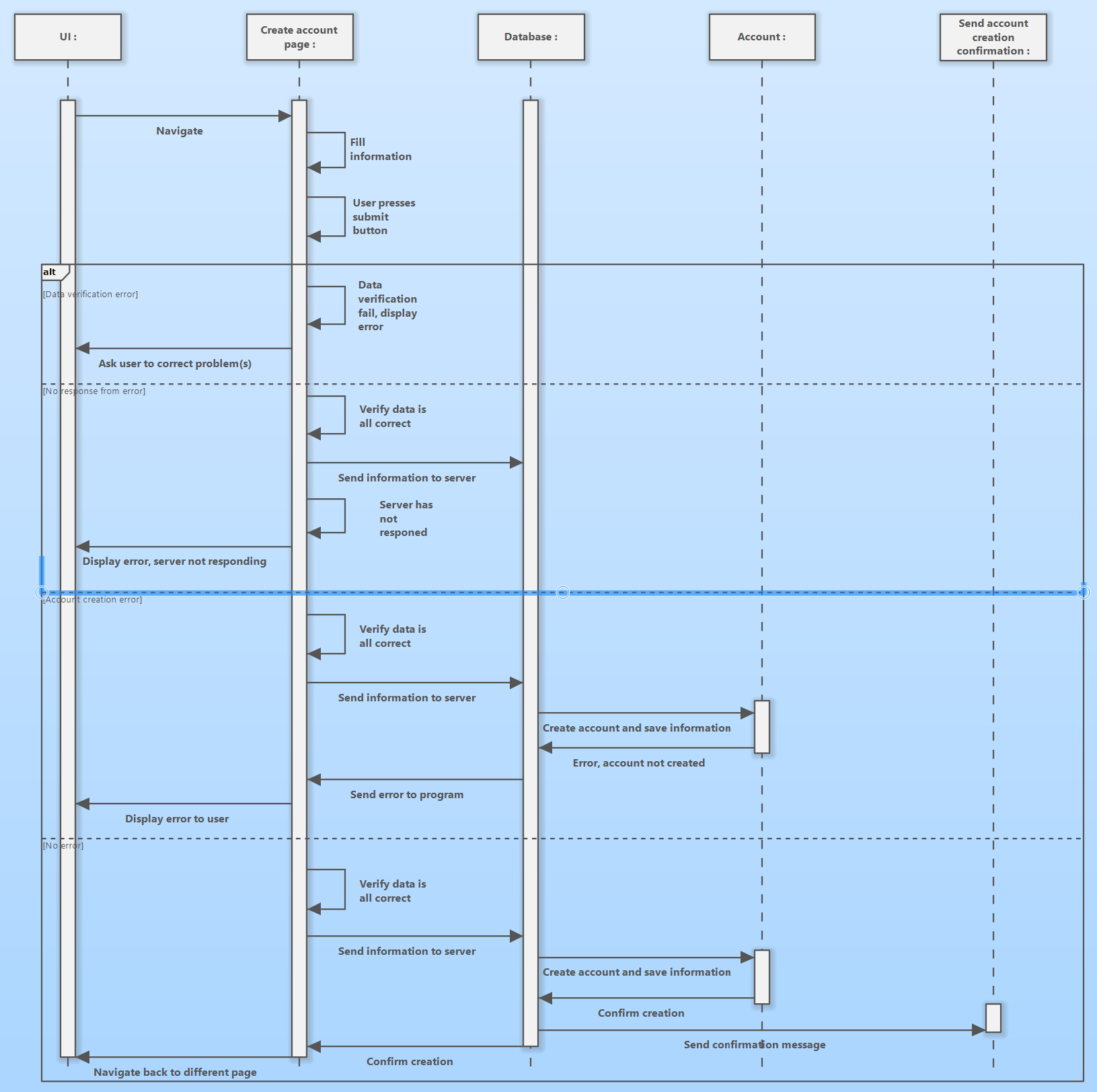
|  |  |
| --- | --- |
| Actor action | System Response |
|  | Notification can not be sent |
|  | Display error on website for user to see |

**Assumptions:** None

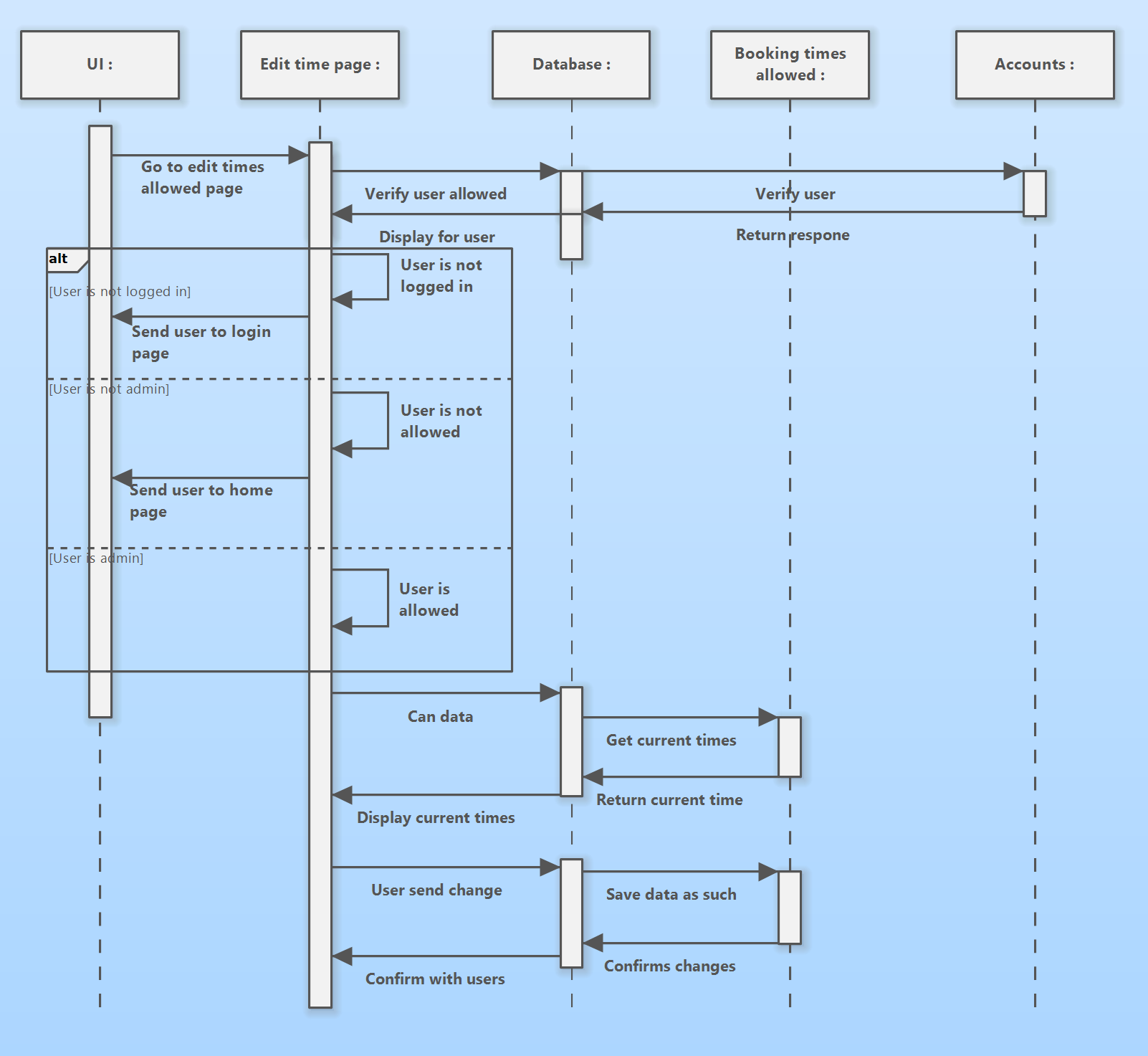
**Notes:**

**Sequence diagrams:**

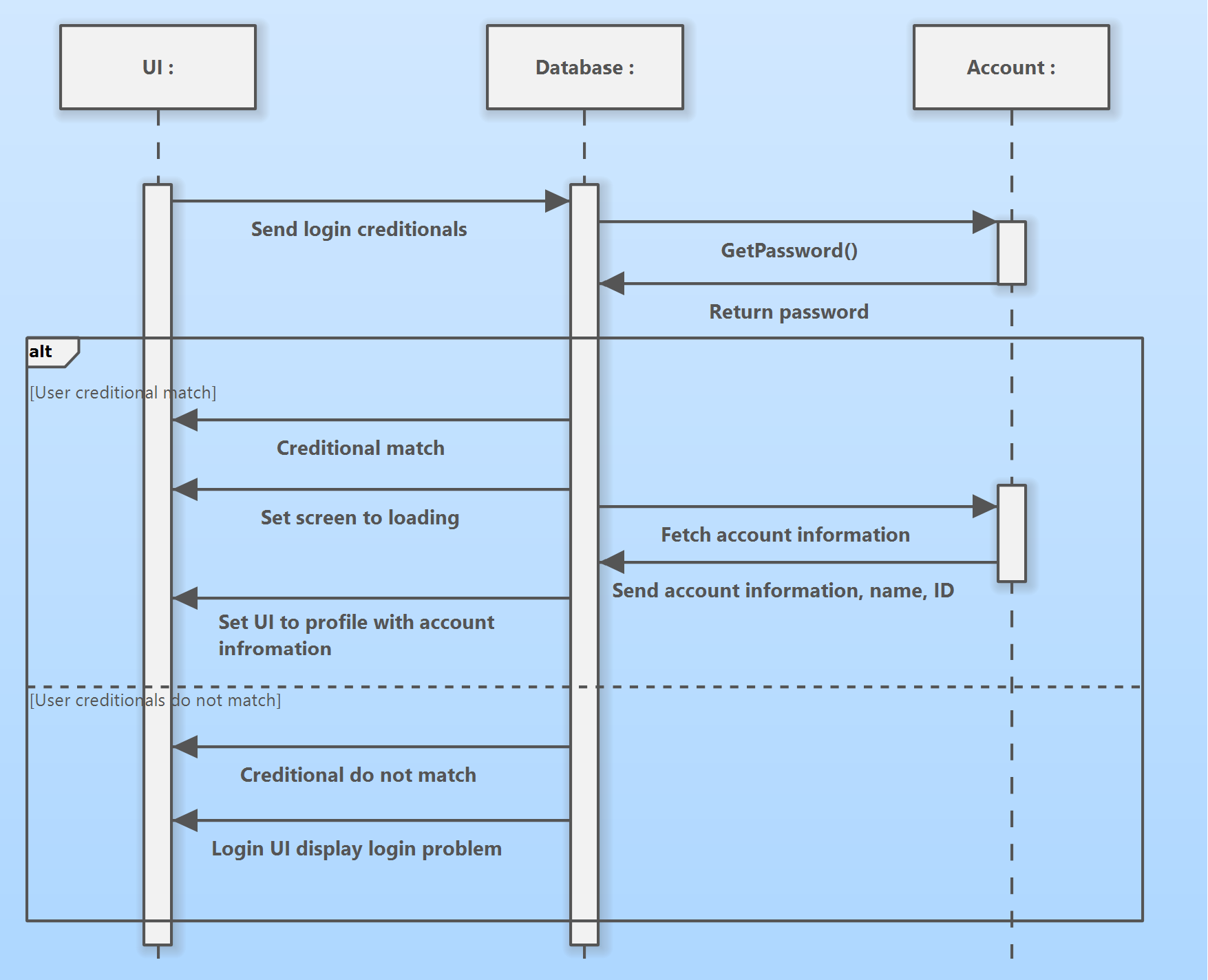
**Sequence diagram 1:**

****

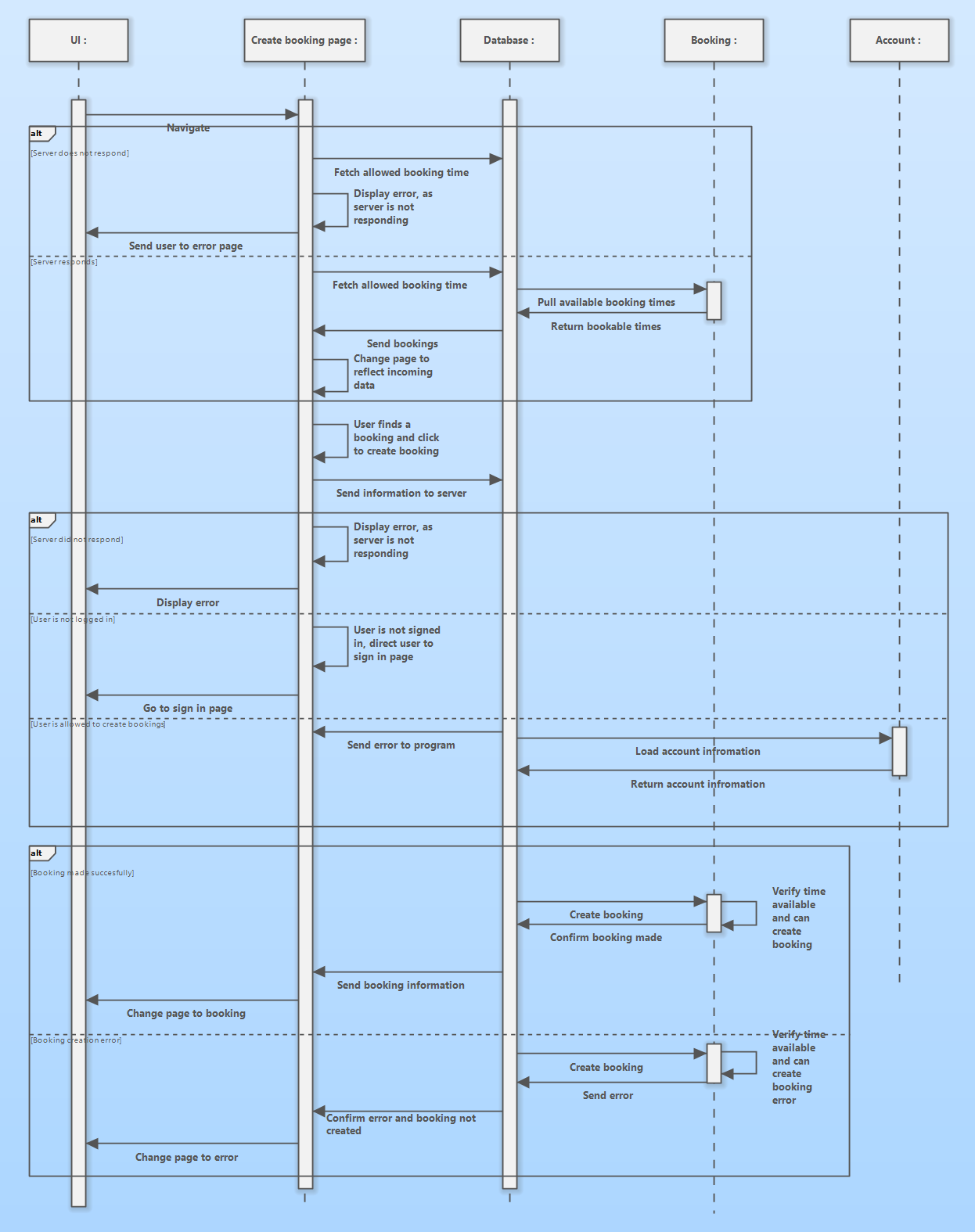
**Sequence diagram 2**



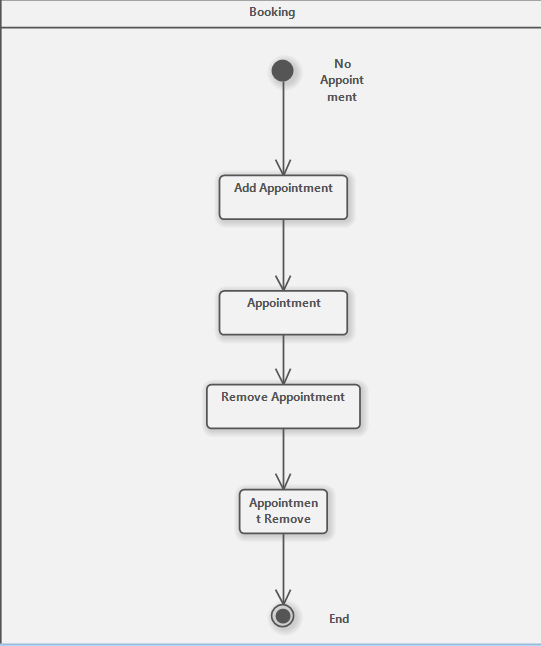
**Sequence diagram 3:**

****

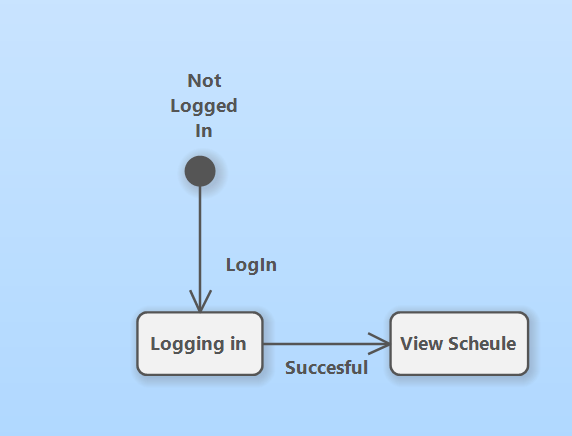
**Sequence diagram 4:**



#### Activity/State Machine Diagrams

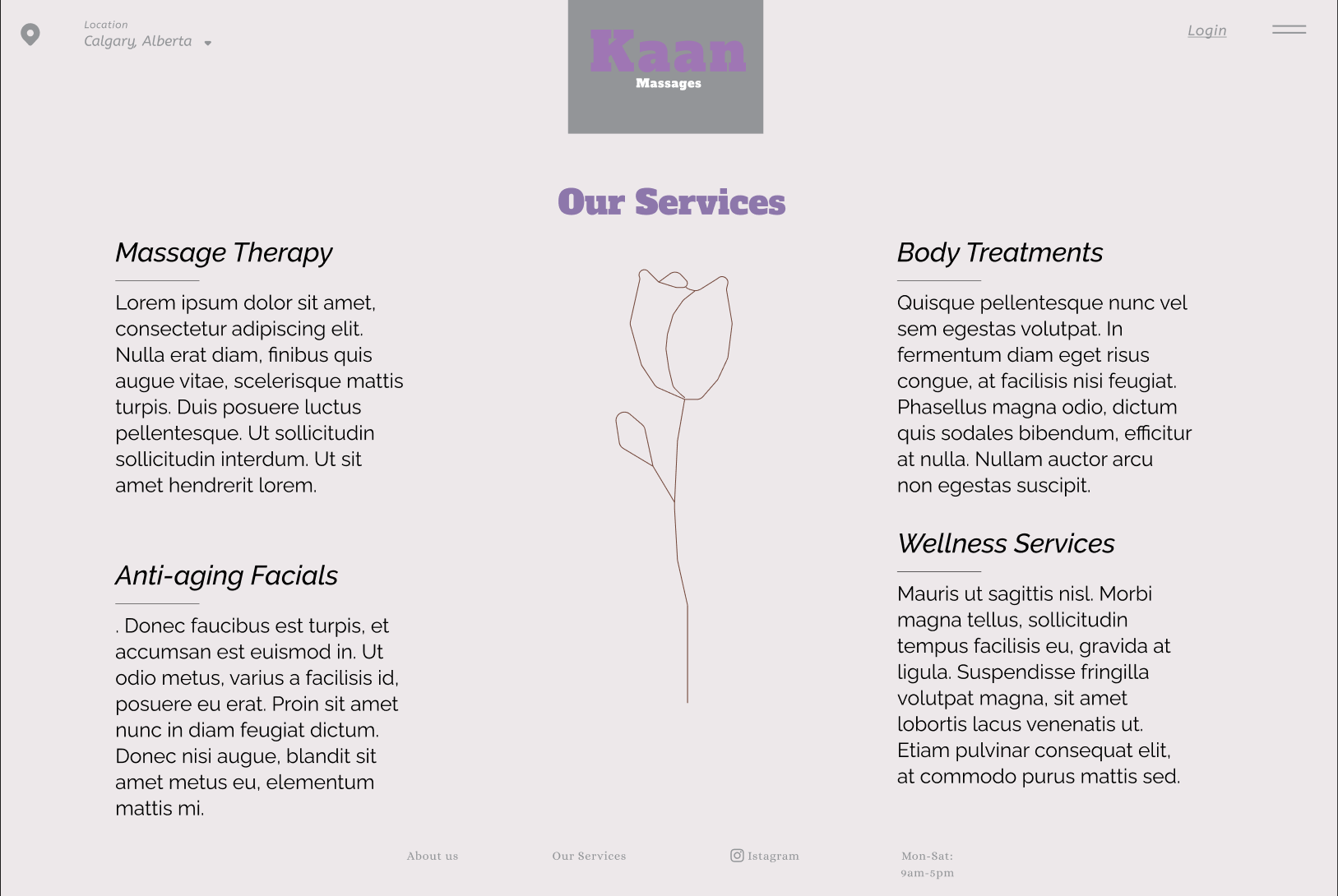
**Activity/state machine diagram 1:**

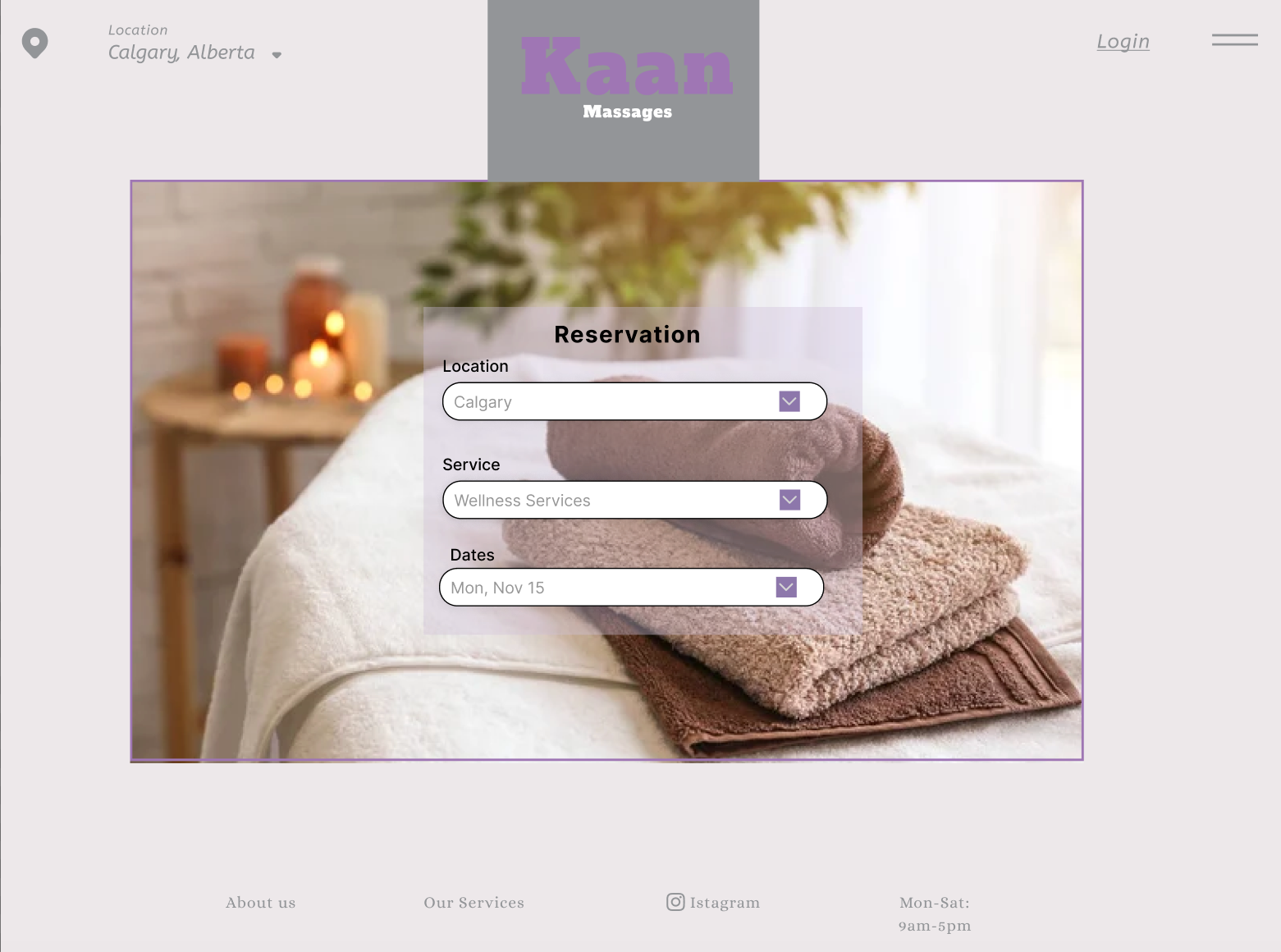
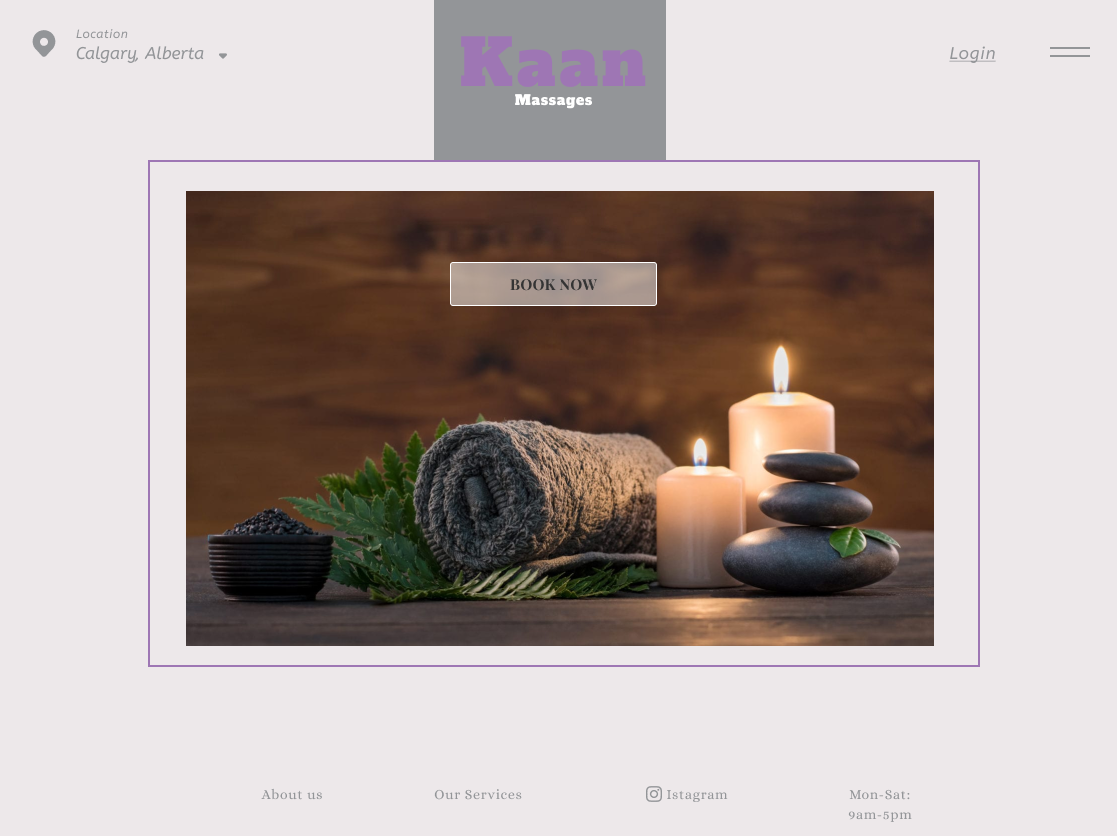
**Activity/state machine diagram 2:**

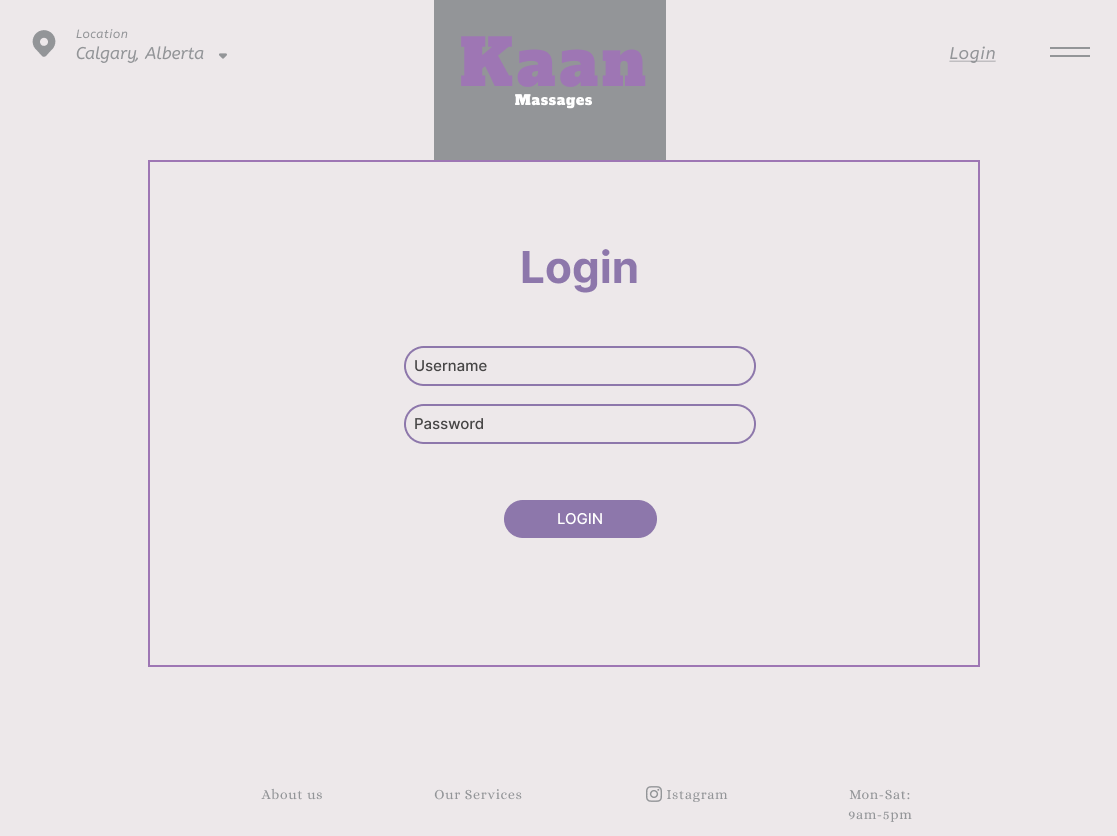
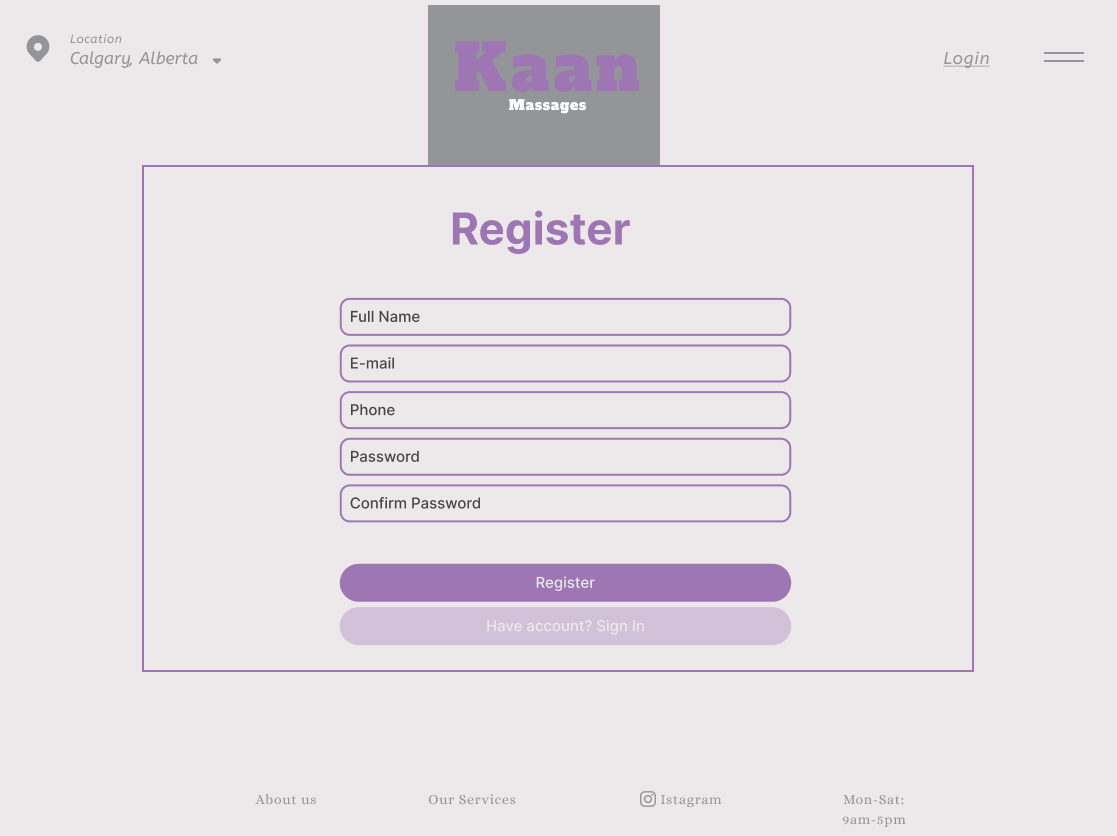
****

#### 

#### Preliminary User Interface Design







#### Data Storage and Persistence

**Requirements**

1. We require a database that will store all booking that are made.
2. It needs to store user account, with user sensitive user data, and keep it all data protected.
3. Booking need to be link bookings to the user account.
4. The database needs to be accessible quickly by the system to check which bookings are coming up.
5. The database need to be able to handle multiple request at a time, as multiple user may access it at once
6. The database needs to be able to scale to user demand and save resource when not in use.

**Options**

**MySQL**

* MySQL is an open source SQL database that is fast and efficient. It is widespread and used. We would be capable of finding a specific version that would be best. The main MYSQL is maintain by oracle but has many different version maintain by other organization. MariaDB is a popular fork of MYSQL with a dedicate user base. It been developed for years and been improving in each of. IT support stored procedure and trigger which also make it useful to run programs on.

**MongoDB**

* MongoDB is a open source document oriented database useful for storing documents. This it makes it especial useful when working with document. It is optimized for quick read and write function. It has strong scaling capabilities and data transformation, making it very useful when needed.

**PostgreSQL**

* PostgreSQL is an opensource object relational database. It could be used to link admins and users to bookings and store all the data as object, removing the need to reconstruct object each time there called. It also has SQL support and is ACID compliant, making transfers between different SQL databases simple.

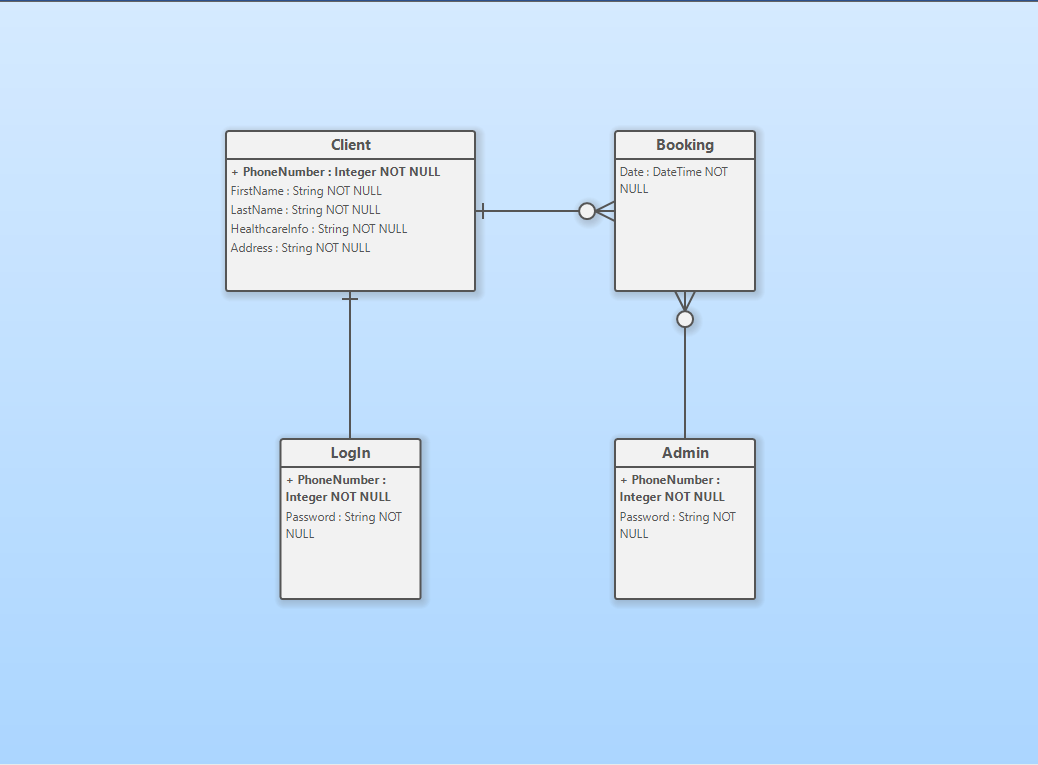
**Selection**

For now, we need a simple database that is fast, efficient and easy to use. We do not need the specialize the database, as a SQL database will be good for us. MySQL is maintained by many organizations, making it reliable and efficient for our uses. Storing documents is a nice feature, but document is currently not needed in the scope of the project. If MySQL not meeting requirements, we can switch to another SQL database with ease.

**Application**

1. There will be a data verification system in the project, it will take the username and password and check them against the database records. If it is a match, it will return true and allow user to view information, else it return an error message to the user as the password username combination did not match.
2. The stored password can be a hash for security reasons, and the incoming password will be hash to check if it matches. Once the user is verified, they can request their user information form the database. That information will include their name and phone number.
3. The user will be allowed to open and view booking as well, only if there are on the reservation. They can create booking with there account and the booking will be linked to there account. The system will verify if the user is allowed access to the booking, and will denied access to any one not on the booking. The user will be able to access the booking and make changes to it.
4. Admin will have access to view all booking and even if not on the booking, as log as they log in.
5. When resetting password, the database will be informed of the phone number and send a reset code if the phone number is on the system.
6. If the booking is coming up soon, a system will get the phone number of the people on the booking and will send them confirmations notifications.

#### Entity Diagram



**Class Diagram**

A screenshot of a computer

Description automatically generated

**System Architecture and Patterns**

System architecture and design patterns are integral facets of software development, each playing a crucial role in shaping robust and scalable systems. System architecture serves as the high-level blueprint, encompassing the organizational structure, component relationships, and fundamental principles governing their interactions. This architectural framework provides a comprehensive view of the system's design, detailing the flow and integration of various elements. Employing diverse viewpoints such as data, process, and deployment ensures a holistic consideration of all aspects.

On the other hand, design patterns offer reusable solutions to common challenges encountered during software development. While not constituting complete architectures, design patterns serve as adaptable templates to address specific design issues. These patterns, such as the singleton pattern, observer pattern, and factory pattern, encapsulate best practices that enhance maintainability, flexibility, and scalability in software systems. Each design pattern is tailored to a particular context, providing proven solutions that can be readily applied to diverse scenarios.

In essence, system architecture delineates the overarching structure of a software system, guiding its organization and integration, while design patterns contribute specific solutions to recurrent design challenges, fostering a more efficient and adaptable development process. Together, these elements form the foundation for creating software systems that are not only structurally sound but also capable of evolving with changing requirements and technological landscapes.

**Appendix**

**Team Name:**

SofTech Innovators

**Group Members:**

**Tyler Gettle:**

Tyler Gettle has two semesters of schooling in the Software Development program at the Southern Alberta Institute of Technology, he brings his knowledge of writing documents as well as his skills with HTML, JavaScript, CSS, and SQL. He also brings his ability to talk to others in a professional manner as well as his ability of public speaking.

**Saihaj Mann:**

Saihaj Mann is an eager student with experience working with many programming languages and frameworks. He is currently a student at SAIT. With an interest in CTF, he works to increases his skills in cybersecurity.

**Vinicius Melo:**

Vinicius Melo is a student enrolled in the Software Development program at SAIT, have acquired skills in several programming languages, including HTML, CSS, JavaScript, and Python. Alongside technical abilities, he has experience working in customer service roles, which has allowed him to develop strong interpersonal skills. He also had the opportunity to gain experience in the field of cybersecurity.

**Victor Odhiambo:**

Victor Odhiambo is a skilled Software Developer at Southern Alberta Insitute of Technology (SAIT), with a passion to solve software related tasks. with a strong foundation in coding languages such as Python, C#, HTML, CSS, JavaScript, and SQL. Including past customer service experience with other jobs he is capable of handling clients and coding development.

**Andrew Wahlers:**

Andrew Wahlers is an eager student at SAIT working towards his diploma in Software Development. He is bringing his experience with customer service in the IT industry, Programming languages including Java, Python, HTML, CSS, JavaScript, SQL, and his growing interest in cyber security.

**Team Goals:**

Our goal as a team is to make a properly designed and well-functioning website. We wish to gain knowledge on speaking with clients in a professional manner, as well as learning how to deal with issues and how to fix them while keeping our client in touch with all of it. We also want to gain more knowledge on HTML, JavaScript, CSS as well as learning how to integrate a database into the code and having them connect and keep information within using SQL. We also want to learn how to work within a team that was put together quickly, where we don’t really know each other's skills and how to utilize them to their fullest potential.

**Intellectual Property Statement:**

SofTech Innovators and our client in Kaan massage have come to an agreement that the client's website will be done by the end of our capstone project and during that time will regular update our client with new information that is discovered during the project.

**Team Roles:**

**Tyler Gettle: Creating the Document**

Tyler will be creating the document as he has knowledge on writing and formatting professional documents to the level that is required.

**Saihaj Mann: Creating the Diagrams**

Saihaj will be creating the use case diagram with their descriptions since our last assignment he had shown he has a better understanding of writing these types of diagrams.

**Vinicius Melo: Editing the Final Document**

Vinicius will be editing the final document when the document is finished as well as working with Tyler to create the document and finish it.

**Victor Odhiambo: Team Lead**

Victor will be our team lead for this assignment as he is the one who has talked face to face with the client and knows more about this project than the rest of the team. So, his knowledge on this will be invaluable.

**Andrew Wahlers: Creating the Diagrams**

Andrew will be creating the diagrams as he has shown that he knows what he is doing when it comes to creating these two types of diagrams as our last assignment has shown.

**Division of Labour:**

We will divide work between members fairly and equally. We understand that when dividing the work between members that some members may get easier sections while others get harder sections. We will have a system that when a member with the easier section is finished with their work, they will either work ahead on the next assignment or they will aid a fellow team member to get their work finished, and this will continue until all the work is completed. This allows for a fair system where no one must deal with more work than the others and we can avoid conflict where one member may feel as though they are doing all the work.

**Team Expectations:**

**Communication:**

We will be using Microsoft Teams as our primary communication format if we cannot speak to one another face to face.

**Meetings:**

Meetings will not have a weekly schedule and instead will be scheduled days in advance through Microsoft Teams where we can either meet online using a call through Microsoft Teams or in person. Meetings will typically take place at SAIT unless otherwise decided by the team.

**Collaboration Tools:**

If we cannot work together in person, we will be using Google Drive to work on assignments, or we will be using Microsoft Teams to stream what we are working on so that all team members can see exactly what is being worked on.

**Resolving Conflict:**

**Team Member Fails to Attend Meetings:**

If a team member fails to attend meetings that were scheduled in advance, they must have a proper explanation as to why they failed to attend. Otherwise, they will be talked with about the failure. If they continue to keep missing meetings, we will discuss this infraction with the instructor and have marks taken off assignments.

If a team fails to attend a meeting that was scheduled last minute, the member will not lose marks or be discussed as last-minute scheduling doesn’t allow team members to always reply right away or they may have plans already scheduled that day.

**Team Member Fails to Meet Deadlines:**

If a team member fails to meet deadlines for work, they will not be discussed, they will immediately lose marks on that assignment, as their failure meant that another team member who already had plenty of work and met the deadline had to cover for them which is unfair to them.

**Team Member is Doing Poor Quality Work:**

If a team member is not putting their all into the work, especially when it comes to working with the client, they will be talked with and will lose marks as another team member will have to fix their poor-quality work. This may lead to another infraction on the level of commitment this team member has to the work and the project.

**Team Member Fails to Communicate:**

If a team member fails to communicate with the rest of the team on what work they are doing or as to why they cannot communicate or do certain work, they will be talked to about their communication, and if they continue to fail to communicate, we will discuss this with the instructor and that team member will lose marks.

**Different Levels of Commitment to The Project:**

If a team member clearly shows they don’t have the commitment to the project as the rest of the team, they will firstly be talked to about this, and if their commitment still doesn’t improve, they will be expelled from the group.

**Personality Conflicts:**

If team members are conflicting through personalities, their behaviour will be discussed with the rest of the team, and then they will be talked to about this conflict and will have these members work with other members, this way they do not collide with one another. If the conflict continues, marks will be taken off each party involved, and if the conflict escalates or continues after mark deduction, all parties involved will be discussed with the instructor and this may lead to expulsion of team members.

**Team Member Feels They are Doing More Work Than Other Team Members:**

If a team member feels as though they are putting in more effort and work than the other team members, we will discuss this with the whole team, and if it's true. We will either all work harder more equally, or the rest of us will be losing marks, except for the member who was carrying the rest of our slack.

**Team Member is Doing Less Work Than Other Team Members:**

If a team member is doing less work than other team members, they will be discussed with and have them help the other team members, but if they refuse to aid the others, they will have marks deducted from them. If it continues and this leads to conflict, they will be expelled from the group.

**Team Member Expulsion:**

If a member does not meet our teams’ standards and continues to let us down when it comes to working on the capstone project, they will firstly be discussed between the other members to determine if it has come to expulsion, then it will be discussed with the team member at fault. We will get an understanding of where they are coming from and as to why they aren’t doing their work. If they can’t come up with a proper explanation as to why they are letting us down and they will fix it. we will discuss this issue with the instructor and have that team member removed from the group and project.

**Conclusion:**

To sum up, the SofTech Innovators team has started working on a project to create a massage booking system for Aisha Carrazo of Kaan Massage, one of our clients. The booking system that our customer now uses is antiquated and devoid of capabilities for user accounts, effective data management, and a contemporary user interface. By switching to a rebuilt website where users can create accounts, log in, manage their appointments, and receive notifications, our suggested solution seeks to alleviate these problems.

In order to accomplish the project's goals, Tyler Gettle, Saihaj Mann, Vinicius Melo, Victor Odhiambo, and Andrew Wahlers will each contribute their special talents and expertise. We are dedicated to creating a functional website that satisfies the needs and goals of our clients.

Our objectives include improving our problem-solving and client-communication abilities as well as becoming proficient in web development technologies including HTML, JavaScript, CSS, and SQL. 

We have set expectations for meetings, communication, and dispute resolution to promote productive collaboration. To collaborate effectively, our team will make use of resources like Google Drive and Microsoft Teams. We also have a well-defined framework in place for handling concerns pertaining to commitment, communication, quality of work, missed deadlines, and absences.

We will send our client regular updates on our work and keep them informed throughout this project. We are committed to creating a great website for Kaan Massage, and the success of our team will largely depend on how well we work as a team and maintain high standards. 

Along with the information and experience this capstone project will provide, we are excited about the chance to collaborate closely with our customer. In order to satisfy the needs of our clients, we are committed to overcoming any obstacles and offering an excellent solution.

**Signatures:**

A group of signature cards

Description automatically generated