Wentworth Institute of Technology

COMP4960 – Software Engineering

Instructor: Dr. Charlie Pham

Software Design Document

for

FiMan (Financial Manager)

Version 2.6

Prepared By

Team: Finance Boyz - Team 3

Michael Marino, marinom1@wit.edu

Mike Depietro, depietrom@wit.edu

Ian Seto, setoi@wit.edu

GitHub Link

https://github.com/marinom1/FiMan-Financial_Manager

4/26/2021

Contents

Revision History

Date	Version	Description	Author(s)
02/09/2021	1.0	Created Document	Michael Marino
02/12/2021	1.1	First Draft of Document	Michael Marino Mike Depietro Ian Seto
2/26/2021	1.2	Minor changes after receiving initial feedback -Included specific apps in Novelty section -Removed "Manage balance" in 1.3 (redundancy) -Reworded requirement and functionality to better connect both to each other, specifically with the stock market features -Modified wording in 2.1.21 "wealth" to "Balance" -Fixed REQ # ordering -Updated REQ 11 from "average user hardware" to " hardware no older than 8 years old"	Michael Marino
3/29/2021	2.0	-Added in new sections, e.g System Architecture, System Design, etc -Added in UI Designs to some sections	Michael Marino Mike Depietro Ian Seto
3/31/2021	2.1	Removed "The system availability shall be 90% (*availability*)" - Does not apply to our software program	Michael Marino
4/4/2021	2.2	-Documented UI Designs and documented Sequence Diagrams for most subfunctions -Documented at least 1 test case for every subfunction -Minor capitalization error fixes	Michael Marino
4/5/2021	2.3	-Finished documenting UI designs for all subfunctions -Documented API's in section 5 -Finalized first submission of document	Michael Marino Ian Seto
4/9/2021	2.4	Addressed feedback -Changed "system requirements specification" to "software design document" -Removed instructional statements (vestigial from copy/pasting the template)	Michael Marino

		-Documented multi-step UI functions instead of having only the first step documented in such cases -Added additional test cases, with at least 1 bad user case and 1 happy user case for every subfunction. Average of 4 test cases for each subfunction currently. -Updated section 2.3.1 to be more specific -Removed Requirement regarding User Agreements -Updated Contents page numbers to reflect above changes	
4/14/2021	2.5	-Addressed Feedback on Section 5 (More specific API documentation)	Ian Seto
4/21/2021	2.6	Addressed 2nd Iteration Feedback -Removed instructional statement -Specified API usage more closely -Updated page numbers on Revision History	Michael Marino Ian Seto

1. Introduction

1.1. Document purpose

This document provides software design documentation for a financial manager application used by everyone who is looking for a tool to better organize and manage their wealth.

1.2. Product overview

1.2.1. - Problem statements

People love money. People love saving money. People don't love losing money. The problem is that there are so many apps and different tools to manage your finances! Therefore, there needs to be a tool that people can use to keep track and manage all your life's earnings, spending, and savings without having to jump from app to app or website to website.

1.2.2. - Proposed solution

Our solution is to create ONE app where you can manage all your finances without any hassle. It will have many different useful features that users will find easy to use, including a balance tracker, a budget manager, stock portfolio manager, a live news tracker, and budget help notifications. The user will be able to take advantage of the modular design so they can pick and choose which features they want to use.

1.2.3. - Novelty

There are many budgeting apps in the market, but many seem to be bloated with features that many users do not actually need. "Mint" is a well-known app, but has so many features that come along with it. There are also apps that are too specialized, where they focus only on one aspect of finance, but users may also want to manage other types of finances, so they must download multiple apps, which is a terrible inconvenience. The app "Wally" is guilty of this, being narrowly specialized in budgeting. Our application will not be overloaded with these extra features and the features we do have are completely customizable so that you don't have to see what you don't need.

1.3. Product functionality

The application shall allow users to:

- Manage profile
- Manage budget
- View stock market news
- View budget help and advice notifications

1.4. Definitions

Profile: The holder of a user's personal information, such as name and balance.

1.5. Acronyms and abbreviations

REQ: requirement

SRS: software requirements specification

GUI: graphical user interface

2. System requirements

2.1. Functional requirements

2.1.1 - Manage Profile

2.1.1.1 - Create a New Profile

[REQ-1] The system shall allow users to create a new profile with their name.

2.1.1.2 - Edit User Profile

[REQ-2] The system shall allow users to edit their personal information.

2.1.1.3 - Login

[REQ-3] The system shall allow users to re-login to their user profile upon exiting application.

2.1.2 - Manage Budget

2.1.2.1 - Set Total Balance

[REQ-4] The system shall allow users to set the total amount of money they currently possess.

2.1.2.2 - Set Target Budget

[REQ-5] The system shall allow users to set a target limit that cannot be exceeded.

2.1.2.3 - Notifications

[REQ-6] The system shall allow users to view notifications that help or advise them regarding their budget.

2.1.2.4 - Add / Remove Expenses

[REQ-7] The system shall allow users to add or remove recurring expenses like daily, weekly, monthly, yearly, etc.

2.1.3 - Stock Market

2.1.3.1 - Select a Sector

[REQ-8] The system shall allow users to select a sector to focus on that will filter for news that is specific to that sector.

2.1.3.2 - View List of Stocks

[REQ-9] The system shall allow users to view a list of stocks and see their recent performance.

2.1.3.3 - View News of Sector

[REQ-10] The system shall allow users to view a news feed regarding recent events in the stock market.

2.2. Non-functional requirements

2.2.1 - Performance

[REQ-11] The system shall take no longer than 10 seconds to start on user hardware no older than 8 years old.

2.2.2 - Ease of Use

[REQ-12] The system should be understandable by the user in under 1 minute.

2.2.3 - Storage

[REQ-13] The system should be able to function with under 1GB of storage space on the user’s computer.

2.3. Other requirements

2.3.1 - Development

[REQ-14] The system shall run with Python 3.7 or newer.

2.3.2 - Interface

[REQ-15] The system should have a GUI.

3. System architecture

3.1. Overall architecture

Figure 1 shows the overall architectural design of FiMan.

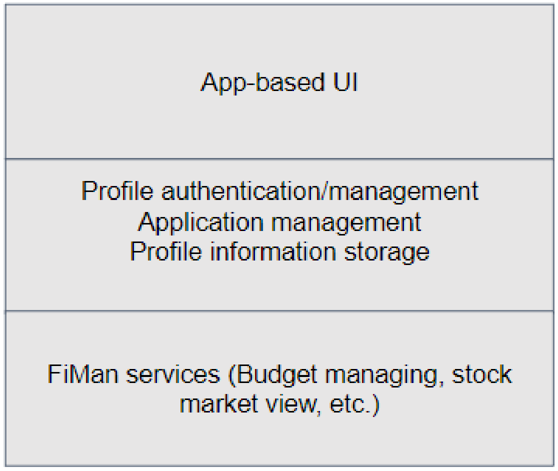

App-based UI
Profile authentication/management
Application management
Profile information storage
FiMan services (Budget managing, stock market view, etc.)

Figure 1. Overall architecture

Layered architectural pattern is used because:

- Not every service is directly dependent on one another, but instead only the ones that are adjacent with each other are dependent

- The program will have the user interface as the top level of the program, where the user will interact with the program’s functionalities

- The organization is ideal for FiMan, as it will have the system sliced into pieces with related functionality associated with each layer i.e the profile’s balance’s layer will be used and accessed by the other layers and will be building on top of that existing layer

- Allows replacement of entire layers so long as the initial interface and profile balance is maintained

Here are the responsibilities of each of the components in Figure 1:

-App-based UI is what the user will see, allows them to select where they want to navigate within the app without difficulty

-Profile authentication/management, Application management, Profile information storage authenticates and allows the viewing of profile information. Will not be seen directly but will be reflected in the UI, gives users only what they need or have rights to. It also holds all of the information relating to profiles including name and budget details

-FiMan services contain the services and applets that users will be able to utilize through their UI. The user will not be able to see what is going on behind the scenes (the backend)

3.2. Components mapping

3.2.1. Functional requirements

Figure 2 shows the components mappin for functional requirements as follows:

- Manage Profile
  - -Create a new profile
  - -Edit user profile
  - -Login
- Stock Market
  - -Select a Sector
  - -View List of Stocks
  - -View News of Sector
- Manage Budget
  - -Set Total Wealth
  - -Set Target Budget
  - -Notifications
  - -Add/Remove Expenses

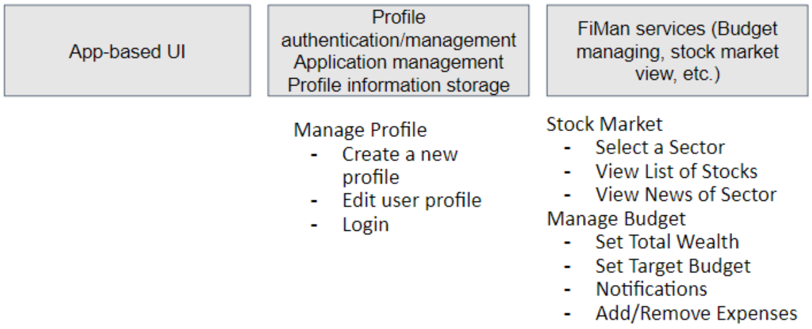


Figure 2. Components mapping for functional requirements

3.2.2. Non-functional requirements

Figure 3 shows the components mapping for non-functional requirements as follows:

- -Understandable by user under 1 minute
- Should be able to function with under 1GB of space
- The system shall take no longer than 10 seconds to start on user hardware no older than 8 years old

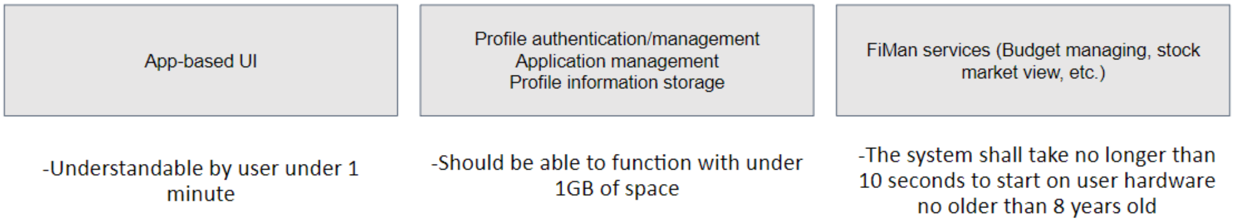

App-based UI
Profile authentication/management
Application management
Profile information storage
FiMan services (Budget managing, stock market view, etc.)
-Understandable by user under 1 minute
-Should be able to function with under 1GB of space
-The system shall take no longer than 10 seconds to start on user hardware no older than 8 years old

Figure 3. Components mapping for non-functional requirements

3.3. Technology stack selection

Backend: Python

Easy to learn and use with our Java programming background

Many available libraries to perform many different tasks

Frontend / GUI: Tkinter

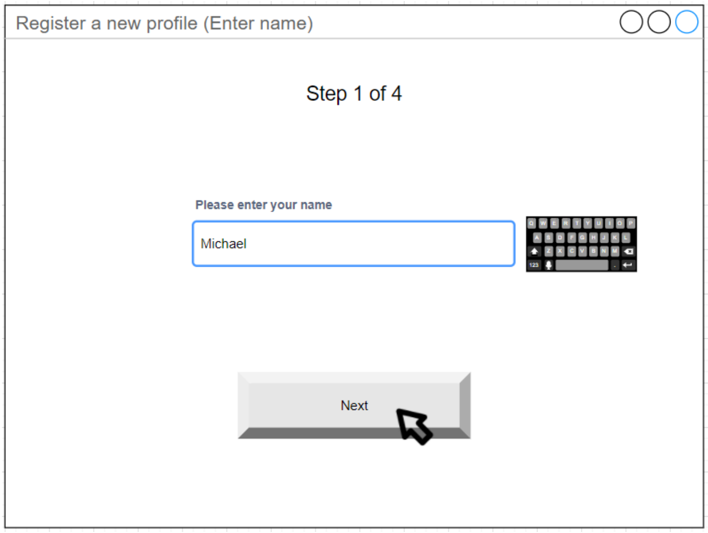
Standard GUI Framework with minimal requirements

Easy to learn

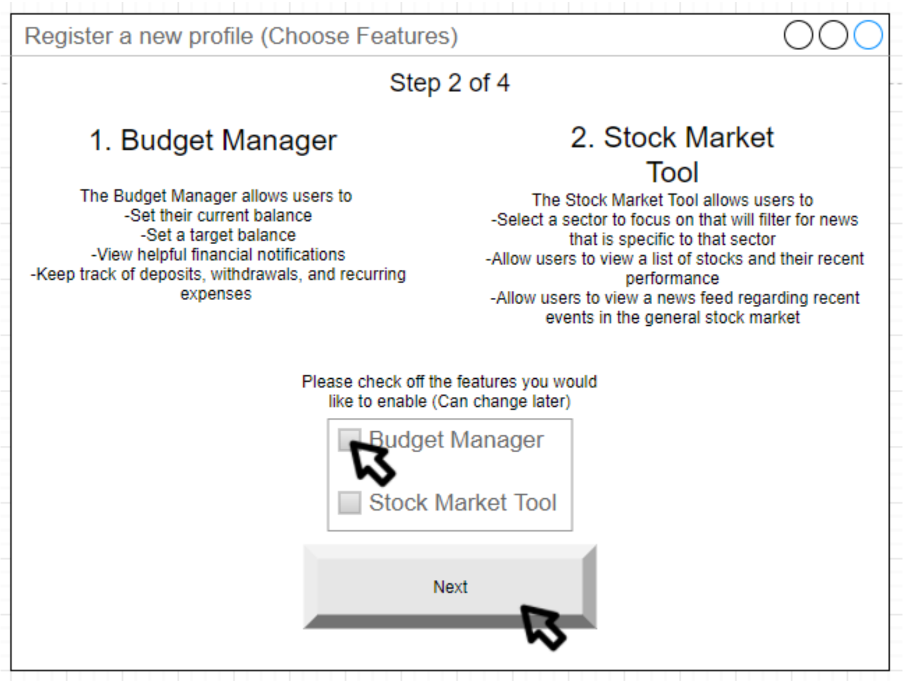
4. System Design

4.1. UI

4.1.1. - Create a New Profile



Please enter your name: Michael



Register a new profile (Choose Features)

Step 2 of 4

1. Budget Manager

The Budget Manager allows users to
-Set their current balance
-Set a target balance
-View helpful financial notifications
-Keep track of deposits, withdrawals, and recurring expenses

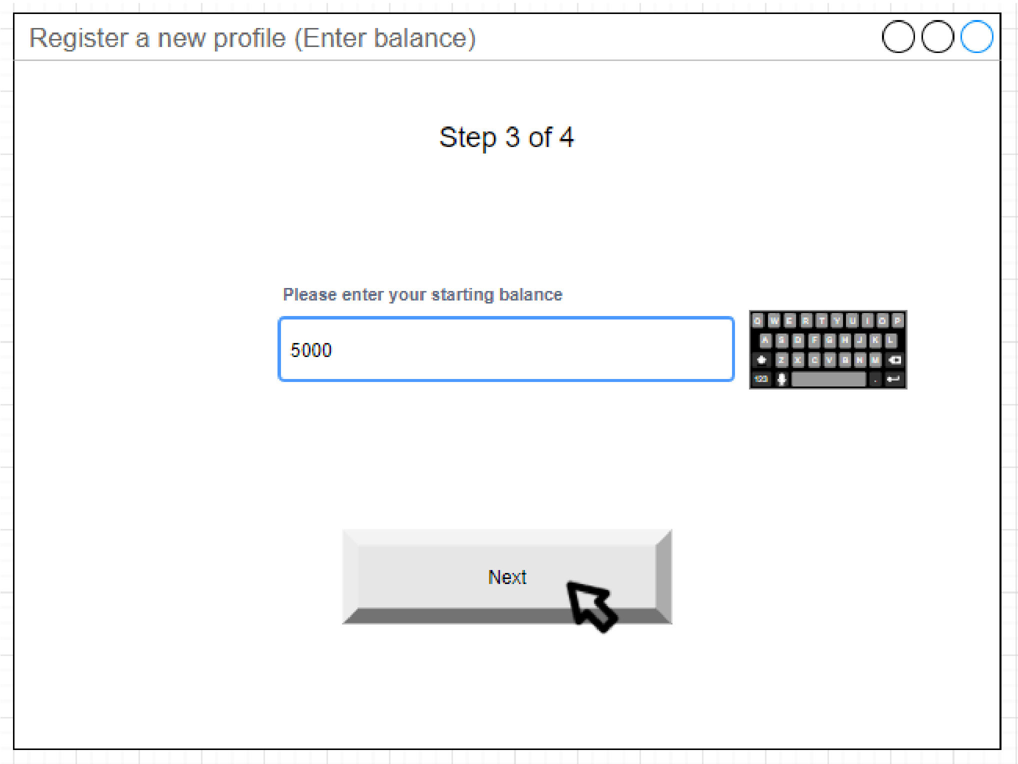
2. Stock Market Tool

The Stock Market Tool allows users to
-Select a sector to focus on that will filter for news that is specific to that sector
-Allow users to view a list of stocks and their recent performance
-Allow users to view a news feed regarding recent events in the general stock market

Please check off the features you would like to enable (Can change later)

□ Budget Manager

□ Stock Market Tool

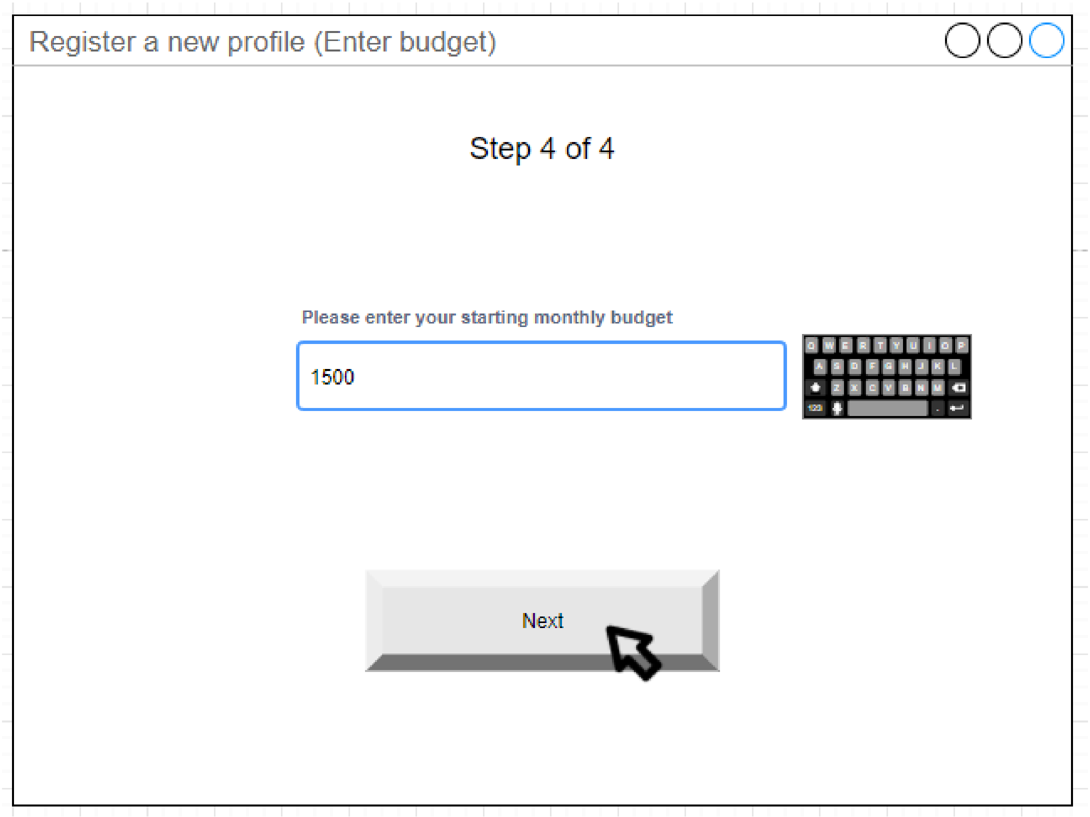


Register a new profile (Enter balance)

Step 3 of 4

Please enter your starting balance

5000



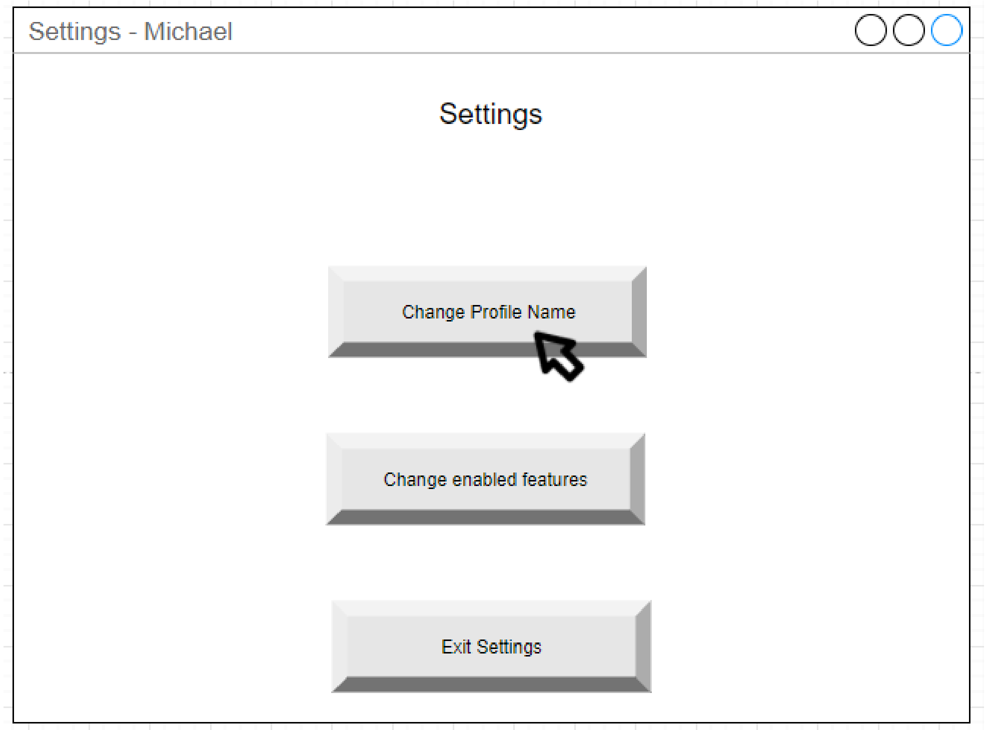
Register a new profile (Enter budget)

Step 4 of 4

Please enter your starting monthly budget

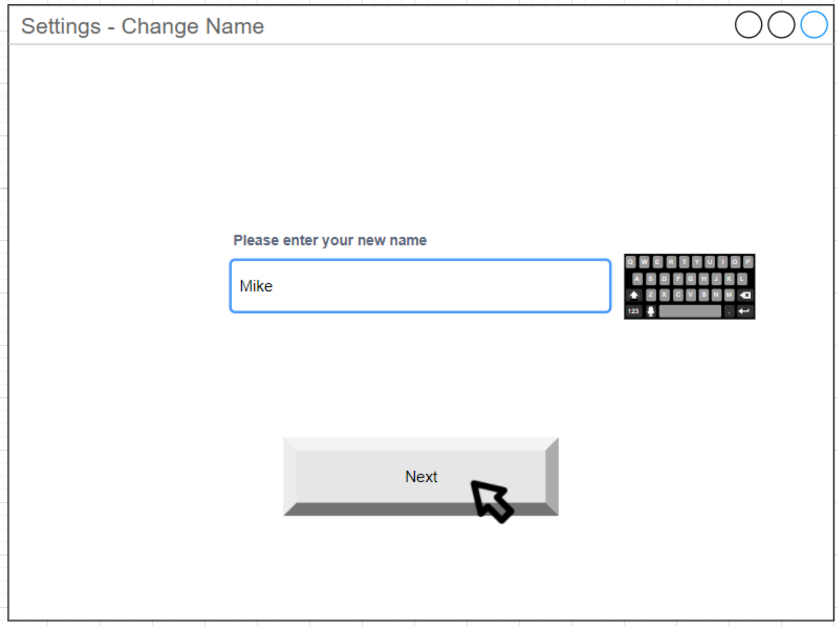
1500

4.1.2. - Edit User Profile



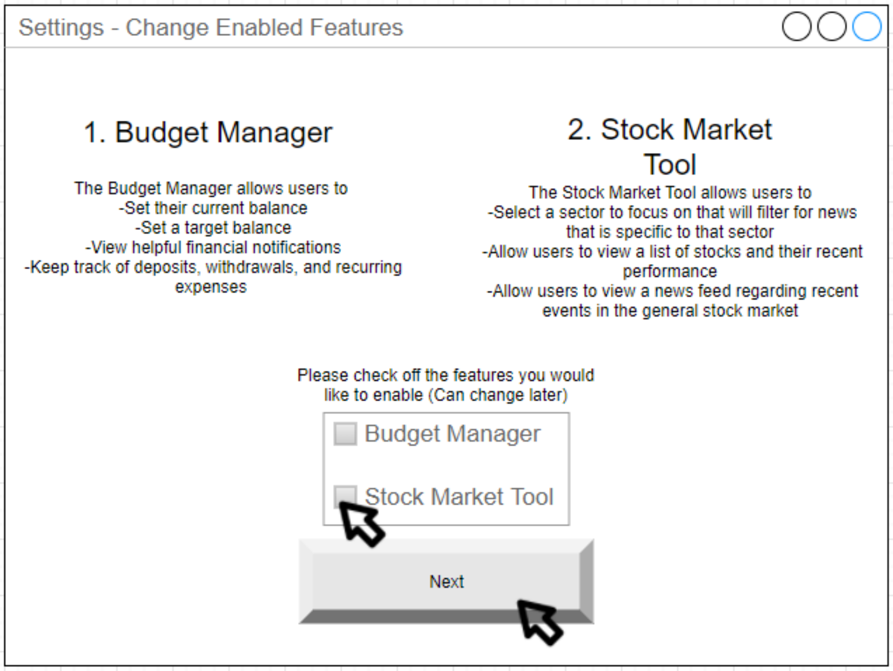
Settings - Michael

Settings


Settings - Change Name

Please enter your new name

Mike


Settings - Change Enabled Features

1. Budget Manager

The Budget Manager allows users to
-Set their current balance
-Set a target balance
-View helpful financial notifications
-Keep track of deposits, withdrawals, and recurring expenses

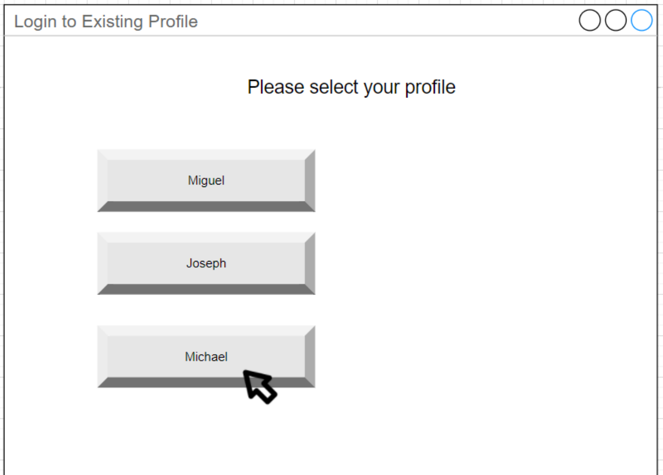
2. Stock Market Tool

The Stock Market Tool allows users to
-Select a sector to focus on that will filter for news that is specific to that sector
-Allow users to view a list of stocks and their recent performance
-Allow users to view a news feed regarding recent events in the general stock market

Please check off the features you would like to enable (Can change later)

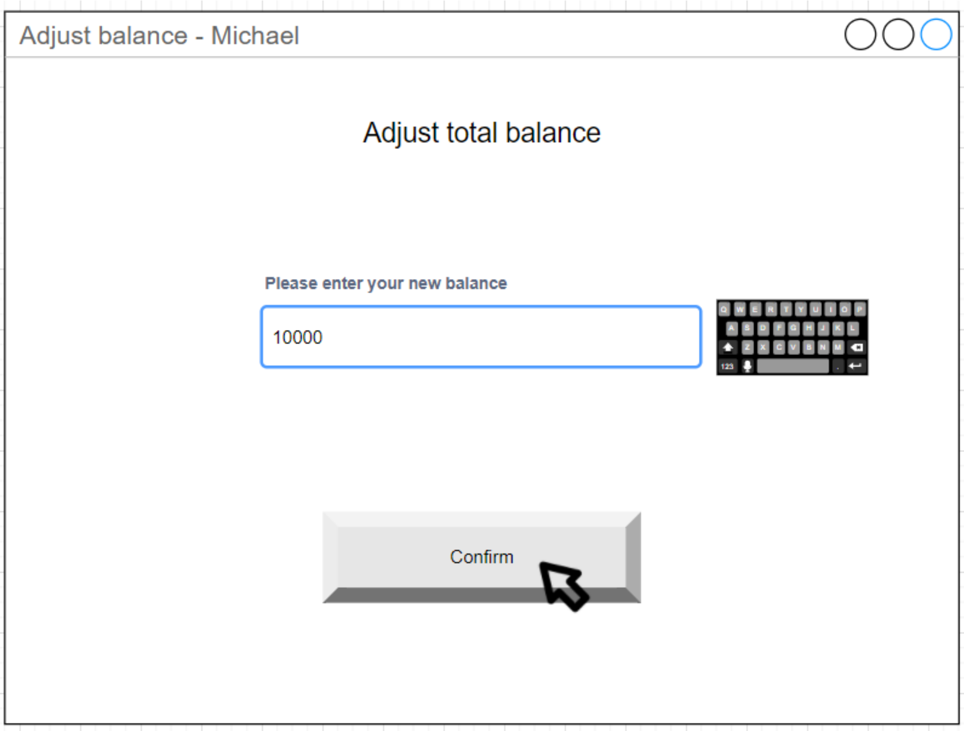
□ Budget Manager
□ Stock Market Tool

4.1.3. - Login


Login to Existing Profile

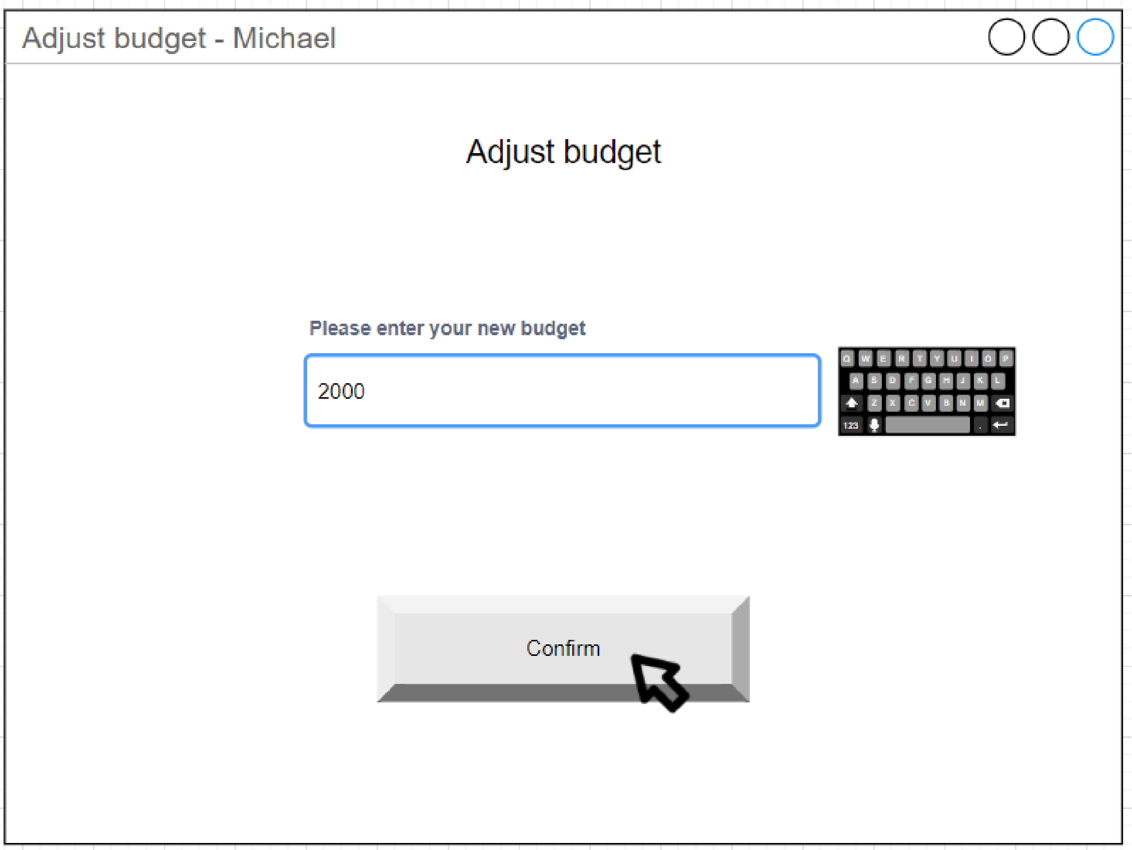
Please select your profile

4.1.4. - Set Total Balance



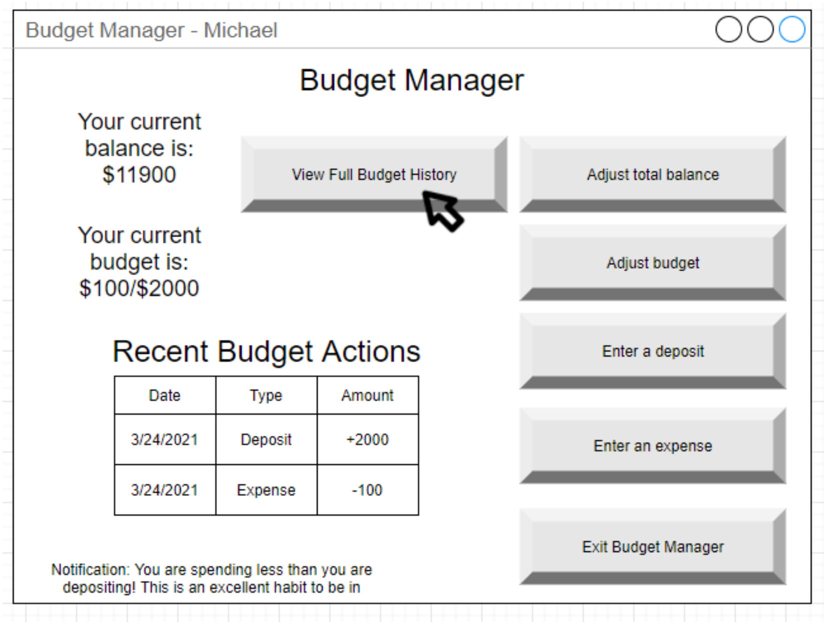
10000

4.1.5. - Set Target Budget



2000

4.1.6. - Notifications



Budget Manager - Michael

Budget Manager

Your current balance is: $11900

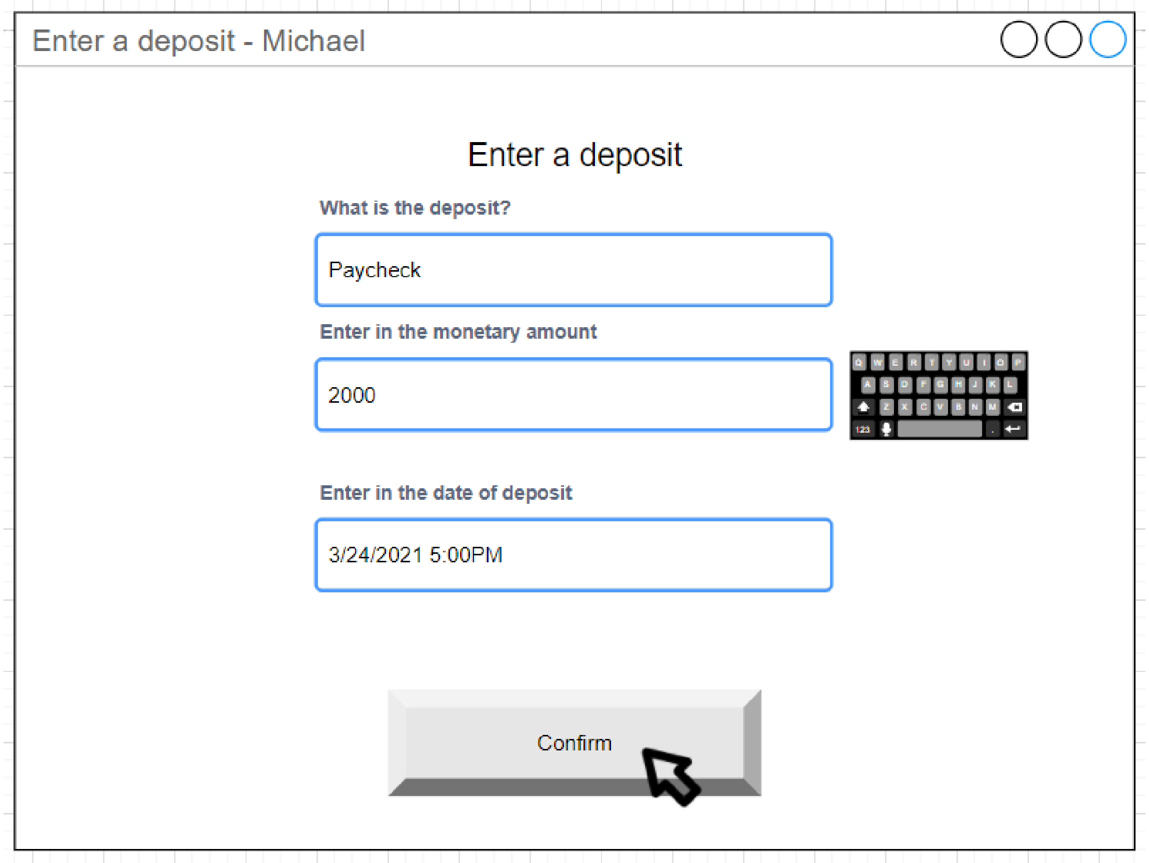
Your current budget is: $100/$2000

Recent Budget Actions

Date	Type	Amount
3/24/2021	Deposit	+2000
3/24/2021	Expense	-100

Notification: You are spending less than you are depositing! This is an excellent habit to be in

4.1.7. - Add / Remove Expenses



Enter a deposit - Michael

Enter a deposit

What is the deposit?

Paycheck

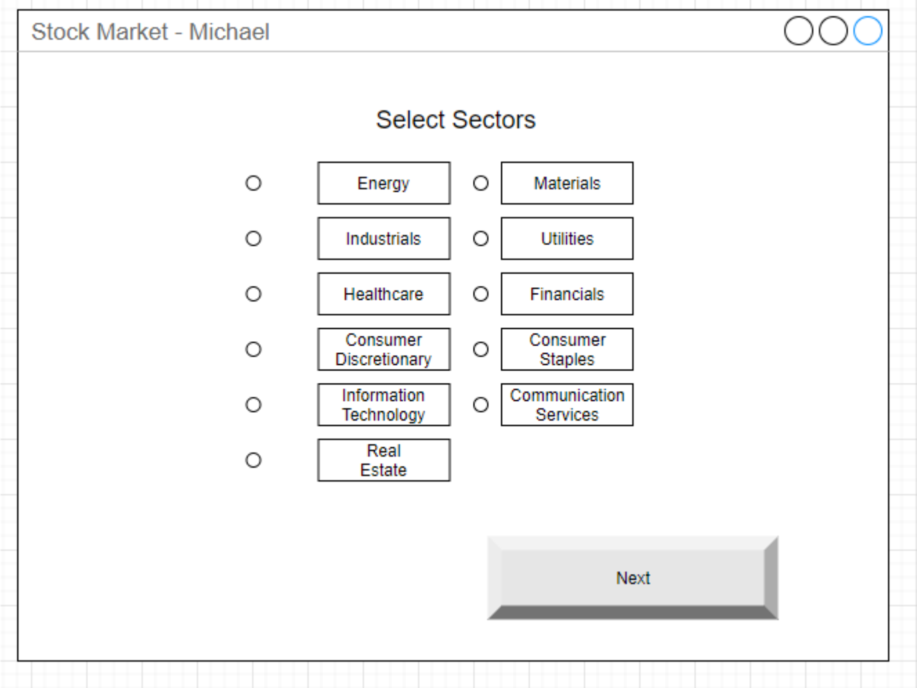
Enter in the monetary amount

2000

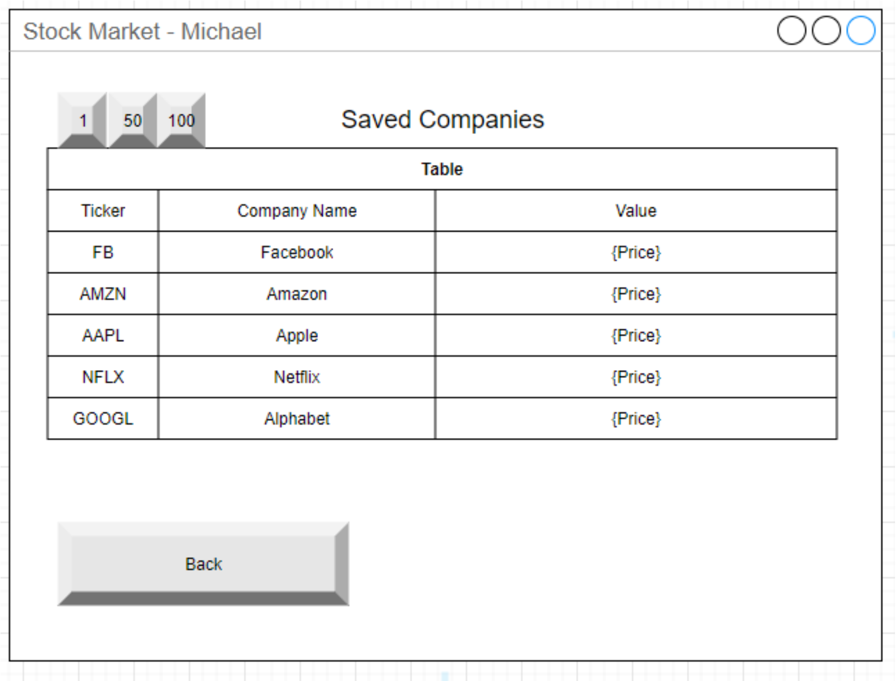
Enter in the date of deposit

3/24/2021 5:00PM

4.1.8. Select a Sector



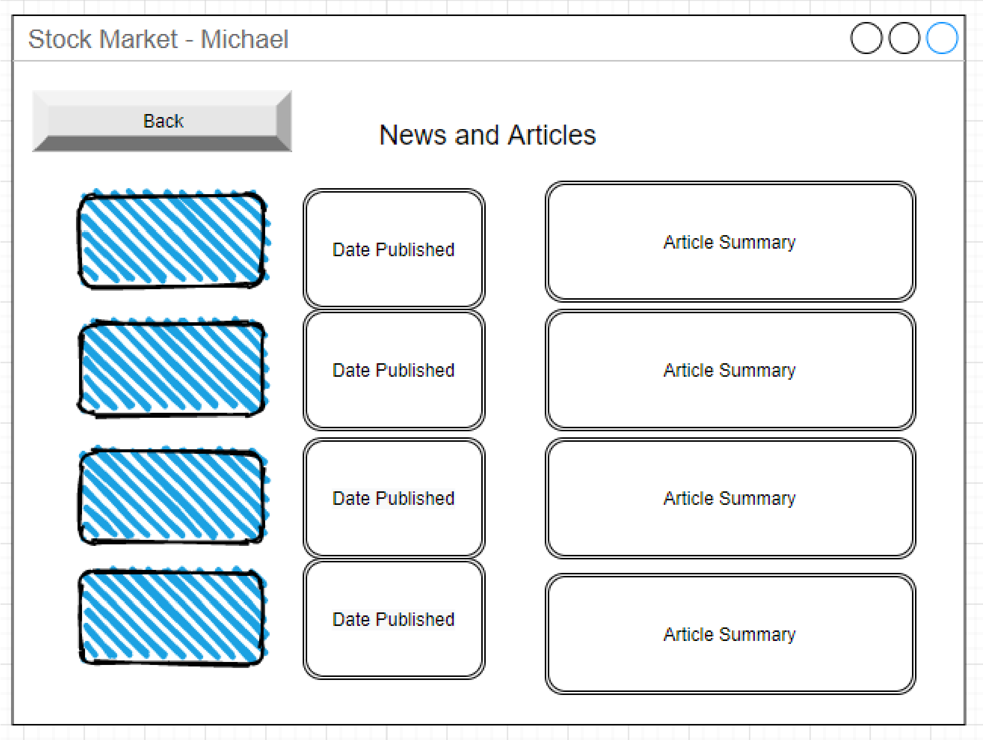
4.1.9. View List of Stocks



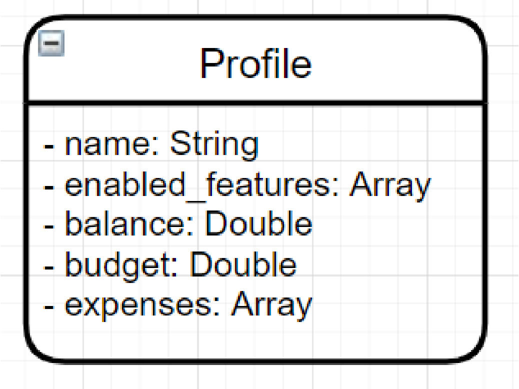
Table

Ticker	Company Name	Value
FB	Facebook	{Price}
AMZN	Amazon	{Price}
AAPL	Apple	{Price}
NFLX	Netflix	{Price}
GOOGL	Alphabet	{Price}

4.1.10. View News of Sector

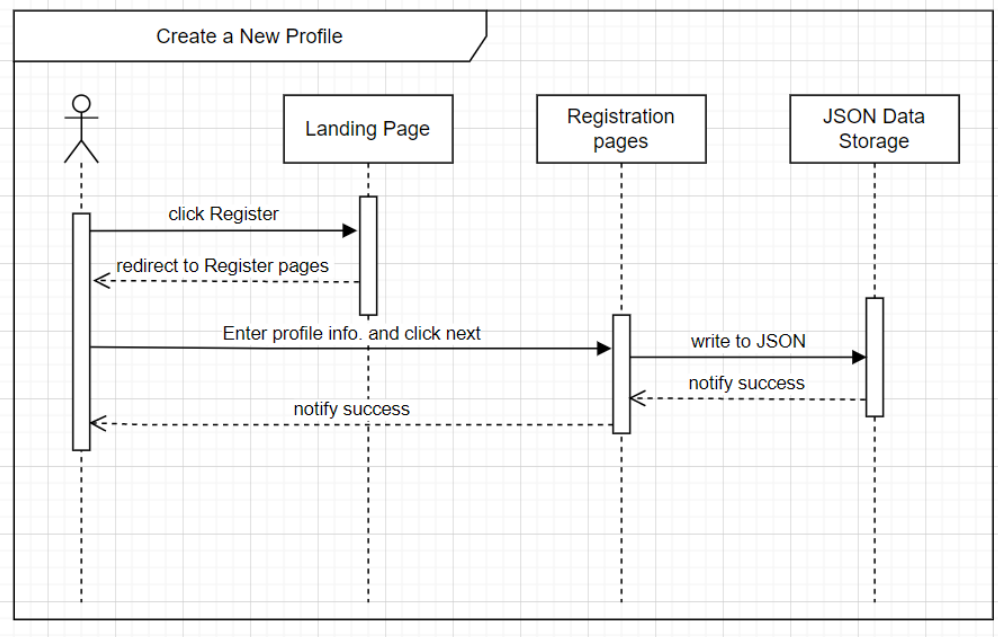


4.2. Class diagram

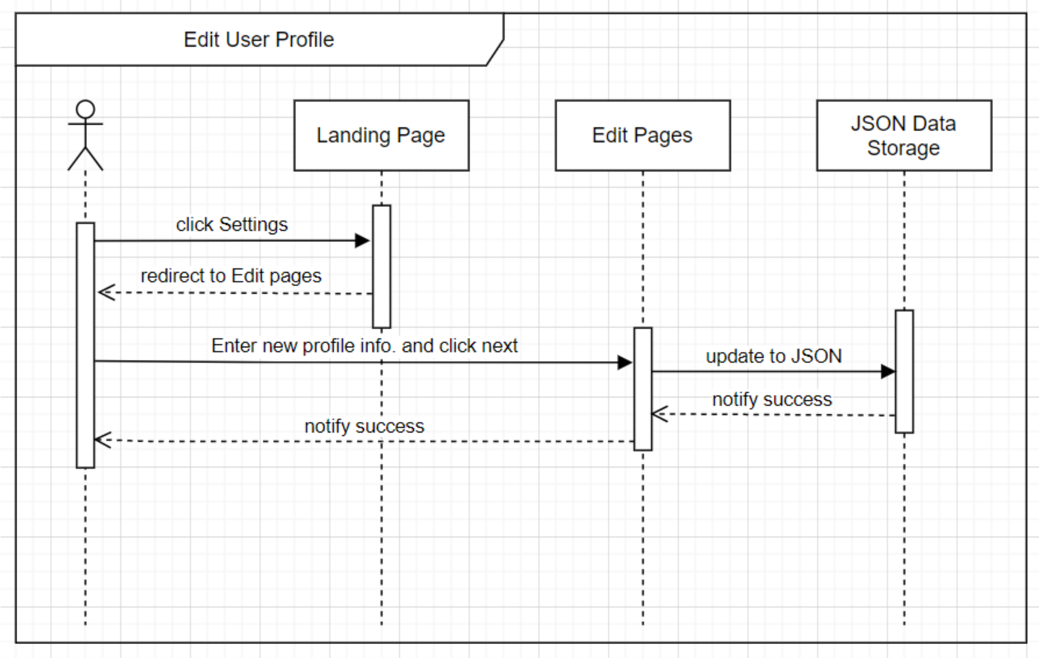


4.3. Sequence/activity diagram

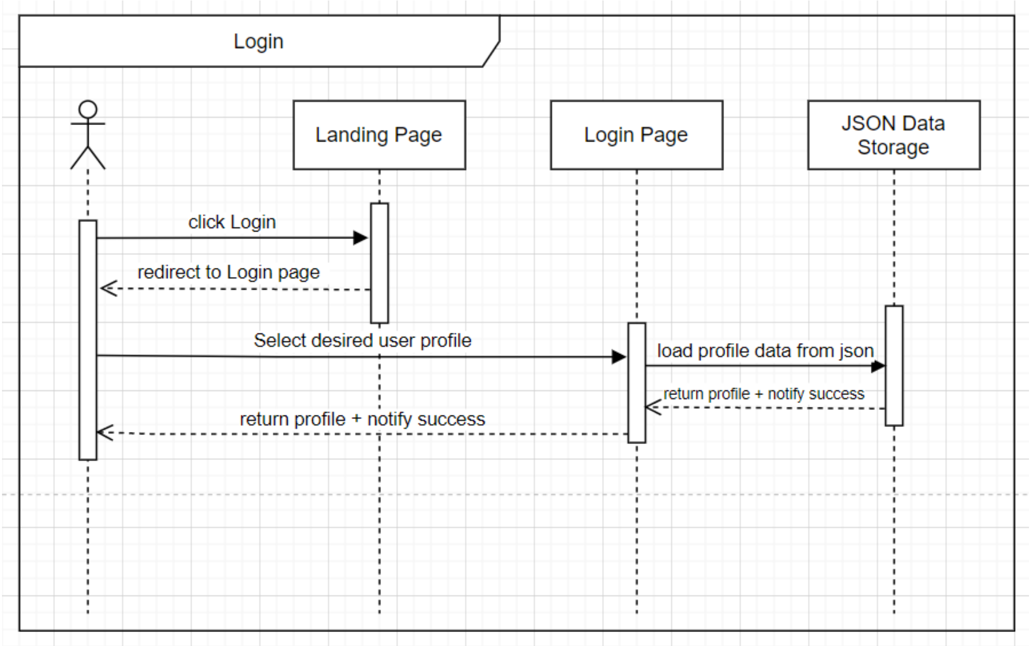
4.3.1. Create a New Profile



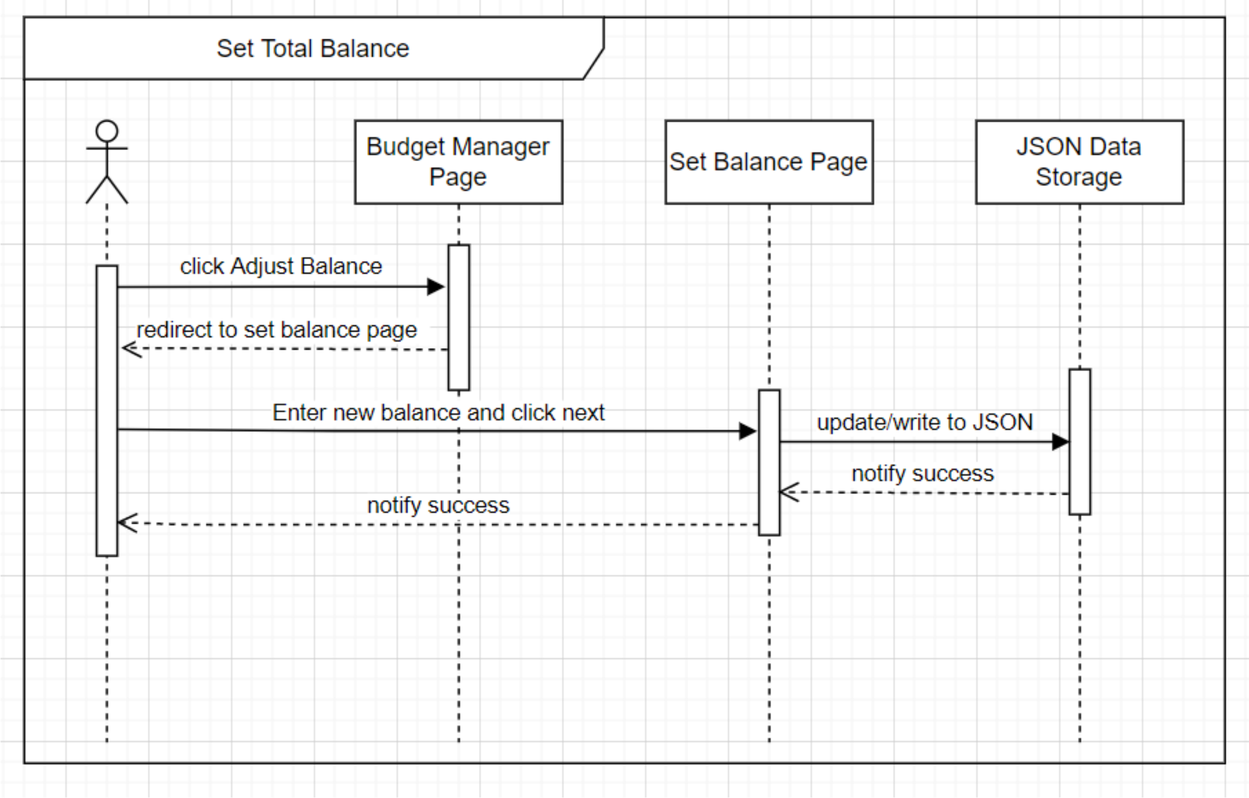
4.3.2. Edit User Profile



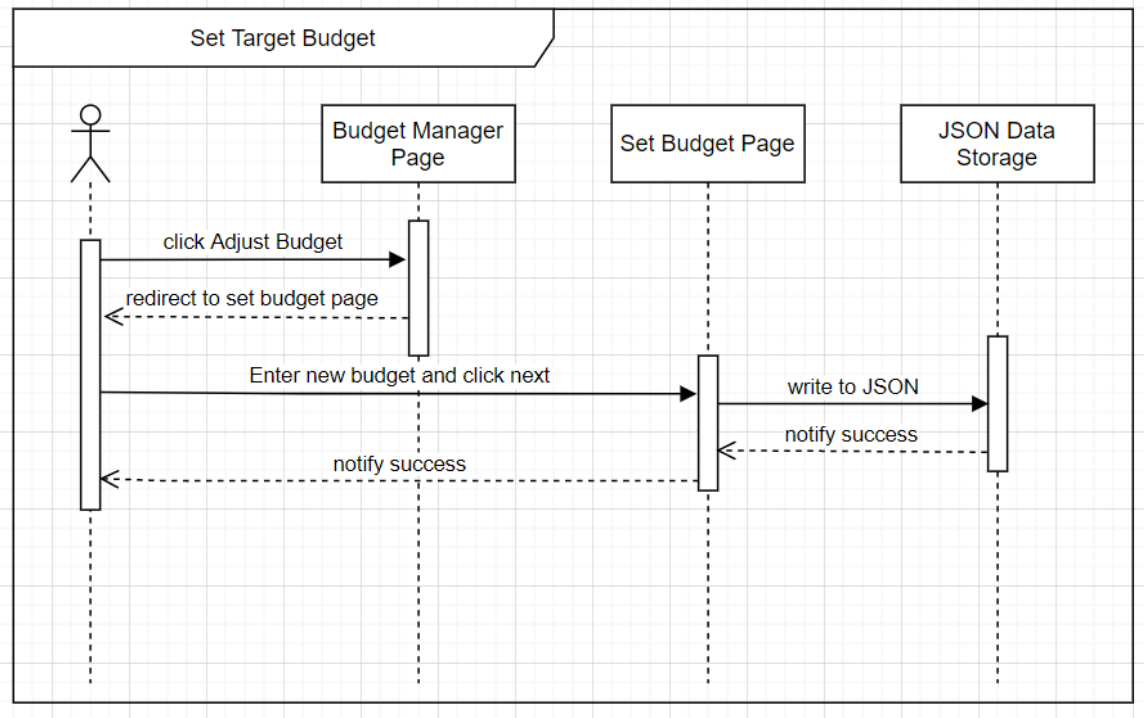
4.3.3. Login


Login
Landing Page
Login Page
JSON Data Storage
click Login
redirect to Login page
Select desired user profile
load profile data from json
return profile + notify success
return profile + notify success

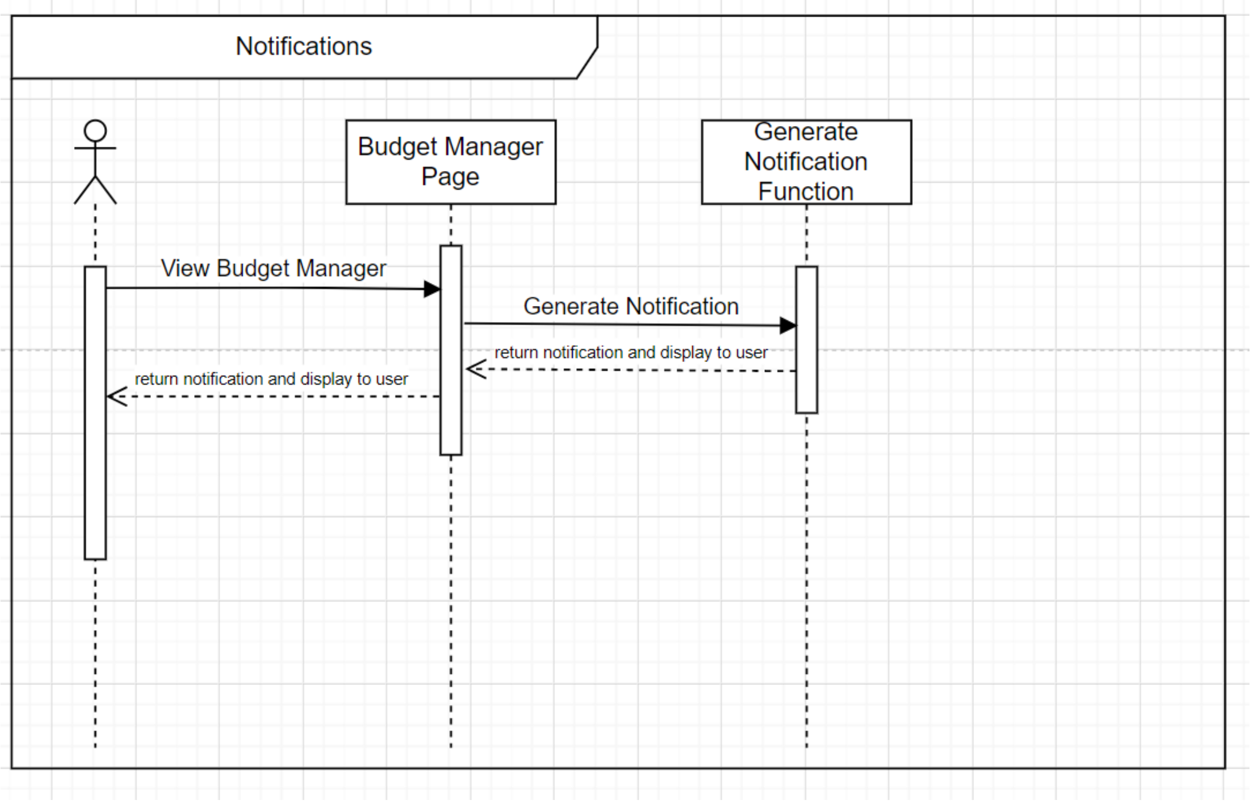
4.3.4. Set Total Balance



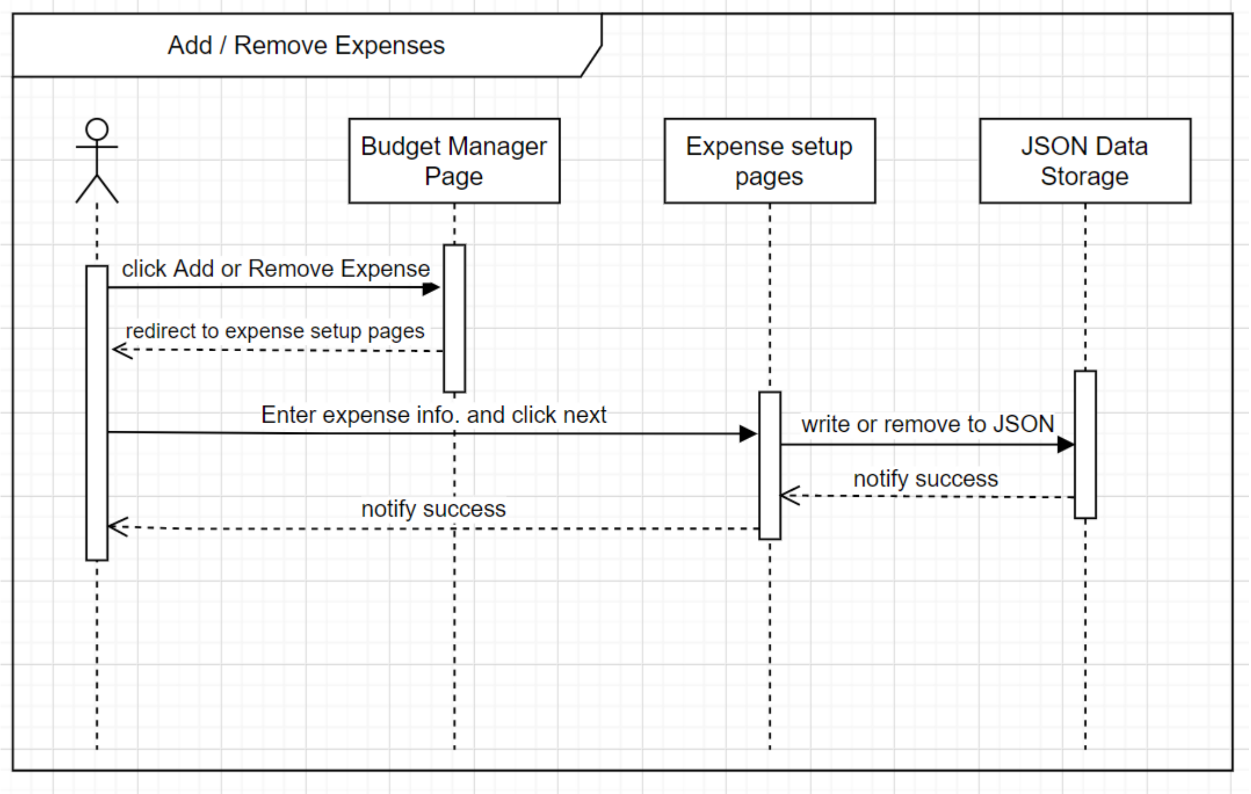
4.3.5. Set Target Budget



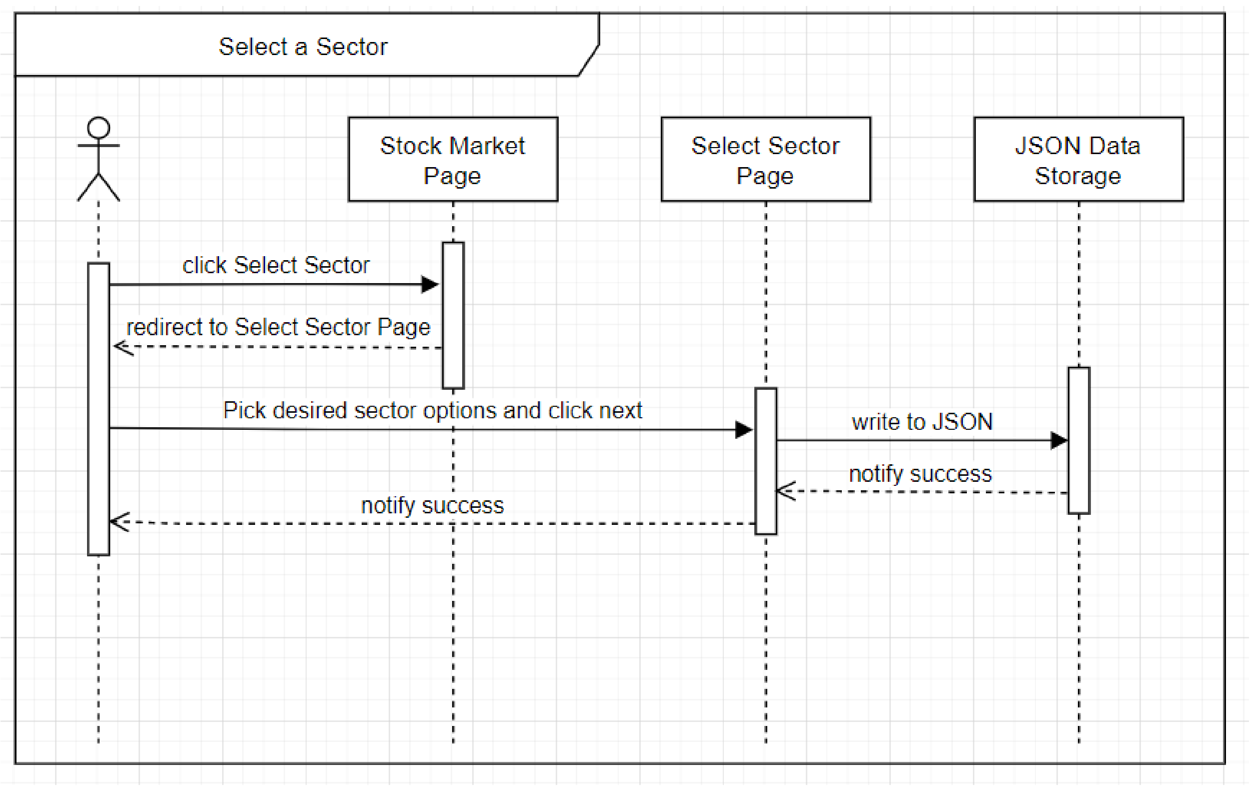
4.3.6. Notifications


Notifications
Budget Manager Page
Generate Notification Function
View Budget Manager
Generate Notification
return notification and display to user
return notification and display to user

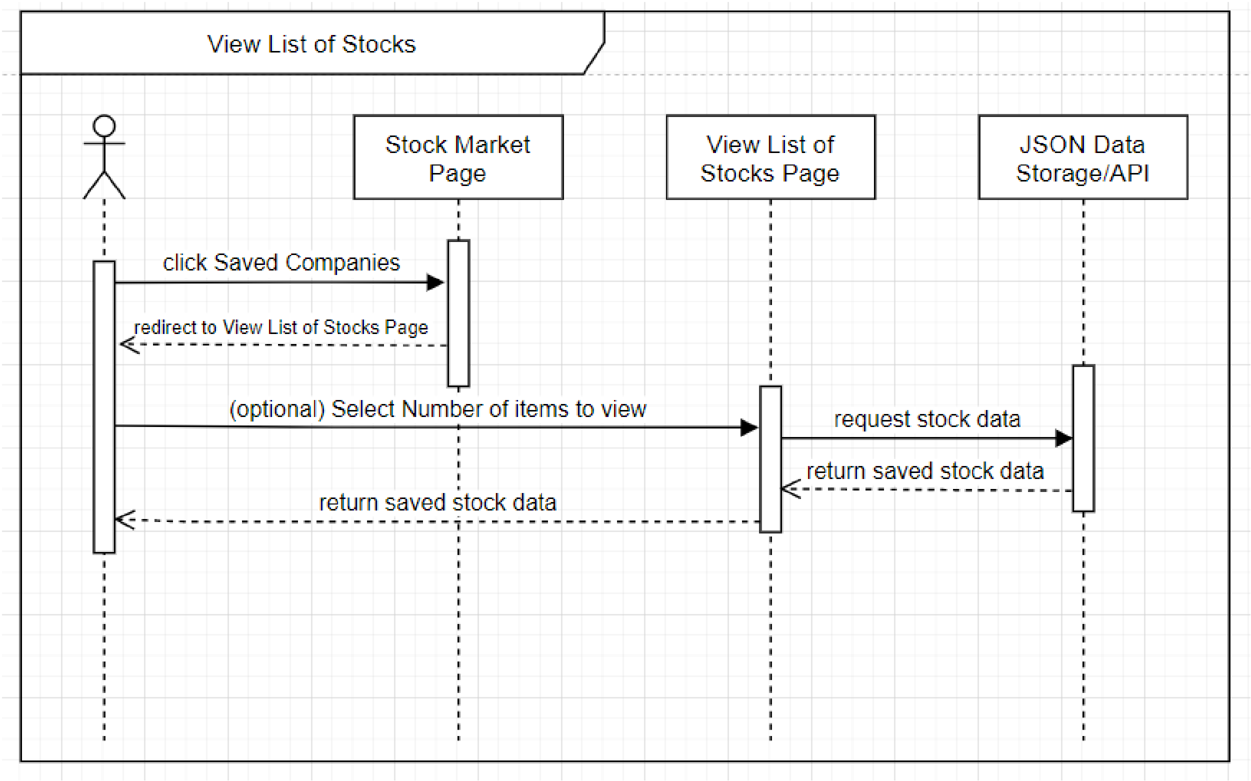
4.3.7. Add / Remove Expenses



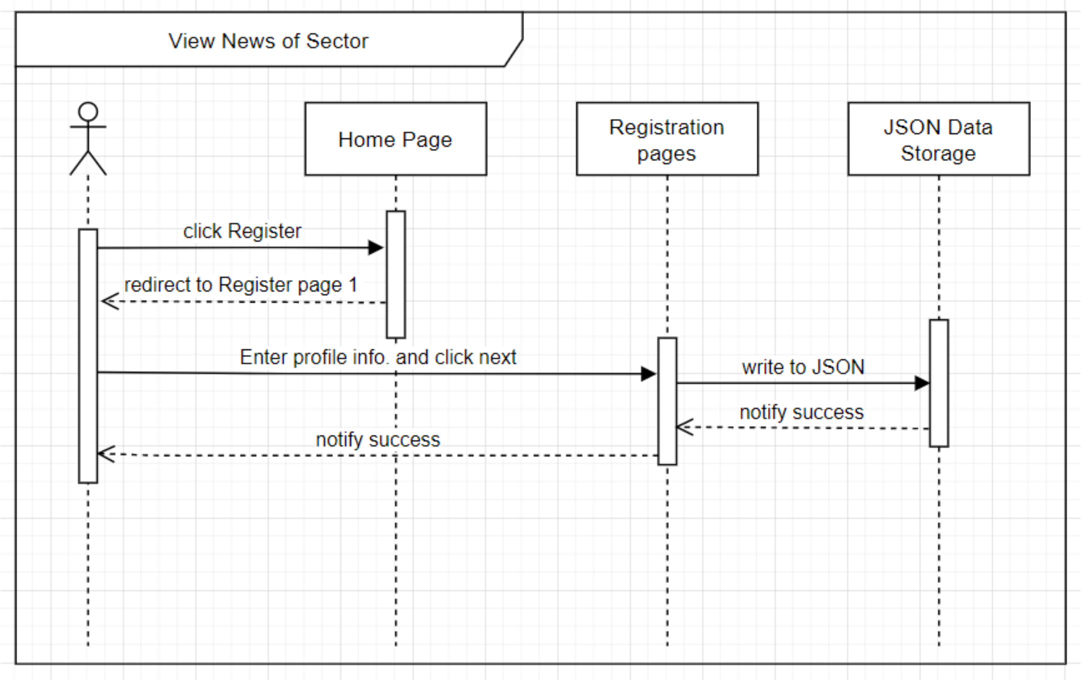
4.3.8. Select a Sector



4.3.9. View List of Stocks



4.3.10. View News of Sector



5. Others

5.1 - FinnHub Stock API

https://finnhub.io/docs/api

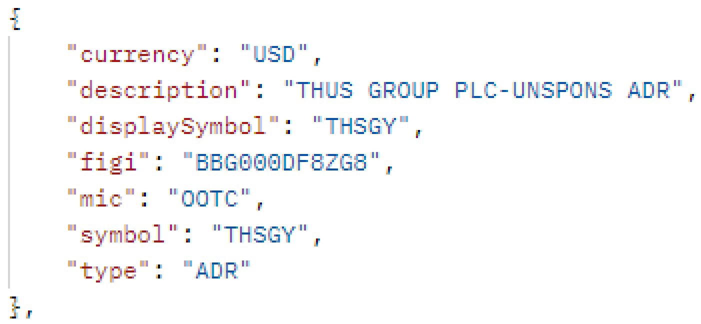
APIs:

Stock Symbols

Request:

GET https://finnhub.io/api/v1/stock/symbol?exchange=US&token=c184np748v6reqlb8lr0

Response:



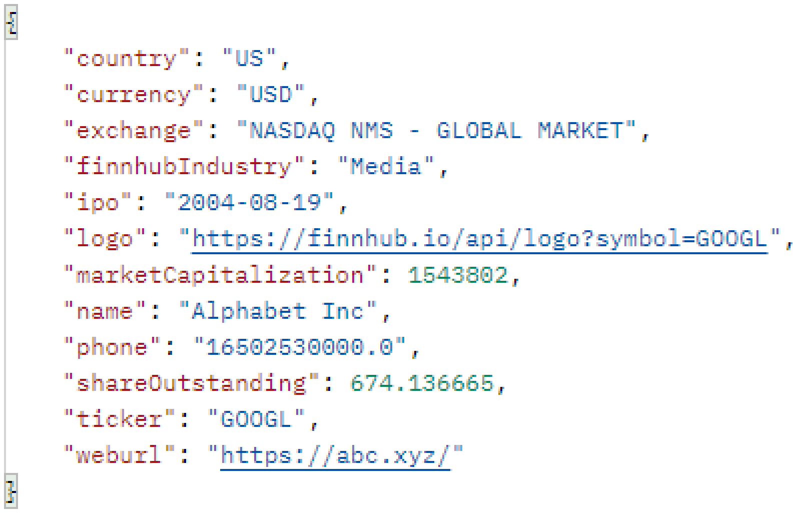
```
{
    "currency": "USD",
    "description": "THUS GROUP PLC-UNSPONS ADR",
    "displaySymbol": "THSGY",
    "figi": "BBG000DF8ZG8",
    "mic": "OOTC",
    "symbol": "THSGY",
    "type": "ADR"
},
```

Company Profile 2

Request:

GET https://finnhub.io/api/v1/stock/profile2?symbol=GOOGL&token=c184np748v6reqlb8lr0

Response:



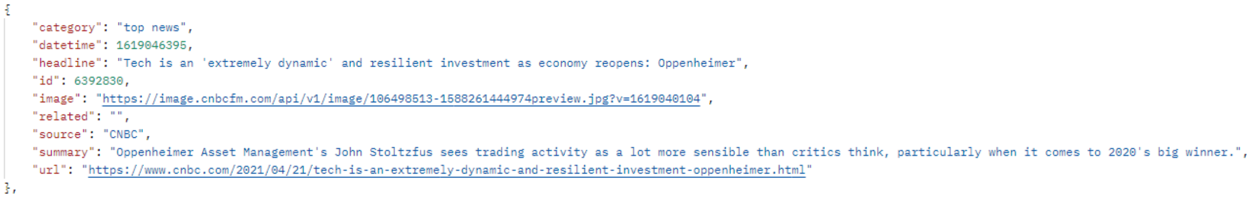
```
{
    "country": "US",
    "currency": "USD",
    "exchange": "NASDAQ NMS - GLOBAL MARKET",
    "finnhubIndustry": "Media",
    "ipo": "2004-08-19",
    "logo": "https://finnhub.io/api/logo?symbol=GOOGL",
    "marketCapitalization": 1543802,
    "name": "Alphabet Inc",
    "phone": "16502530000.0",
    "shareOutstanding": 674.136665,
    "ticker": "GOOGL",
    "weburl": "https://abc.xyz/"
}
```

Market News

Request:

GET https://finnhub.io/api/v1/news?category=general&token=c184np748v6reqlb8lr0

Response:



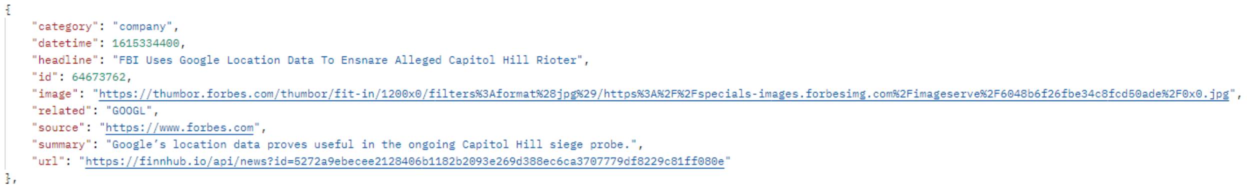
```
{
    "category": "top news",
    "datetime": 1619046395,
    "headline": "Tech is an 'extremely dynamic' and resilient investment as economy reopens: Oppenheimer",
    "id": 6392830,
    "image": "https://image.cnbcfm.com/api/v1/image/106498513-1588261444974preview.jpg?v=1619040104",
    "related": "",
    "source": "CNBC",
    "summary": "Oppenheimer Asset Management's John Stoltzfus sees trading activity as a lot more sensible than critics think, particularly when it comes to 2020's big winner.",
    "url": "https://www.cnbc.com/2021/04/21/tech-is-an-extremely-dynamic-and-resilient-investment-oppenheimer.html"
},
```

Company News

Request:

GET https://finnhub.io/api/v1/stock/symbol?exchange=US&token=c184np748v6reqlb8lr0

Response:



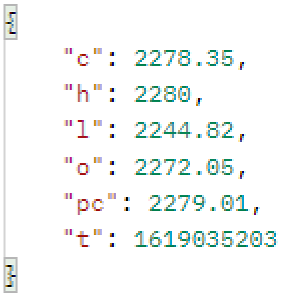
```
{
    "category": "company",
    "datetime": 1615334400,
    "headline": "FBI Uses Google Location Data To Ensnare Alleged Capitol Hill Rioter",
    "id": 64673762,
    "image": "https://thumbor.forbes.com/thumbor/fit-in/1200x0/filters%3Aformat%28jpg%29/https%3A%2F%2Fspecials-images.forbesimg.com%2Fimageserve%2F6048b6f26fbe34c8fcd50ade%2F0x0.jpg",
    "related": "GOOGL",
    "source": "https://www.forbes.com",
    "summary": "Google’s location data proves useful in the ongoing Capitol Hill siege probe.",
    "url": "https://finnhub.io/api/news?id=5272a9ebecee2128406b1182b2093e269d388ec6ca3707779df8229c81ff080e"
},
```

Quote

Request:

GET https://finnhub.io/api/v1/quote?symbol=GOOGL&token=c184np748v6reqlb8lr0

Response:



```
{
    "c": 2278.35,
    "h": 2280,
    "l": 2244.82,
    "o": 2272.05,
    "pc": 2279.01,
    "t": 1619035203
}
```

6. Test plan

6.1. Manage Profile

6.1.1. Create a New Profile

Table 1 . Test Cases for creating a new profile

No.	*Test case*	*User input*	*Pass criteria*
1	User enters in blank or username with only spaces	Username:	Display error message “Select valid username” and let user choose again
2	User enters in an already existing profile username	Username: Ian	Display error message “Username already taken” and let user choose again
3	User enters in a username with at least 1 character	Username: Michael	Program accepts name, and brings user to next page to select features they want to enable
4	User does not select at least 1 feature to enable during registration	Feature 1: [] Feature 2: []	Display error message “Please select at least 1 feature to enable” and let user choose again
5	User selects at least 1 feature to enable during registration	Feature 1: [X] Feature 2: []	Program accepts chosen features, and brings user to next page to set their balance

6.1.2. Edit User Profile

Table 2. Test Cases for editing user profiles

No.	*Test case*	*User input*	*Pass criteria*
1	User enters in blank or new username with only spaces	Username:	Display error message “Select valid username” and let user choose again
2	User enters in an already existing profile username	Username: Ian	Display error message “Username already taken” and let user choose again
3	User enters in a username with at least 1 character	Username: Miguel	User profile is created and display success message to user
4	User does not select at least 1 feature to enable	Feature 1: [] Feature 2: []	Display error message “Please select at least 1 feature to enable” and let user choose again
5	User selects at least 1 feature to enable	Feature 1: [X] Feature 2: [X]	User profile is modified to reflect the user’s changes

6.1.3. Login

Table 3. Test Cases for logging into profiles

No.	*Test case*	*User input*	*Pass criteria*

1	User attempts to login but they did not create any profiles beforehand	Clicks on Login	Display error message “Please register a profile first”
2	User attempts to login to their existing profile	Clicks on Login - Clicks on their profile name	Display successful login message and then display the user’s homepage

6.2. Manage Budget

6.2.1. Set Total Balance

Table 4. Test Cases for setting a total balance

No.	*Test case*	*User input*	*Pass criteria*
1	User enters in a non-numeric input	Balance: Joe	Display error message “Enter in a valid balance” and let user choose again
2	User enters in a negative number	Balance: -100	Display error message “Enter in a valid balance” and let user choose again
3	User enters in a number with more than 3 decimal places	Balance: 150.625	Program rounds the balance amount to the nearest hundredth,accepts balance amount, and displays success message to user
4	User enters in a positive numeric input	Balance: 100	Program accepts balance amount, and displays success message to user

6.2.2. Set Target Budget

Table 5. Test Cases for setting a target budget

No.	*Test case*	*User input*	*Pass criteria*
1	User enters in a non-numeric input	Budget: Joe	Display error message “Enter in a valid budget” and let user choose again
2	User enters in a negative number	Budget: -100	Display error message “Enter in a valid budget” and let user choose again
3	User enters in a number with more than 3 decimal places	Budget: 150.625	Program rounds the budget amount to the nearest hundredth,accepts budget amount, and displays success message to user
4	User enters in a positive numeric input	Budget: 100	Program accepts budget amount, and displays success message to user

6.2.3. Notifications

Table 6. Test Cases for notifications

No.	*Test case*	*User input*	*Pass criteria*
1	Visit the budget manager page 8+ times	View budget manager and exit budget manager 8 times	All notifications displayed are valid and truthful (and likely different each time)

6.2.4. Add / Remove Expenses

Table 7. Test Cases for adding and removing expenses

No.	*Test case*	*User input*	*Pass criteria*
1	User does not type anything for expense description	Expense description:	Display error message “Enter in a valid expense description” and let user choose again
2	User enters in at least 1 character for the expense description	Expense description: Netflix Subscription	Program accepts expense description
3	User does not type anything for expense amount	Expense amount:	Display error message “Enter in a valid expense amount” and let user choose again
4	User enters in a non-numeric input for expense amount	Expense amount: Joe	Display error message “Enter in a valid expense amount” and let user choose again
5	User enters in a negative number for expense amount	Expense amount: -100	Display error message “Enter in a valid expense amount” and let user choose again
6	User enters in a number with more than 3 decimal places for expense amount	Expense amount: 150.625	Program rounds the expense amount to the nearest hundredth, accepts expense amount
7	User enters in a positive numeric input for expense amount	Expense amount: 100	Program accepts expense amount
8	User leaves date field empty	Date:	Display error message “Enter in a valid date” and let user choose again
9	User enters in an invalid date	Date: Joseph	Display error message “Enter in a valid date” and let user choose again

10	User enters in a valid date	Date: 3/24/2021	Program accepts date, and displays success message to user assuming all other fields are valid

6.3 Stock Market

6.3.1 Select a Sector

Table 8. Test Cases for selecting a sector

No.	*Test case*	*User input*	*Pass criteria*
1	User does not select at least 1 sector	Energy[] Industrials[] …[]	Display error message “Select at least 1 sector” and let user choose again
2	User selects at least 1 sector	Energy[] Industrials[X] …[]	Program accepts input

6.3.2 View List of Stocks

Table 9. Test Cases for viewing list of stocks

No.	*Test case*	*User input*	*Pass criteria*
1	User attempts to view list of stocks with none selected	User clicks view stocks	Display error message “Select stocks to view first” and let user select stocks
2	User attempts to view list of stocks with at least 1 selected	User clicks view stocks	Display desired stocks to user

6.3.3 View News of Sector

Table 10. Test Cases for viewing news of sector

No.	*Test case*	*User input*	*Pass criteria*
1	User attempts to view news of sector without an internet connection	User clicks view news of a sector	Display error message “Could not retrieve news. Do you have an internet connection?” and let user try again
2	User attempts to view news of sector with an internet connection	User clicks view news of a sector	Display desired sector news to user

7. References

1. https://stocknews.com/