**Product name:** FreeGap App

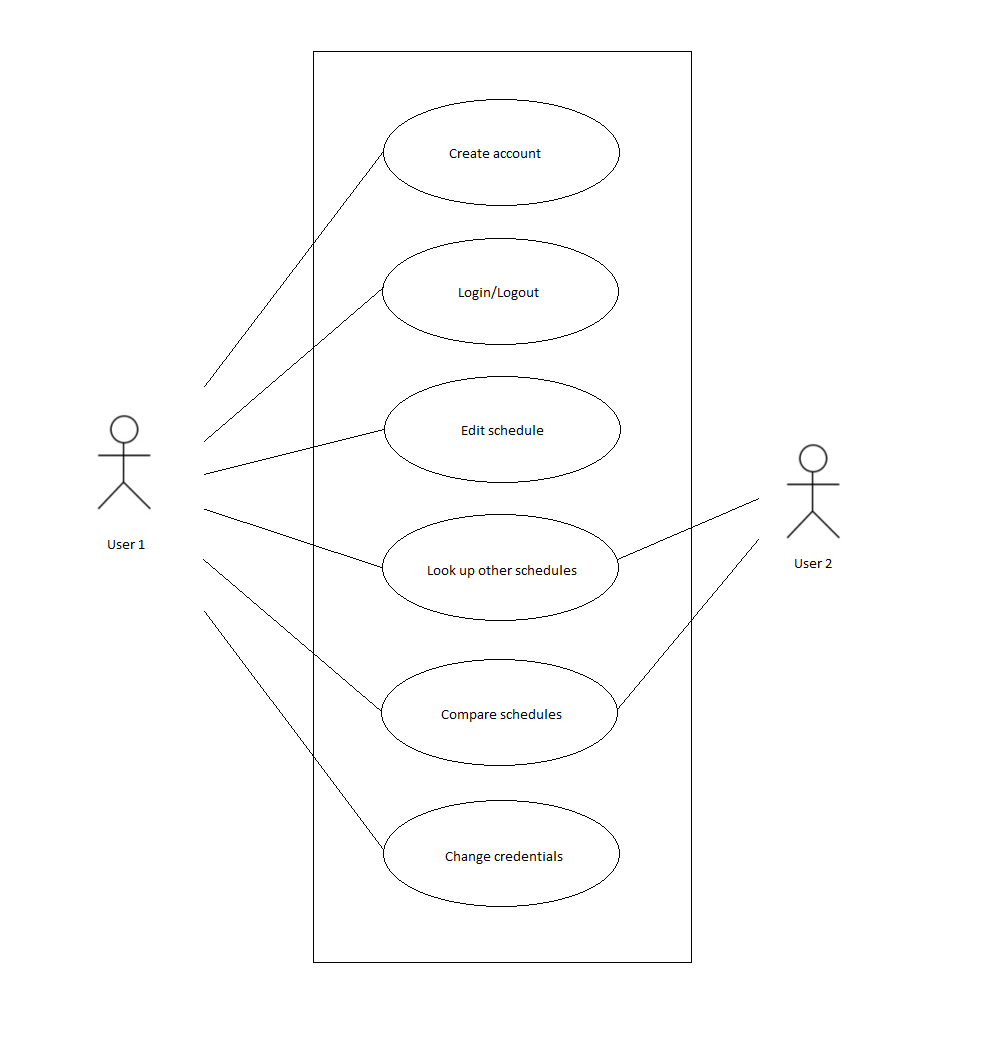
**Problem statement:** Many people have varying schedules and as a result, finding times when both parties are available may become quite a hassle. The FreeGap App provides a solution to reducing the amount of time it takes for a person to figure out times when they are available in compliance with another person’s schedule.

**Objectives** - This application shall provide a quick and efficient way to compare gap periods.  
 - This application must save users schedules.  
 - This application shall allow users to modify their schedules.

**Functional requirements** - A user shall be able to create an account. **LOCKED.**

- A user shall be able to sign in and out of their account, given the account has been made. **LOCKED.** - A user shall be able to edit their schedule, such as adding and removing events. **LOCKED.** - A user shall be able to look up another user’s schedule, provided that the other user has granted permission. **LOCKED.** - A user must be able to compare schedules with another user, provided that the other user has granted permission.  
- A user shall be able to change their credentials, such as their username, e-mail, and password.   
  **Nonfunctional requirements** -The application will be coded with Java backend. **LOCKED.**

-The application’s error messages will be displayed in English. **LOCKED.** -The application must run on Windows and Mac OS.  
 -The application shall compare a user’s schedule with another user’s schedule within ten seconds.

**Use Case Diagram:  
  
**

**Use Case Description 1:** Comparing Schedules

**Preconditions:**

1. The user enters a valid username and the correct password pertaining to their account.
2. Schedule is entered and stored in system; existing schedule is provided if the schedule has been entered before.
3. User searches for username of the person(s) that the user wants to compare schedules with and selects their username.
4. User must obtain a permission number if the user that is searched has their schedule set to private.
5. User must enter permission number to compare schedules and see the other user’s schedules if the other user(s) schedules are set to private.

**Trigger:** The user selects the “Compare Schedules” option.

**Primary Sequence:**

1. The system prompts the user for who they want to compare their schedule with.
2. The user is able to add multiple users to compare at once which will result in the system finding a time when all users entered are available.
3. The user confirms the users that he/she wants to compare schedules with.
4. The system displays the times where all users selected are available regarding their schedule.

**Alternative Sequences:**

1. The user has not entered a schedule into the system.

a. The system displays a message to the user.

b. The system prompts the user to enter their schedule.

2. The user’s schedule or the other user’s schedule has no periods of time where both parties are available/free.

a. The system displays a message to the user.

b. The system prompts the user that there is no period of time in which both parties are available/free.

3. The user has not entered a permission number when trying to compare schedules with another user that has their schedules set to private.

a. The system displays a message to the user.

b. The system prompts the user to enter the permission number in order to compare schedules.

a. If the user is unable to enter the permission number within one minute or the wrong permission number is entered, the system displays a message to the user.

b. The system prompts the user to request a permission number from the user(s) in which the user is trying to compare their schedule with.

**Post Conditions:**

The user is shown times in which the parties whose schedules were compared are available/free.

* The user gets to see when both/all parties are available.

**OR**

The user is shown times in which the parties whose schedules were compared are available/free.

* The user is informed with a message that there exists no times in which both parties are available/free based on the schedules entered.
* The user is informed with a message that they have not entered a schedule or the user(s) they are trying to compare schedules with have not entered a schedule.

**Use Case Description 2:** Adding an item to your schedule.  
**Preconditions:**

1. The user has an account.
2. The user is currently logged into their account.

**Trigger:** The user selects the “Submit” button.

**Primary Sequence:**

1. The user selects the “Add To Schedule” button.
2. The user fills in all requirements boxed for a schedule item. (Date, Time, Place, Recurrence)
3. The app returns to your schedule with your newly added event.   
   **Alternative Sequences:**
4. The user clicks “Add to Schedule” but is not logged in
   1. The app prompts the user to log in
5. The user tries to add an event in a past period of time.
   1. The app prompts the user to put a future date/time
6. The user adds an event at the same time as another event.
   1. The app merges two events into one block on the schedule.

**Post Conditions:**

1. The user is shown their schedule with newly added item.

**Use Case Description 3:** Making an Account

**Preconditions:**

1. The user access the main page of the website.

**Trigger:** The user selects the “Create an Account” window on the main page.

**Primary Sequence:**

1. The system prompts the user for a username, password, and email address.
   1. Unaccepted fields are outlined in red
2. The username is added to the system.
   1. Message to pick new username pops up
3. The password is encrypted and added to the system
   1. Message to use different email pops up
4. The app send an email to the email address to confirm account signup
   1. Message prompts user to check email and confirm
5. User confirms sign-up in email and account is created with blank schedule.
   1. An email that was inputted will receive the confirmation and a link that will lead to the “create new schedule” page.

**Alternative Sequences:**

1a.) Username already exists in the system

2a.) Email already exists in the system

3a.) Password is too short/ does not contain required letters, numbers, or symbols.

4a.) Does not confirm email address

**Post Conditions:**

* The user’s account is created and they can sign in

**OR**

* The user’s account is not created