Team 2: Habit Tracking

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

Table of Contents

Project Description

Overview

Tracker is an application on mobile and desktop platforms to help people track their habits. The application can use other apps (apple watch/health-tracking app). This app offers a highly tailorable experience for the user to track their habits. The application will be sold through existing app stores and will work cross-platform.

Key Features

Core objectives of the project that must be completed for the application are:

- Available on mobile
- Method of creating a user account and having the user account saved in the database
- Method of the user creating habits they want to track and inputting data for the habits they want to track
- Able to transition from the UI display of habits tracked over a week to the UI where the user can input data for habits they are tracking in less than .5 seconds and have the transition be less than .5 seconds 99% of the time.

Market Value

Applications such as ours need to fight against a pretty heavy competition to gain any recognition on the Android and Apple app stores. Asking users to pay upfront usually discourages them from trying the app at all. So, the only viable option to monetize the app is to show the user advertisements. A competitive app isn't overrun by ads, as this discourages even the newest of users.

Displaying ads on apps effectively is tough, as the more ads you have, the more annoying they become. The revenue from ads is directly proportional to how intrusive they are and how much engagement they get. Google pays around 2 cents per impression, but about 16 cents per interstitial ad.

Our app will mostly utilize interstitial ads. For users who really enjoy the app and would like an ad-free experience, the user can pay 99 cents per month to remove all ads. While this might not make as much revenue in the long run, it creates a more enjoyable experience for the most dedicated user base.

Scope

The larger goal of the project might be to get acquired by a larger company. Often, apps such as this hit a ceiling with how much revenue they can create, and maintenance costs overtake the profit. Companies such as Nike, Apple, or Sworkit own a variety of apps relating to lifestyle management and fitness regimens. While our application isn't meant exclusively for fitness and health, such companies might find value in certain aspects of the app. As a part of their larger ecosystem of such apps and devices, it might have more value to them than it would as a standalone app.

A technical challenge to the scope of this project is maintaining performance with the potential of millions of users accessing the app servers all at once. The architecture design will play a key role in scalability of the app. Another technical challenge is allowing the app to interface with other apps on the users desktop or phone such as the apple watch or a health tracking app. This would require a license agreement with the other app creators of what data could be used from their apps to supplement the Tracker app.

Stakeholders

The stakeholders of this project include:

- Frontend Developer
    - Create and implement the app functionalities and the interaction between the app and the user.
- UI designer
    - Focus on the user interface and graphic interaction between the user and the app.
- Backend Developer
    - Create and manage the app server and server system database of the app
- Marketer
    - Focus on finding the audience and platform to advertise the game effectively
- Project Manager
    - Manage the team and resources for the project to stay on the right track and deliver the project to the client on time before the deadline.
- Client
    - Test and verify if the app meets the requirement without any possible error and bug before it is released to the audience.

Two people may have the same role.

Feasibility:

Technical

- Hardware
    - Limited by internet connection to database/server
    - Large amounts of data can be updated if the app is not allowed to upload consistently
    - A week of data takes up around 40 MB.
- Server takes care of processing
- Database takes care of long term data storage
    - Will need to let phone hold onto data during moments without a connection
- Phone Application takes care of data collection
- Data sharing between apps for passive data collection of habits user wants to track
- Easy distribution (AppStore for phone)
- Easy to use (one click analysis unless specific settings wanted)
- Core analysis code will require a bit of fine-tuning as it will need to adjust according to the user
- Can use the native UI elements on each platform as the initial build does not need to be stylish

Organizational

- Agile SDLC will be used during the development
- Medium-sized team of 3-4 people with a specialist in software development.
- Size of the team will encourage getting essential parts developed and into testing in an efficient manner.

Operational

- Apple watch gathers data from heart rate and pulse
- App uses device storage even when the device is not connected to the internet. Once connected to the internet, data stored on the device gets pushed to the app database.
- The app takes in user's input for data
- The app is not expected to take up a substantial amount of battery usage, as the only work it needs to do in the background includes notification scheduling. All data gathered by external devices such as Fitbits or Apple watches spends their respective batteries, and is independent of the app's usage

Economics

- At the current stage, we are planning to make an ad-based version of the app to generate revenue and an ad-free paid version of the app.
- Cost
    - Developing cost
    - Marketing costs (for advertisements)
    - Maintaining cost
    - Server hosting costs
- Benefit
    - Developing experience
    - Income from in-app ads and app uses from users
    - Project management experience
    - Kickstarter

Support/maintainability

- Tech support through email
- Tech support through the website

- Tech support from the app's feedback

Marketability

- Main audience
    - Must have phone
    - Student
    - Worker
    - Parent
    - No restricting age
- Free application
    - Gain money through ads
- Easy accessible
- Ads
    - Social media: Instagram, Facebook, podcast, Pinterest
    - Ask for application review

Developability

- Effort to add more functionalities
- Effort to update app based on the customer feedback
- Possible patches and fixes upon launch
- App optimization

Schedule

- We could finish this project in approximately four weeks (40-50 hour weeks) since we have seven people; including making policy, procedure, and testing.

Formal Software Specifications

Functional Requirements:

This section focuses on the Habit Tracking App 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 server 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 the 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 the internet is available
HT-FR-19	The user can provide feedback on 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 receive software updates remotely from the 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 are 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 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, *only* 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 server 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 off-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

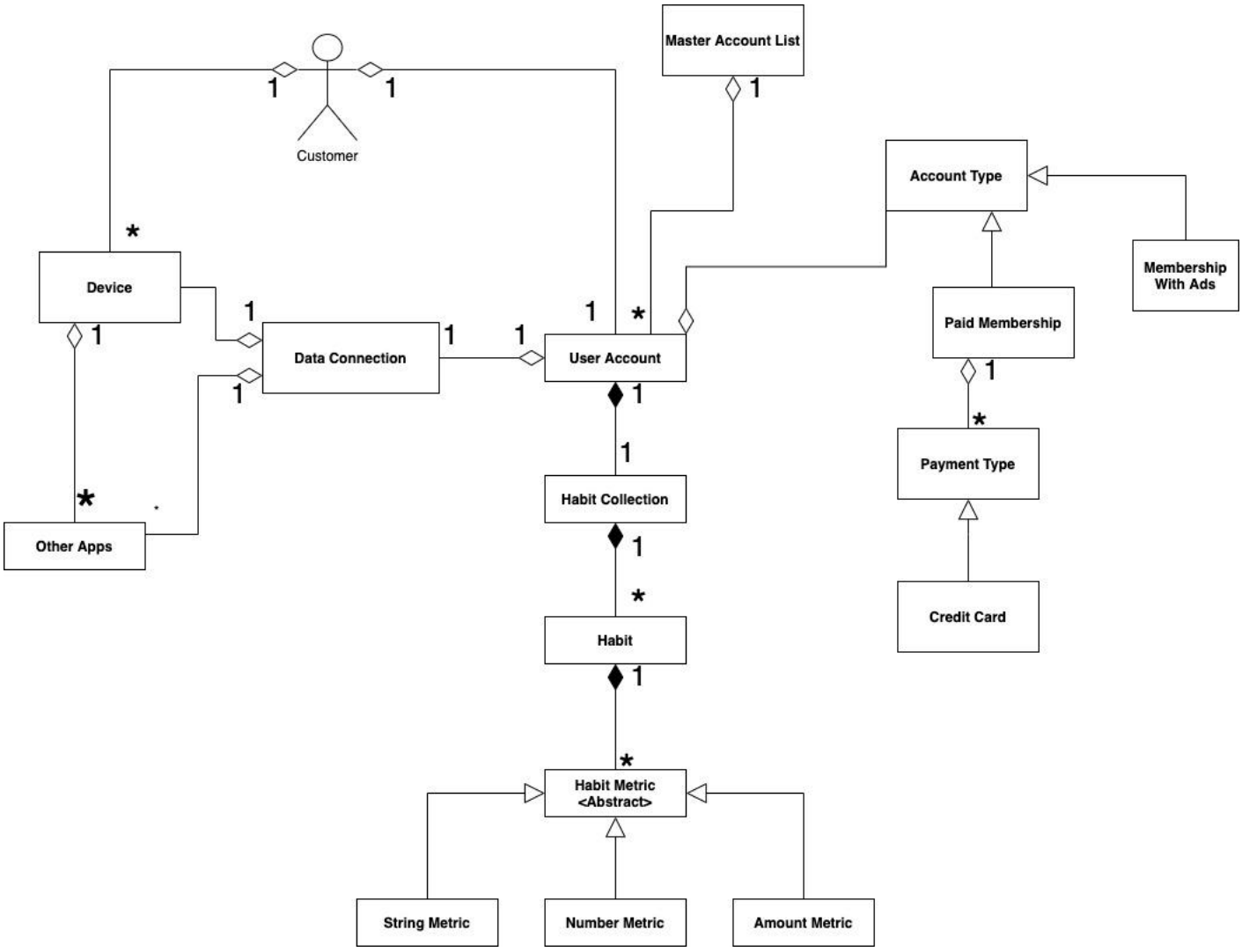
ID	Non-Functional Requirement
HT-NFR-20	The application should allow one-click analysis and tracking of user habits

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 an account for new user

Actor(s) : Anyone

Preconditions: User has the habit tracking app on the 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 the user to enter information
    a. Required information - Username, password, email
    b. Optional information - Phone number, backup email
4. Server system database checks if username or email have already been used
5. Server system database creates the account and locks it
6. Server system database sends the user a confirmation email
7. User accepts the confirmation email and the account is unlocked

Alternatives:

1a) No Internet connection
1b) Display note about requiring internet for account creation
4a) Username or email already exists
4b) Display small notification about duplicates and prompt change
6a) Email never confirmed
6b) Account remains locked for 7 days then removed.

End Condition: Users account is created

UC 2: Display History of User Activity

Goal: Display history of user activity

Actor(s): Anyone with an account

Preconditions: User is logged in and has data either in local storage or uploaded to the server system database

Trigger: User clicks check the history

Main Scenario :

1. Program checks internet connection
2. Program sends the request to the server system database to acquire history of user activity
3. Program displays history of user activity

Alternatives:

1a) No internet connection

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

End Condition: History of user activity displayed

UC 3: Upload data

Goal: Uploads local user data to the server system database

Actor(s): Anyone with an account

Preconditions: Is logged in to the account

Trigger: User clicks “Upload data”

Main Scenario :

1. Program checks internet connection
2. Server system database pulls data from phone-based off last recorded upload date

Alternatives:

1. a. No internet connection

   b. Display error message and ask for the user to get to the location with internet
2. a. Server system database unavailable

   b. Display error message

End Condition: User data now backed up in server system database

UC 4: Display Data

Goal: Display habit tracking data to the user

Actor(s): Anyone with an account

Preconditions: Has data on server system database and user is logged in

Trigger: User clicks habit tracking data

Main Scenario :

1. Program pulls server system database 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

Actor(s): Anyone with an account

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

Trigger: Clicks the add activities button on habit 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)
4. Program takes data and uploads it to the server system database

Alternatives:

1a) No internet connection

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

End Condition: User is informed of date saved

UC 6: Delete habit

Goal: Delete habit on application

Actor(s): Anyone with an account

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

Trigger: User clicks the delete habit button

Main Scenario:

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

Alternatives:

1a) No internet connection

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

End Condition: Habit is deleted from the user account

UC 7: User shares data through the email

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

Actor(s): Anyone with an account

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

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

Main Scenario:

1. Program checks internet connection
2. Program displays user habits and asks the user to click the habits they want to share
3. Options to move between the time frame of last week, last month, or the last three months
4. Prompts the user to type in the email they want to send their habit info to
5. User Habit information will be shared via email address.
6. User action of sharing habit is stored in Account History

Alternatives:

1a) No internet connection
1a1) Display error message and ask for the user to get to the location with internet
5a) invalid email address
5b) highlight in red the email text box field showing the error message "invalid email entered"

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 the user to get to the location with internet

2) Invalid payment info

2a) Program displays error and lists of invalid fields

2b) user corrects fields until Program validates them

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

UC 9: Logout:

Goal: User can log out of 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

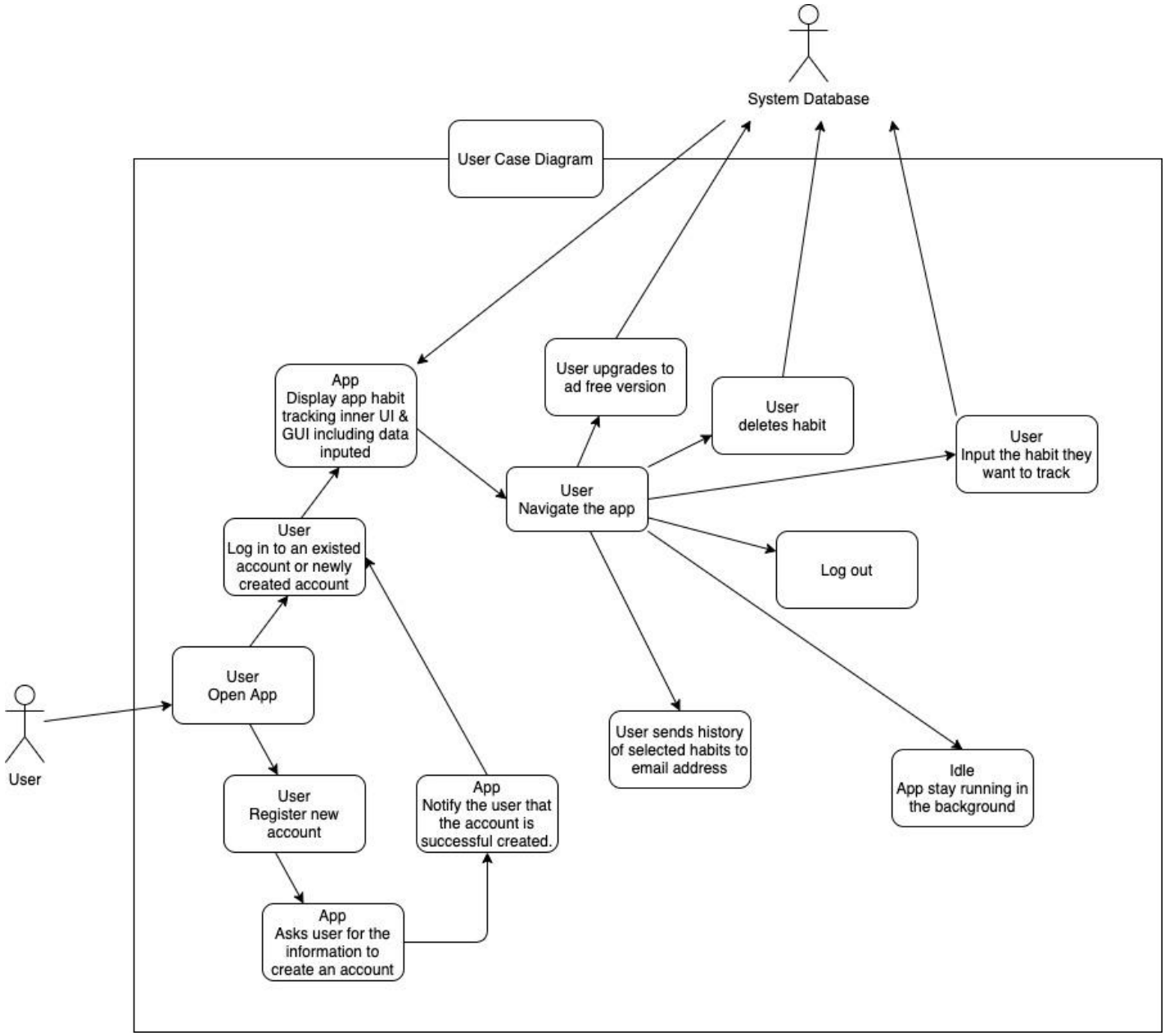
3a1a) User presses save work and work is saved

3a1b) User presses discard and work is not saved

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

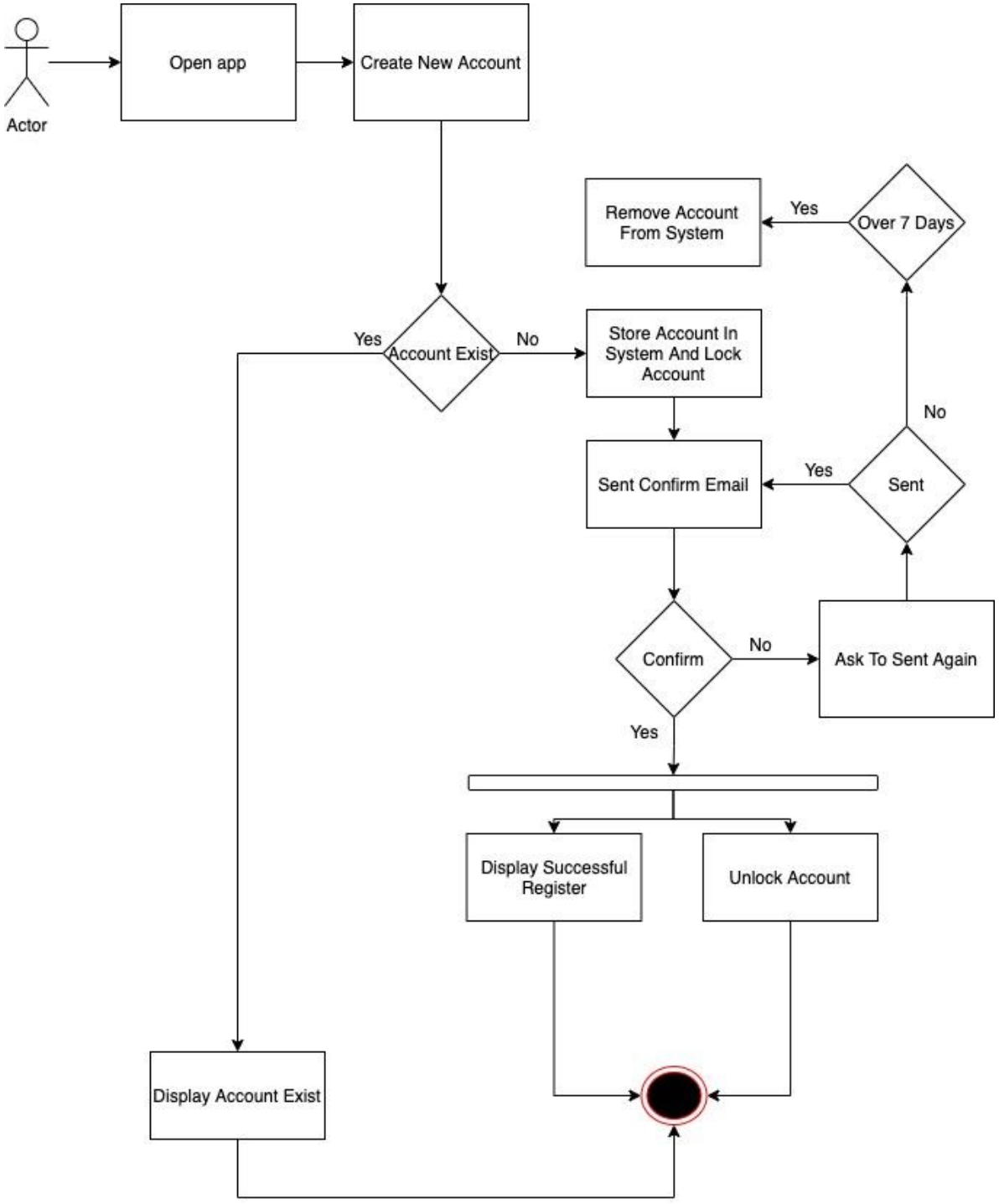
End Condition: User is logged out of the account.

Use case Diagram:

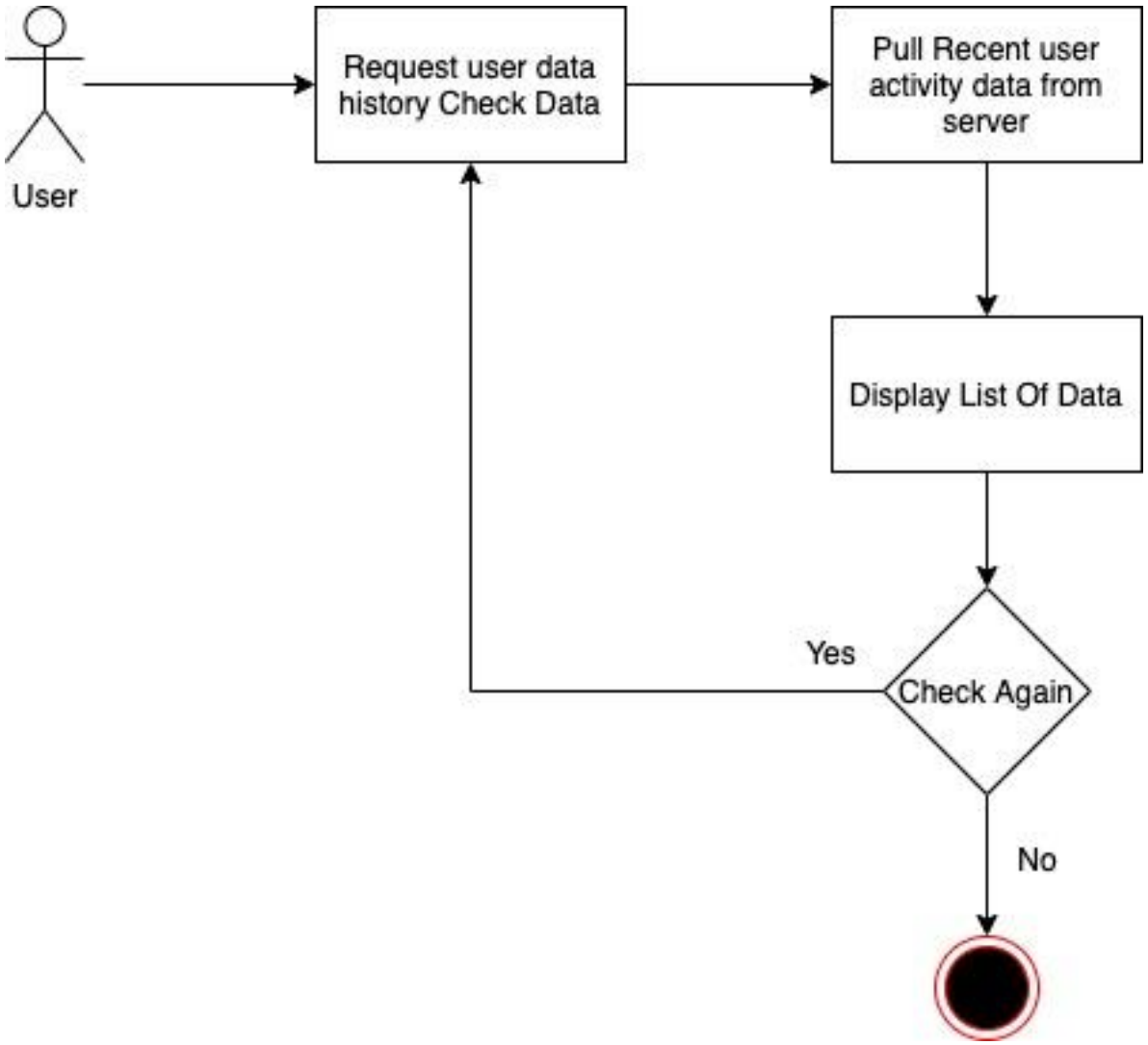


Activity Diagrams:

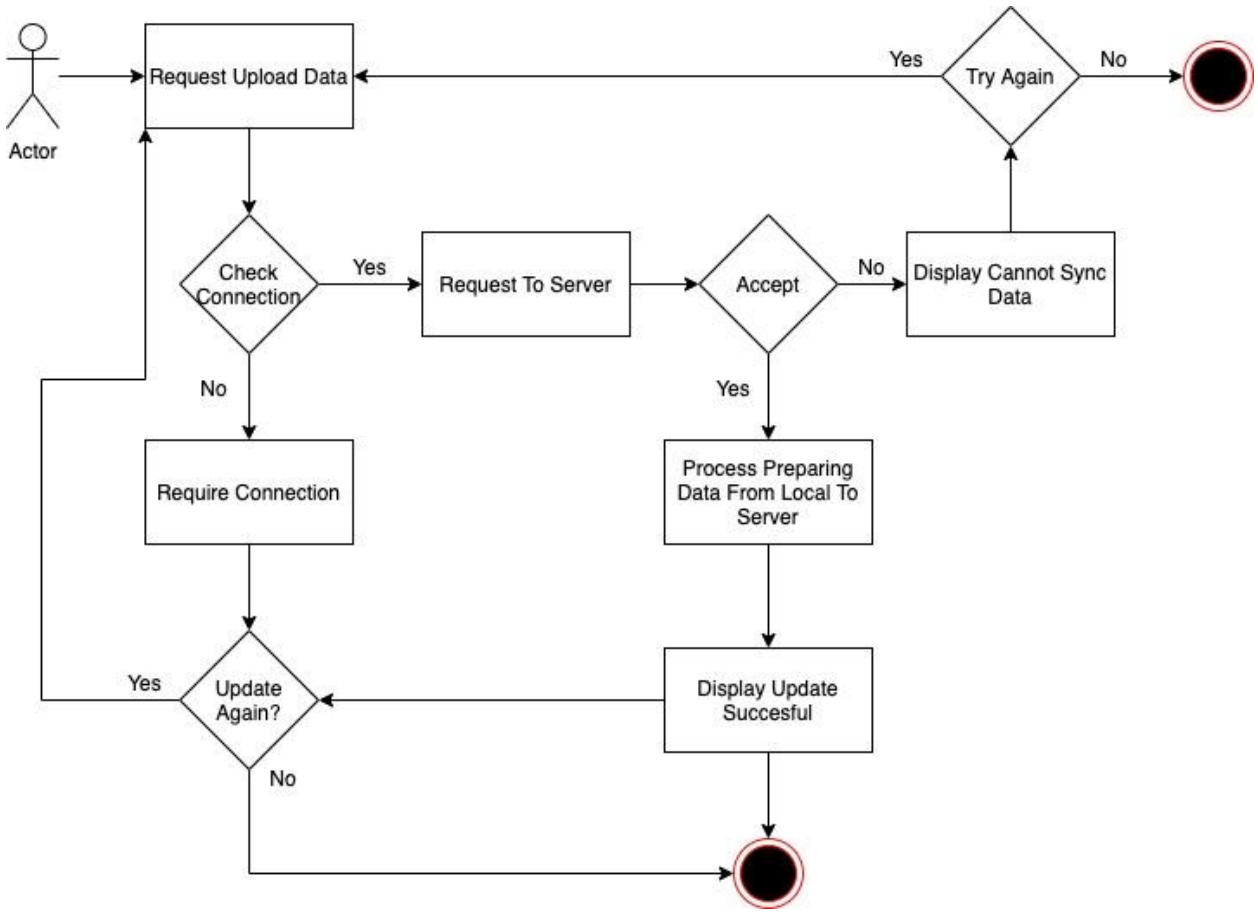
Activity Diagram of UC 1: Register an account



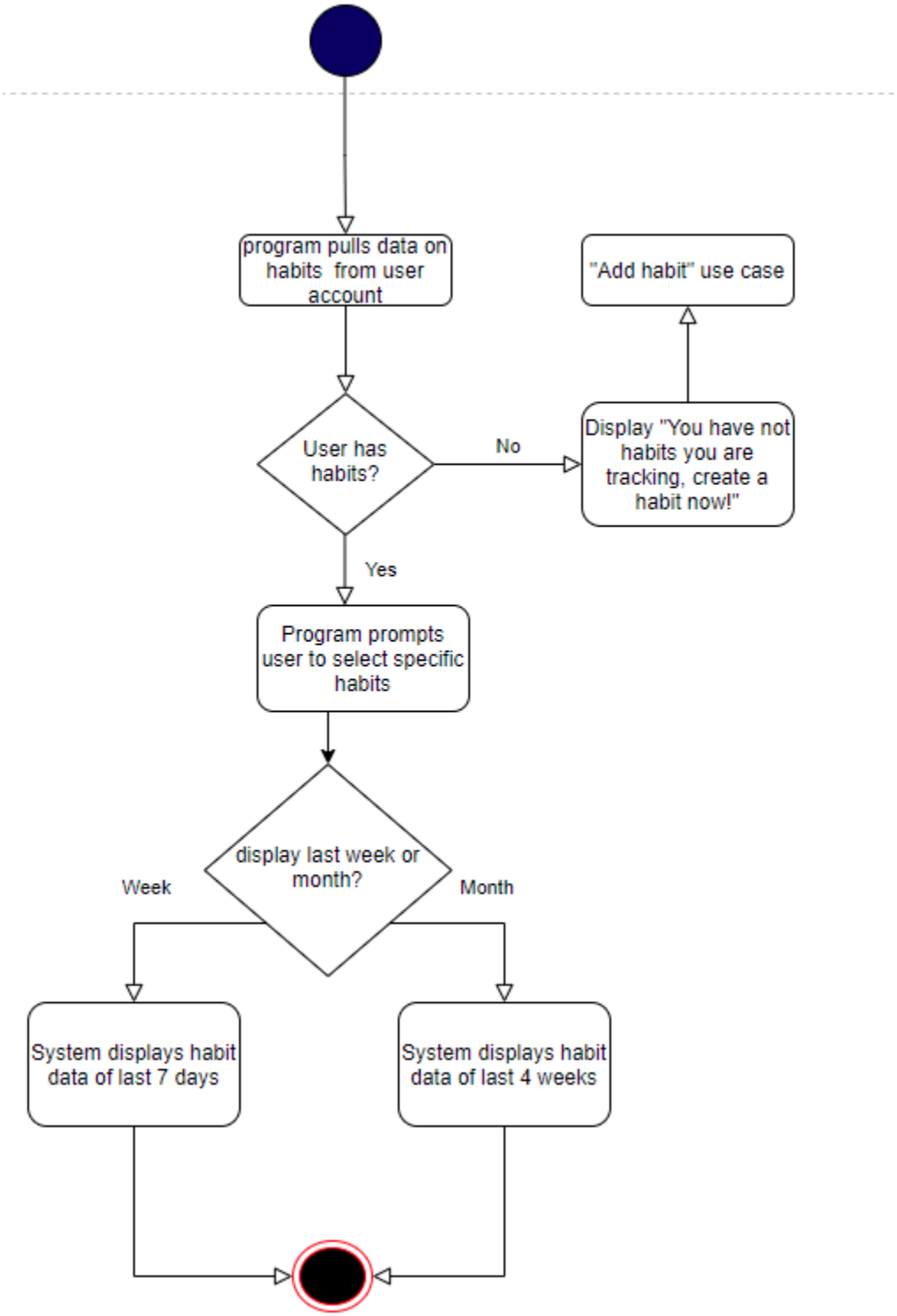
Activity Diagram of UC 2: Check Recent Data History



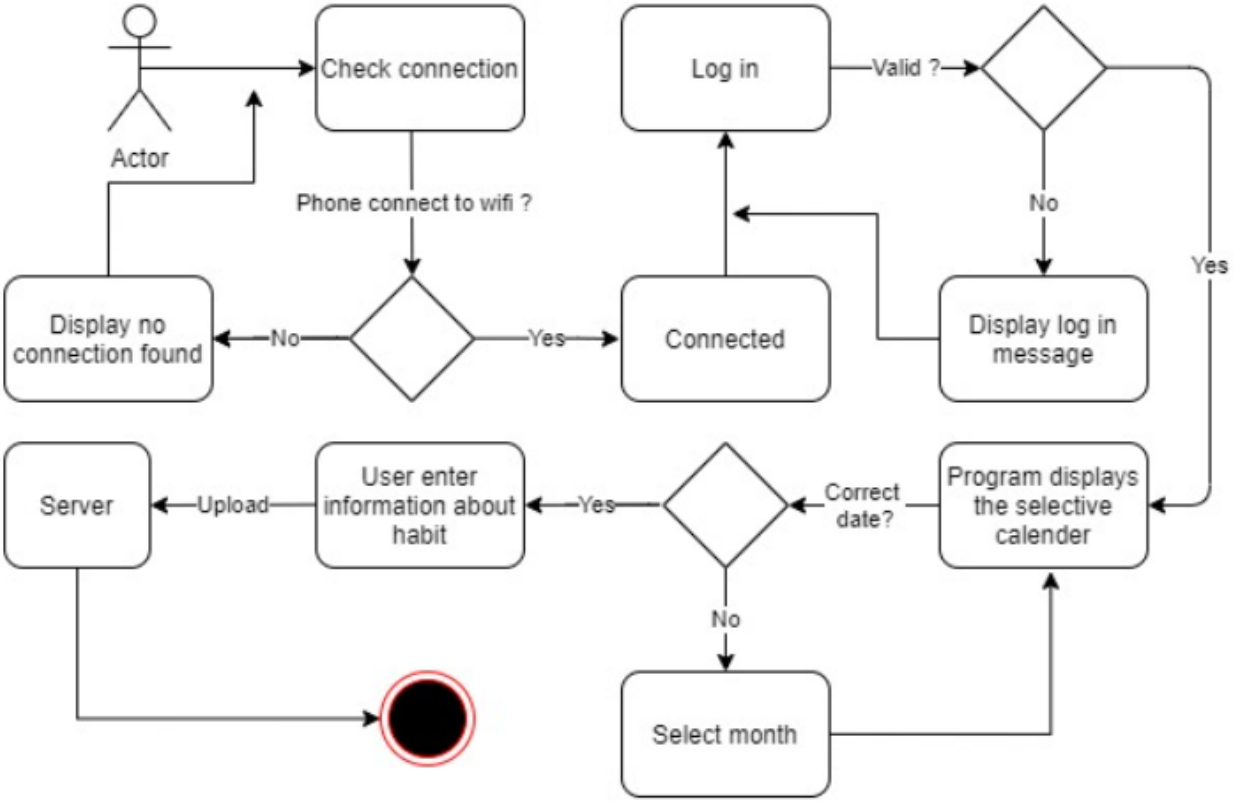
Activity Diagram of UC 3: Upload Data



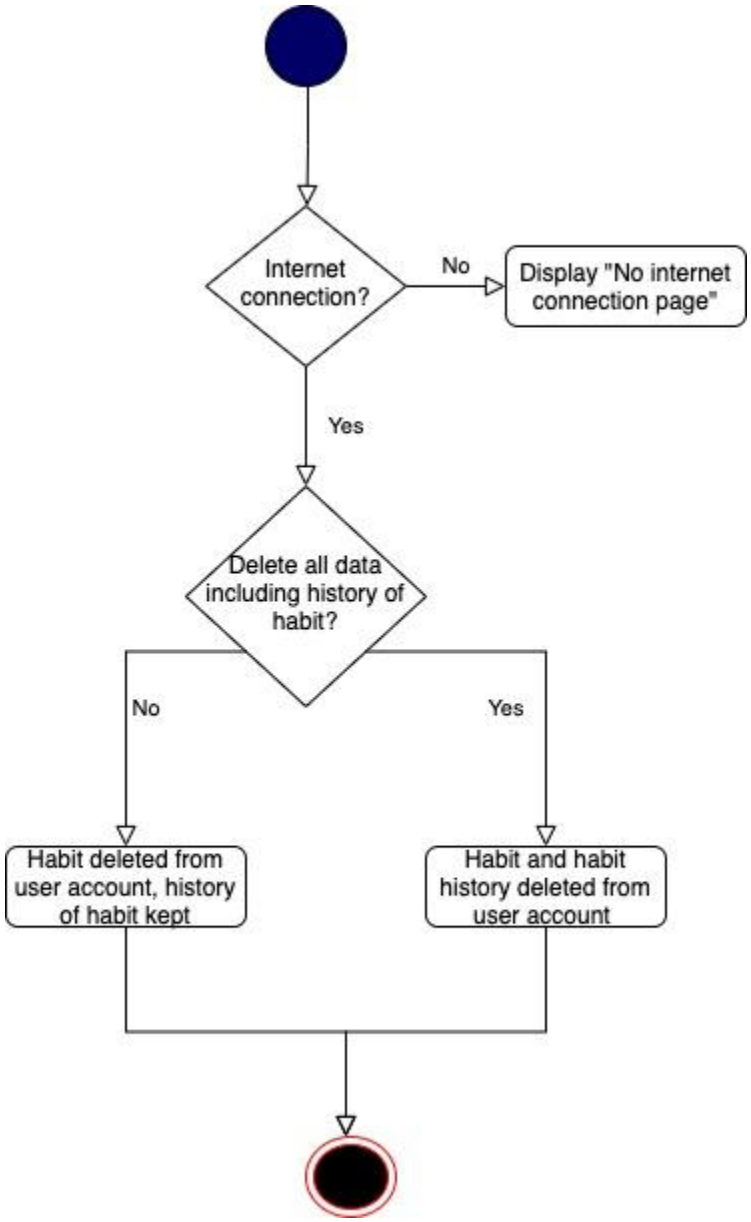
Activity Diagram of UC 4: Display Data



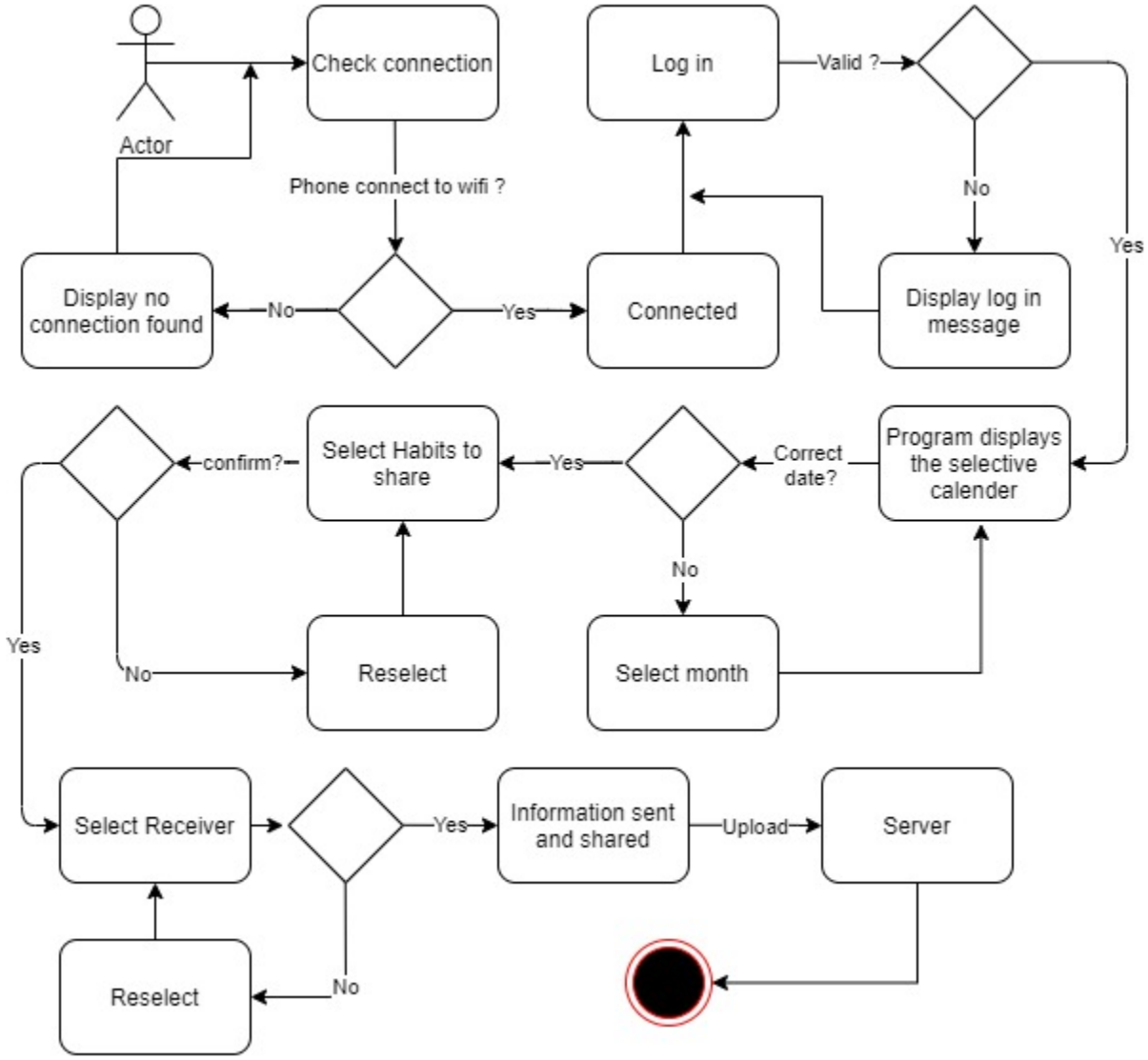
Activity Diagram of UC 5: Add Habit



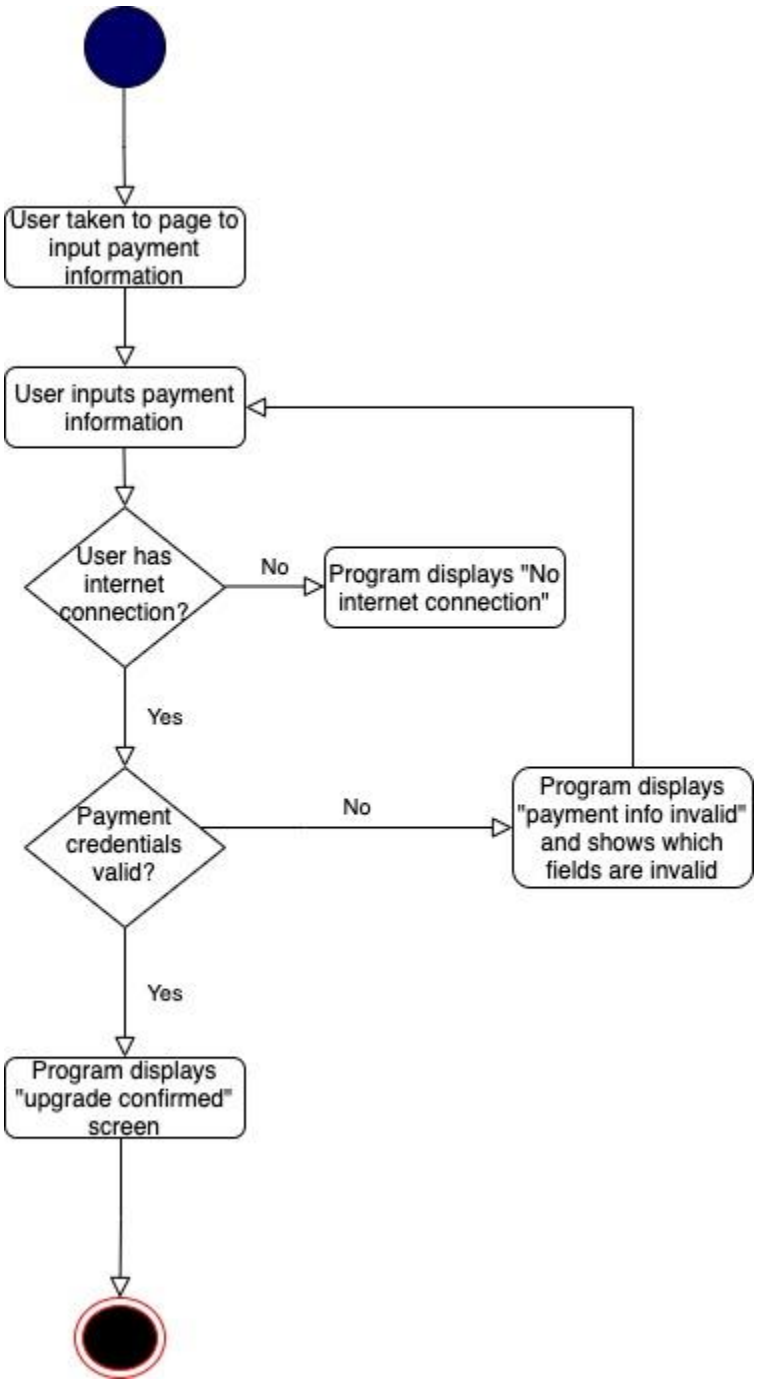
Activity Diagram of UC 6: Delete Habit



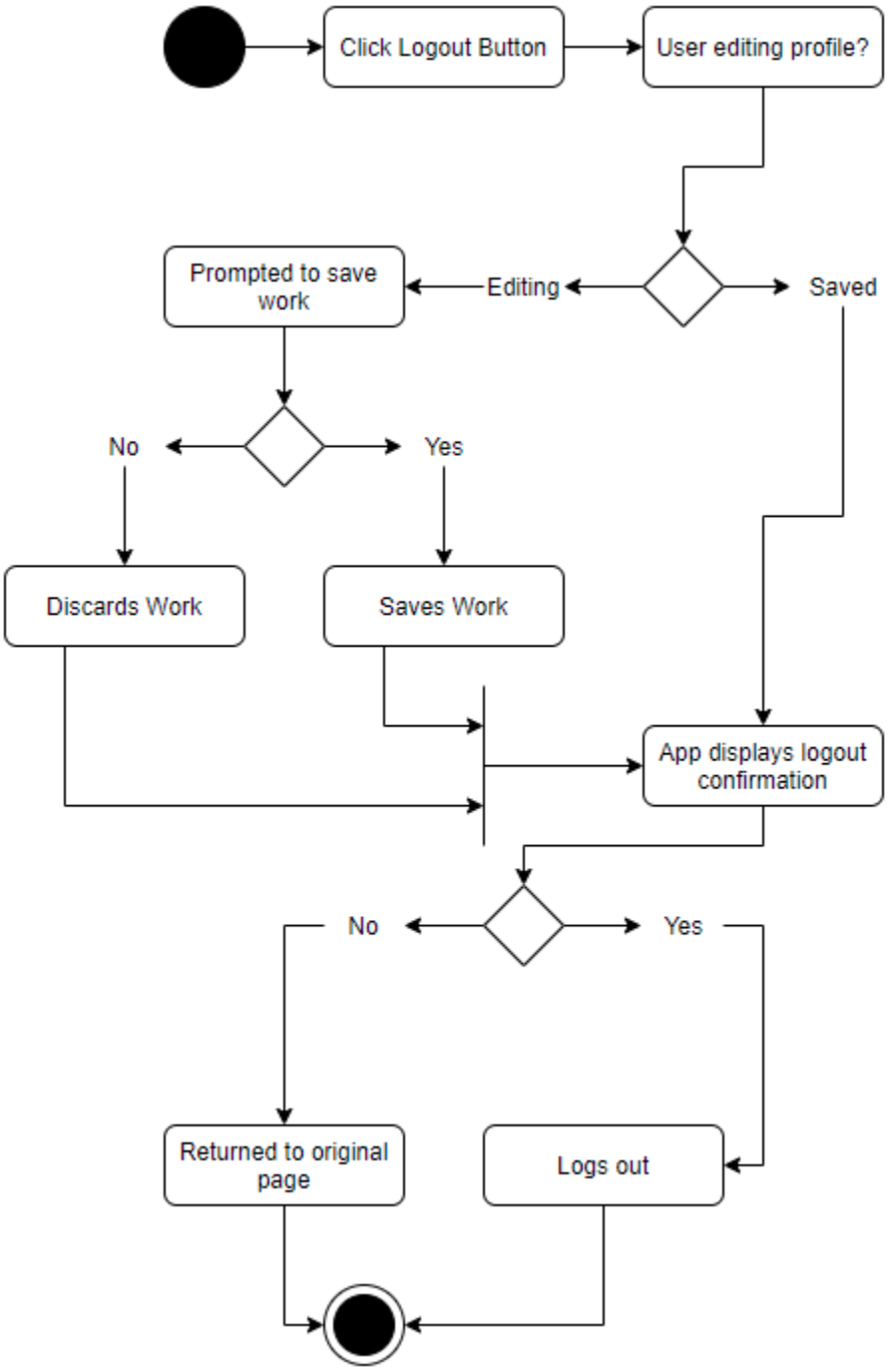
Activity Diagram of UC 7: User Shares Data Through Email



Activity Diagram of UC 8: Account Subscription

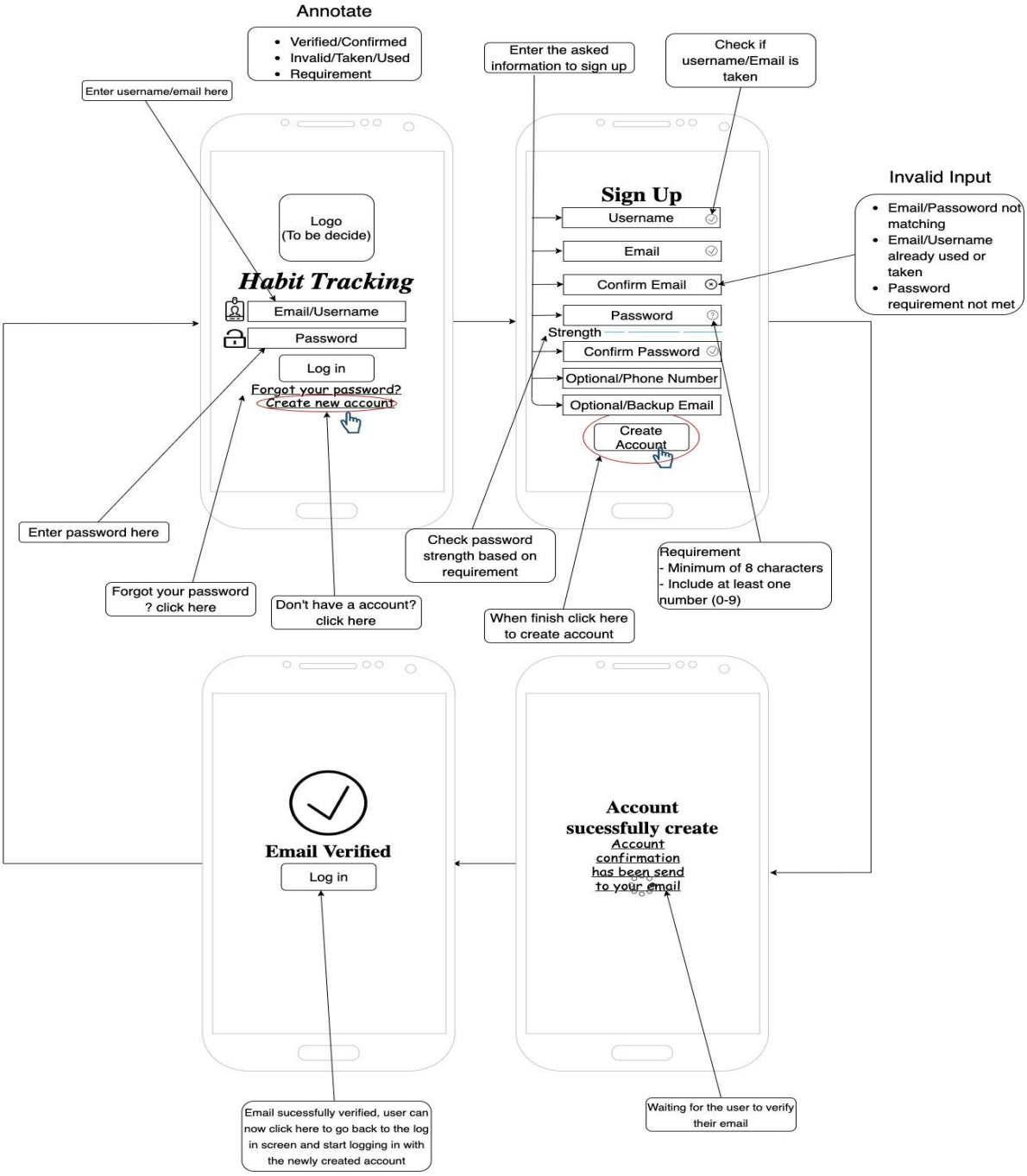


Activity Diagram of UC 9: Logout

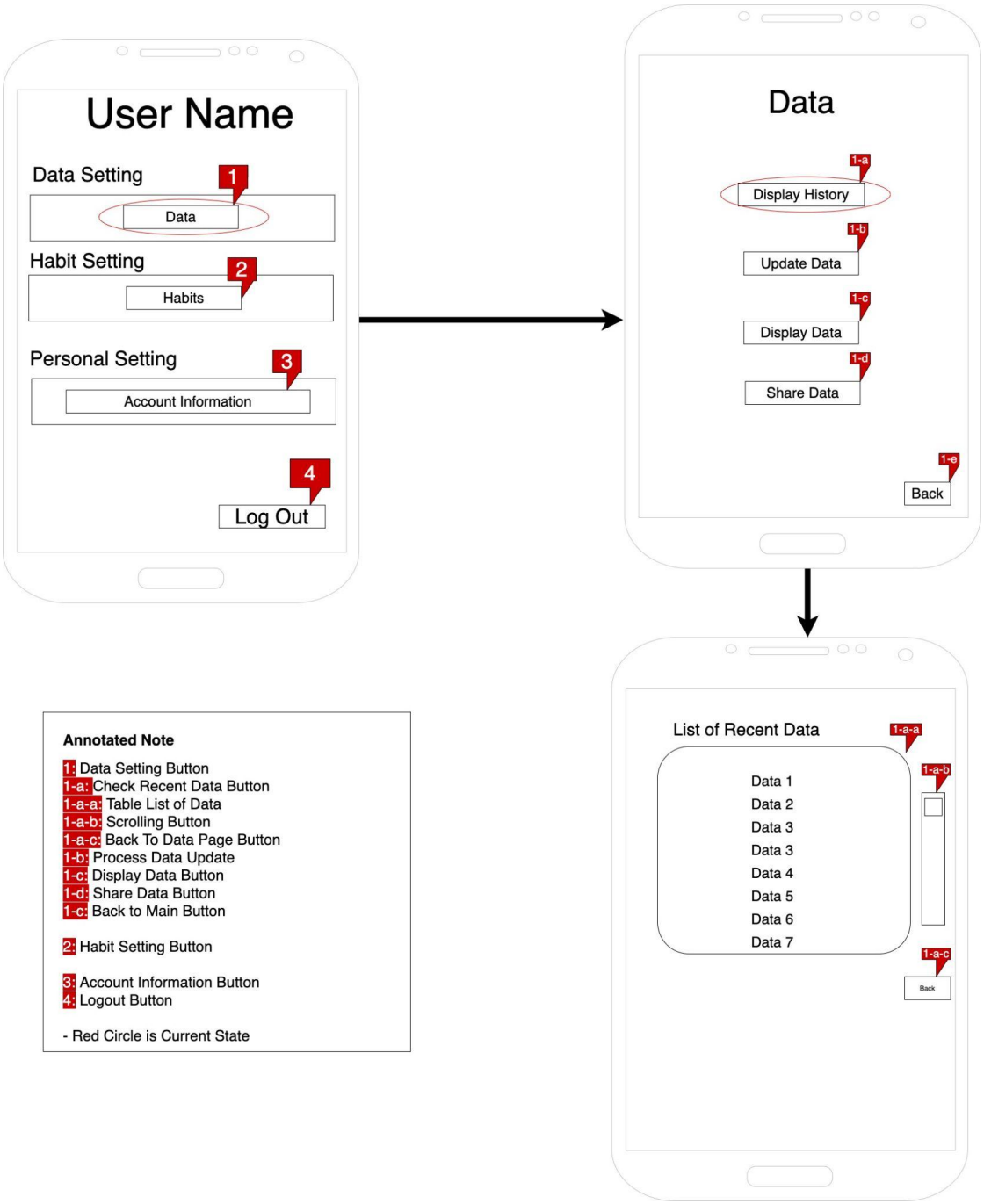


Wireframe:

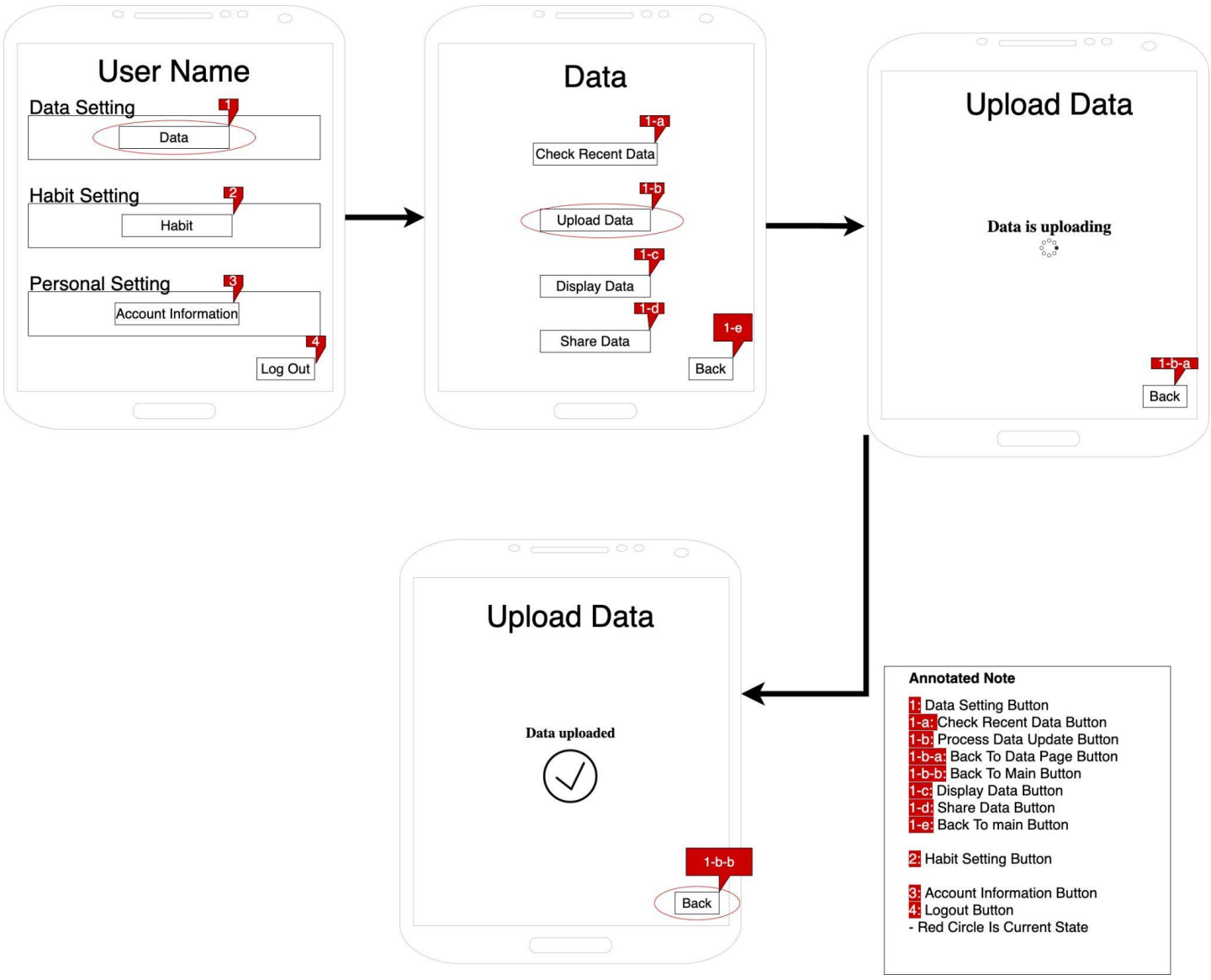
Wireframe of UC 1: Register an Account



Wireframe of UC 2: Display History of User Activity

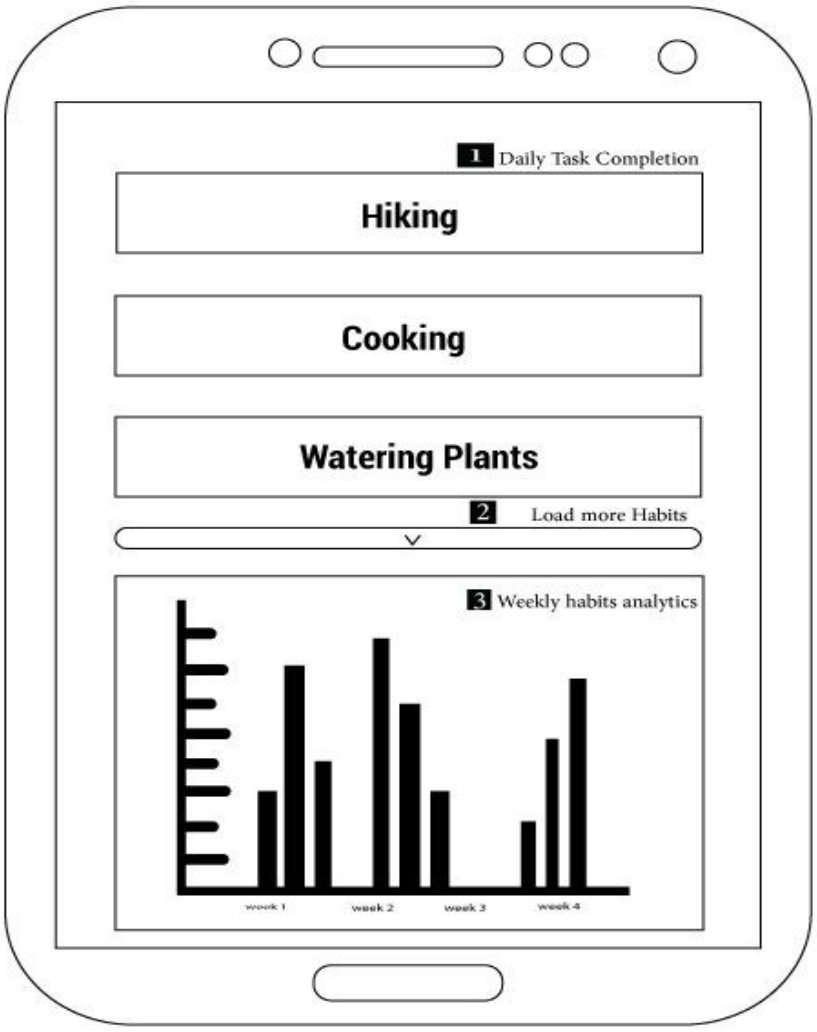


Wireframe of UC 3: Upload Data

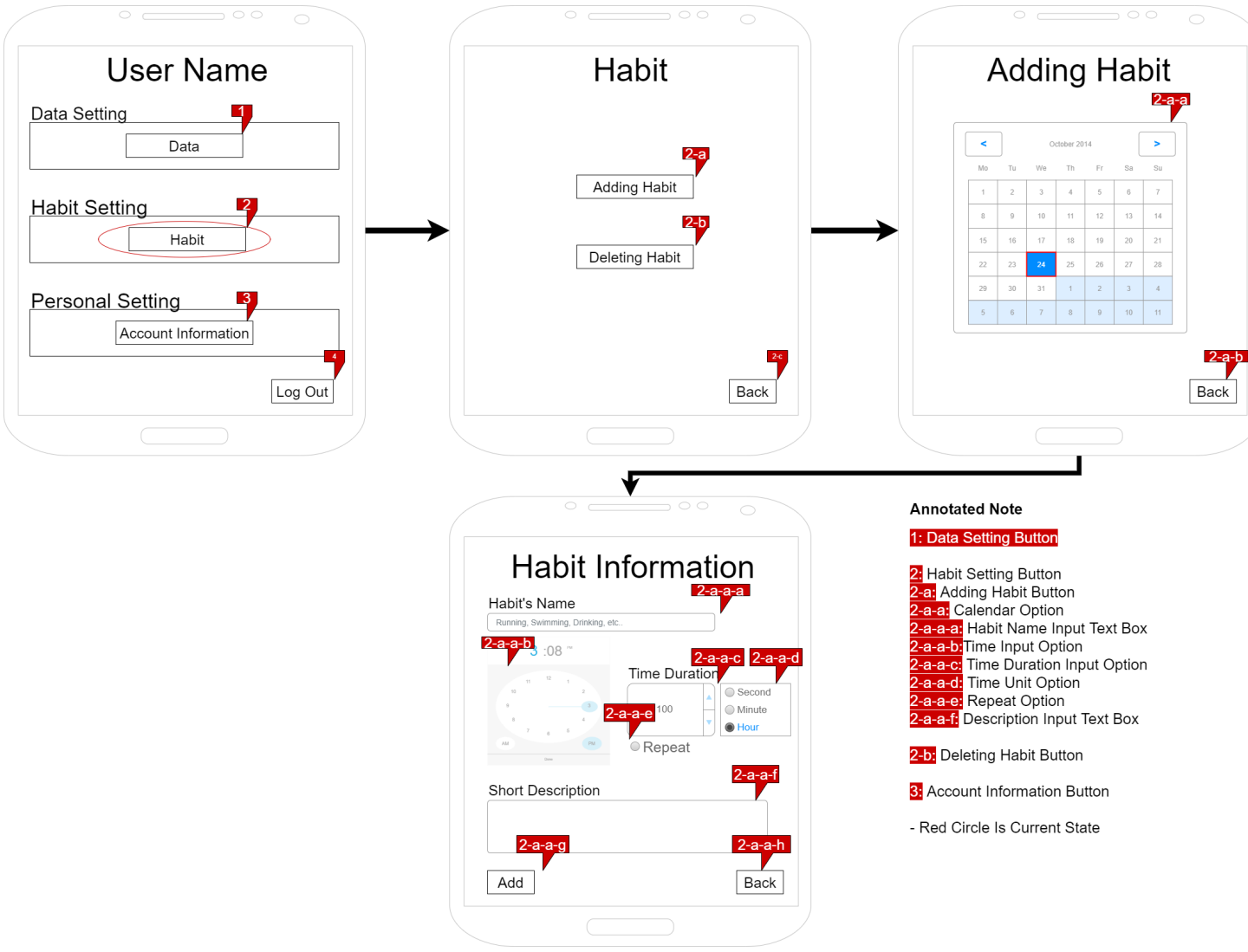


Wireframe of UC 4: Display Data

The Display data screen shows the list of every added habit and waits for the user to select some. It notifies the user about the daily completion of the tasks. The analytics graph shows the weekly report on selected habits. 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



Annotated Note

1: Data Setting Button

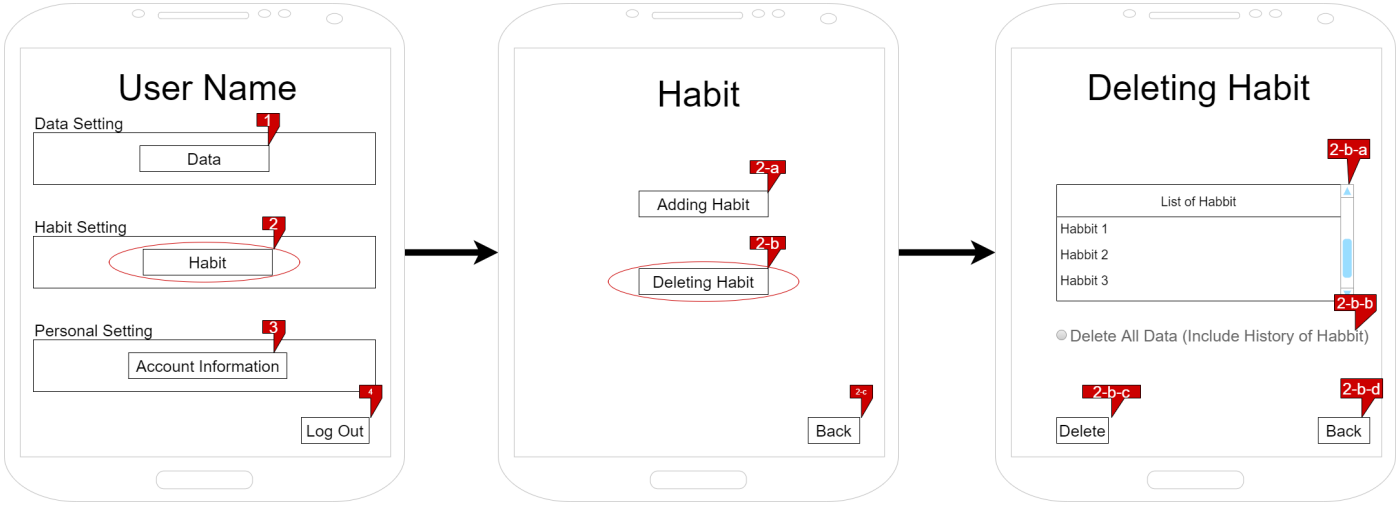
2: Habit Setting Button
2-a: Adding Habit Button
2-a-a: Calendar Option
2-a-a-a: Habit Name Input Text Box
2-a-a-b: Time Input Option
2-a-a-c: Time Duration Input Option
2-a-a-d: Time Unit Option
2-a-a-e: Repeat Option
2-a-a-f: Description Input Text Box

2-b: Deleting Habit Button

3: Account Information Button

- Red Circle Is Current State

Wireframe of UC 6: Delete Habit



Annotated Note

1: Data Setting Button

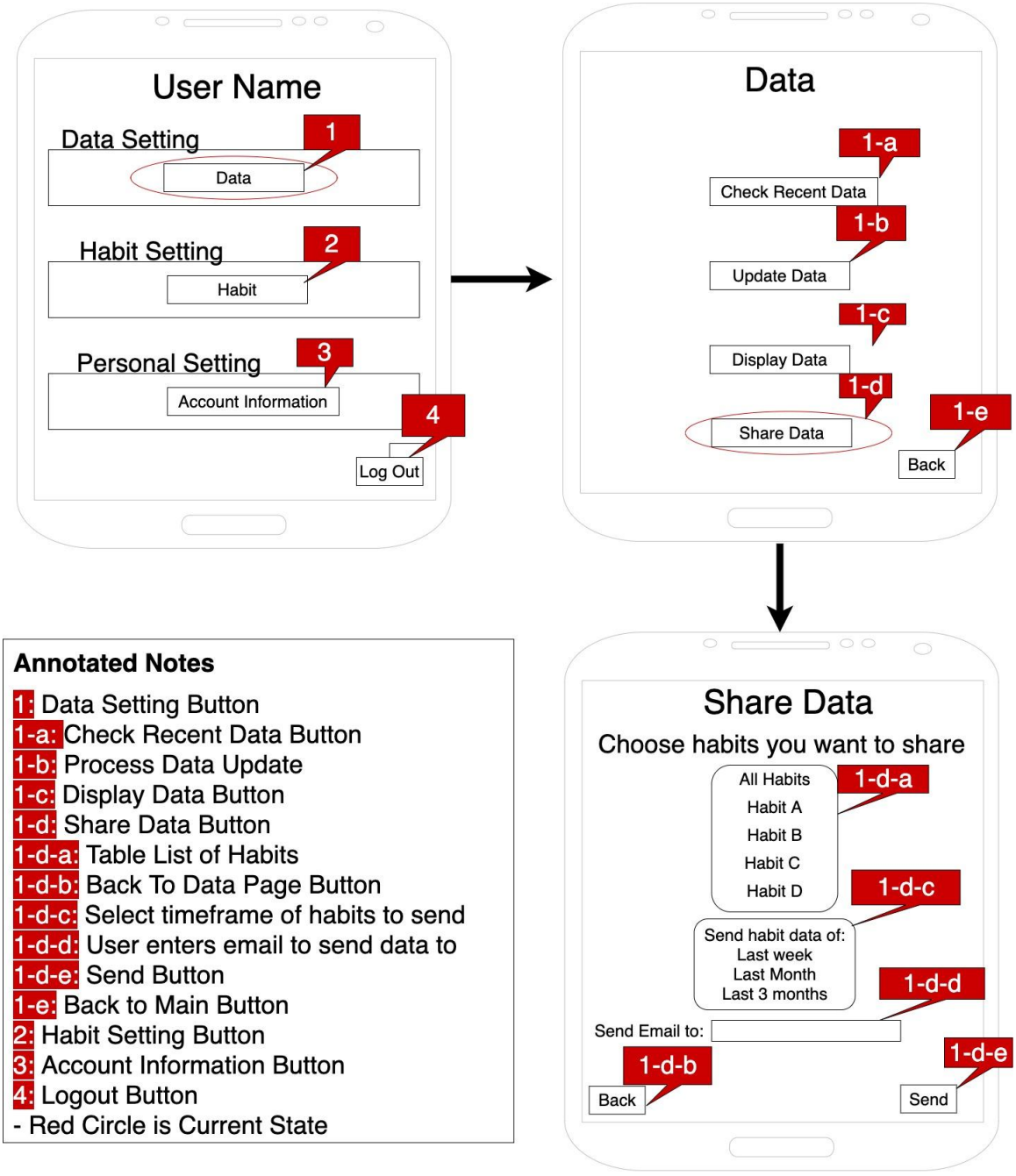
2: Habit Setting Button
2-a: Adding Habit Button
2-b: Deleting Habit Button
2-b-a: List of Habit
2-b-b: Delete All Data Option
2-b-c: Delete Button
2-b-d: Back To Habit Button
2-c: Back To Main Setting Button

3: Account Information Button

4: Logout Button

- Red Circle Is Current State

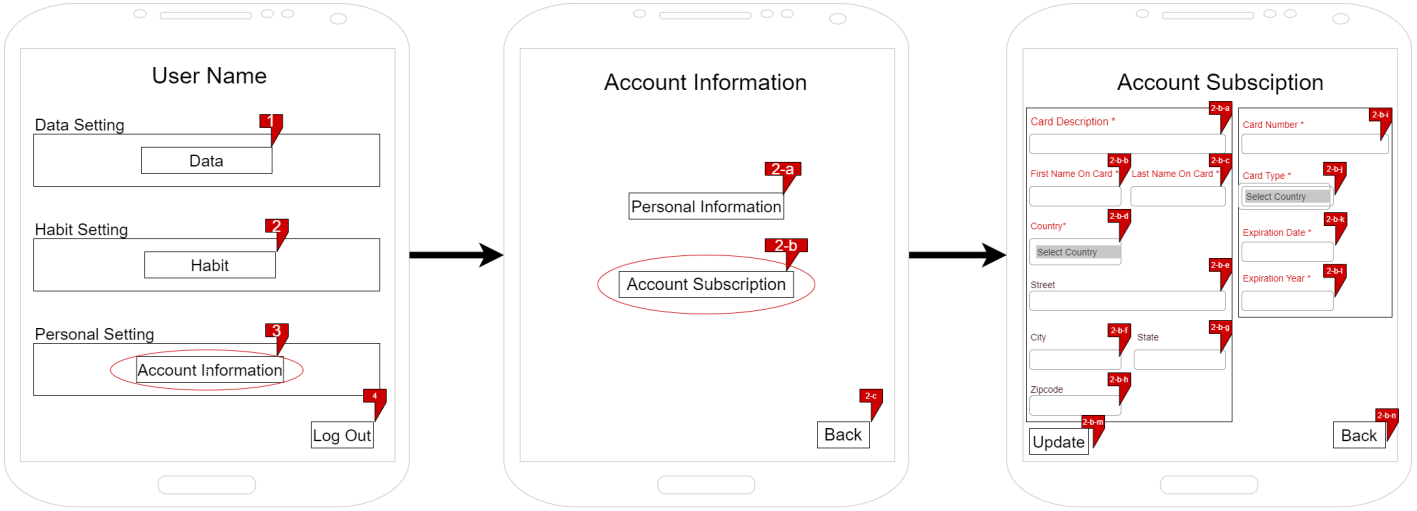
Wireframe of UC 7: User Shares Data Through Email



Annotated Notes

1: Data Setting Button
1-a: Check Recent Data Button
1-b: Process Data Update
1-c: Display Data Button
1-d: Share Data Button
1-d-a: Table List of Habits
1-d-b: Back To Data Page Button
1-d-c: Select timeframe of habits to send
1-d-d: User enters email to send data to
1-d-e: Send Button
1-e: Back to Main Button
2: Habit Setting Button
3: Account Information Button
4: Logout Button
- Red Circle is Current State

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

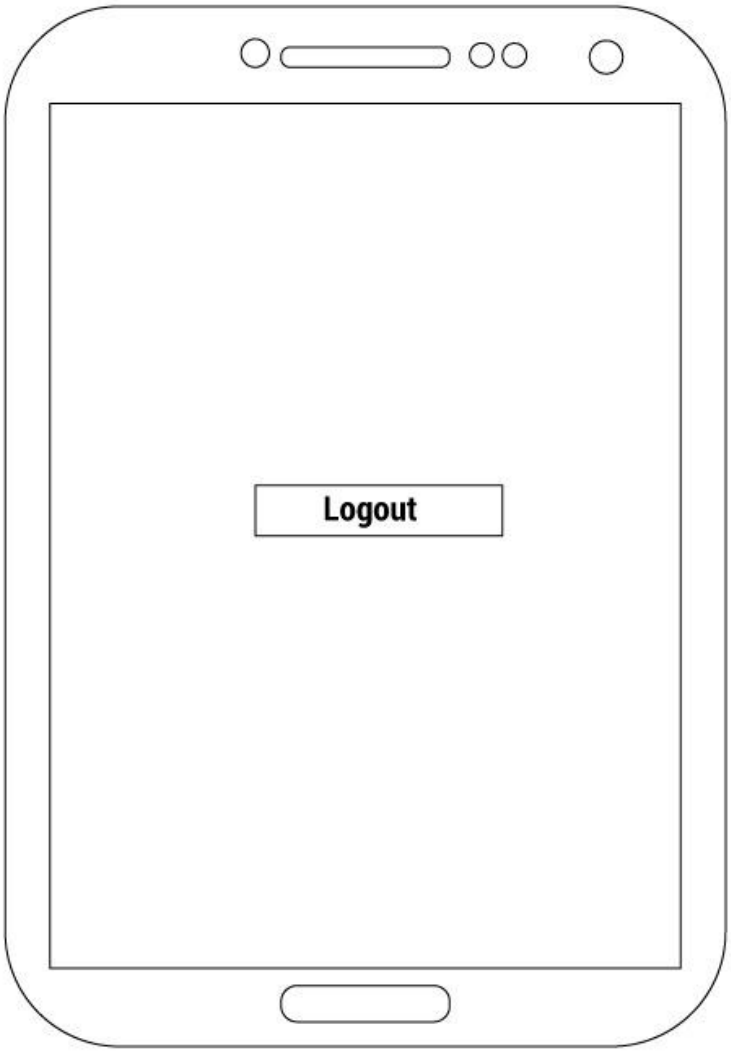
3: Account Information Button

4: Logout Button

- Red Circle is Current State

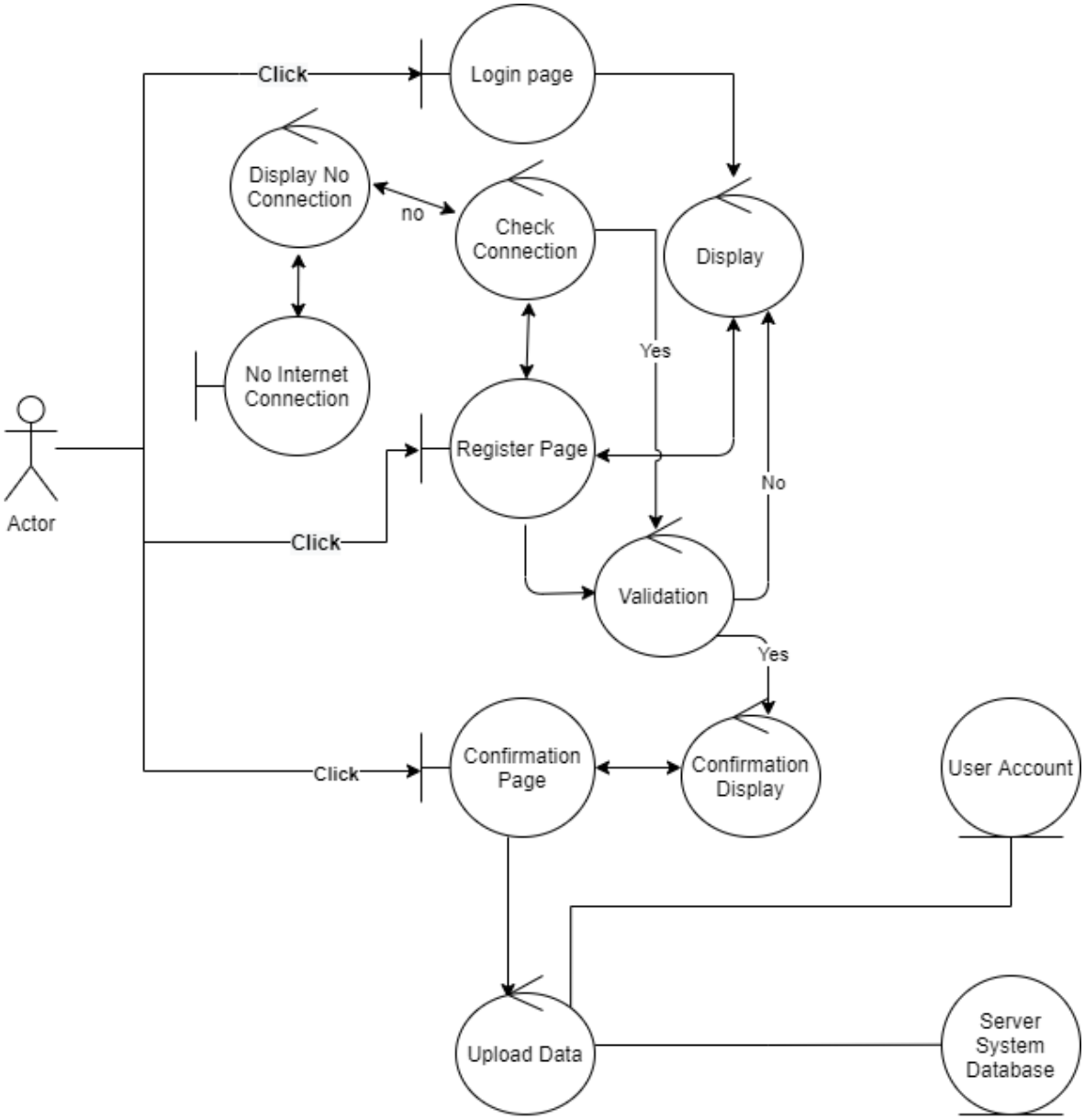
Wireframe of UC 9: Log out

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

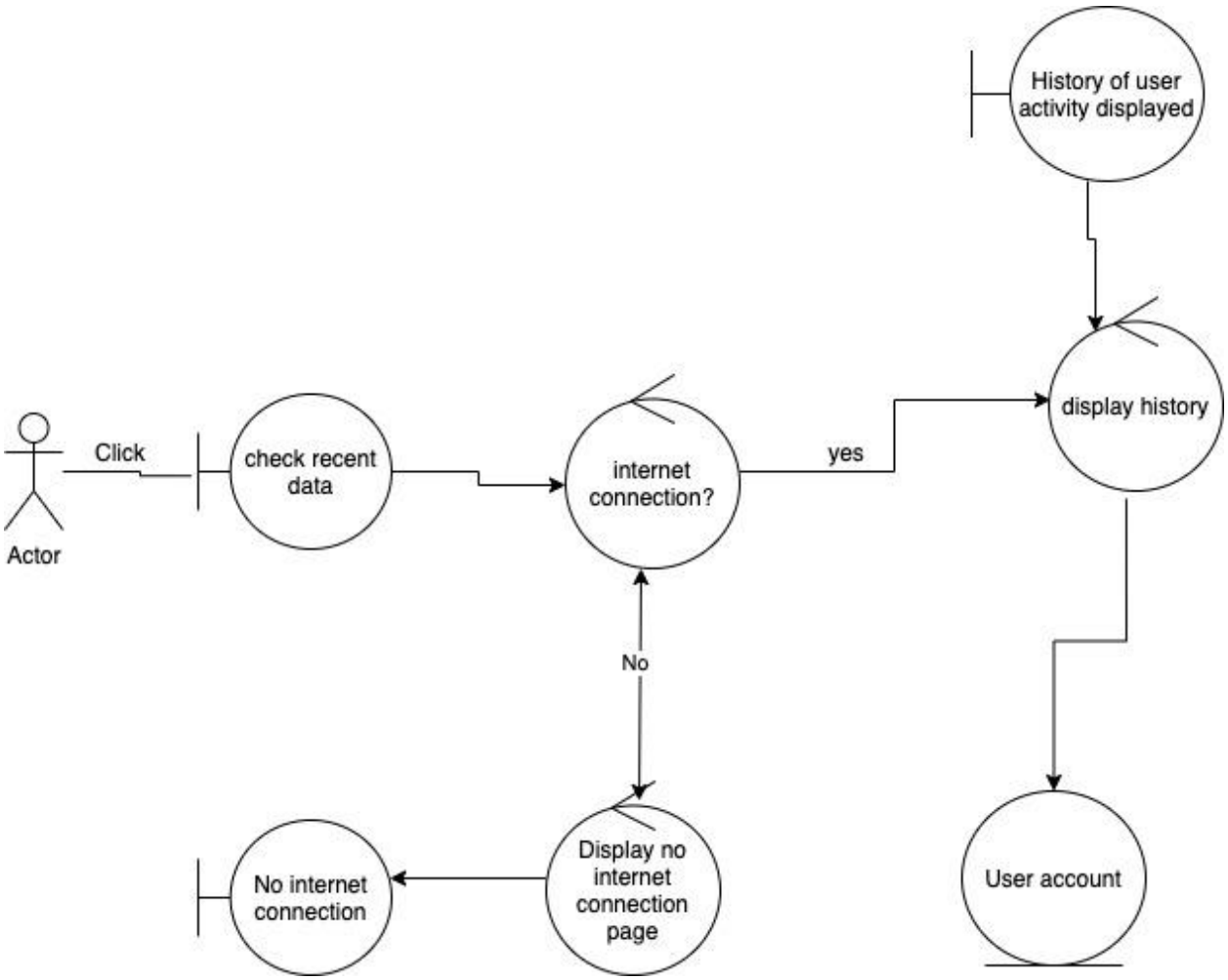


ROBUSTNESS DIAGRAMS

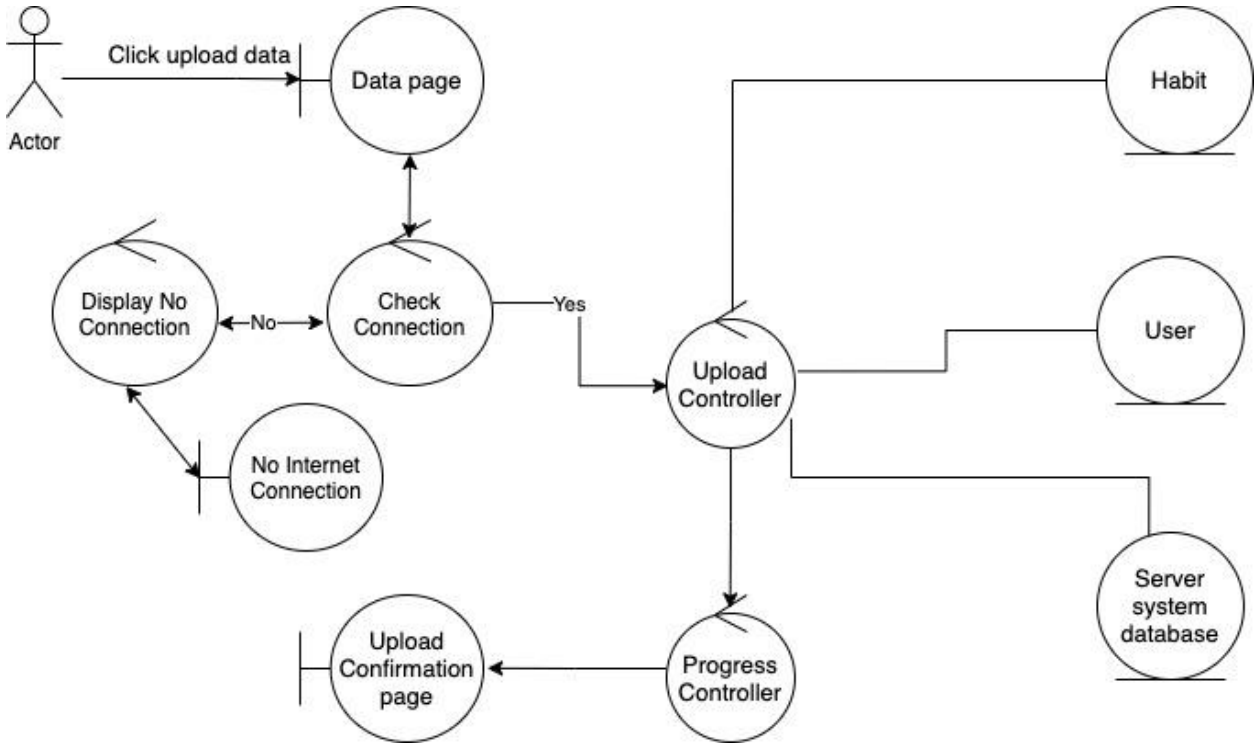
Robustness Diagram UC-1: Register An Account



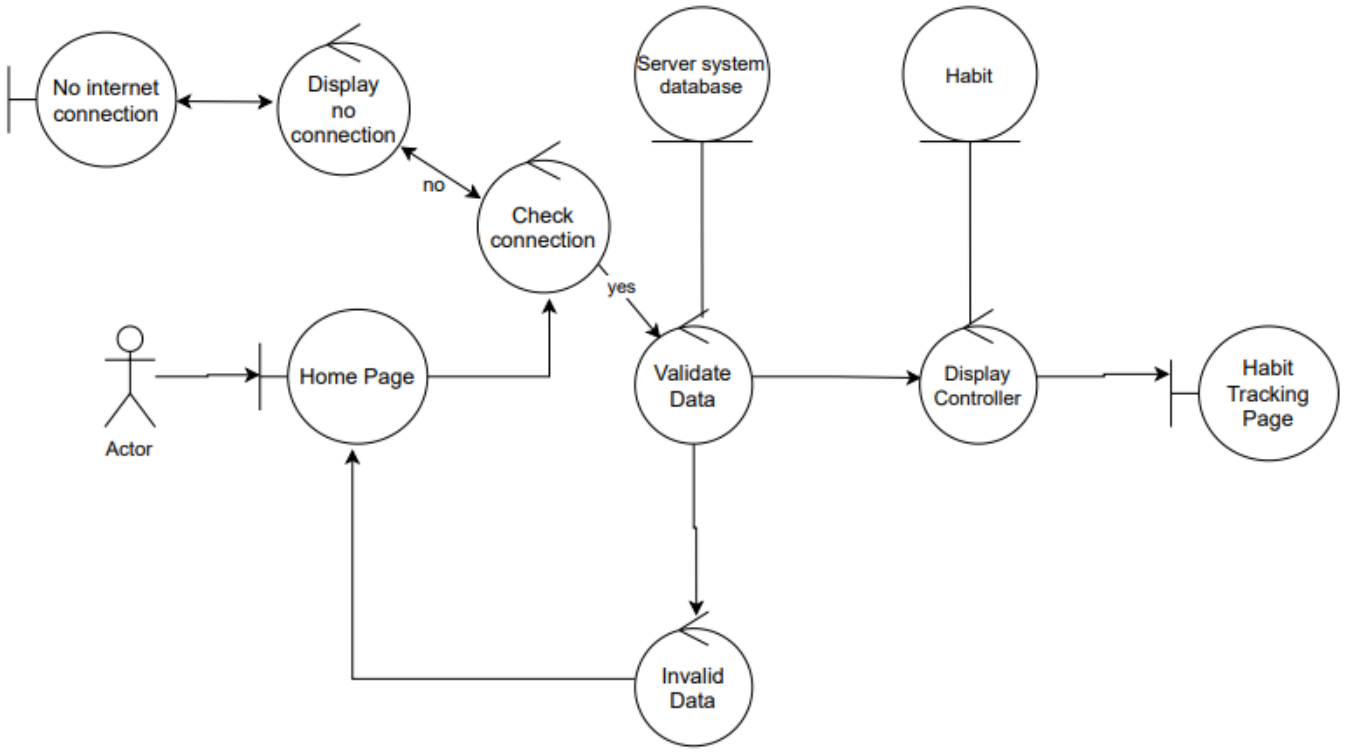
Robustness Diagram UC-2: Display History of User Activity



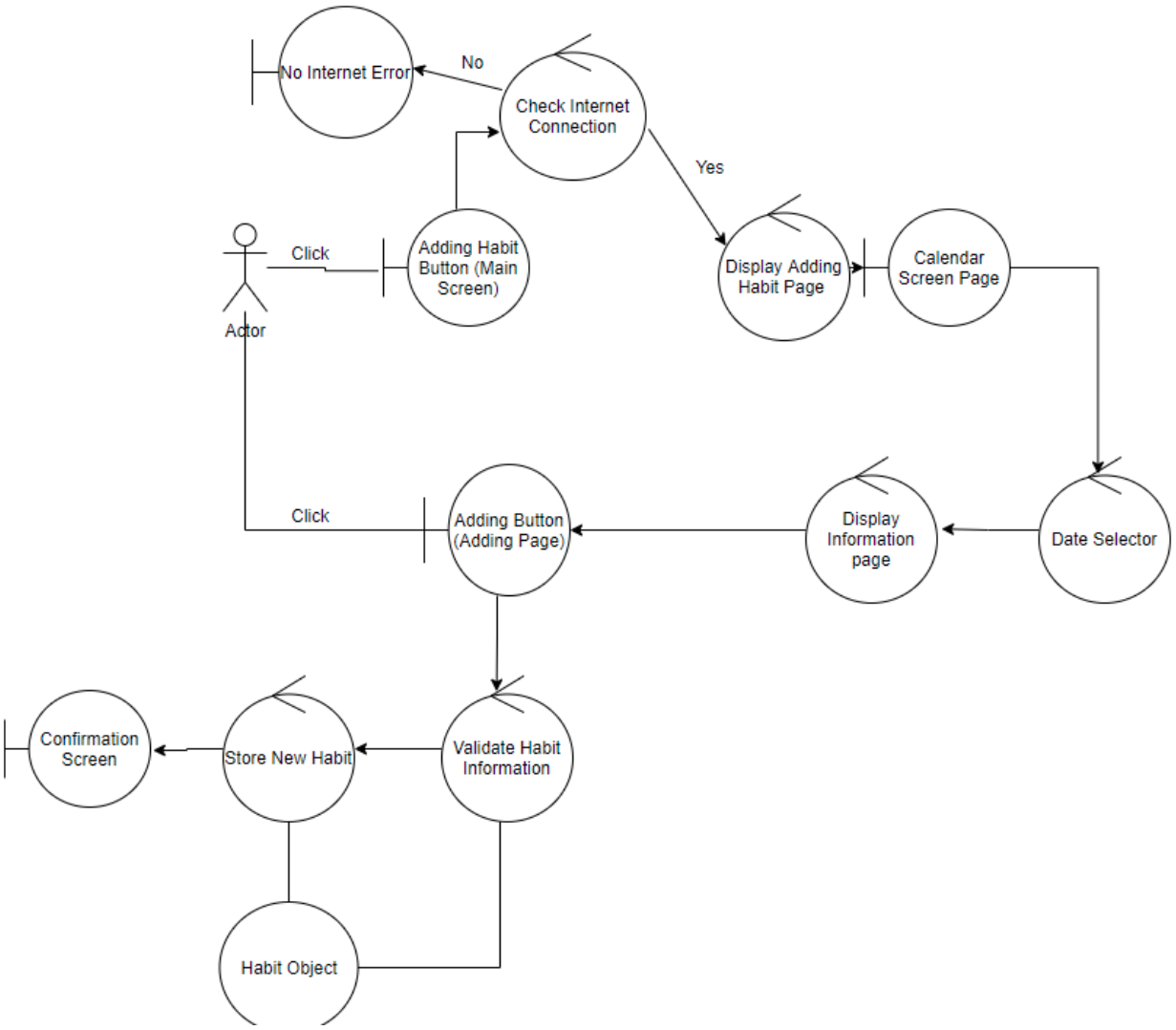
Robustness Diagram UC-3: Upload Data



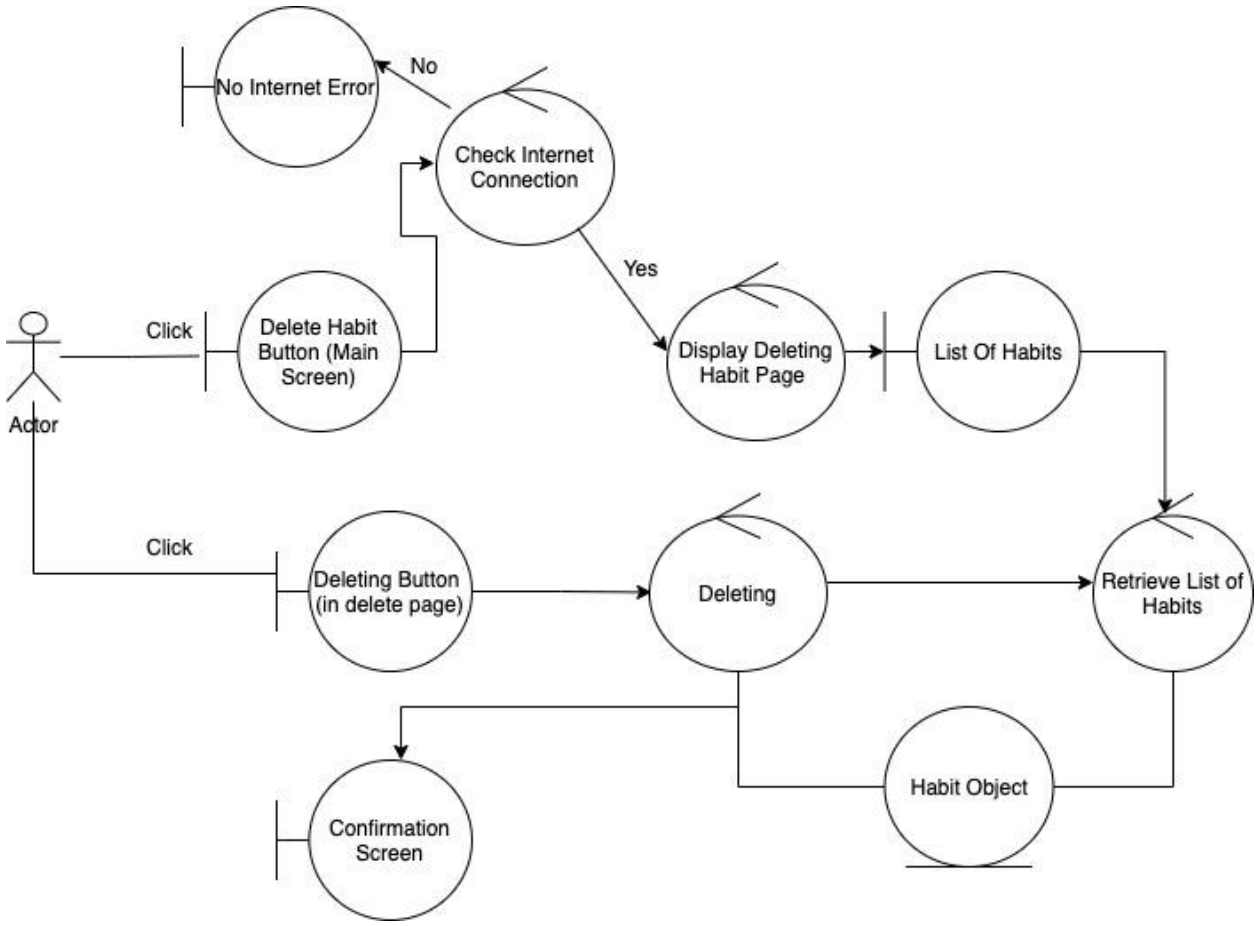
Robustness Diagram UC-4: Display Data



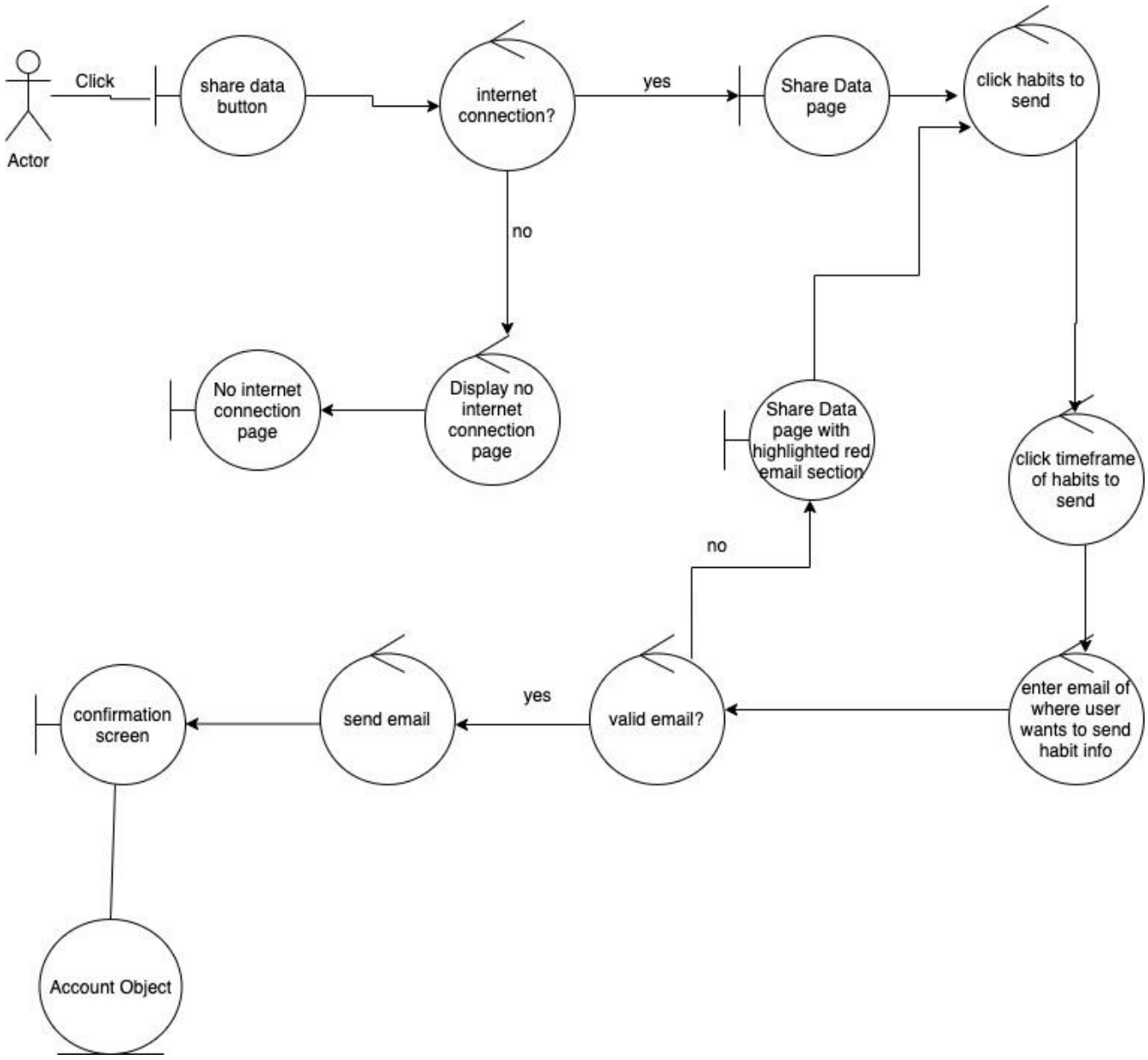
Robustness Diagram UC-5: Add Habit



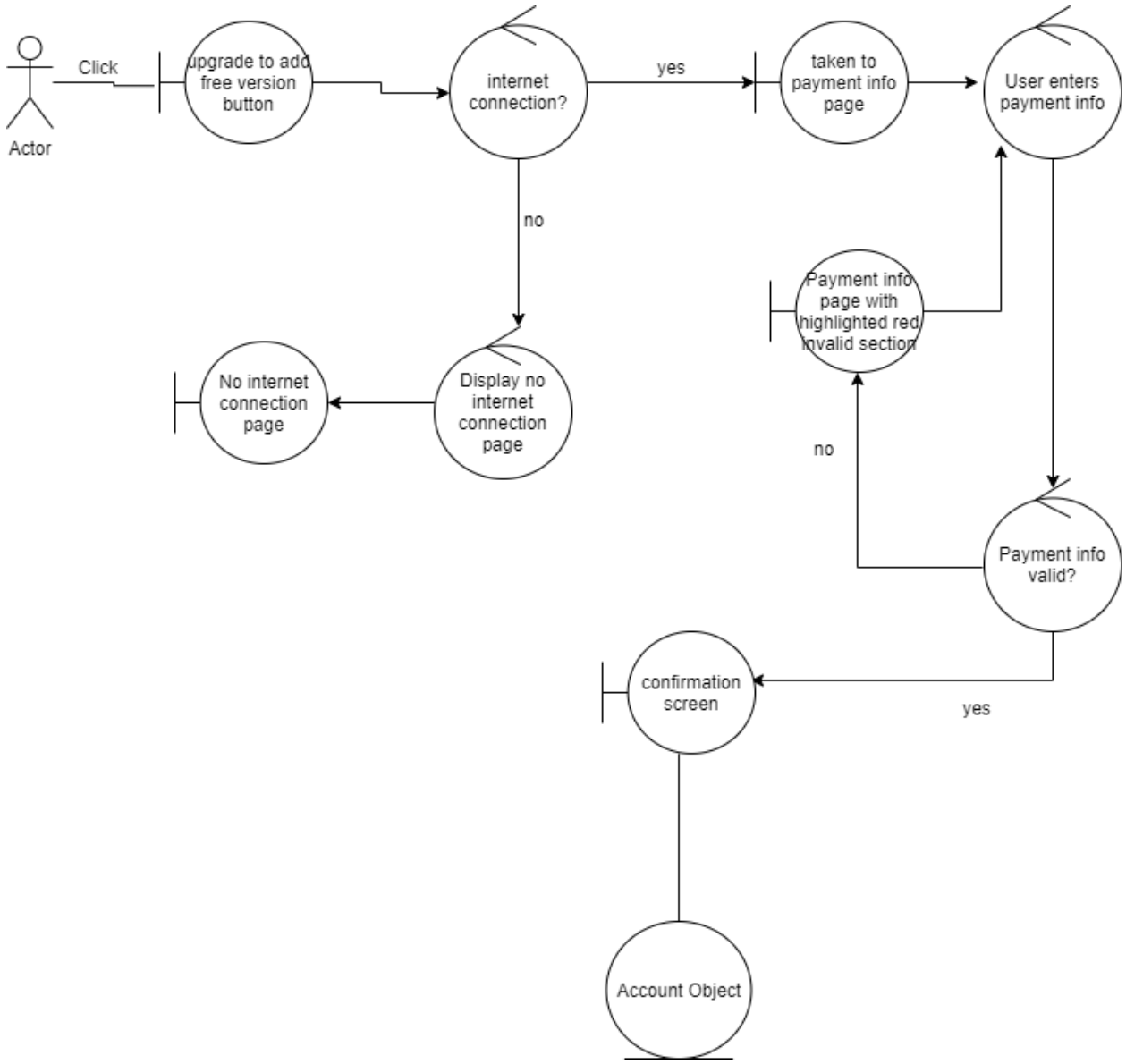
Robustness Diagram UC-6: Delete Habit



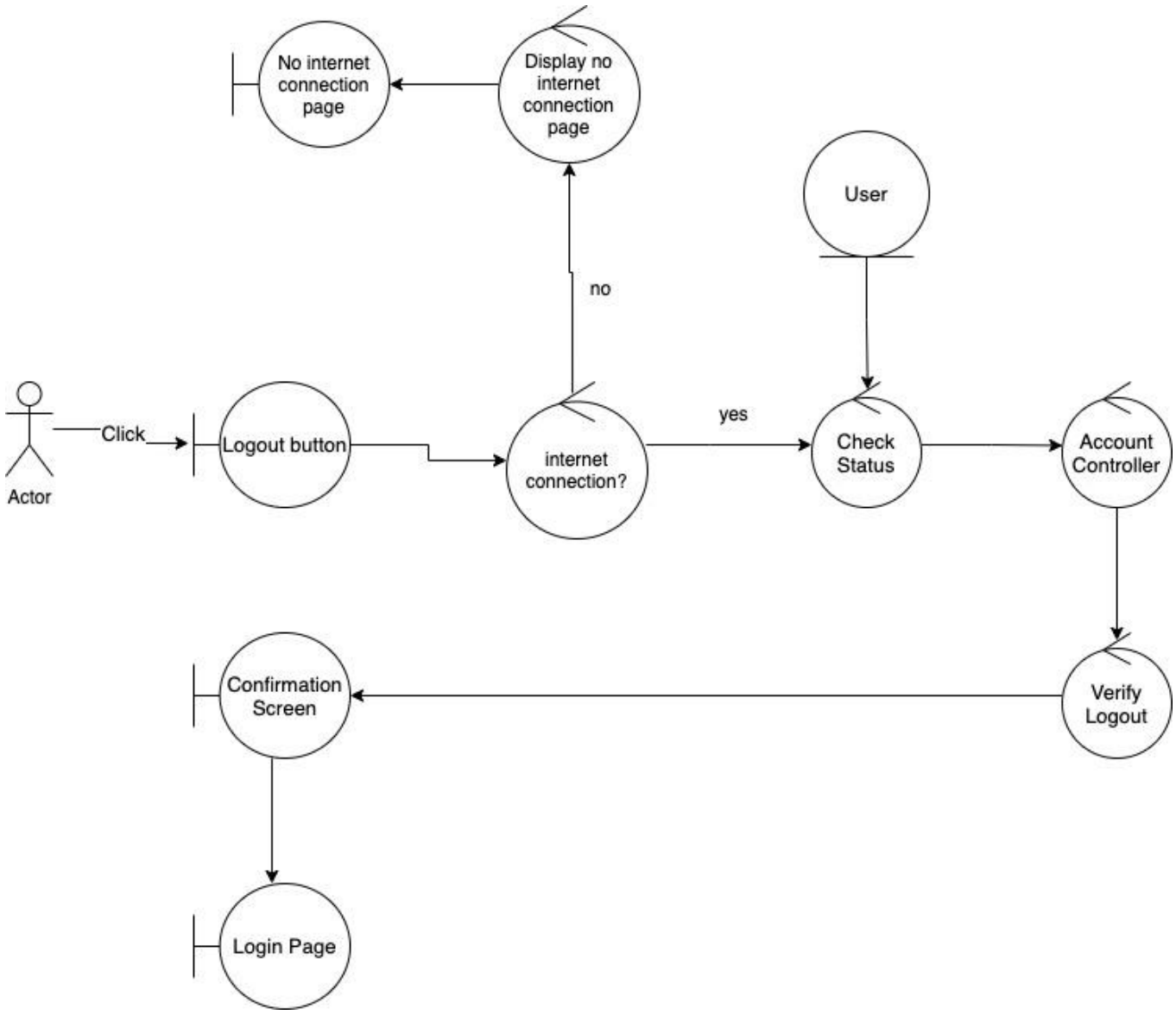
Robustness Diagram UC-7: User Shares Data Through Email



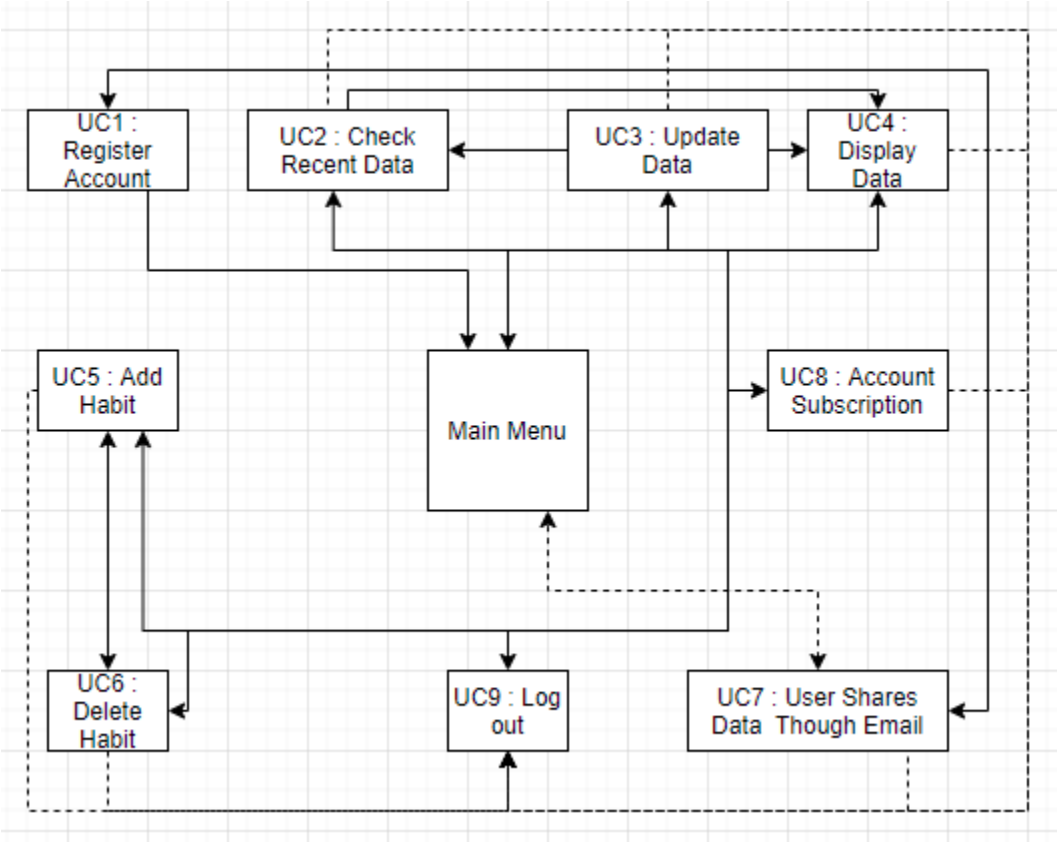
Robustness Diagram UC-8: Account Subscription



Robustness Diagram UC-9: Log out

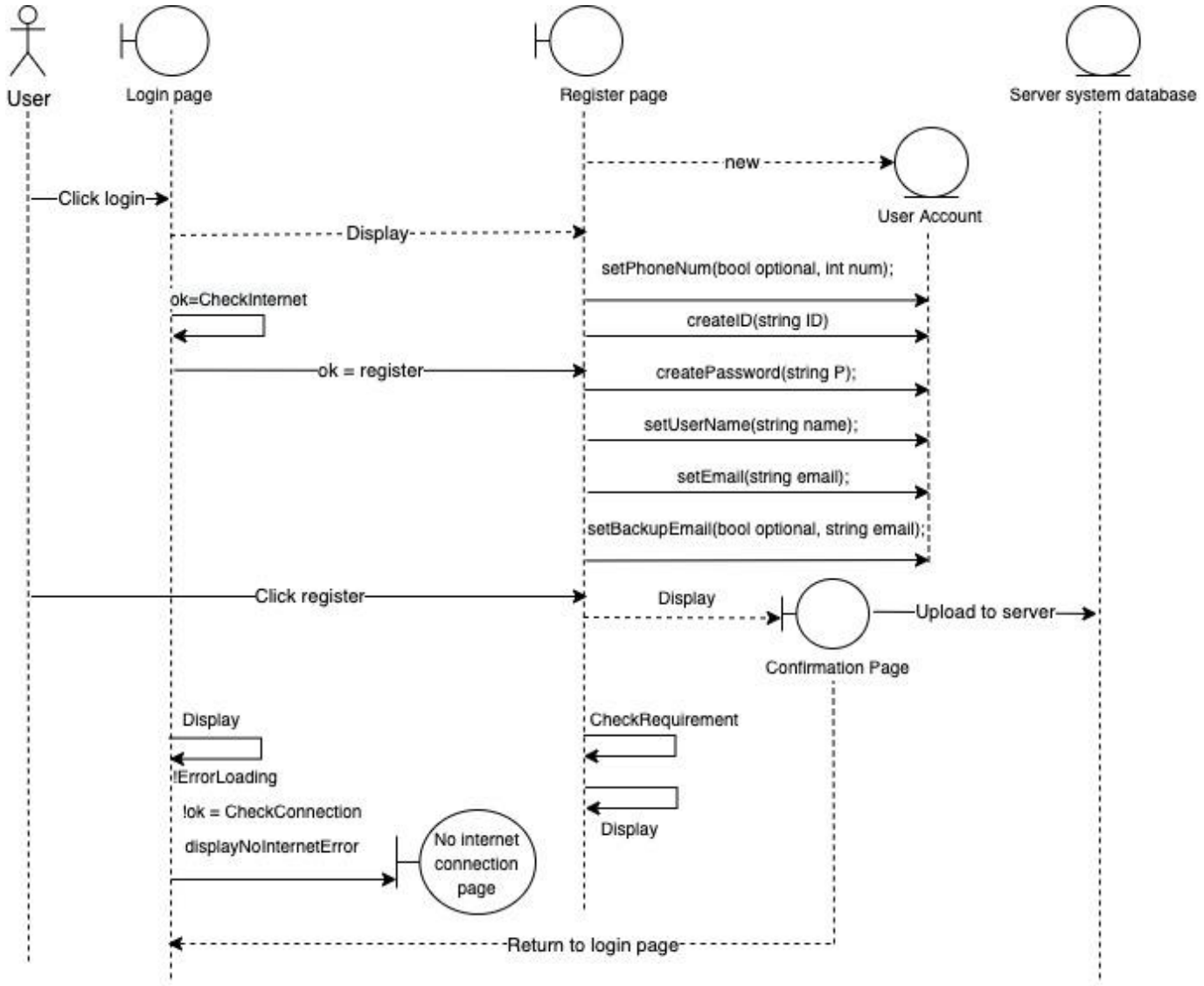


Use Case Connections

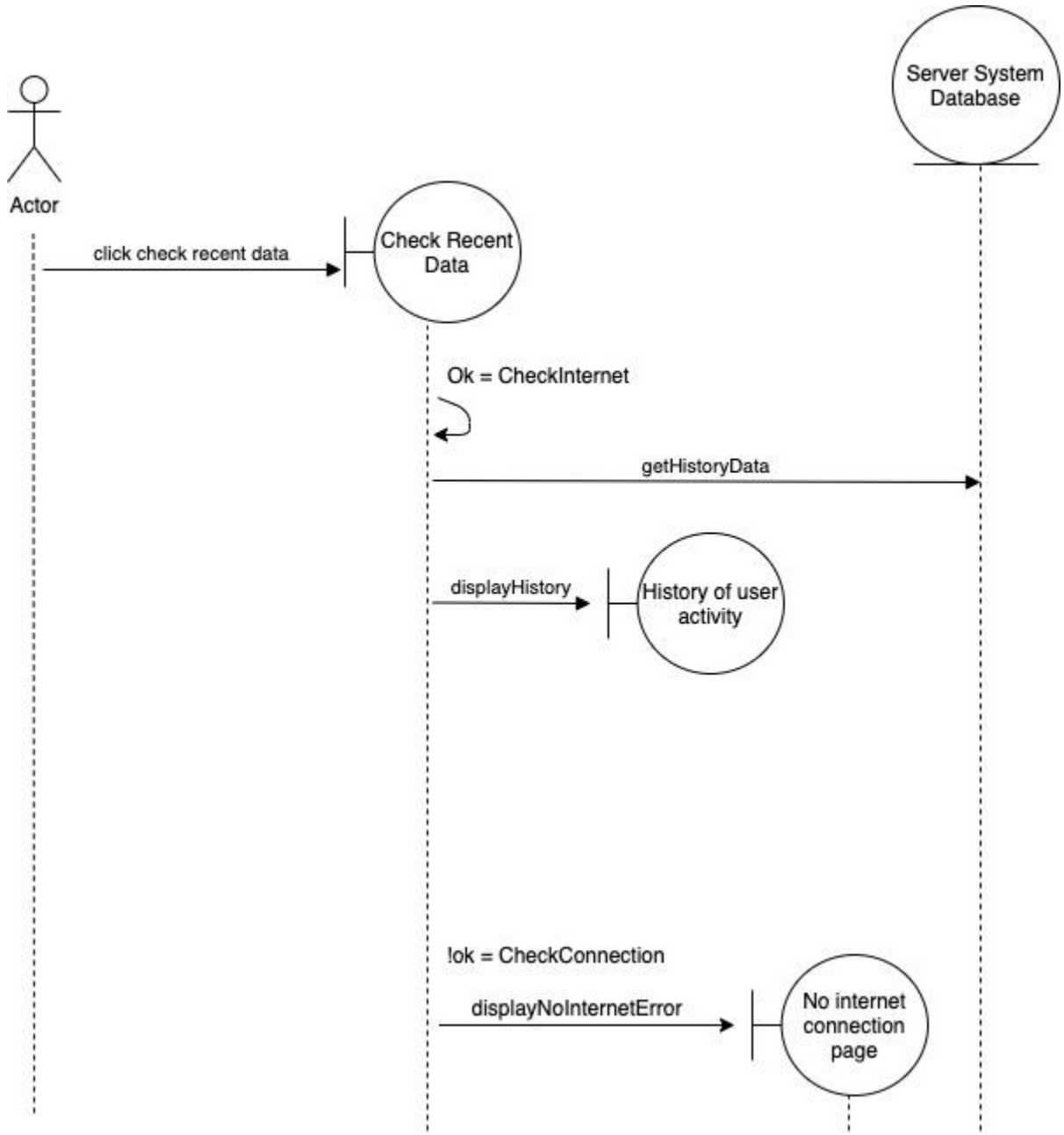


Sequence Diagrams

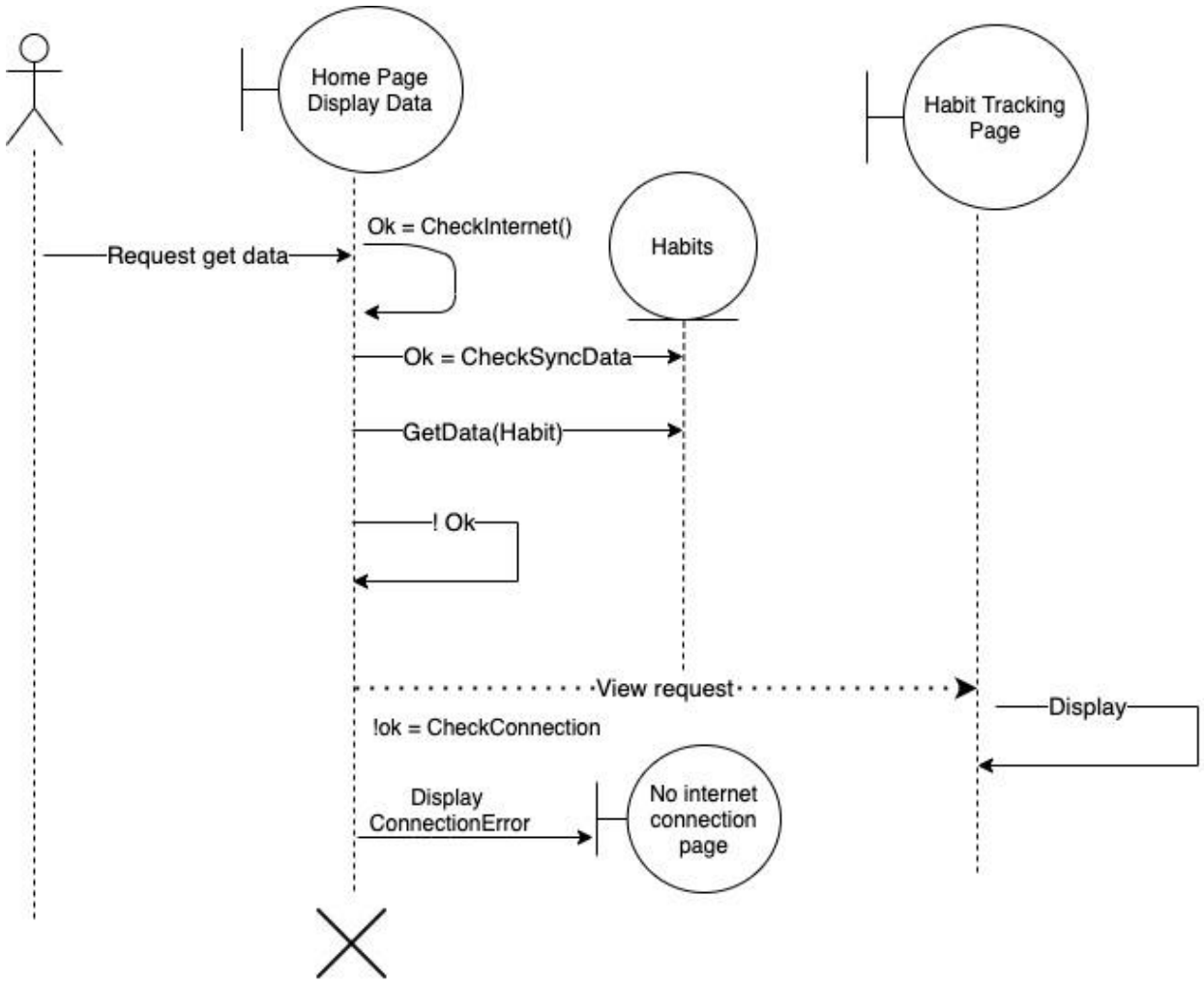
Sequence Diagram UC-1: Register An Account



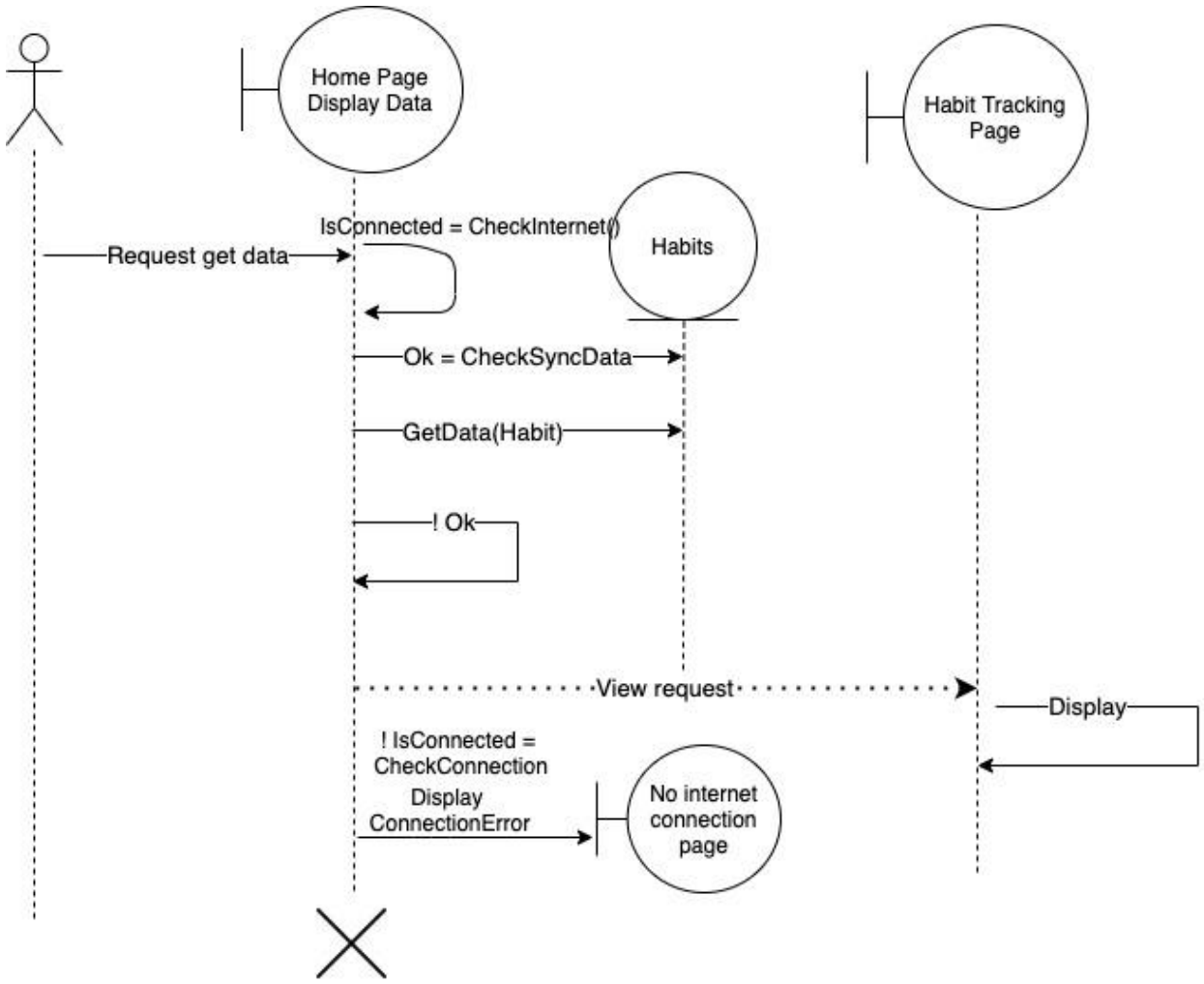
Sequence Diagram UC-2: Display history of user activity



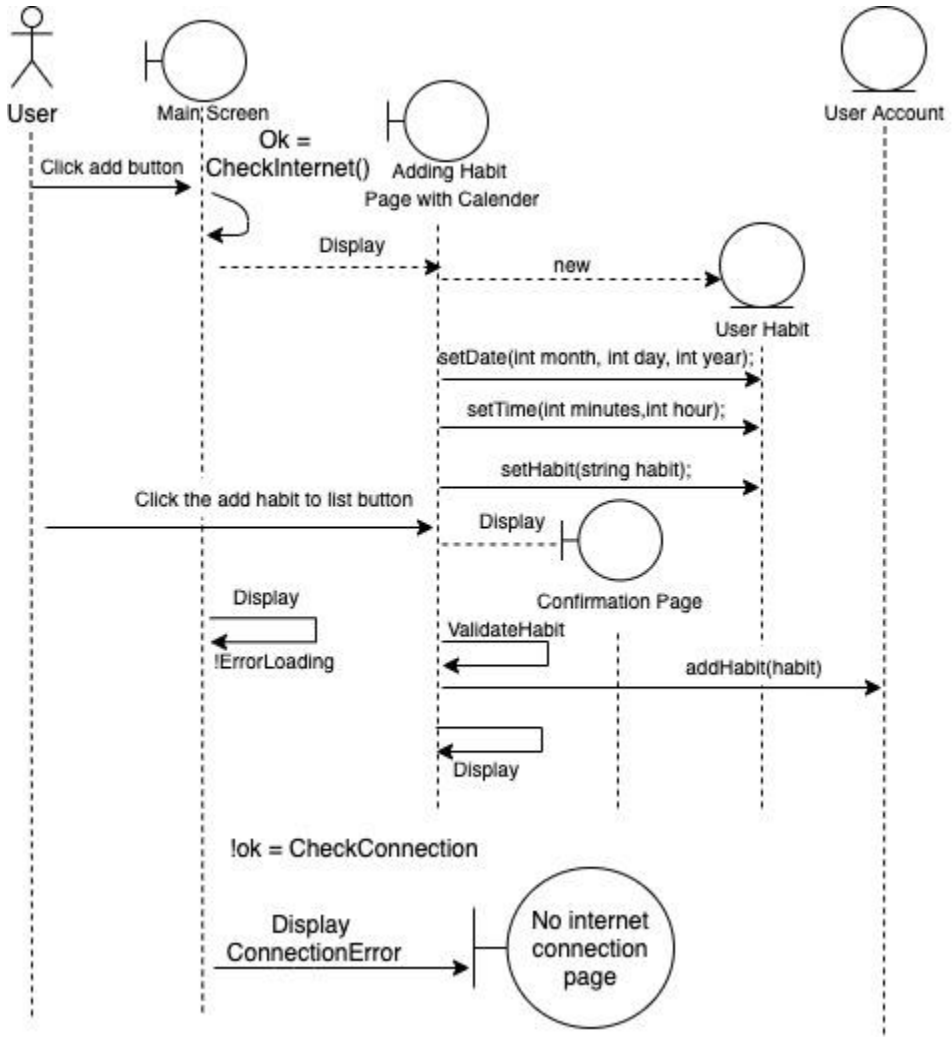
Sequence Diagram UC-3: Upload Data



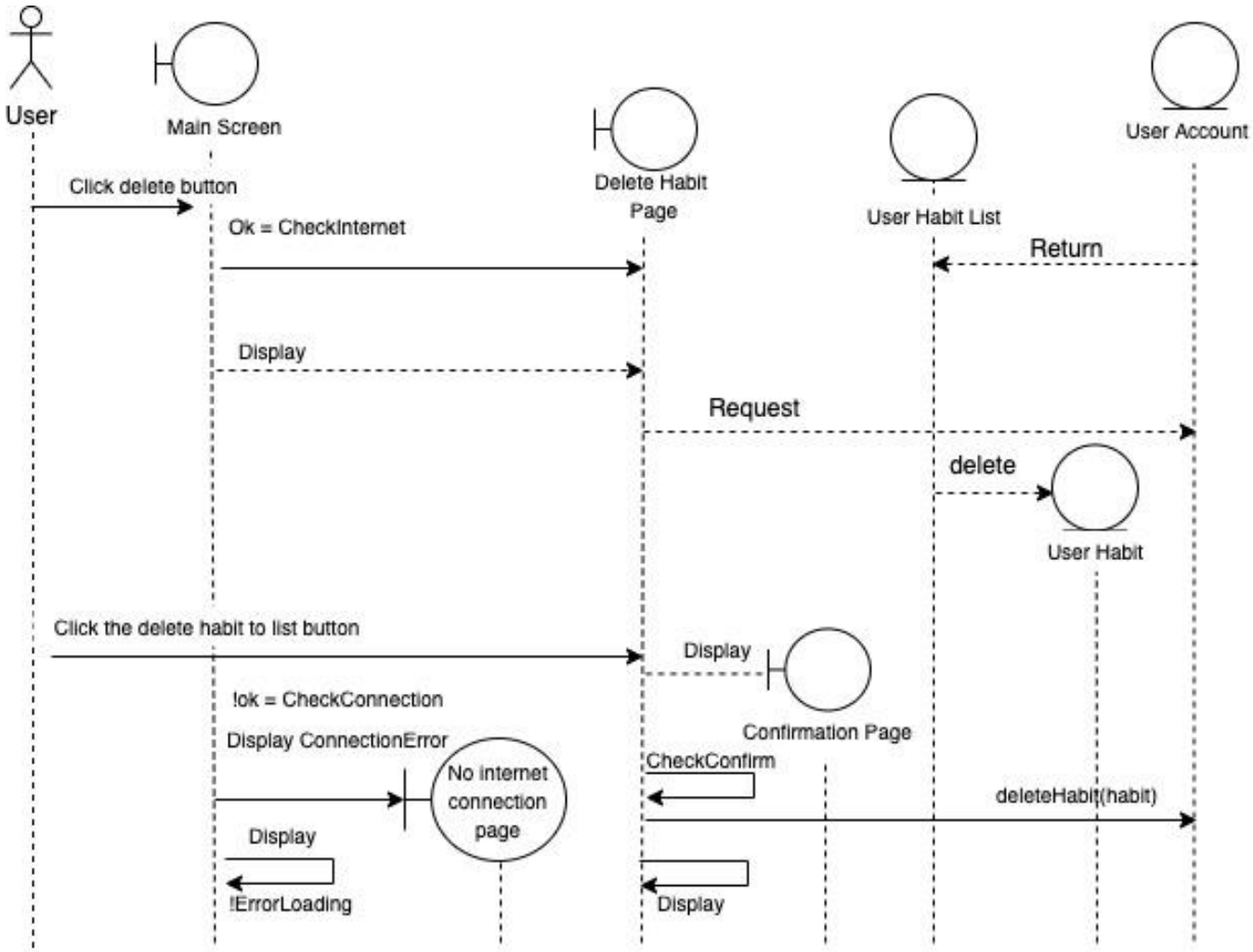
Sequence Diagram UC-4: Display Data



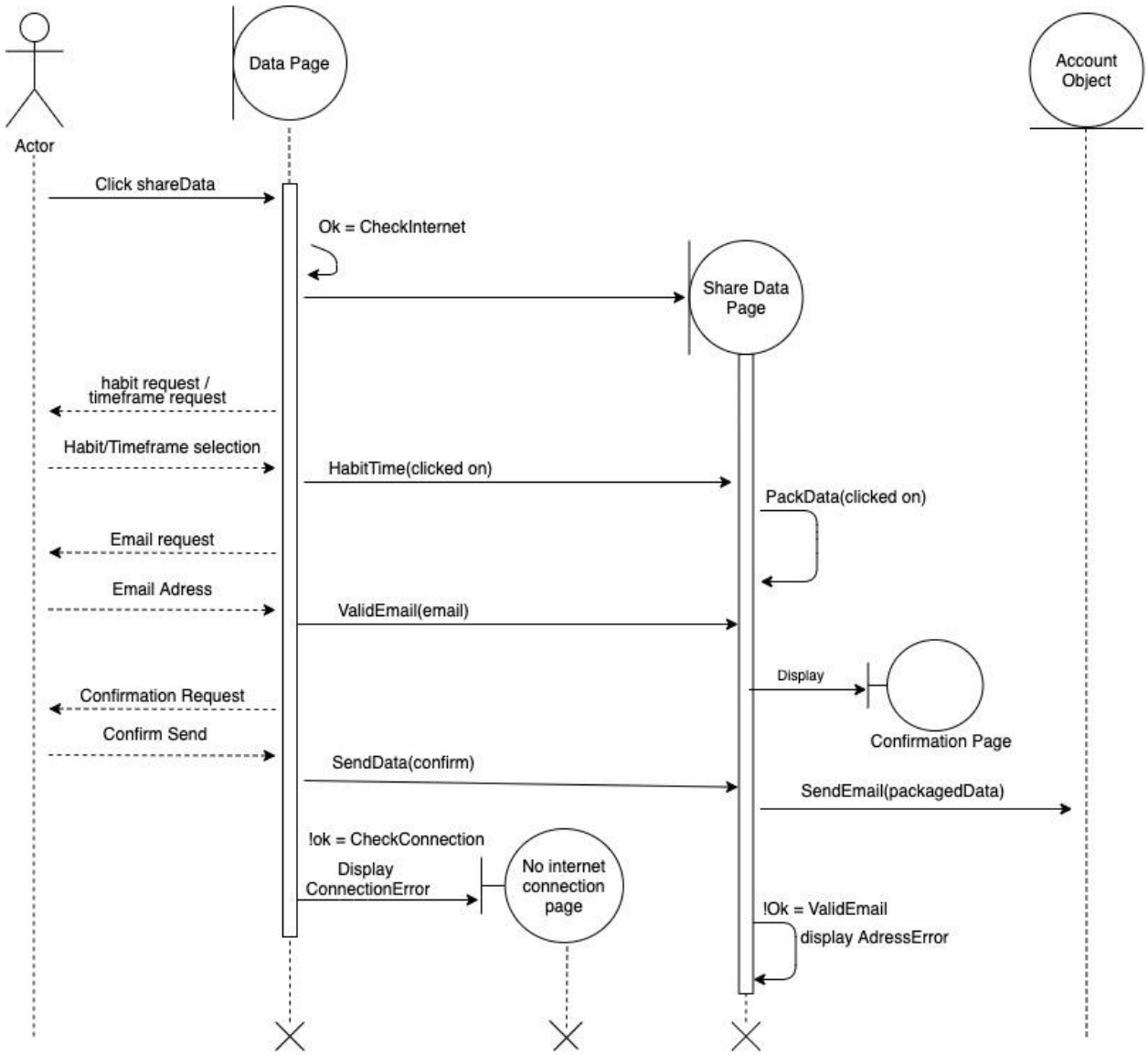
Sequence Diagram UC-5: Add Habit



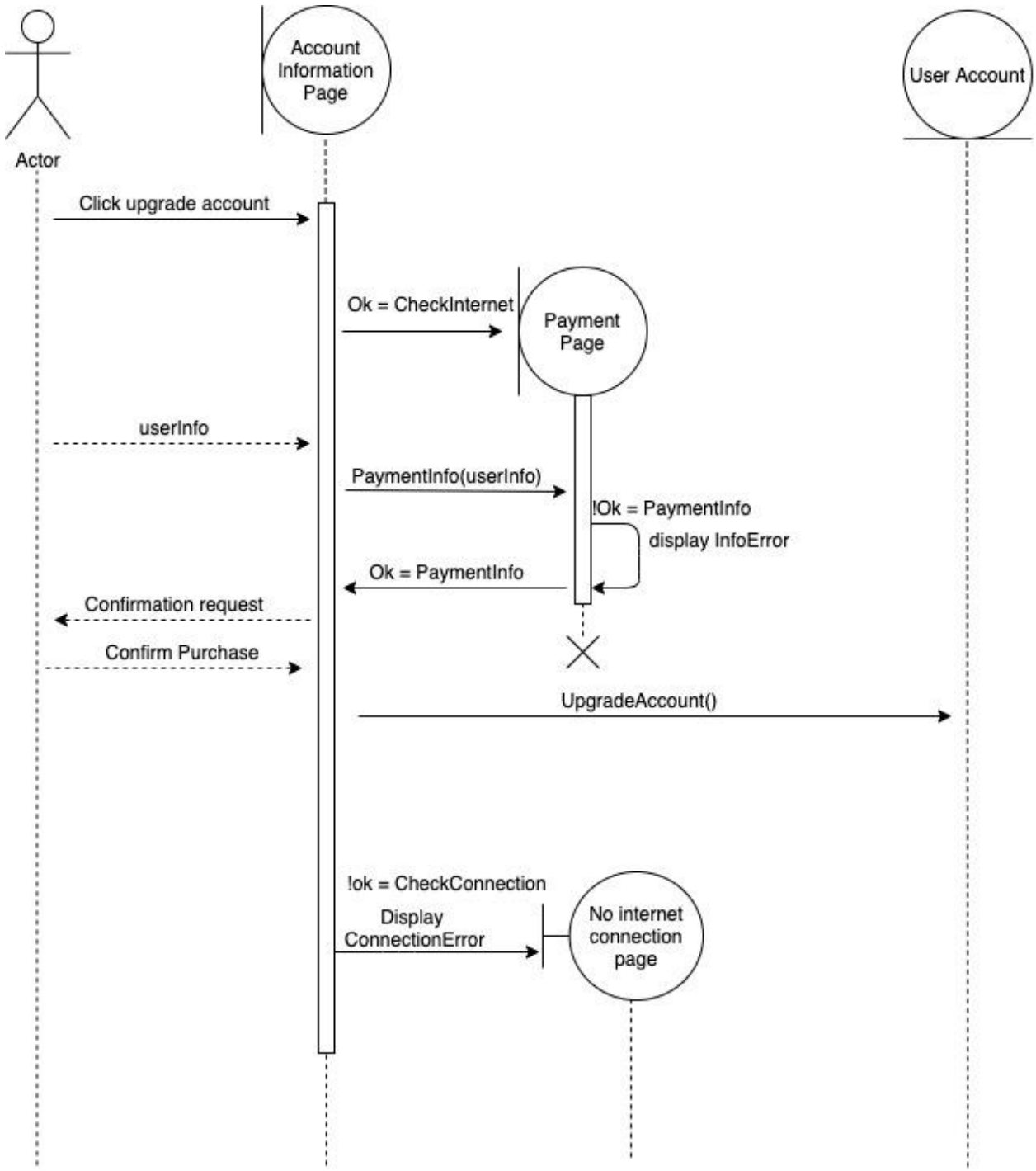
Sequence Diagram UC-6: Delete Habit



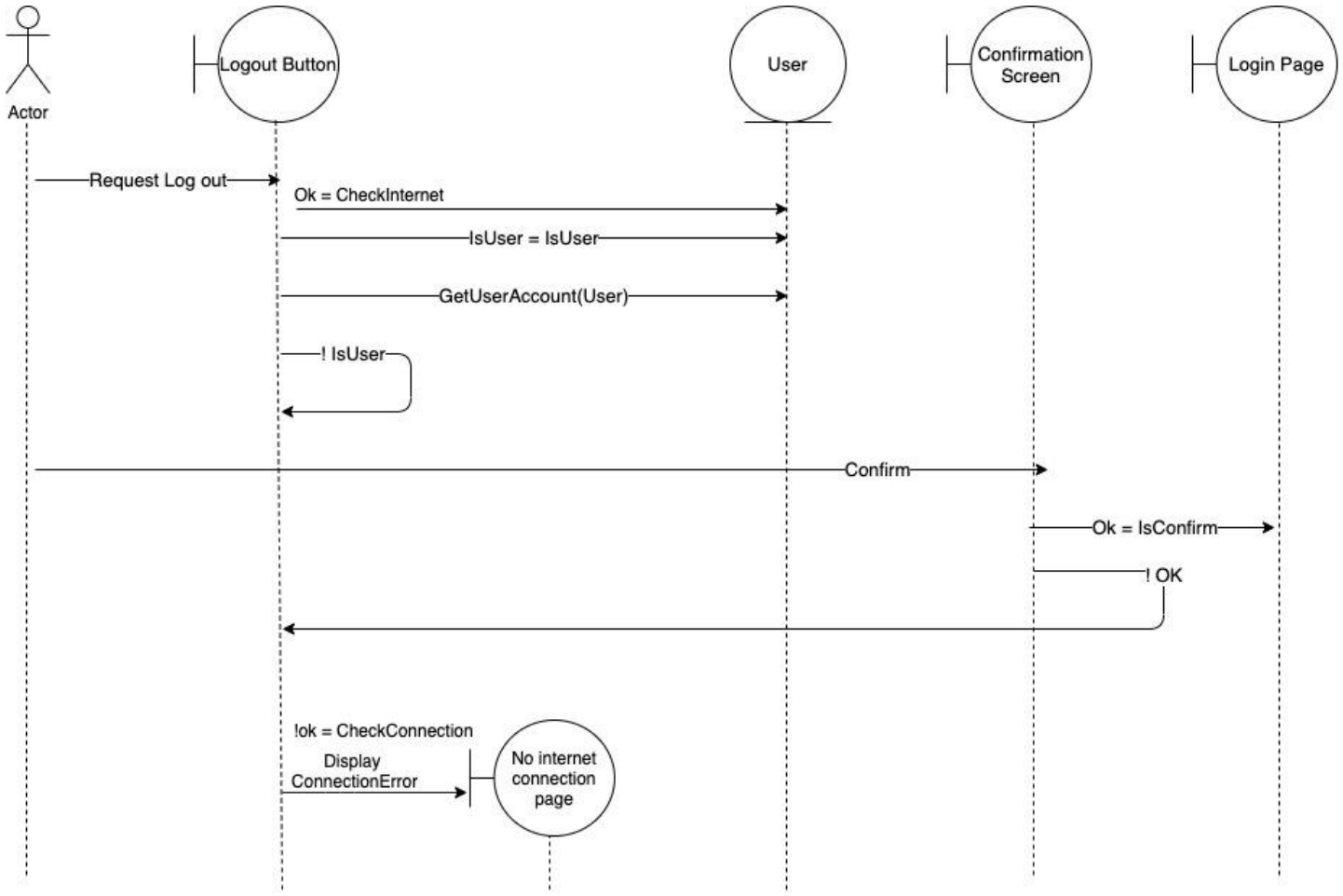
Sequence Diagram UC-7: User Shares Data Through Email



Sequence Diagram UC-8: Account Subscription

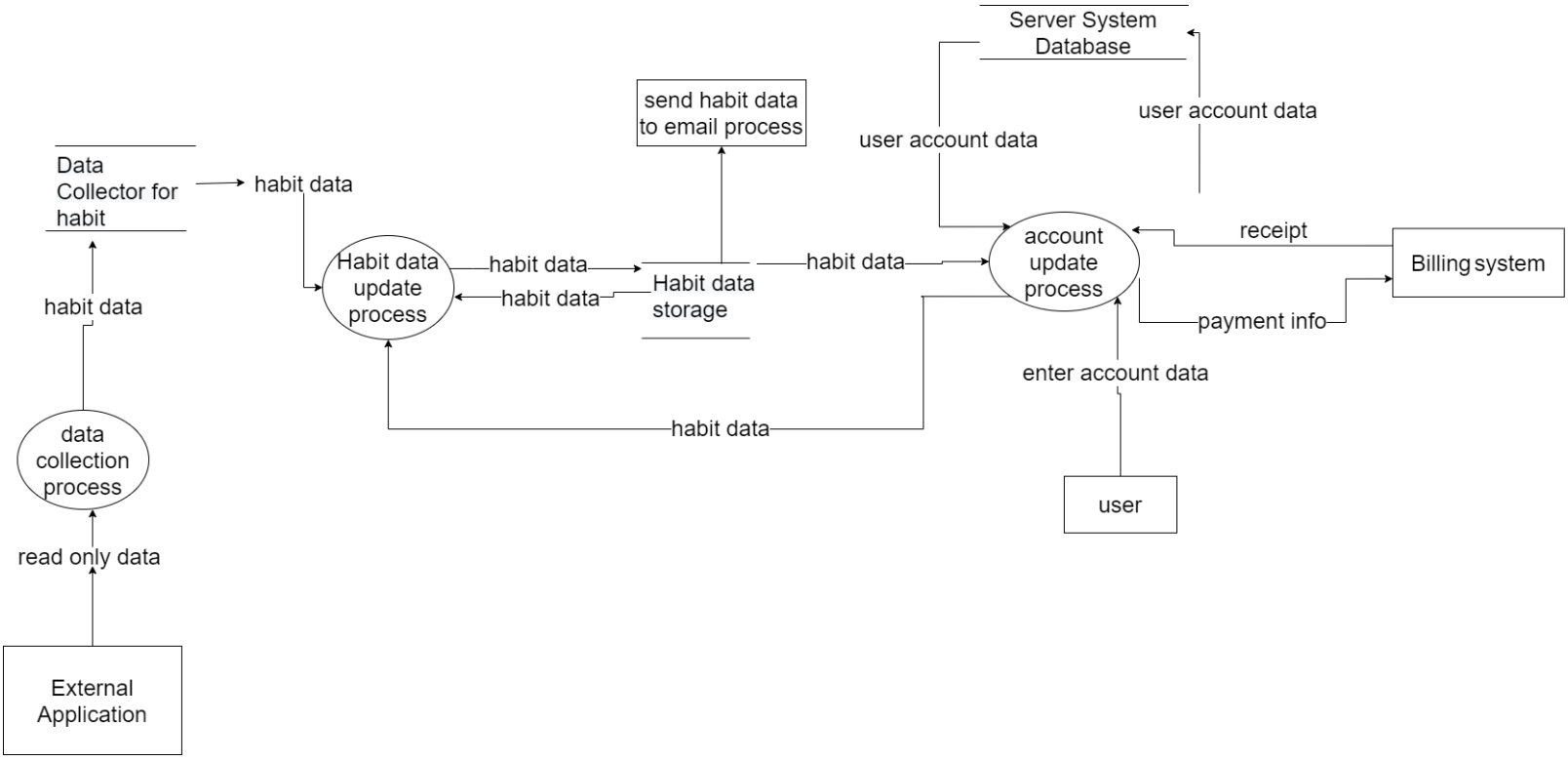


Sequence Diagram UC-9: Log Out

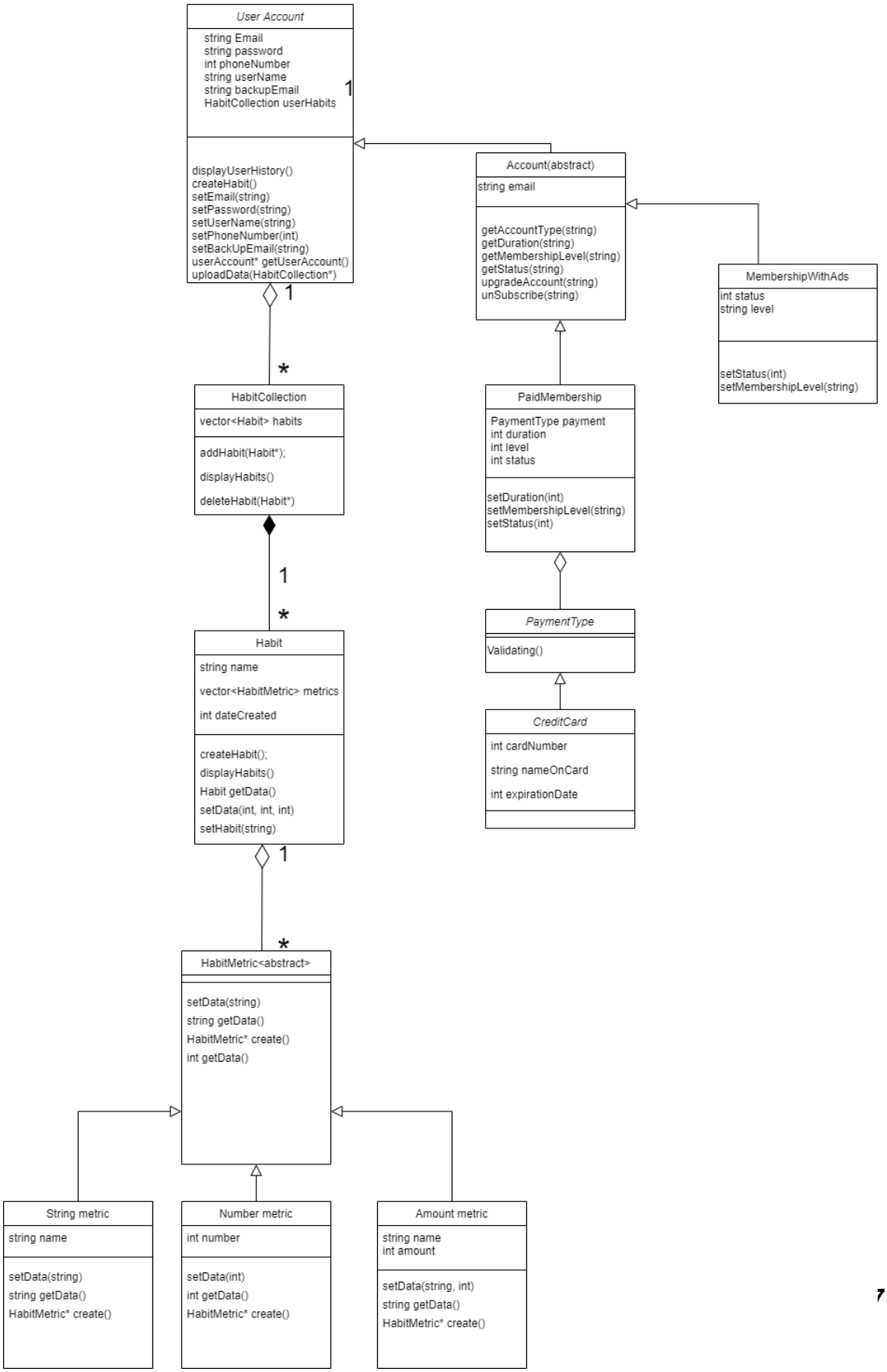


Data Flow Diagram

Level 1 Diagram



Class Diagram



Glossary

Term	Definition
Account	Specifies a user, and contains all their details, including email id and membership level.
Billing System	Controls all accounting and billing, and is the only point of contact to the third-party payment processing vendors.
Dashboard	The main page of the app, which shows a concise view of all the habits being tracked, and some of the important related metrics.
External Application	An external data collecting entity that supplies the habit tracker with relevant data. Could be something like a smartwatch or another app.
Habit	A recurring tendency or behavioral pattern that the user wants to track. Eg: smoking, workout, or sleep.
HabitCollection	A set of habits that the user wants to track. These can be added or removed, each with its own properties. All habits are displayed on the Dashboard.
HabitMetric	An abstract class that allows the user to measure each of their habits in one or more different metrics.
Master Account List	List of all user accounts, stored on the server system database.
Membership Level	Specifies whether a user has a paid membership or a free one (with ads).
Membership Status	Specifies whether a user's membership is active or inactive.
MembershipWithAds	The base membership level. It is free but contains regular ads for monetization.
Paid membership	The higher membership level, which contains no ads, but costs a regular fee.
Server System Database	Online repository of all the users' account data. All sensitive transactions and account changes are validated on the server. Regular backups of all recorded data and analytics are also maintained.
Upgrade Account	An option that allows users to switch from MembershipWithAds to PaidMembership.
Username	Identifying string used as a primary key for the Master Account List. Dispensed on a first-come-first-serve basis.

Bibliography

Rosenberg, Doug, and Matt Stephens. *Use Case Driven Object Modeling With UML: Theory and Practice*. Apress, 2016.

"Diagrams.net - Free Flowchart Maker and Diagrams Online." *Flowchart Maker & Online Diagram Software*, app.diagrams.net/.

Contact information

Name: Vivek Gandhi
Email: gandhiv@uw.edu

Name: Evan Kyle Jensen
Email: jensen55@uw.edu

Name: Majid Khoshakhtar
Email: khoshm@uw.edu

Name: Connor Wilding Leonie
Email: cwleonie@uw.edu

Name: Thong Ton
Email: tonthong@uw.edu

Name: Duy Vu
Email: duyvu263@uw.edu