

Team 2: Habit Tracking

Team members: Connor Wilding Leonie, Duy Vu, Evan Kyle Jensen, Majid Khoshakhtar, Ton That Quynh Thong, Vivek Gandhi, Zaid Mohammed Sikander

Formal Software Specifications

- Functional Requirements:

This section focuses on the system features and the major services provided by the application. Details of all requirements necessary for development are stated below. This provides a framework for developers to create the product in accordance with requirements.

ID	Functional Requirement
HT-FR-01	User can create an account to sign into the mobile application
HT-FR-02	User can sign into their previously created account on the application database
HT-FR-03	User accounts data must be stored onto the system database
HT-FR-04	User can create or remove their habits on the application interface
HT-FR-05	User can input data for the habits they want to track

HT-FR-06	User can track their habits through the dashboard
HT-FR-07	Application should be able to generate visual analytics, in forms of graphs, showing the user habit tracks
HT-FR-08	Application should display the generated analytics to the user
HT-FR-09	User can change the statistics displayed from weekly to monthly and vice versa
HT-FR-10	Application should be able to integrate with devices outside of local device such as Apple Watch and other fitness tracking devices and software in order to passively read data from these other devices and use the data to track a habit e.g time heart rate elevated via apple watch
HT-FR-11	Application should be able to read data from other applications on local device such as Reminders app on iphone, screen time tracker on mobile devices in order to acquire data for a habit when the device is used
HT-FR-12	Application should show advertisements to the user

HT-FR-13	The user can upgrade their account for an ad-free experience
HT-FR-14	The user should be able to pay for the premium ad-free account via the application
HT-FR-15	The application should proceed the payment process promptly
HT-FR-16	Users can choose monthly or annual billing from the available subscriptions
HT-FR-17	The application should store the data on local device storage when the user is offline
HT-FR-18	The application should push the locally stored data to the cloud database when internet is available
HT-FR-19	The user can provide feedback of the system through the application
HT-FR-20	The user can tweak the settings to best suit them through the application settings menu
HT-FR-21	The user can log out of their account
HT-FR-22	User can share their habits with other people through email

HT-FR-23	Application can be receive software updates remotely from IT team
----------	---

- Non-functional: Requirements

This section has its focus on the software quality attributes of the Habit Tracking application being developed. The performance, safety and security of the application is discussed.

2.1. Performance

ID	Non-Functional Requirement
HT-NFR-01	The initial load time of the application should .2 - .4 seconds
HT-NFR-02	The transition time between different screens should be 0.5 seconds
HT-NFR-03	The communication between the application and cloud database should be take less than 1 second, allowing immediate and efficient storage and retrieval of data
HT-NFR-04	The payment procedure should continue promptly, allowing users to know what is going on and avoid frustration

2.2. Security

ID	Non-Functional Requirement
HT-NFR-05	User data, including email, password, habits and payment information should be SHA-2 encrypted
HT-NFR-06	Habit Tracking application can read from linked fitness/health tracking devices.
HT-NFR-07	User sign in procedure should be protected via two-factor authentication
HT-NFR-08	The data communication between supplement devices should be one way, <i>only</i> allowing Habit Tracking application to read data from other devices and applications to the Habit Tracking application.
HT-NFR-9	The communication between application and system database will be over the internet, through a locally secured Wi-fi or data connection

2.3. Availability

ID	Non-Functional Requirement
----	----------------------------

HT-NFR-10	The application should be available 99.999% of the time
HT-NFR-11	Scheduled maintenance can be done after notifying the user through push notification at of-peak hours, and should not last longer than an hour
HT-NFR-12	The application should handle 100,000 concurrent users, making it available for all users at the same time

2.4. Legal and Licensing

ID	Non-Functional Requirement
HT-NFR-13	The application should have a legal license agreement with the supplement devices, to access user data

2.5. Transparency

ID	Non-Functional Requirement
HT-NFR-14	The application should have a transparent payment process, allowing users to track

	the transaction history and no hidden charges should be involved
--	--

2.6. Reusability

ID	Non-Functional Requirement
HT-NFR-15	The application should make use of existing software components and assets. The software should be reusable, allowing it to be integrated into other applications, as a feature, in future

2.7. Usability

ID	Non-Functional Requirement
HT-NFR-16	The application should be simple and user friendly
HT-NFR-17	The application should provide a tour of features upon first time use
HT-NFR-18	The application will be cross platform, but use native platform design and theme

HT-NFR-19	No prior or specific knowledge/ skill should be required to use the application
-----------	---

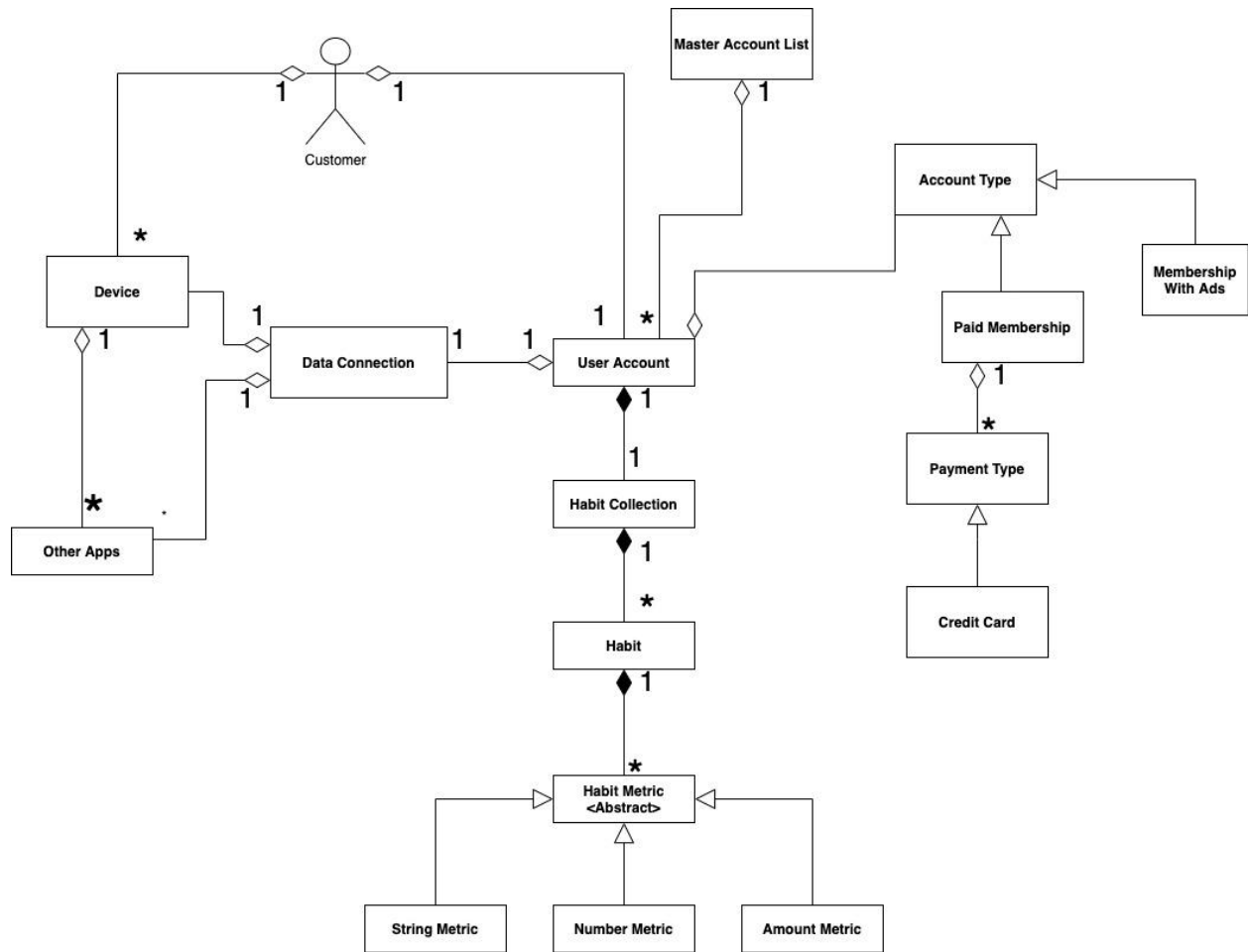
2.8. Accessibility

ID	Non-Functional Requirement
HT-NFR-20	The application should allow one click analysis and tracking of user habits

2.9. Maintainability

ID	Non-Functional Requirement
HT-NFR-21	The application should allow expansion of features with changing customer demands
HT-NFR-22	The application should make use of Agile development techniques to cope with changing environment
HT-NFR-23	Frequent updates should be released to improve user experience and hence the product life

Domain Diagram:



Use Case Description:

UC 1 : Register an account

Goal : Create account for new user

Actor(s) : Anyone

Preconditions : User has the habit tracking app or program on mobile device

Trigger : User selects Create Account

Main Scenario :

1. Program checks internet connection
2. Program loads the account creation page
3. Account creation page prompts user to enter information
 - a. Required information - Username, password, email
 - b. Optional information - Phone number, backup email
4. System checks if username or email have already been used
5. System creates the account and locks it
6. System sends user a confirmation email
7. User accepts the confirmation email and account is unlocked

Alternatives:

- No Internet connection
 - Display note about requiring internet for account creation
- Username or email already exists
 - Display small notification about duplicates and prompt change
- Email never confirmed
 - Account remains locked for 7 days then removed.

End Condition : Users account is created

UC 2 : Check Recent Data

Goal : Run algorithm and produce habit results

Actor(s) : Anyone with an account

Preconditions : Has data either in storage or uploaded to server and has an active account

Trigger : User clicks check habits

Main Scenario :

1. Program sends request to server to execute the habit algorithm
2. Server executes habit algorithm
3. Server stores data for future use (improving algorithm or if they click again without data update)
4. Server returns results from habit algorithm
5. Program displays data to the user on a specific page

Alternatives:

- Algorithm triggered by upload to server
- If the algorithm was triggered by upload it will not display the data but hold it

End Condition : Habit algorithm results are displayed

UC 3 : Update Data

Goal : Upload data since last server upload to server

Actor(s) : Anyone with an account

Preconditions : Has data on phone different from data stored on server

Trigger : User clicks upload

Main Scenario :

1. Program checks internet connection
2. Program states the size of the upload
3. User confirms upload
4. Server pulls data from phone based off last recorded upload date
5. Program displays progress bar page
 - a. When complete it either vibrates phone or displays a message to the desktop if user is not in program

Alternatives:

- No internet connection
 - Display error message and ask for user to get to location with internet

End Condition : Servers data is now up to date with users data and ask if user wants to trigger U2

UC 4 : Display Data

Goal : Display habit tracking algorithms data to user

Actor(s) : Anyone with an account

Preconditions : Has data on server and user is logged in

Trigger : User clicks habit tracking data

Main Scenario :

1. Program pulls server data
2. Program displays the data to the user

Alternatives:

- Data mismatch
 - If different prompts U3 followed by U2

End Condition : User is provided a page visualizing the habit tracking data

UC 5 : Add habit

Goal : Add pre planned activities to the schedule as to not have the algorithm consider them

Actor(s) : Anyone with an account

Preconditions : Has the app installed, has made an account and user is logged in

Trigger : Clicks the add activities button on manage algorithm page

Main Scenario:

1. Program checks internet connection
2. Program displays a calendar of the current month
 - a. Options to move between months
3. User enters information about Habit
 - a. (date, time, repeated or not, short description, exclude from algorithm)
4. Program takes data and uploads it to server

Alternatives:

- No internet connection
 - Display error message and ask for user to get to location with internet

End Condition : User is informed of date saved, and algorithm excludes specific times if notified

UC 6 : Delete habit

Goal: Delete habit on application

Actor(s) : Anyone with an account

Preconditions: User has app installed, has made an account and user is logged in

Trigger: User clicks the delete habit button

Main Scenario:

1. Program checks internet connection
2. Program asks user if they want to delete all previous data of habit or just delete habit from being tracked
- 3.
4. System displays habit was successfully deleted

Alternatives:

1a) No internet connection

1a1) Display error message and ask for user to get to location with internet

End Condition: Habit is deleted from user account

UC 7 : User shares data through email

Goal: planned activities can be shared through mail if you want to tell or share it with someone.

Actor(s) : Anyone with an account

Preconditions: User has app installed, user is logged in and has habits on account

Trigger: Clicks the share activities button on the manage algorithm page and will provide an option to share via email option.

Main Scenario:

1. Program checks internet connection
2. Program displays a calendar of the current month
3. Options to move between months and select habits user wants to share
4. User Habit information will be shared via email address. (date, time, repeated or not, short description, exclude from algorithm)
5. Program takes data and uploads it to the server and after on send it via email to some other user having this app.

Alternatives:

1a) No internet connection

1a1) Display error message and ask for user to get to location with internet

End Condition: User is informed that its data is shared via email.

UC 8: Account Subscription:

Goal: User can get rid of ads by account subscription.

Actor: User, IT Department

Preconditions: user has free version of the app installed which has ads displayed to user and user is logged in

Trigger: User clicks "Upgrade to ad-free version"

Main Scenario:

1. User is prompted to enter payment information via credit/debit card or paypal
2. Payment info is validated
3. User is displayed "upgrade confirmed" screen

Alternatives:

1a) No internet connection

1a1) Display error message and ask for user to get to location with internet

2a) Invalid payment info

2a1) system displays error and lists of invalid fields

2a2) user corrects fields until system validates them

End Condition: User is informed that its subscription is done and account will be ad free.

UC 9: Logout:

Goal: User can log out his/her account.

Actor(s) : Anyone with an account

Preconditions: Customer is logged into an account

Trigger: User can logout from account.

Main Scenario:

1. User Presses "logout" button
2. Program checks status of user editing profile
3. App displays "logout?" with a yes or no button
4. User presses the yes button and is logged out from the account.

Alternatives:

3a) user is editing unsaved work

3a1) User is prompted to save work or discard work

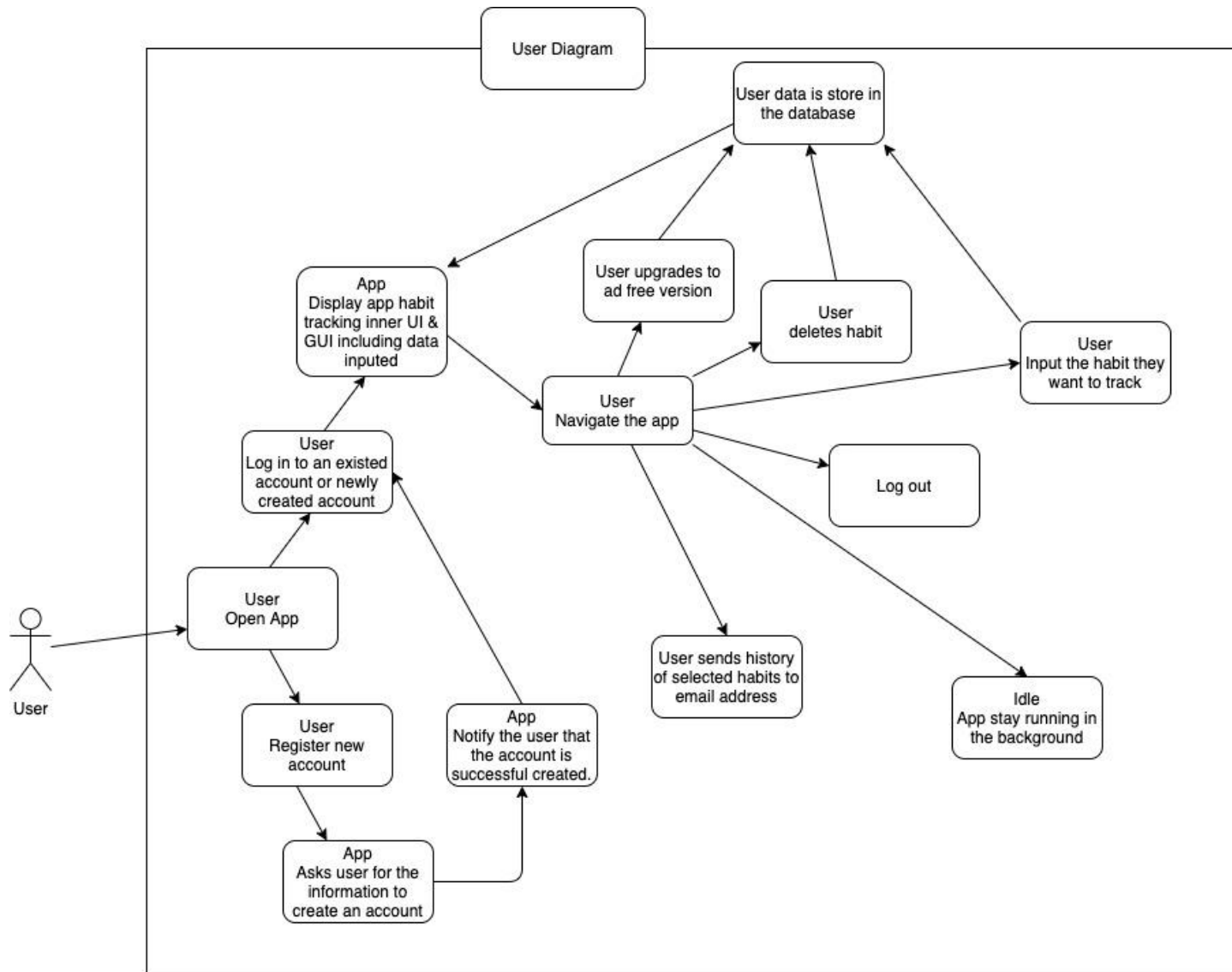
3a1a) User presses save work and work is saved

3a1b) user presses discard and work is not saved

4a) User clicks no, is returned to original page

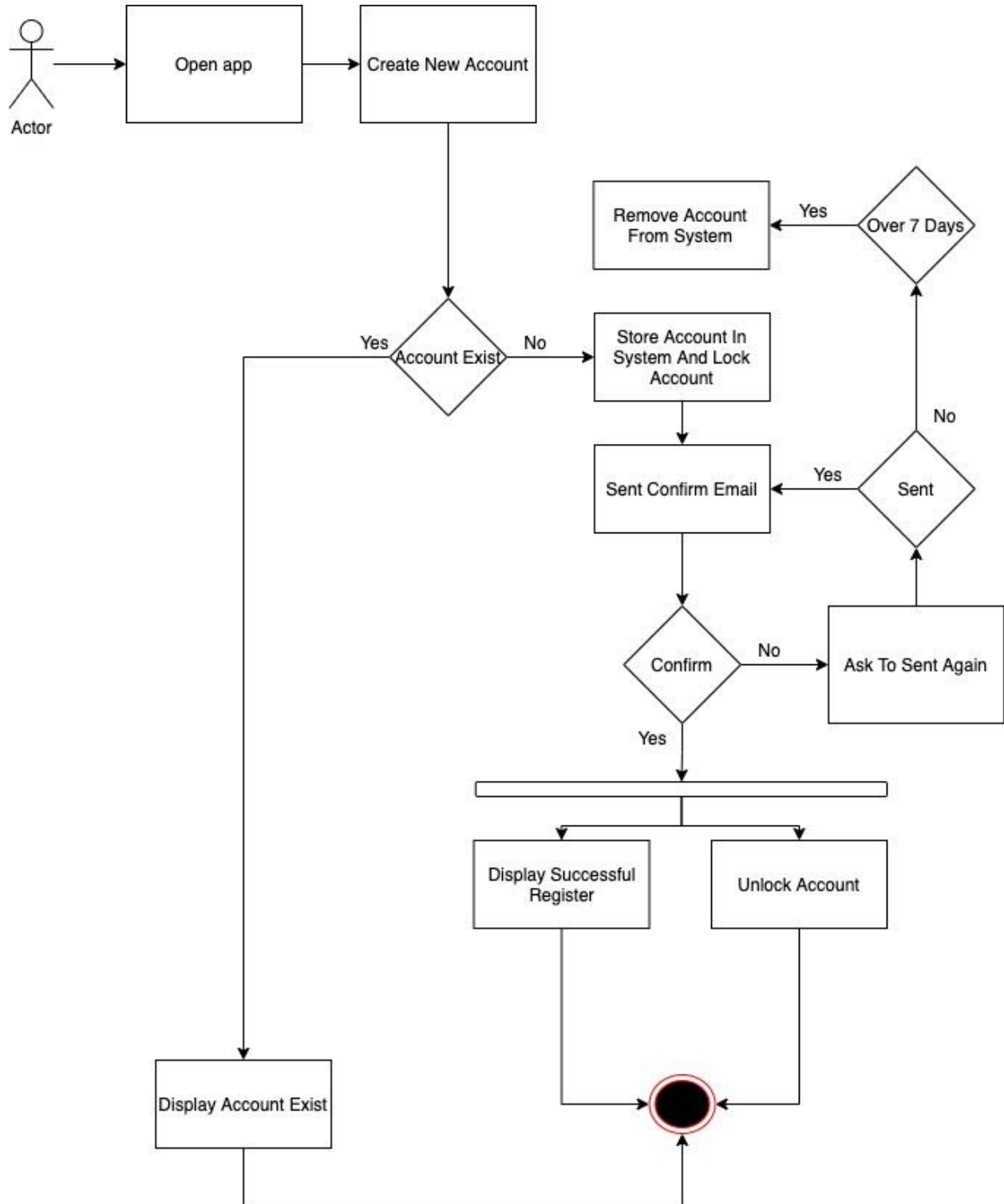
End Condition: User is logged out of the account.

Use case Diagram:

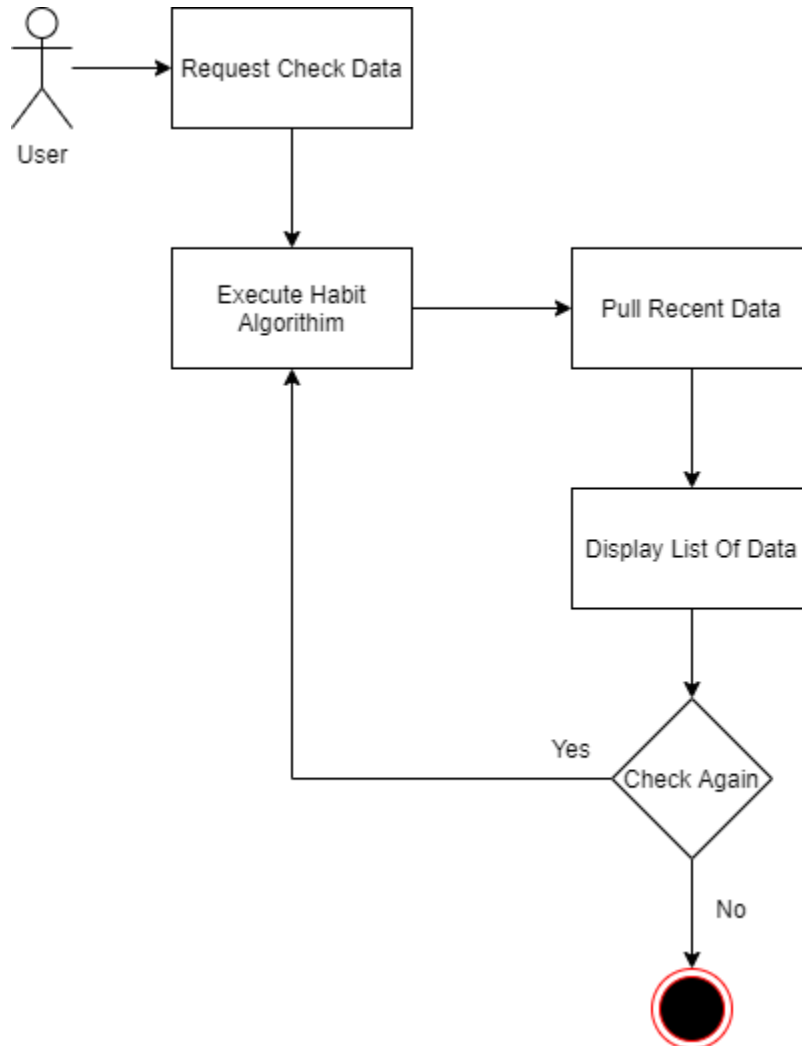


Activity Diagrams:

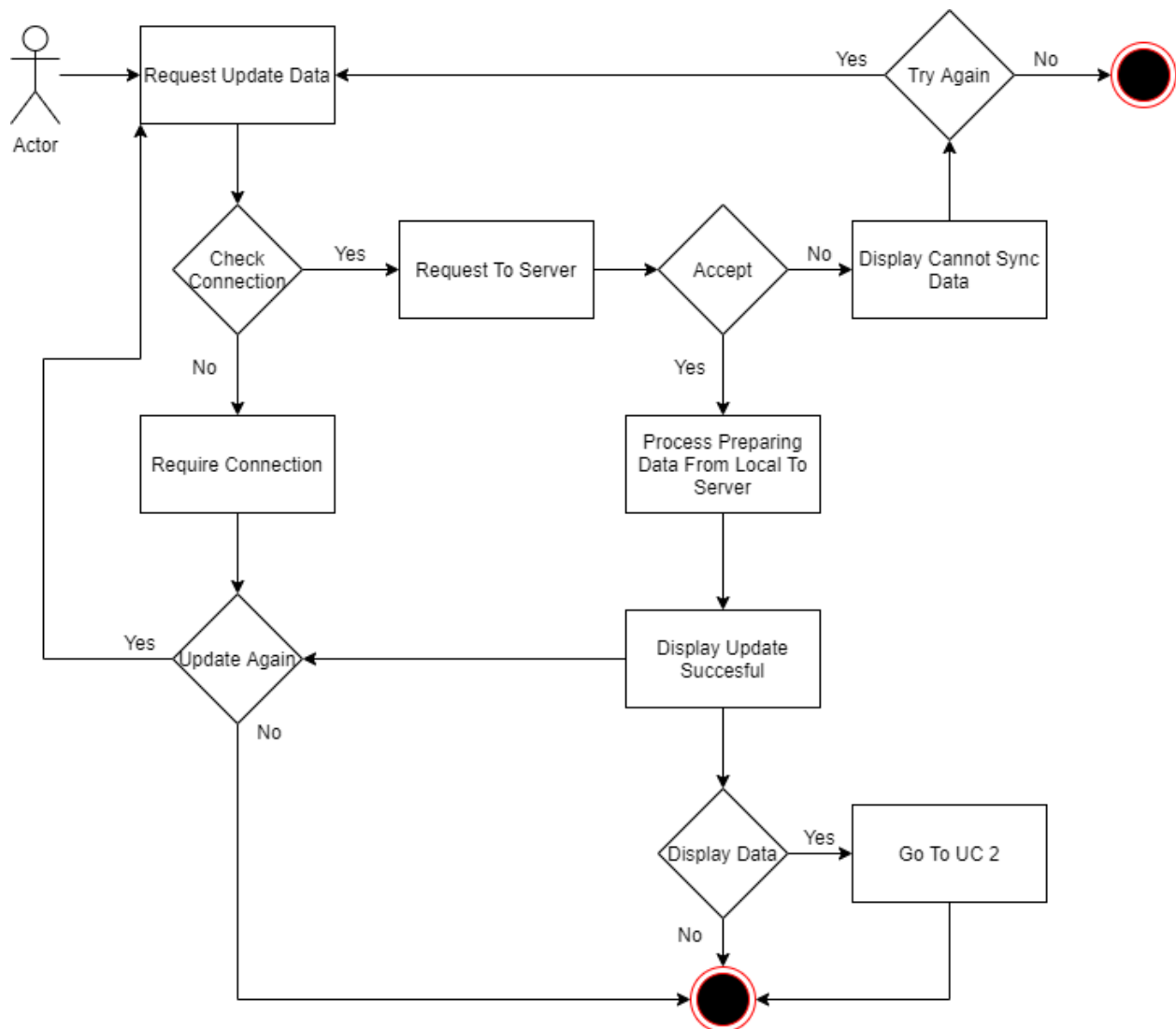
Activity Diagram of UC 1: Register an account



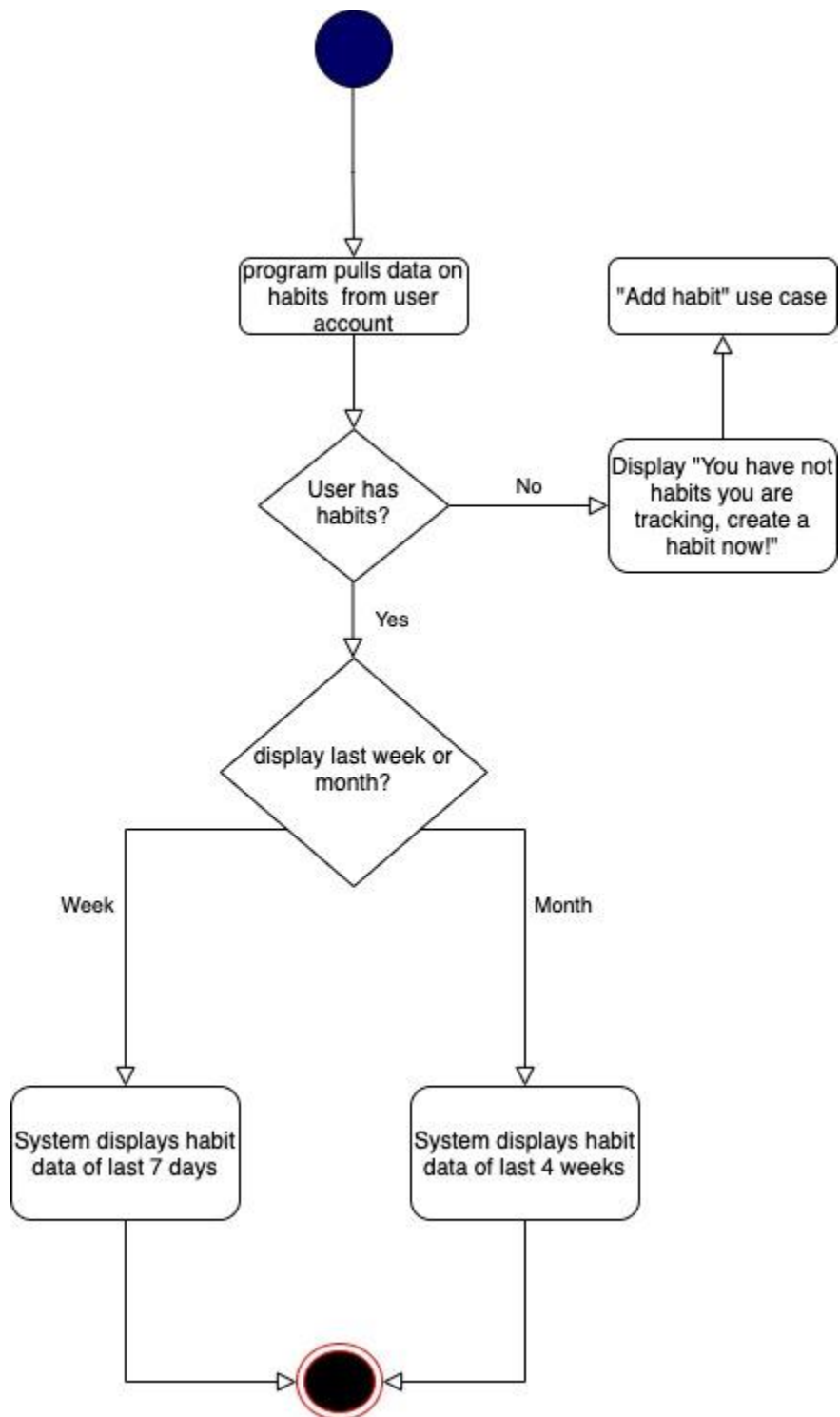
Activity Diagram of UC 2: Check Recent Data



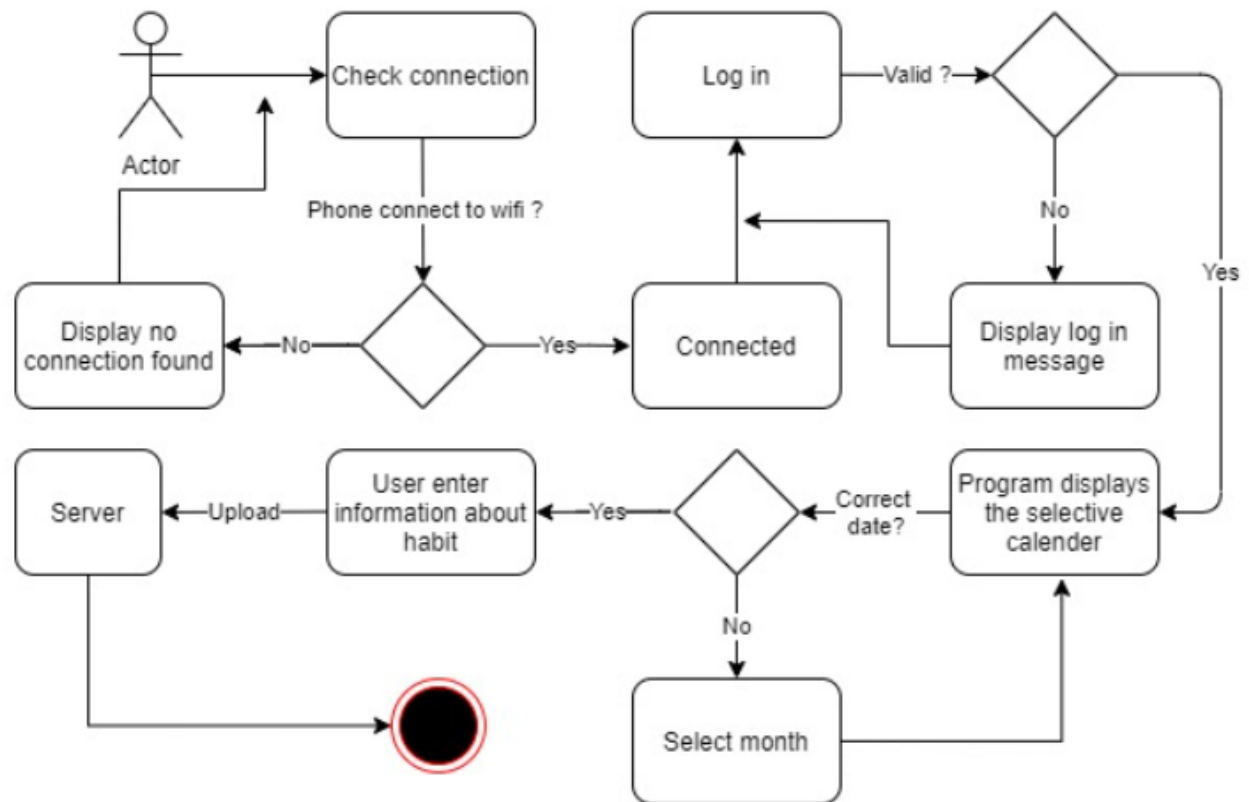
Activity Diagram of UC 3: Update Data



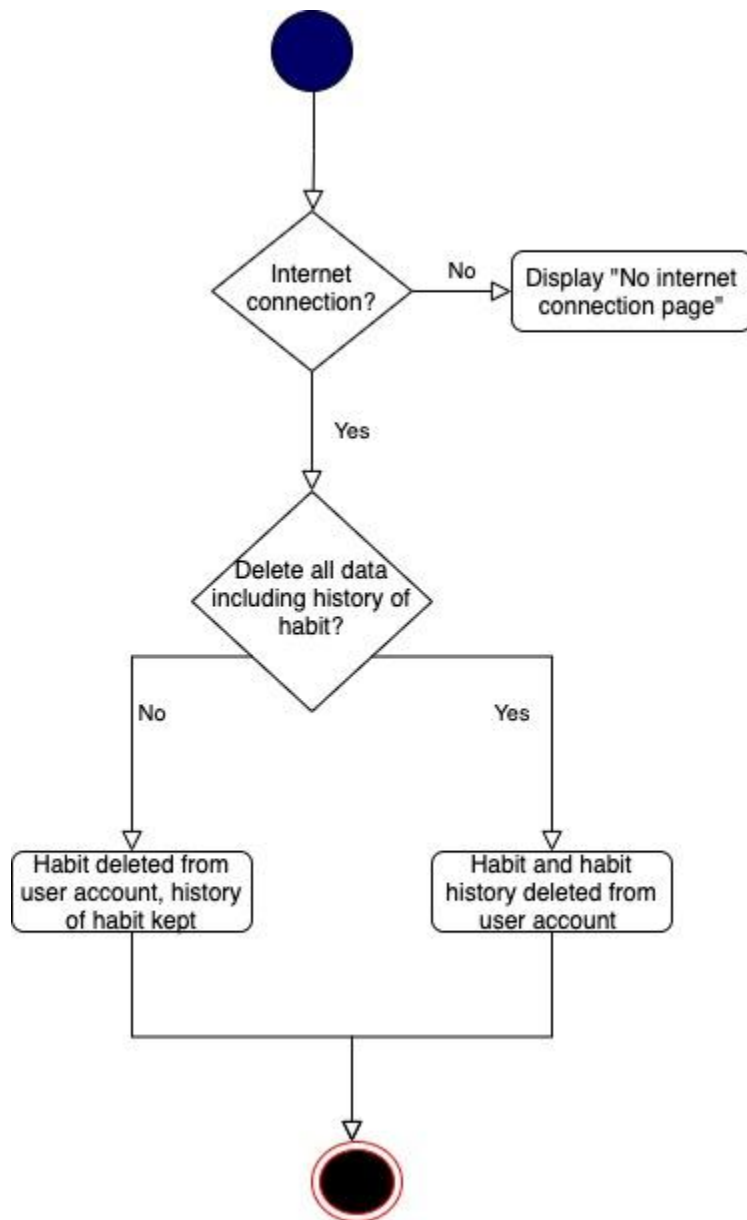
Activity Diagram of UC 4: Display Data



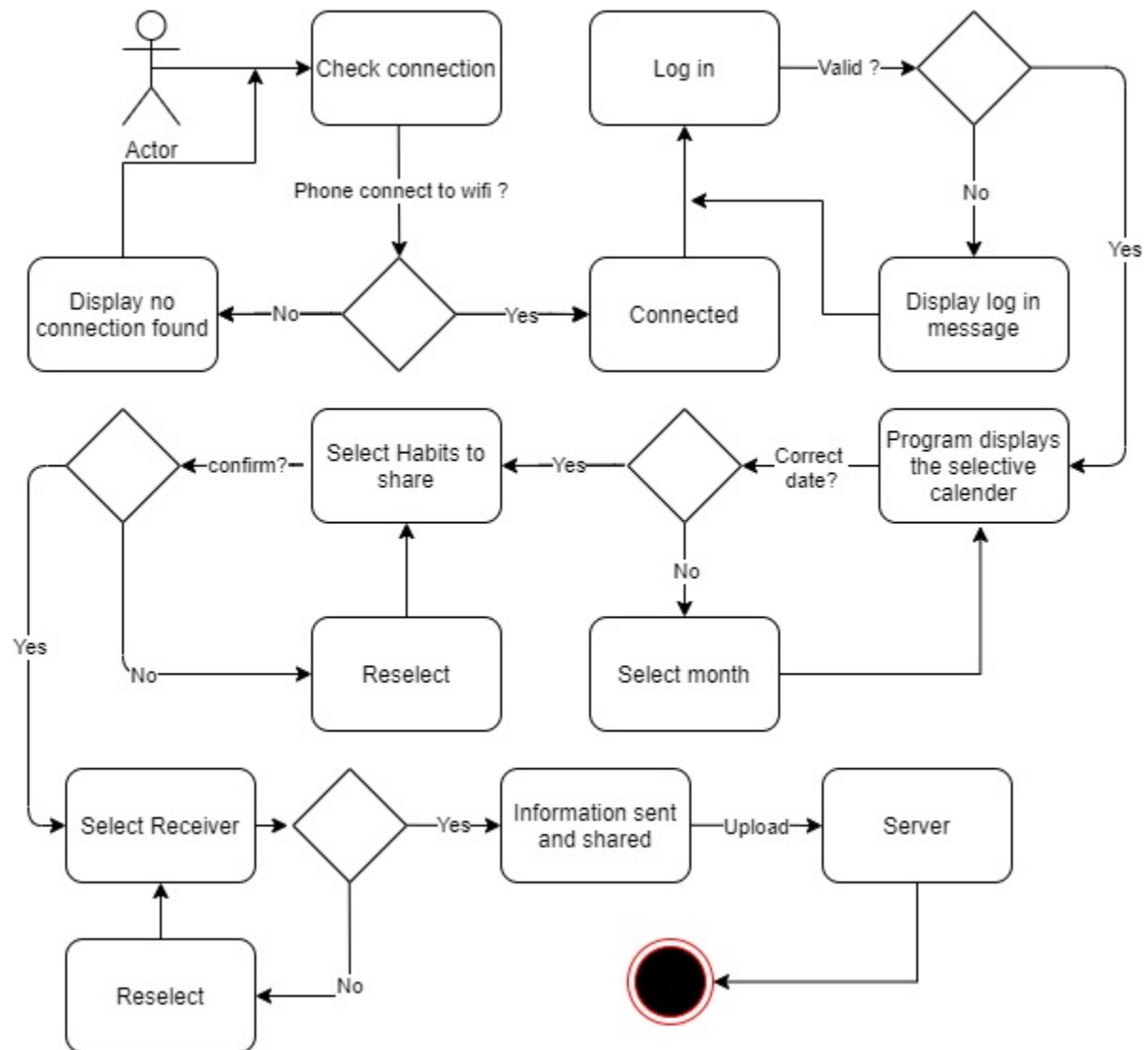
Activity Diagram of UC 5: Add Habit



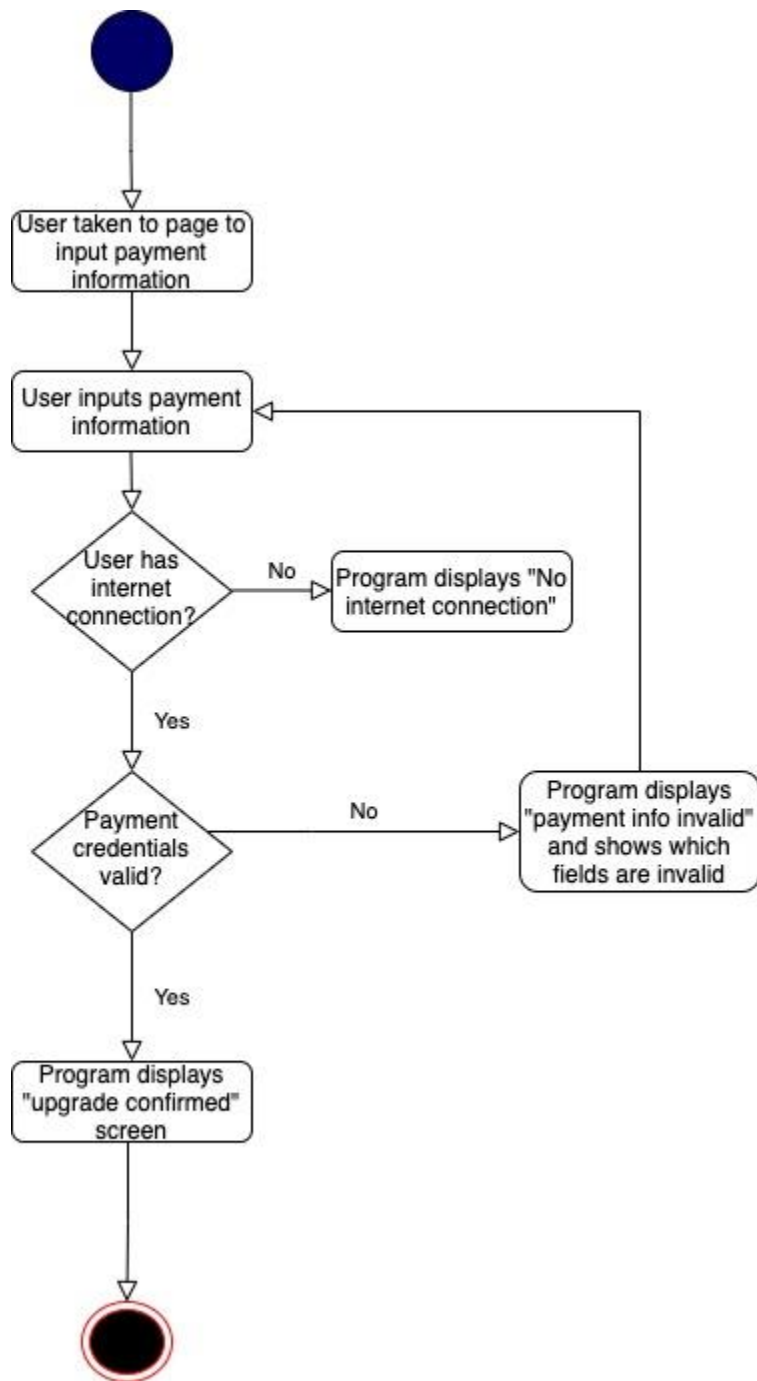
Activity Diagram of UC 6: Delete Habit



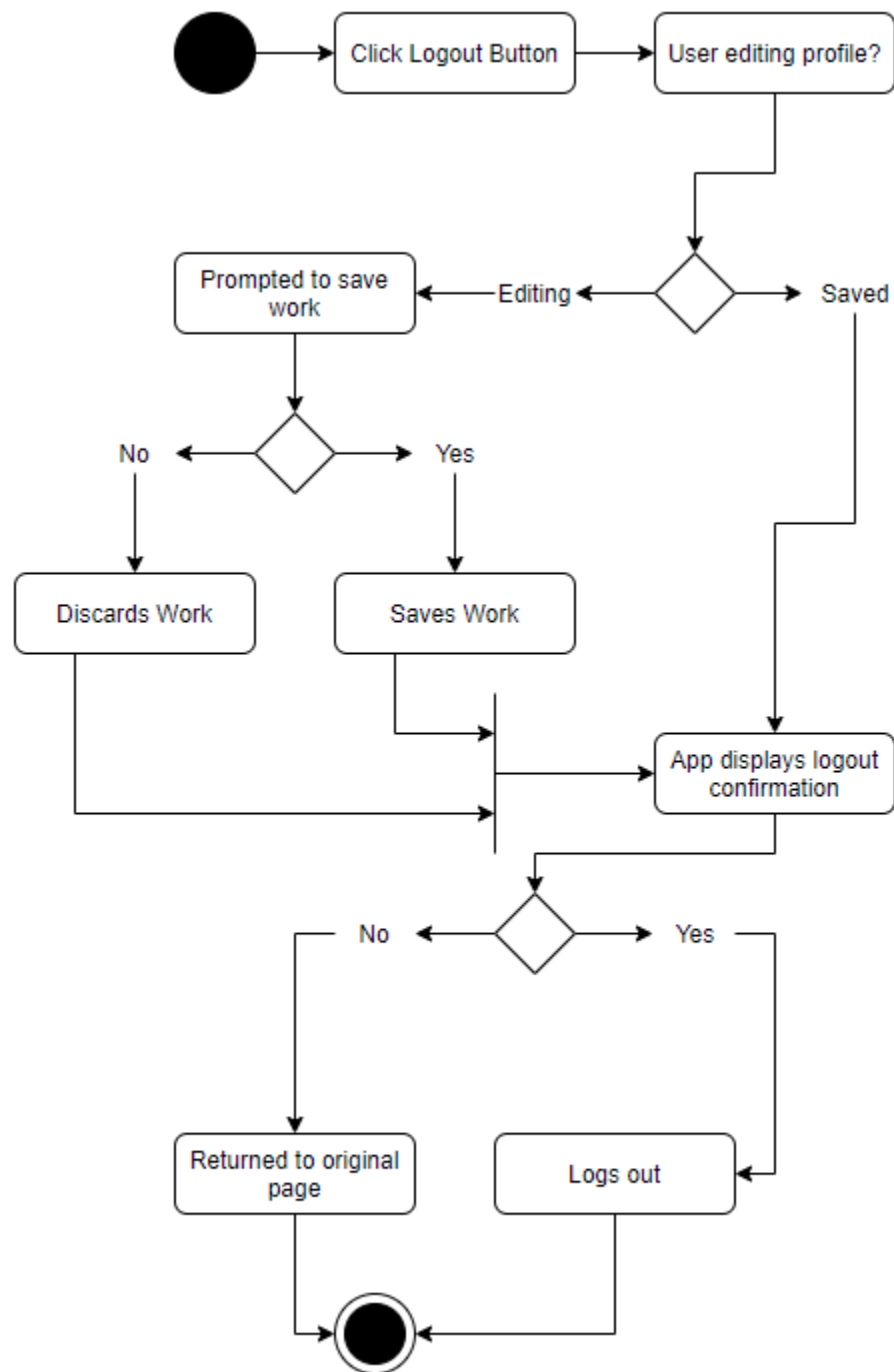
Activity Diagram of UC 7: User Shares Data Through Email



Activity Diagram of UC 8: Account Subscription

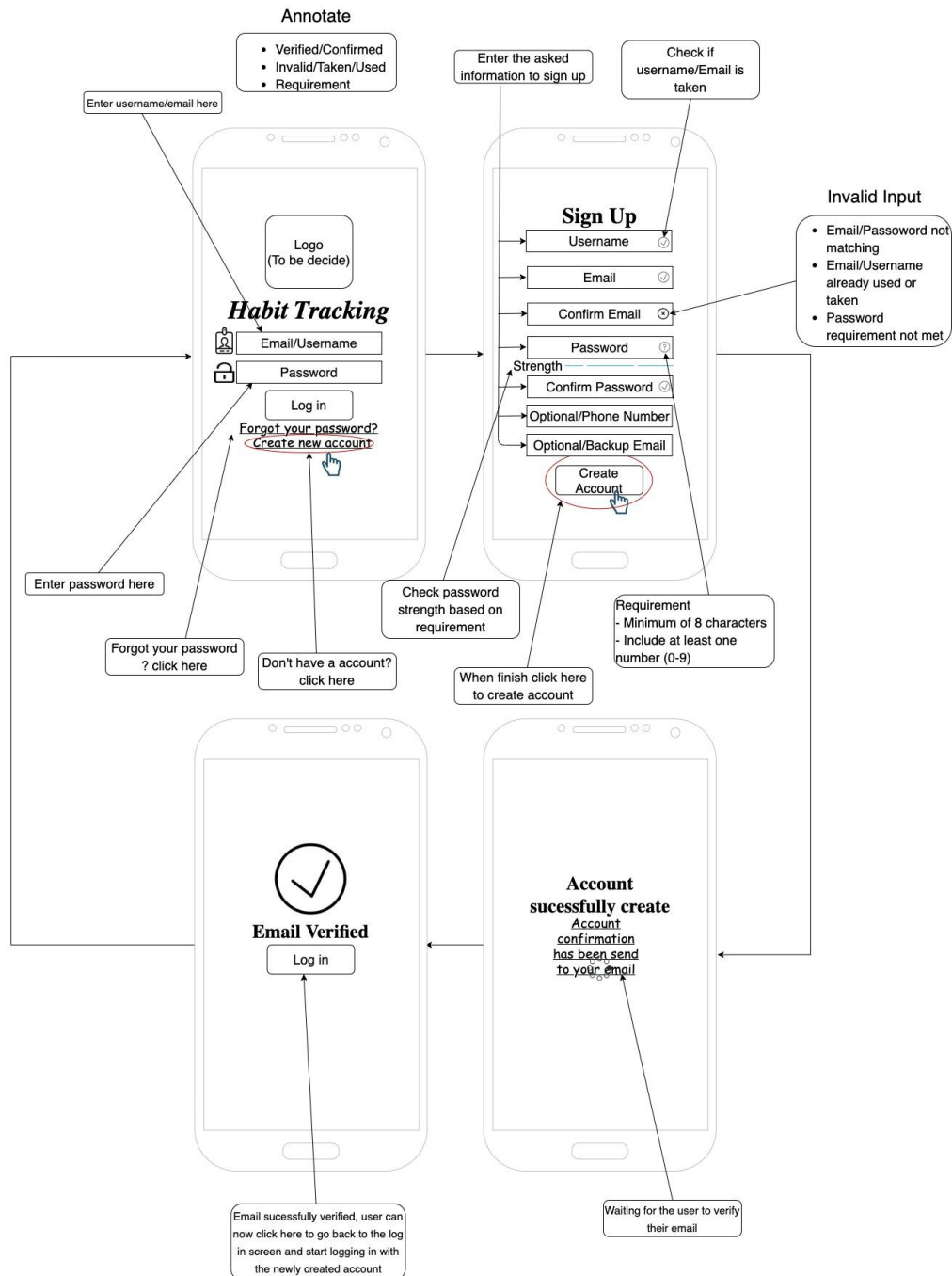


Activity Diagram of UC 9: [Logout](#)

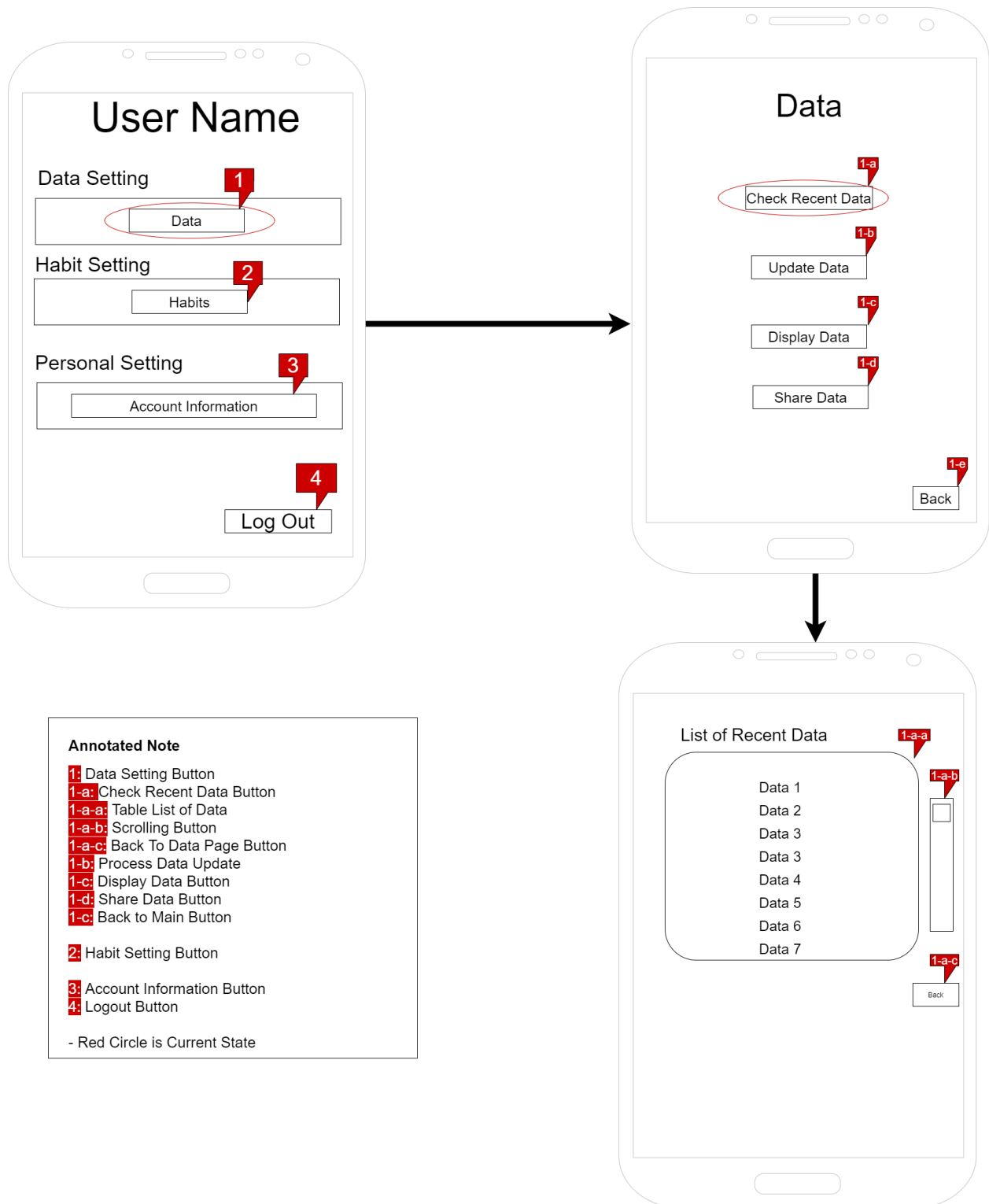


Wireframe:

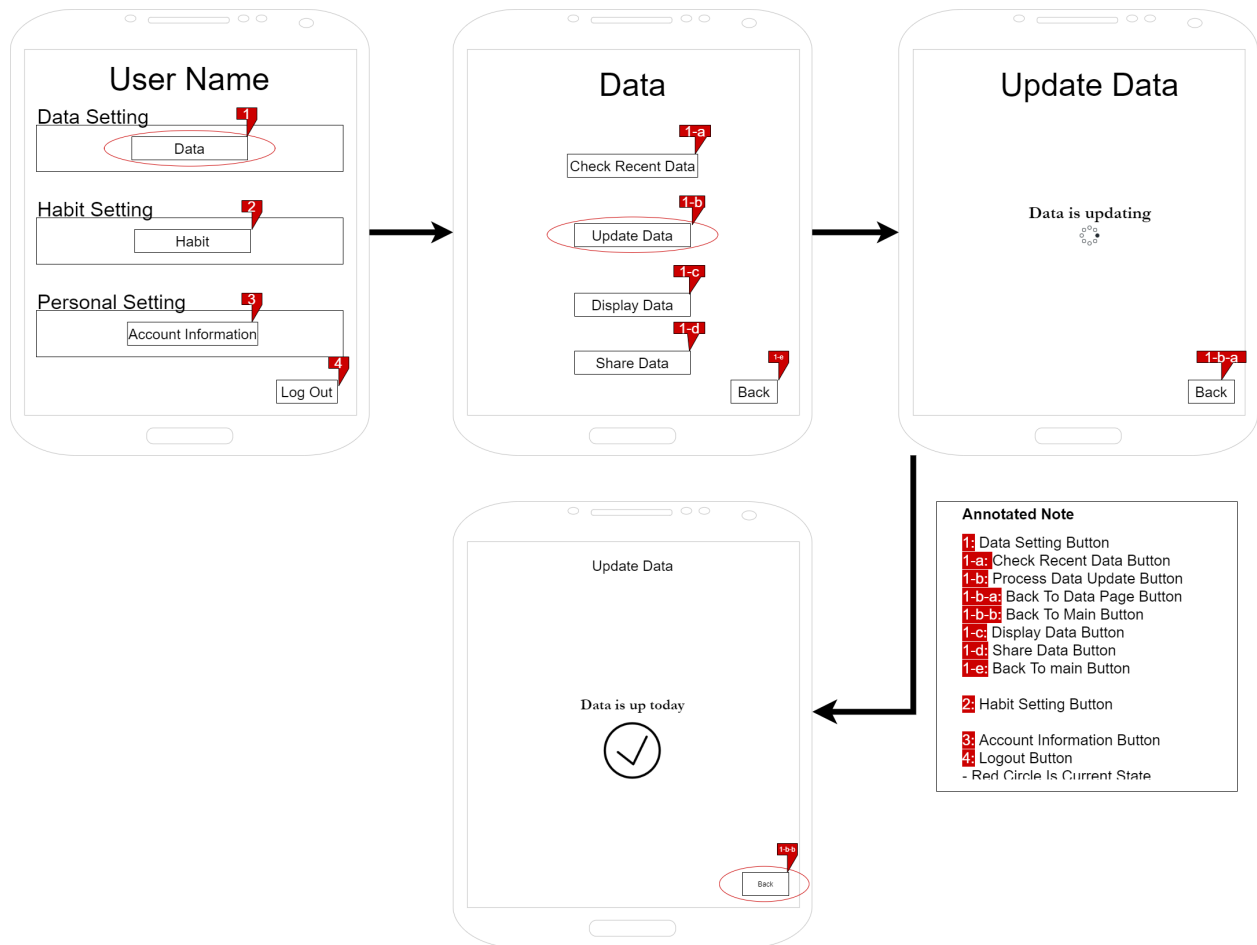
Wireframe of UC 1: Register an account



Wireframe of UC 2: Check Recent Data

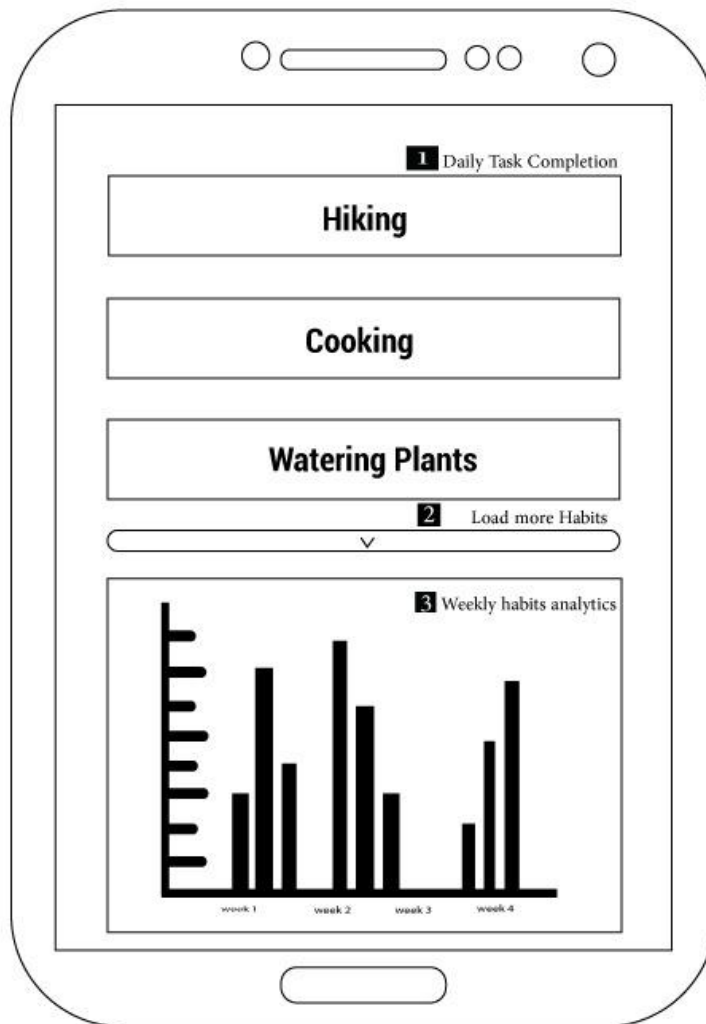


Wireframe of UC 3: Update Data

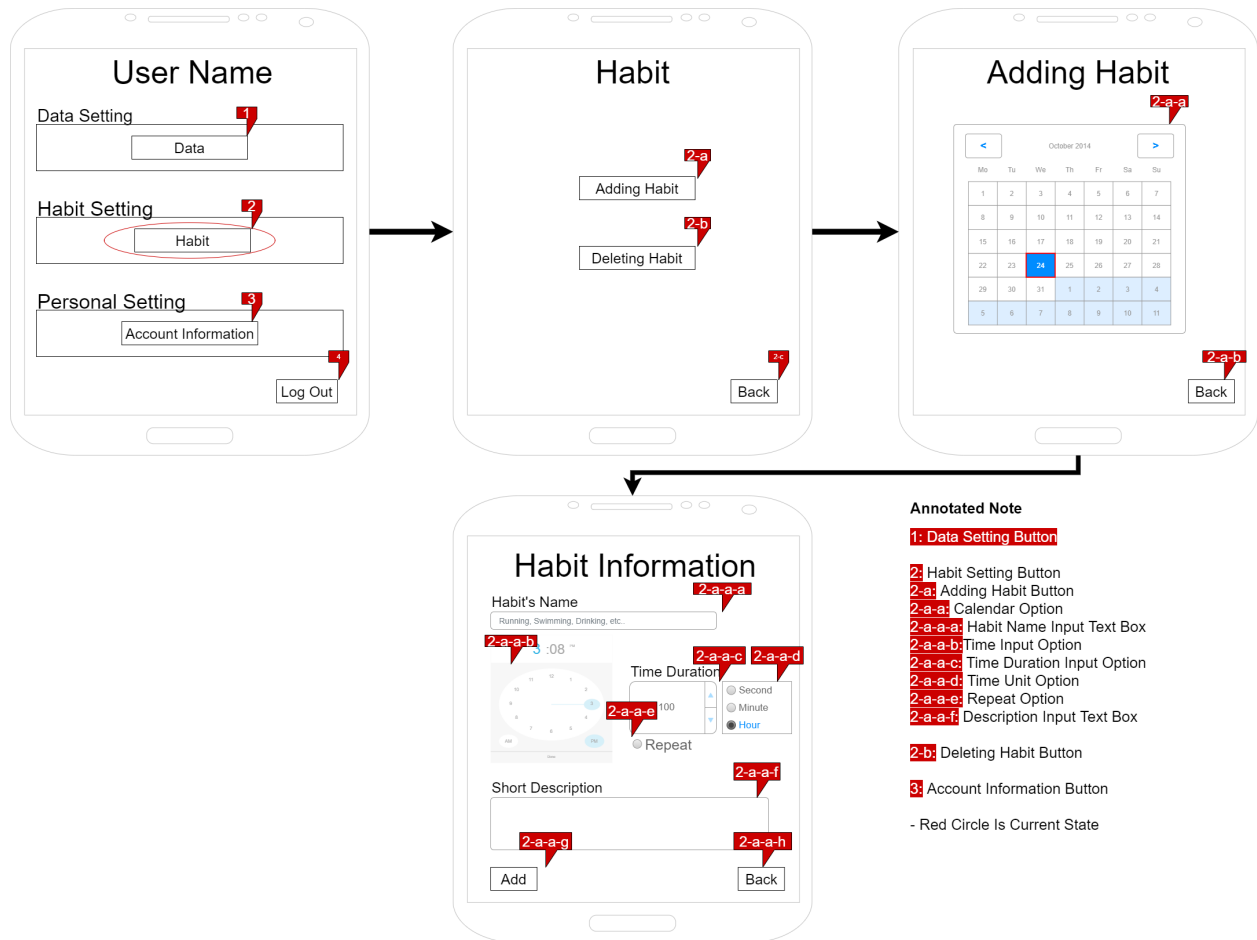


Wireframe of UC 4: Display Data

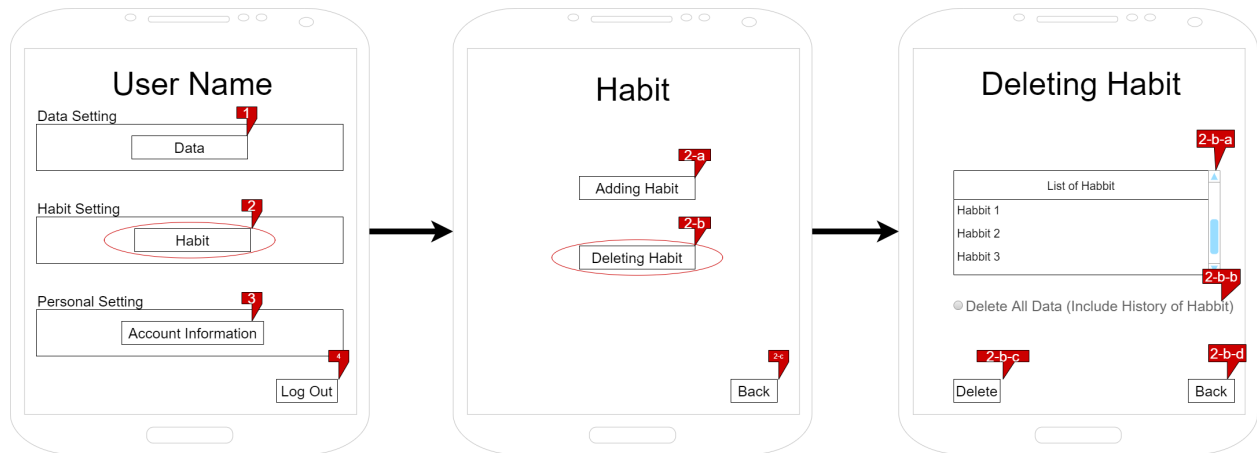
The Display data screen shows the data entered about every added habit, it notifies the user about the daily completion of the tasks. The analytics graph shows the weekly report on each habit. The users can keep track of their progress on each habit and how much time they spent on each of them.



Wireframe of UC 5: Add Habit



Wireframe of UC 6: Delete Habit



Annotated Note

1: Data Setting Button

2: Habit Setting Button

2-a: Adding Habit Button

2-b: Deleting Habit Button

2-b-a: List of Habbit

2-b-b: Delete All Data Option

2-b-c: Delete Button

2-b-d: Back To Habit Button

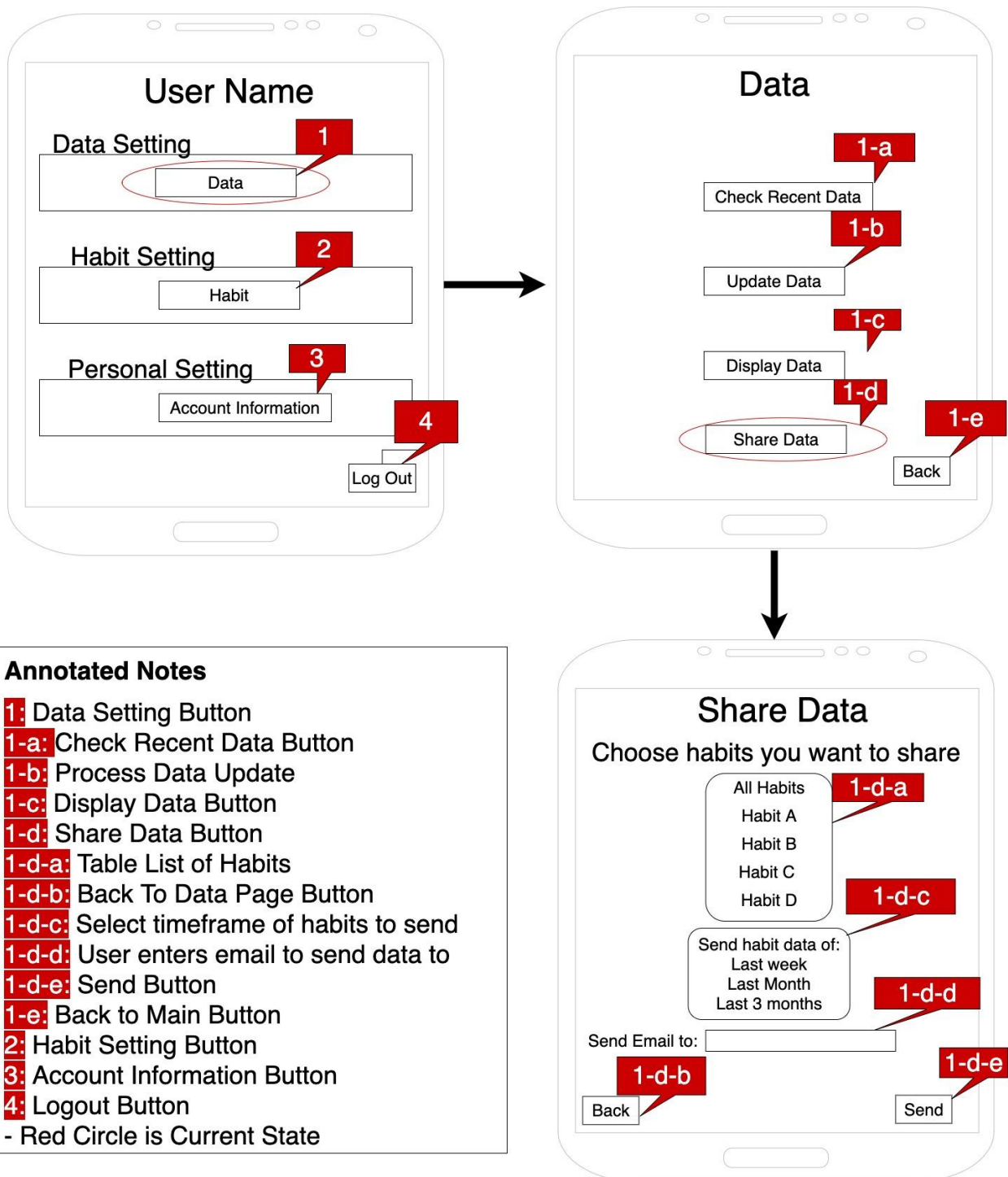
2-c: Back To Main Setting Button

3: Account Information Button

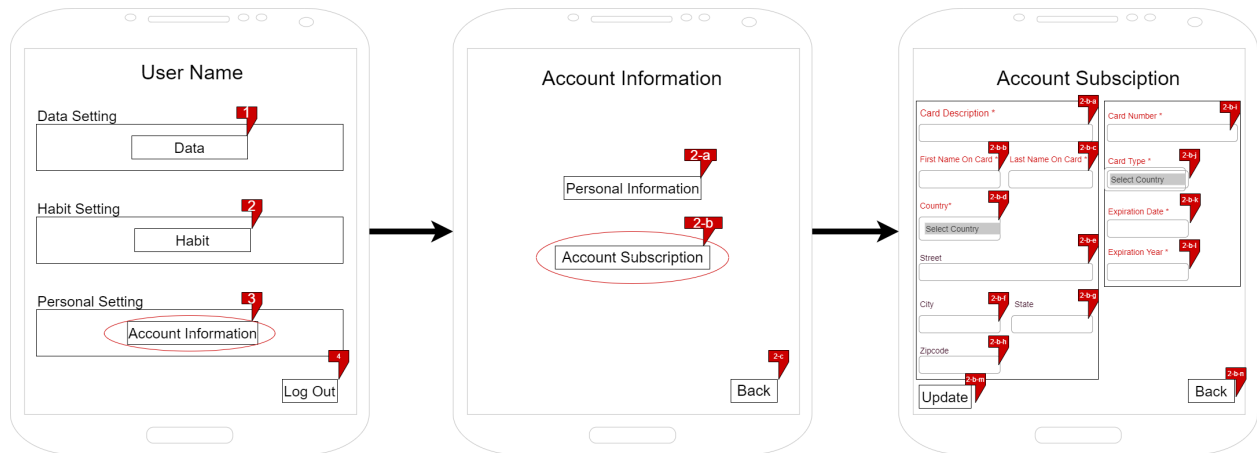
4: Logout Button

- Red Circle Is Current State

Wireframe of UC 7: User Shares Data Through Email



Wireframe of UC 8: Account Subscription



Annotated Note

1: Data Setting Button

2: Habit Setting Button

2-a: Personal Information Button

2-b: Account Subscription Button

2-b-a: Card Description Text Box

2-b-b: First Name Input Text Box

2-b-c: Last Name Input Text Box

2-b-d: Country Input Text Box

2-b-e: Street Input Text Box

2-b-f: City Input Text Box

2-b-g: State Input Text Box

2-b-h: Zipcode Input Text Box

2-b-i: Card Number Input Text Box

2-b-j: Card Type Select Option

2-b-k: Card Expiration Date

2-b-l: Card Expiration Year

2-b-m: Update Payment Button

2-b-n: Back to Account Page Button

3: Account Information Button

4: Logout Button

- Red Circle is Current State

Wireframe of UC 9: Log out

The user will press the logout button to logout of the account. Which will lead the user back to login page

