# 

# 

# 

# 

# 

# CSC 431 FundsBook Software Requirements Specification (SRS)

**Team 16**

|  |  |
| --- | --- |
| Luis Diaz | Developer |
| Julio Ojalvo | Developer |
| Temuulen Ganbold | Developer |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Comments |
| 1 | 2/11/21 | Luis Diaz,  Temuulen Ganbold,  Julio Ojalvo | * Added functional and non-functional system requirements |
| 2 | 3/9/21 | Luis Diaz,  Temuulen Ganbold,  Julio Ojalvo | * Corrections and additions made to functional and non-functional system requirements. * Added Use Case Models * Added System constraints and evolutionary requirements * Added Table of Tables * Added Table of Figures |
| 3 | 4/30/21 | Luis Diaz,  Temuulen Ganbold,  Julio Ojalvo | * Fixed NFRs that were misallocated * Added a main menu * Fixed errors in some requirement’s flows * Removed inappropriate system constraints |
|  |  |  |  |

# Table of Contents

[1.](#_3dy6vkm) System Requirements 6

[1.1](#_1t3h5sf) Functional Requirements 6

[1.2](#_2s8eyo1) Non-Functional Requirements 13

[2.](#_3rdcrjn) System Constraints 13

[2.1](#_26in1rg) Tool Constraints 13

[2.2](#_35nkun2) Language Constraints 14

[2.3](#_44sinio) Platform Constraints 14

[2.4](#_z337ya) Hardware Constraints 14

[2.5](#_1y810tw) Network Constraints 15

[2.6](#_2xcytpi) Deployment Constraints 15

[2.7](#_3whwml4) Transition & Support Constraints 15

[2.8](#_qsh70q) Budget & Schedule Constraints 16

[2.9](#_1pxezwc) Miscellaneous Constraints 16

[3.](#_2p2csry) Requirements Modeling 17

[4.](#_3o7alnk) Evolutionary Requirements 21

[4.1](#_32hioqz) Non-Functional Requirements 21

# Table of Tables

1. System Requirements 6

1.1.1 Profile Creation 6

1.1.2 Profile Editing 6

1.1.3 Personal and Auxiliary Budgets 7

1.1.4 Followers 8

1.1.5 Friends 8

1.1.6 Close Friends 8

1.1.7 Searching 9

1.1.8 Achievements 9

1.1.9 Notifications 10

1.1.10 Privacy 10

1.1.11 Bank Account Linking 11

1.1.12 Suspicious Activity Detection 12

1.1.13 Main Menu 12

1.2.1 Information Encryption 13

1.2.2 Database Management System (DBMS) 13

1.2.3 Search Speed 13

2. System Constraints 13

2.1.1 Environment 13

2.2.1 Java 14

2.2.2 Swift 14

2.3.1 Android 14

2.3.2 IOS 14

2.4.1 Equipment 14

2.5.1 Connectivity 15

2.6.1 Publishing 15

2.7.1 Transitionary Requirements 15

2.7.2 Continued Maintenance 15

2.8.1 Time Management 16

2.9.1 Communication 16

4. Evolutionary Requirements 21

4.1.1 Data logging

# Table of Figures

3. Requirements Modeling

[3.1.1 Profile login 1](#_147n2zr)7

3.1.2 Profile editing 17

3.1.3 Budget management 18

3.1.4 Auxiliary budget 19

3.1.4 Socialization 20

### 1. System Requirements

#### 1.1 Functional Requirements

##### 1.1.1 Profile creation

|  |  |
| --- | --- |
| Title | Creating a Profile |
| Description | Users will create their personal profile that others will be able to see and where they can look through their budget plans and badges |
| Priority | 0 |
| Precondition(s) | The app must be downloaded from the app store |
| Basic Flow | * Upon entering the app, the user is prompted with a screen to choose what language they want to see the app in * After this, the user is prompted with a page to either sign in to an existing one or create a profile * If they choose to sign in, they will be prompted with an email or username and password box and will be entered into their profile if the credentials match a profile on record * If the user has forgotten their password, an option will be given to email them a link to reset it * If the user is new, they will create an account and choose an email and username that hasn’t been taken as well as a password with at least 8 characters, a lowercase and uppercase letter, and one special character * On the last step of creating the profile, users will receive a confirmation email and when the link in the email is clicked, they will be allowed into the app |
| Postconditions(s) | The user can now edit their profile and use the major functionalities of the app |
| Use Case Diagram | 3.1.1 |

##### 1.1.2 Profile editing

|  |  |
| --- | --- |
| Title | Editing a Profile |
| Description | Users can edit their profile by adding budget plans and badges to be displayed for others as well as a profile picture and banner behind the profile picture |
| Priority | 0 |
| Precondition(s) | A profile must be created and logged into |
| Basic Flow | * Once the user creates a profile, they will be presented with a blank one that they can edit * A blank profile picture when clicked prompts the user to add one of their own, same as the banner behind it * Another button to add and delete budget plans will be present * An add button for badges will allow the user to look through all their badges and add up to 7 to their page for others to see |
| Postconditions(s) | Once the user makes an edit, it will be updated so everyone else can see their changes when they go into that person’s profile |
| Use Case Diagram | 3.1.2 |

##### 1.1.3 Personal and Auxiliary Budgets

|  |  |
| --- | --- |
| Title | Creating a personal budget |
| Description | Ability to create a personal budget plan and put money into certain savings accounts based on the plan created by the user. Auxiliary budget plans can also be created that others can see and that can be swapped out for the personal one at any moment |
| Priority | 0 |
| Precondition(s) | The user must have completed the registration process and gone to their profile |
| Basic Flow | * User clicks a button to enter into a screen for editing budgets * User has a box labeled “category” to type in the name of one of the aspects of the plan e.g. car payments, cell phone bill, credit cards * Under each category, the user can input what percent of the total budget or how much money in dollars is going towards it * User can choose what color that category will be represented in * A pie chart will be shown to the side to demonstrate the totals of each category in the color chosen * If the user inputs a percent that brings the total over 100%, a warning is given, and the number is removed * Once, the user is done and clicks the finish button they are given a notification if their budget plan does not reach 100% total, but they are allowed to continue, and that remaining money will be left untouched * Personal budget plans are the ones currently being used by the user while auxiliary ones are on their profile either public or private as they choose * At any moment, the user can switch an auxiliary one with their personal one and use that budget plan instead |
| Postconditions(s) | Once the main budget plan is made the user will be returned to the profile page where they can see the status of their budget plan |
| Use Case Diagram | 3.1.3 |

##### 1.1.4 Followers

|  |  |
| --- | --- |
| Title | Ability to Follow and use Others’ Budget Plans |
| Description | In the given case that successful millionaires/users decide to interact with this app and be some sort of guidance for other users, users should be able to follow those influencers and financial advisor to be able to use as a reference to meet their budgeting plans |
| Priority | 4 |
| Precondition(s) | Must have a profile to follow from |
| Basic Flow | * Users can click on other’s profiles to see their plans as long as the profile is public * If the profile is private, an option to request to follow them is given and a request is sent to that profile owner where they can accept or deny it * If the profile is public, an option to follow is given which when clicked will go through without any further processing |
| Postconditions(s) | User now receives notifications when the profile they followed makes a change if they choose to be notified of it in the settings |
| Use Case Diagram | 3.1.5 |

##### 1.1.5 Friends

|  |  |
| --- | --- |
| Title | Ability to Add Others as Friends |
| Description | Users should be able to interact with other users, having the option to add other users as “friends” and be able to see how they are doing with their budget plans and in a way give each other motivation to reach those goals. |
| Priority | 4 |
| Precondition(s) | Two people must follow each other in order to become friends |
| Basic Flow | * Once two users have chosen to follow each other, they will become friends |
| Postconditions(s) | Once friends, users can privately message each other and choose who will be on their close friends list |
| Use Case Diagram | 3.1.5 |

##### 1.1.6 Close Friends

|  |  |
| --- | --- |
| Title | Ability to designate close friends |
| Description | Users can choose who on their friends list is a close friend |
| Priority | 4 |
| Precondition(s) | User must be friends with someone in order to have the choice of adding them to their own close friends list |
| Basic Flow | * When a user looks at their own profile, there will be a tab to see your close friends * Initially it will say “No close friends yet” and prompt them to click on a button that will take them to a list of their friends * From there they can click on all the friends they would like to add to their close friends list and allow for more detailed customization   + Allow notifications only from close friends   + Allow only close friends to see these budget plans   + Allow only close friends to see all my badges   + Send this message out to only close friends |
| Postconditions(s) | A user that has been added to another’s friends list can see more detailed information when they check their profile |
| Use Case Diagram | 3.1.5 |

##### 1.1.7 Searching

|  |  |
| --- | --- |
| Title | Ability to search for users or budget plans |
| Description | Users will have a search bar they can use to find either a certain type of budget plan or another user’s profile |
| Priority | 1 |
| Precondition(s) | Must have a profile |
| Basic Flow | * One of the main tabs on the app will bring the users to the search bar * On the search bar, user can perform three types of searches   + Users can search by tag to find a certain type of budget plan e.g., long term investments, saving for an expensive car in a short amount of time, paying off high interest loans or credit card   + Users can search for other users based on their profile name   + Users can search for budget plans by their names |
| Postconditions(s) | Once a user is found, they can be followed and once a plan is found it can be saved as an auxiliary plan that can be switched over to the main budget plan at any moment |
| Use Case Diagram | 3.1.1 |

##### 1.1.8 Achievements

|  |  |
| --- | --- |
| Title | Achievement system |
| Description | An achievement system that will reward users for making budget plans, using tags in them, and following their own budget plans for long periods of time |
| Priority | 3 |
| Precondition(s) | A main budget plan must be made and a profile to display the achievements from |
| Basic Flow | * Users will receive a small badge on their profile if they fulfill certain achievements such as   + Following a budget plan for one week, one month, one year, and every subsequent year   + Making 5 different budget plans   + Adding 10, 20, and 50 different tags among all the plans made * Once a user sets a budget plan as their main one, a counter will start to keep track of how long since it’s been the main one. That counter will be used to determine when to give the badges out * A counter for total number of budget plans made and tags used will be used for those badges * Total number of followers and profiles followed will be counted as well for their own badges |
| Postconditions(s) | Other users can now see that badge on their profile if the user who earned it chose to display it |
| Use Case Diagram | 3.1.2 |

##### 1.1.9 Notifications

|  |  |
| --- | --- |
| Title | Send notifications to the users |
| Description | User will select what notifications they want to receive if any and they will be sent one whenever what they chose to see occurs |
| Priority | 1 |
| Precondition(s) | Notifications must be enabled when the profile is created and the settings for what else the user wants to be notified about must be adjusted |
| Basic Flow | * Once the app is downloaded and a profile is created, the user will be prompted to allow the app to send them notifications * If they say yes, they will only be notified if a friend or close friend sends them a private message, if someone they follow creates a new budget plan, and when someone follows them or accepts their follow request * They will have notification settings where they can choose what notifications to receive   + Only show notifications from close friends   + Show notifications if a followed profile makes a change to a budget plan   + Only show notifications on activity of friends/close friends, etc. |
| Postconditions(s) | Users will be sent the notifications they chose in the settings |
| Use Case Diagram | 3.1.1 |

##### 1.1.10 Privacy

|  |  |
| --- | --- |
| Title | Privacy Settings |
| Description | Private profiles will have no available information on them if the user wants to be very careful of others knowing what they’re doing |
| Priority | 0 |
| Precondition(s) | An account must have been created and logged in to. |
| Basic Flow | * Once the user creates their account, on the main menu page there will be a settings menu * Within that menu there will be a privacy settings button * One button will be to make the account private, meaning if anyone wants to view their account they must request to follow it and wait to be accepted by the user * Additional settings will be available to customize who (anyone, followers, friends, or close friends) can see the budget plans and what details can they see   + They can choose to only show categories to whichever of the above groups of people   + They can choose to show percentage of monthly income   + They can choose to show direct monetary value (amount of money set aside for each category and amount of money spent their so far) |
| Postconditions(s) | User is returned to the profile menu where they can see the results of their changes (private profile, private budget plans, …) |
| Use Case Diagram | 3.1.5 |

##### 1.1.11 Bank Account Linking

|  |  |
| --- | --- |
| Title | Link bank account to FundsBook account |
| Description | The user’s FundsBook account will be linked to their bank account (if they choose), and budgeting information will be updated automatically |
| Priority | 1 |
| Precondition(s) | The user must have created and verified their account, or they can choose to log in with their bank account information for the first time and an account will be created and verified at the same time |
| Basic Flow | * One way to link the bank account is to click on the “Login with bank account button” on the first screen of the app   + This will allow the user to Login to their bank account and subsequently make their FundsBook account to have them linked simultaneously. * Another way is to do it once the account has already been created   + Once a profile is created and the user is on the main menu page, they can select the settings button   + In the settings menu, they can choose to Login to their bank account   + If the credentials are correct and the user agrees to link the account, that data will be stored locally on the device and that FundsBook profile will be linked to the bank account they Logged into. |
| Postconditions(s) | User will be taken to the main menu once their account is linked and will see a small icon on their personal budget to show that their account was linked successfully |
| Use Case Diagram | 3.1.1 |

##### 1.1.12 Suspicious Activity Detection

|  |  |
| --- | --- |
| Title | Detect suspicious activity |
| Description | Detects suspicious activity to protect the user from an intruder trying to access their account. |
| Priority | 0 |
| Precondition(s) | Users must have an account that they have created and registered on a certain device or series of devices to identify which devices are owned by the user and safe. |
| Basic Flow | * Once a profile is created, the device used to Login will be registered * Any Login attempt made from an unregistered device will alert the user and lock the account * The user must then verify their identity through email to unlock their account * Accounts will also be locked when too many failed Login attempts are made, even on a registered device * Users can choose to not have the account lock automatically when one of these events are triggered but that is the default setting if nothing is changed |
| Postconditions(s) | Users will now receive email and text notifications (if chosen in settings) to protect their account and have access to their account again once verified |
| Use Case Diagram | 3.1.1 |

##### 1.1.13 Main Menu

|  |  |
| --- | --- |
| Title | Main menu of app |
| Description | Provide a main menu where all the functionalities of the app are accessible from |
| Priority | 0 |
| Precondition(s) | Users account must be logged into |
| Basic Flow | * This will be the first page the user sees after the sign in page once they login to their account * From here, all the functionalities of the app can be accessed   + Personal profile   + Search   + Settings |
| Postconditions(s) | Users will now have a section to be able to user the entire app |
| Use Case Diagram | 3.1.1 |

#### 1.2 Non-Functional Requirements

##### 1.2.1 Information Encryption

|  |  |
| --- | --- |
| Title | Information Encryption |
| Description | Since the app deals with sensitive information, passwords and log-in information will be heavily encrypted using hashing functions to make it as close to impossible as we can to being hacked |
| Priority | 0 |
| Applicable FR(s) | Profile creation, Personal budget |

##### 1.2.2 DBMS

|  |  |
| --- | --- |
| Title | Database System |
| Description | The app should be able to accurately store a user’s information using hash tables for easy encryption and safety from hackers and to make accessing the information seamless and quick for the user |
| Priority | 2 |
| Applicable FR(s) | Personal budget |

##### 1.2.3 Search Speed

|  |  |
| --- | --- |
| Title | Speed of search function |
| Description | Our search function will operate using a binary search algorithm as our budget plans, tags, and profiles will be organized in alphabetical order in our database. |
| Priority | 2 |
| Applicable FR(s) | Searching |

### 2. System Constraints

#### 2.1 Tool Constraints

##### 2.1.1 Environment

|  |  |
| --- | --- |
| Title | Visual Studio IDE |
| Description | We will work on the code using the Visual Studio IDE and share iterations on it using GitHub |
| Priority | 0 |

#### 2.2 Language Constraints

##### 2.2.1 Java

|  |  |
| --- | --- |
| Title | Java |
| Description | The primary language used for programming the android version of the app will be Java |
| Priority | 0 |

##### 2.2.2 Swift

|  |  |
| --- | --- |
| Title | Swift |
| Description | The primary language used for programming the IOS version of the app will be Swift |
| Priority | 0 |

#### 2.3 Platform Constraints

##### 2.3.1 Android

|  |  |
| --- | --- |
| Title | Android |
| Description | One version of the app will be programmed and designed to function on an android operating system |
| Priority | 0 |

##### 2.3.2 IOS

|  |  |
| --- | --- |
| Title | IOS |
| Description | The other version of the app will be made to work on an iPhone or IOS operating system |
| Priority | 0 |

#### 

#### 2.4 Hardware Constraints

##### 2.4.1 Equipment

|  |  |
| --- | --- |
| Title | Cross Platform testing |
| Description | Lack of distinct devices for cross platform testing. Devices include iOS phones, iOS tablets, android phones and android tablets. |
| Priority | 3 |

#### 2.5 Network Constraints

##### 2.5.1 Connectivity

|  |  |
| --- | --- |
| Title | Connectivity among platforms |
| Description | Although the app will be available to both IOS and Android devices, all must be connected to the same network and function with each other seamlessly |
| Priority | 0 |

#### 2.6 Deployment Constraints

##### 2.6.1 Publishing

|  |  |
| --- | --- |
| Title | Publishing an app |
| Description | Number of tedious steps to deploy an app on two different app stores. |
| Priority | 0 |

#### 2.7 Transition & Support Constraints

##### 2.7.1 Transitionary Requirements

|  |  |
| --- | --- |
| Title | Transitionary Requirements |
| Description | Once the user makes an account and edits their profile, our service must download and package all the data in a convenient way for the user |
| Priority | 0 |

##### 2.7.2 Continued Maintenance

|  |  |
| --- | --- |
| Title | Continued Maintenance |
| Description | This is a term project, ending all maintenance one the grading period is over, or a new team can come in and continue maintenance |
| Priority | 5 |

#### 2.8 Budget & Schedule Constraints

##### 2.8.1 Time Management

|  |  |
| --- | --- |
| Title | Spring 2021 Semester |
| Description | Due to class schedule, we only have until May 1st to complete this project |
| Priority | 0 |

#### 2.9 Miscellaneous Constraints

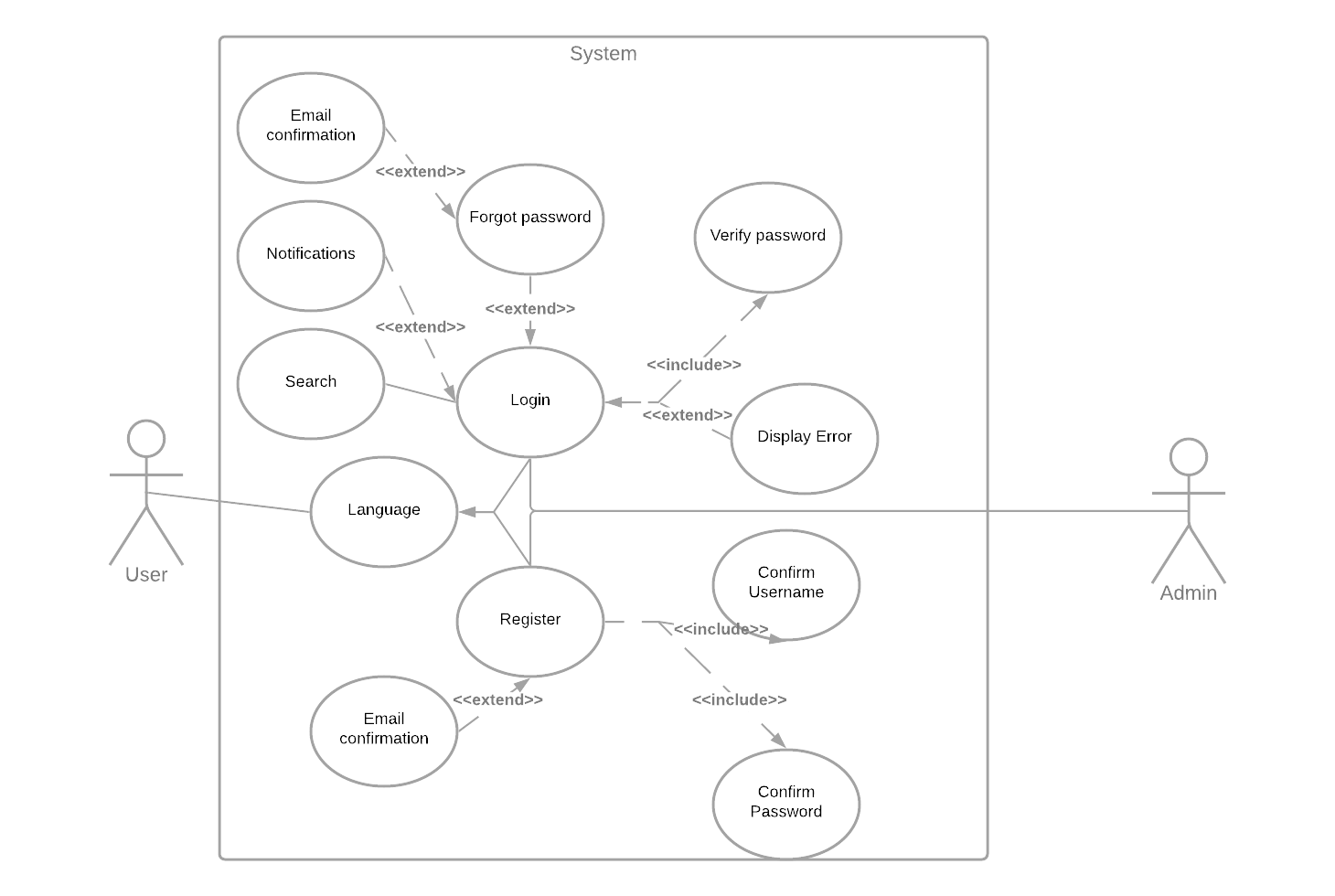
##### 2.9.1 Communication

|  |  |
| --- | --- |
| Title | Communication between members |
| Description | Time Zone difference between members and difficulty of communication. |
| Priority | 0 |

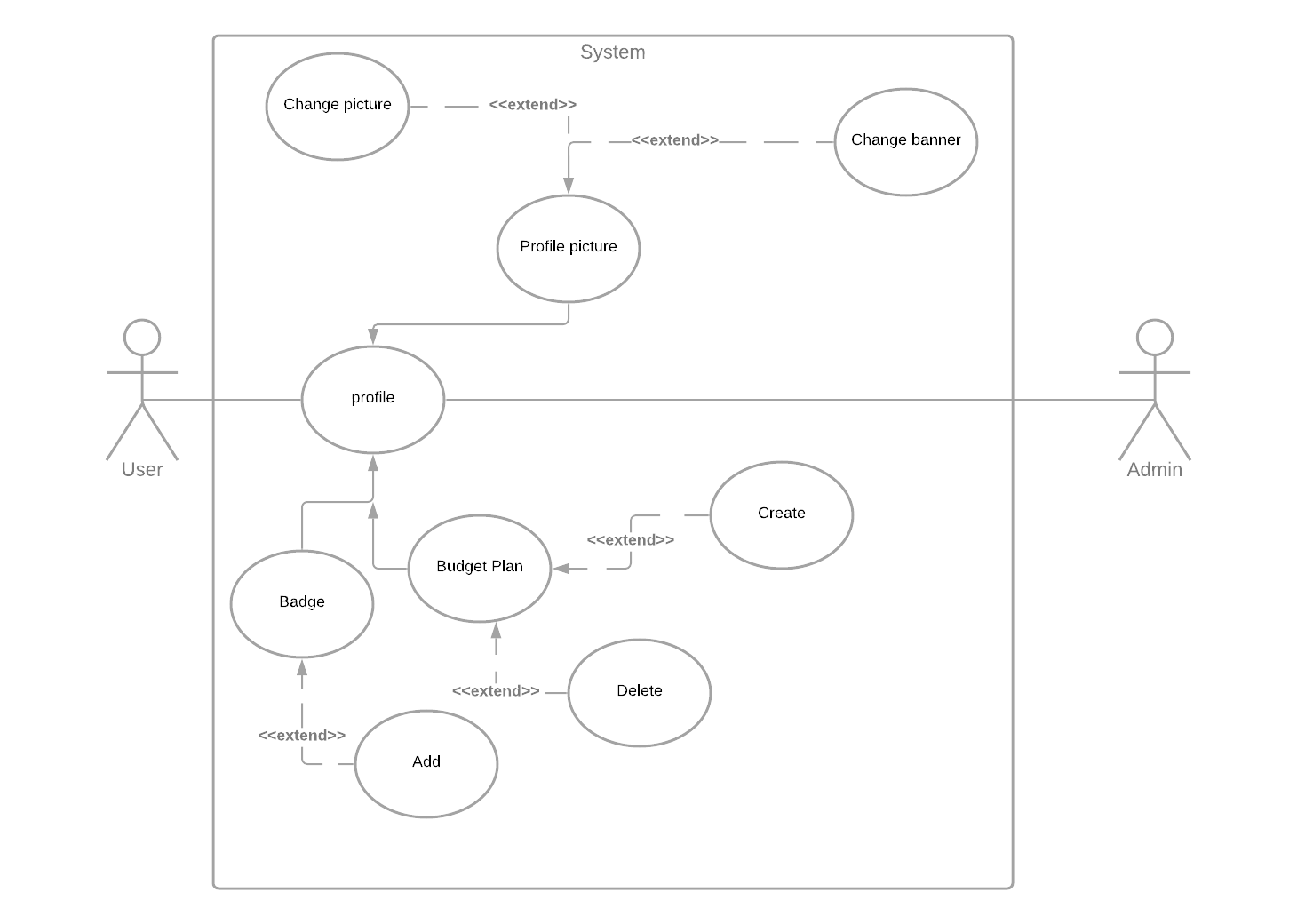
### 3. Requirements Modeling

*< List all Use-case diagrams for the functional requirements in the following format>*

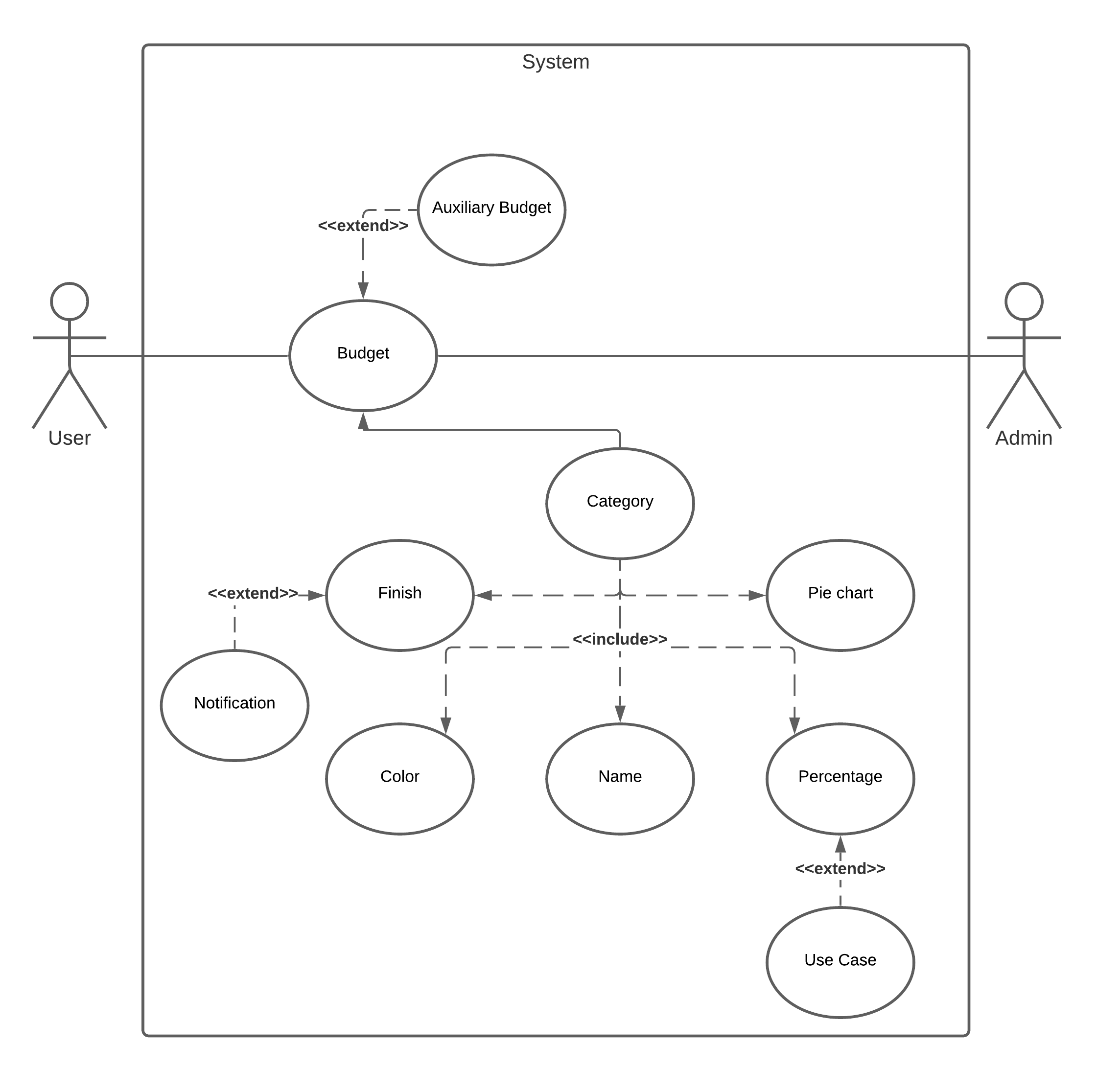
##### 3.1.1 Profile Login



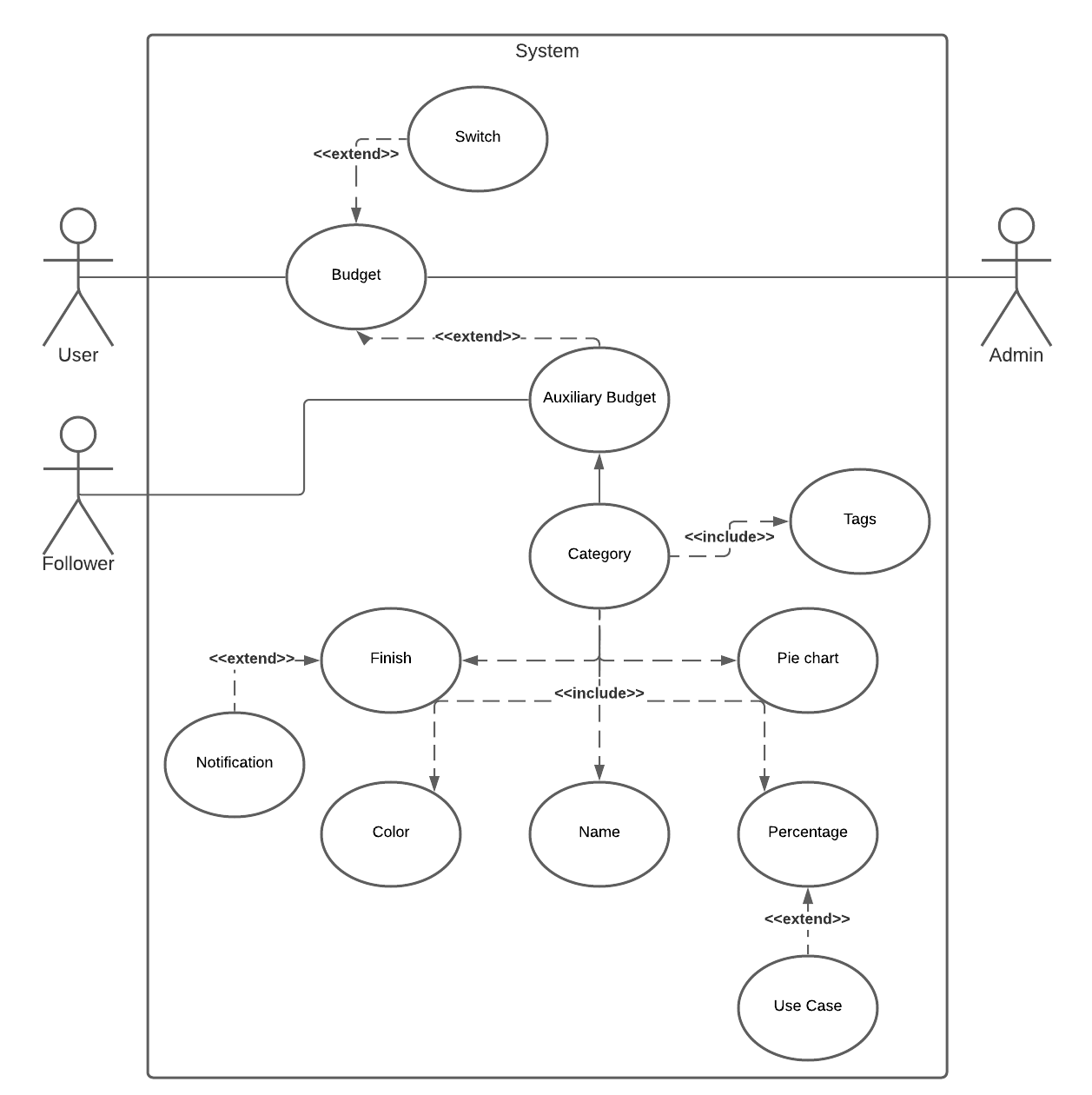
##### 3.1.2 Profile Editing



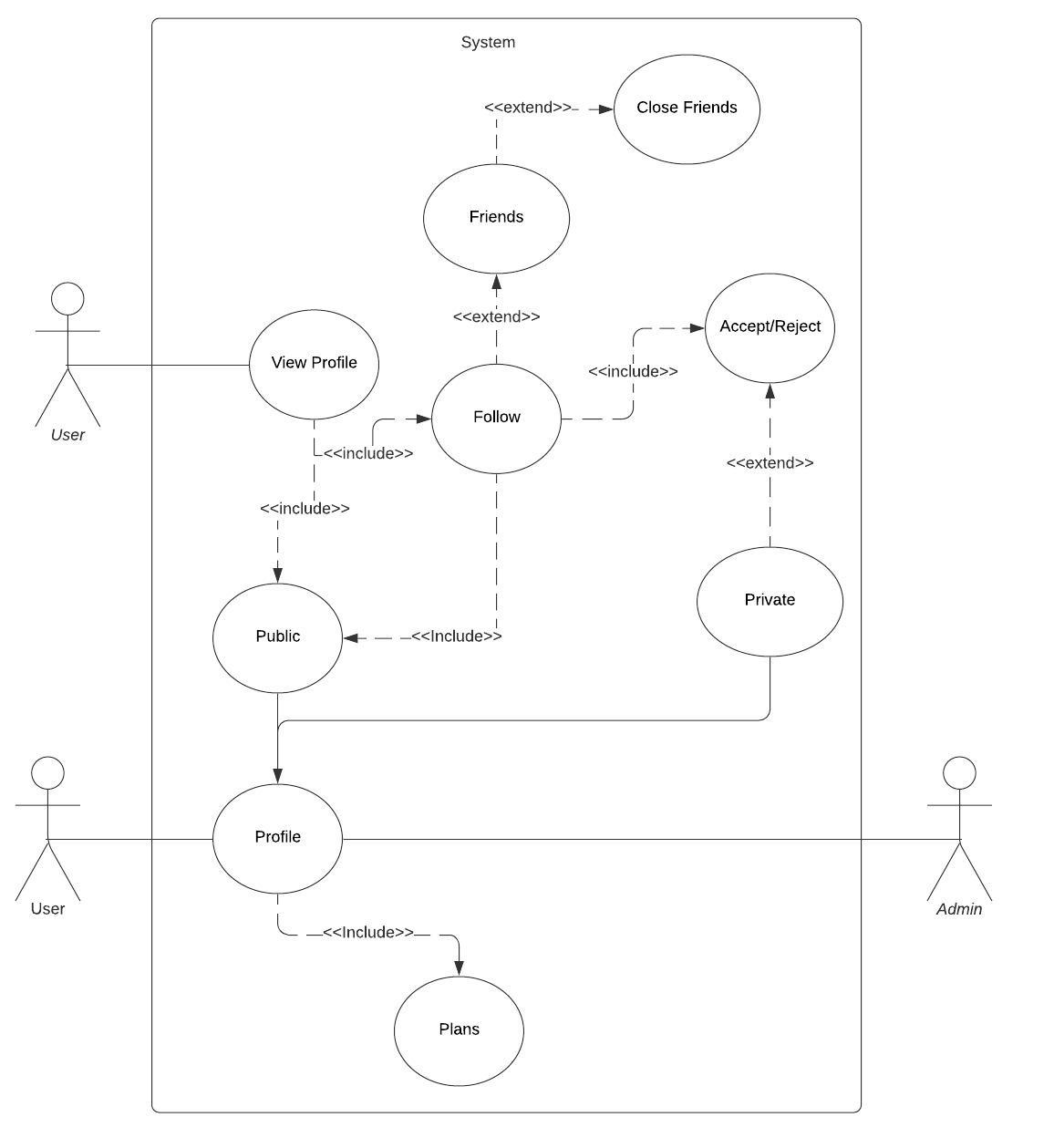
##### 3.1.3 Budget Management



##### 3.1.4 Auxiliary Budget



##### 3.1.5 Socialization



### 4. Evolutionary Requirements

#### 4.1 Non-Functional Requirements

##### 4.1.1 Data logging

|  |  |
| --- | --- |
| Title | Data logging |
| Description | Logging all the data stored by the app and any changes or previous versions in order to update as well as look through previous versions in case of a bug to figure out where the issue started |
| Priority | 1 |
| Applicable FR(s) | N/A |