



# **Software Engineering project 2023/2024**

## **Ca' Foscari University of Venice**



**Design Document**  
**2.0**

**AgileMasters**

**15/01/2024**



### Document Informations

ProjectName	FinanceFlow	Acronym	FinFlow
Deliverable	Requirements Analysis		
Delivery date	15/01/2024		
Team leader	Alberto Tomasin		<a href="mailto:892614@stud.unive.it">892614@stud.unive.it</a>
Team members	André Ramolivaz Simone Dinato Mirco De Zorzi		<a href="mailto:891923@stud.unive.it">891923@stud.unive.it</a> <a href="mailto:892539@stud.unive.it">892539@stud.unive.it</a> <a href="mailto:891275@stud.unive.it">891275@stud.unive.it</a>

### Document History

Version	Issue Date	Stage	Changes	Contributors (reviewer)
0.1	07/11/2023	Draft	Finish chapter 1	Alberto Tomasin, (Andrè Ramolivaz)
0.2	09/11/2023	Draft	Finish chapter 2	Mirco De Zorzi, (Andrè Ramolivaz)
0.3	10/11/2023	Draft	Finish chapter 3	Alberto Tomasin, (Simone Dinato)
0.4	15/11/2023	Draft	Finish chapter 4	Andrè Ramolivaz, (Simone Dinato)
0.5	20/11/2023	Draft	Finish chapter 5	Simone Dinato, Alberto Tomasin, (Andrè Ramolivaz)
0.6	21/11/2023	Draft	Finish chapter 6	Mirco De Zorzi, (Simone Dinato)
1.0	23/11/2023	Final	Finish Document	(Simone Dinato, Andrè Ramolivaz, Alberto Tomasin, Mirco De Zorzi)
2.0	15/01/2024	Final	Documentation Alignment	Simone Dinato, (Alberto Tomasin)



# Index

<b>1. Introduction</b>	<b>4</b>
<b>1.1. Document Overview</b>	<b>4</b>
<b>1.2. Glossary</b>	<b>4</b>
<b>2. System architecture</b>	<b>5</b>
<b>2.1. Model and System Structure</b>	<b>5</b>
<b>2.2. Data management</b>	<b>5</b>
2.2.1. PostgreSQL	5
2.2.2. Redis	5
2.2.3. Integration and Synchronization	5
<b>3. Data and Control Model</b>	<b>7</b>
<b>4. UML Model</b>	<b>8</b>
<b>4.1. Class Diagram</b>	<b>8</b>
<b>4.2. Activities Diagram</b>	<b>9</b>
<b>4.3. Sequence Diagram</b>	<b>13</b>
<b>5. User interface design</b>	<b>17</b>
<b>5.1. First Access</b>	<b>17</b>
5.1.1. Registration	17
5.1.2. Sign-In	17
<b>5.2. Bottom Bar</b>	<b>18</b>
<b>5.3. Feed Page</b>	<b>18</b>
5.3.1. Account Balance	19
<b>5.4. Stocks Page</b>	<b>20</b>
5.4.1. ChatBot	21
<b>5.5. News Page</b>	<b>21</b>
<b>5.6. Profile Page</b>	<b>22</b>
<b>6. References</b>	<b>23</b>



# 1. Introduction

## 1.1. Document Overview

This section outlines the primary objective of the document, which is to provide a clear and structured overview of the Android application's architecture. The document aims to detail the software's technical aspects, including how the front-end (user interface) and back-end (server-side processing) are designed and integrated. A notable focus is placed on the application's structural organization, highlighting how it's divided into distinct components:

- **Packages:** These are collections of related classes and interfaces, providing a hierarchical organization for the app's codebase.
- **Classes:** The building blocks of the app, representing the objects within the software and their interactions.
- **Fragments and Activities:** Essential elements of Android UI, where Activities represent single screens, and Fragments are reusable UI components within these screens.

The use of diagrams in this document serves to visually represent the application's architecture, aiding in understanding the relationships and workflows between different components.

## 1.2. Glossary

This section is crucial for clarifying technical terms that are specific to Android development and might not be familiar to all readers. Key terms include:

**Activity:** Central to Android apps, activities are essentially screens that facilitate user interaction. Each activity serves a specific purpose and presents a unique user interface.

**Fragment:** Representing a modular section of an interface, fragments are used for more dynamic and flexible UI designs. They can exist within activities and contain their own layout and behavior.



## 2. System architecture

### 2.1. Model and System Structure

The system structure will adopt a client-server architecture approach, a common choice in distributed systems. This approach is particularly suitable for our application, since the data critical to its operation is hosted on a central server and is transmitted to the application on demand.

The client-server architecture facilitates efficient distribution of data across an organized network. This configuration not only makes it easy to add new services, but also allows improvements to existing services to be made smoothly.

Specifically, the application will forward requests to the server to perform read and write operations on the data stored in the database. This mechanism ensures a constant and secure flow of data between the client and the server, optimizing the user experience and information management.

### 2.2. Data management

Data management in FinanceFlow is critical to ensure optimal performance and reliability. To achieve these goals, the application uses a combination of relational and nonrelational databases: PostgreSQL for user storage and Redis for transaction management.



#### 2.2.1. PostgreSQL

PostgreSQL, a relational database management system (RDBMS), is used to store structured and persistent data, such as user information.

It provides a solid foundation for storing data that requires referential integrity and support for complex queries.

Essential details such as users' first name, last name, email address, and date of birth are managed in a secure and structured manner.

#### 2.2.2. Redis

Redis, an in-memory data store, is used to manage all transactions within the application.

Its ability to provide high-speed, low-latency access makes it ideal for managing frequent and temporary transaction data.

#### 2.2.3. Integration and Synchronization

Although the two databases handle different types of data, there is a synchronization mechanism that ensures consistency and reliability between the systems. Users' information in PostgreSQL is linked to their transactions in Redis.



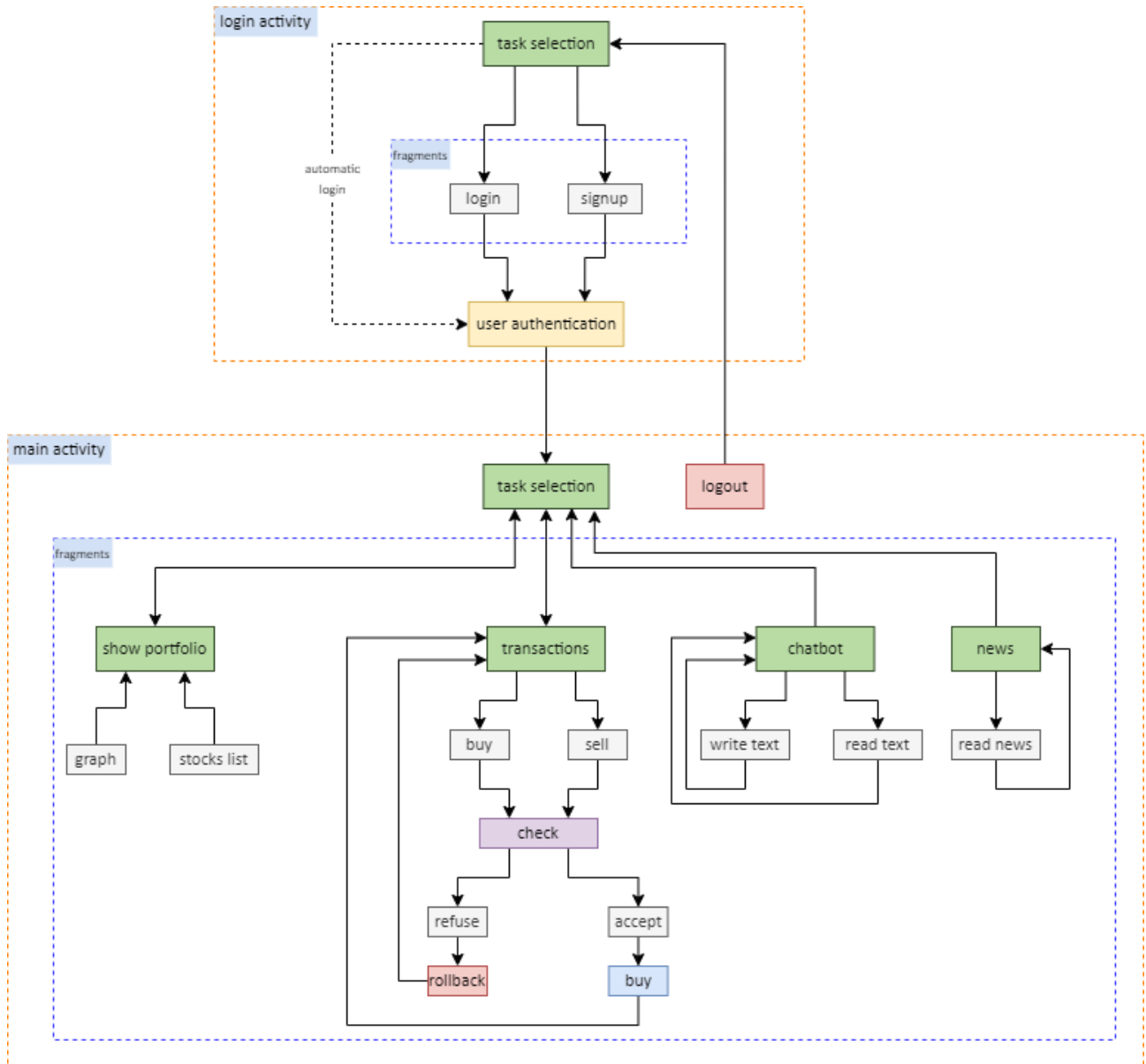
The use of PostgreSQL and Redis in synergy allows FinanceFlow to effectively manage user data and transactions. This configuration ensures security, speed and scalability, providing users with an optimal online trading experience.

Data management in FinanceFlow is critical to ensure optimal performance and reliability. To achieve these goals, the application uses a combination of relational and nonrelational databases: PostgreSQL for user storage and Redis for transaction management.



### 3. Data and Control Model

Control will be structured by a centralized system that will manage the entire application. A *call return* model will be used where control is managed in a top-down manner, as illustrated in the diagram below.



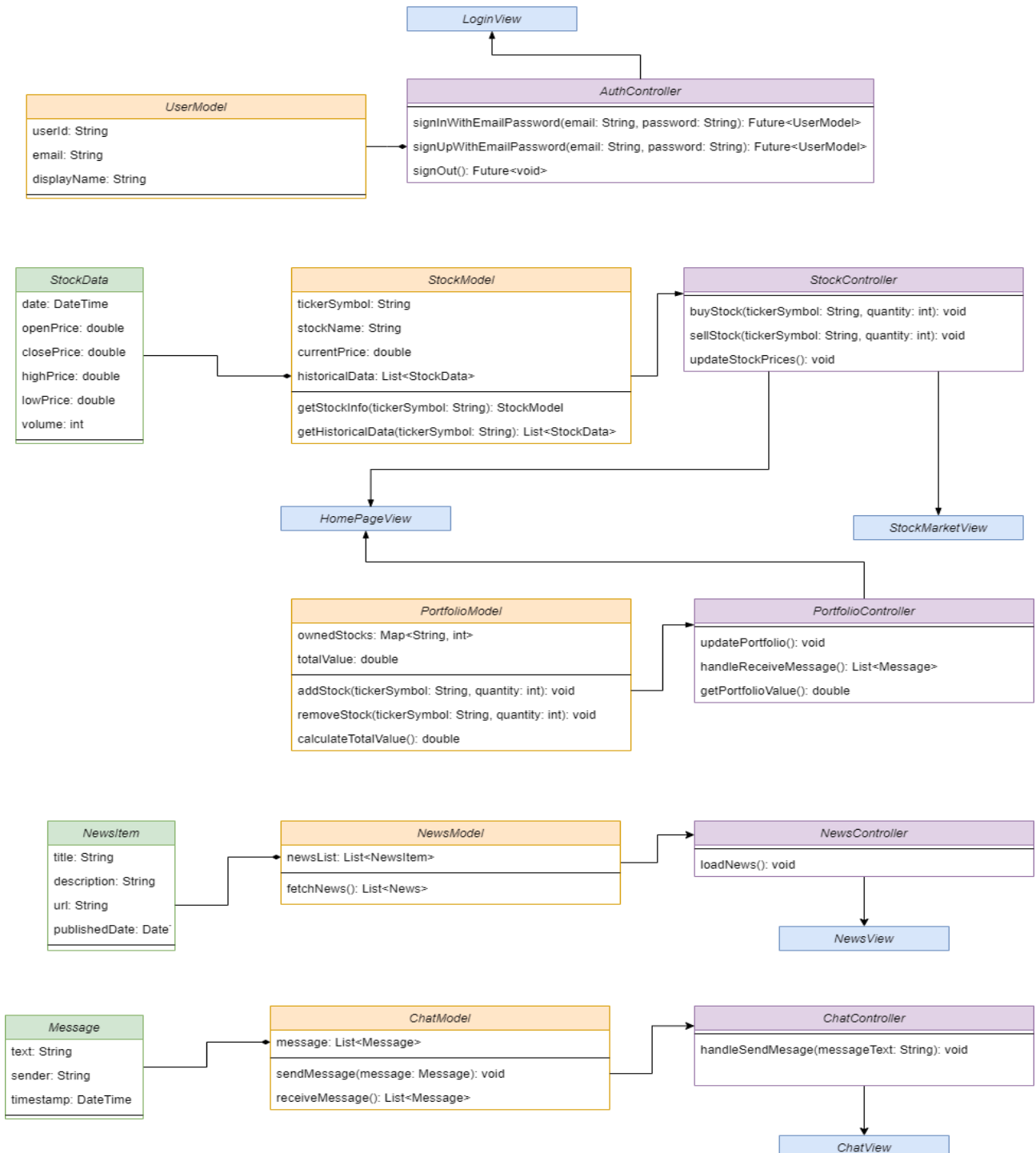
The application is structured to have two activities, login and main, and several fragments related to them. Unidirectional arrows indicate the sequential flow of actions executed, while bidirectional ones indicate a flowing flow in both directions. The single dashed arrow indicates the performance of an automatic action performed by the system.



## 4. UML Model

### 4.1. Class Diagram

The class diagram represents the object classes of the system with their attributes and methods. It also shows the relationships between the various classes, if any. The views, containing a number of classes used to generate the interface, have been represented simply by name to avoid burdening the visualization.

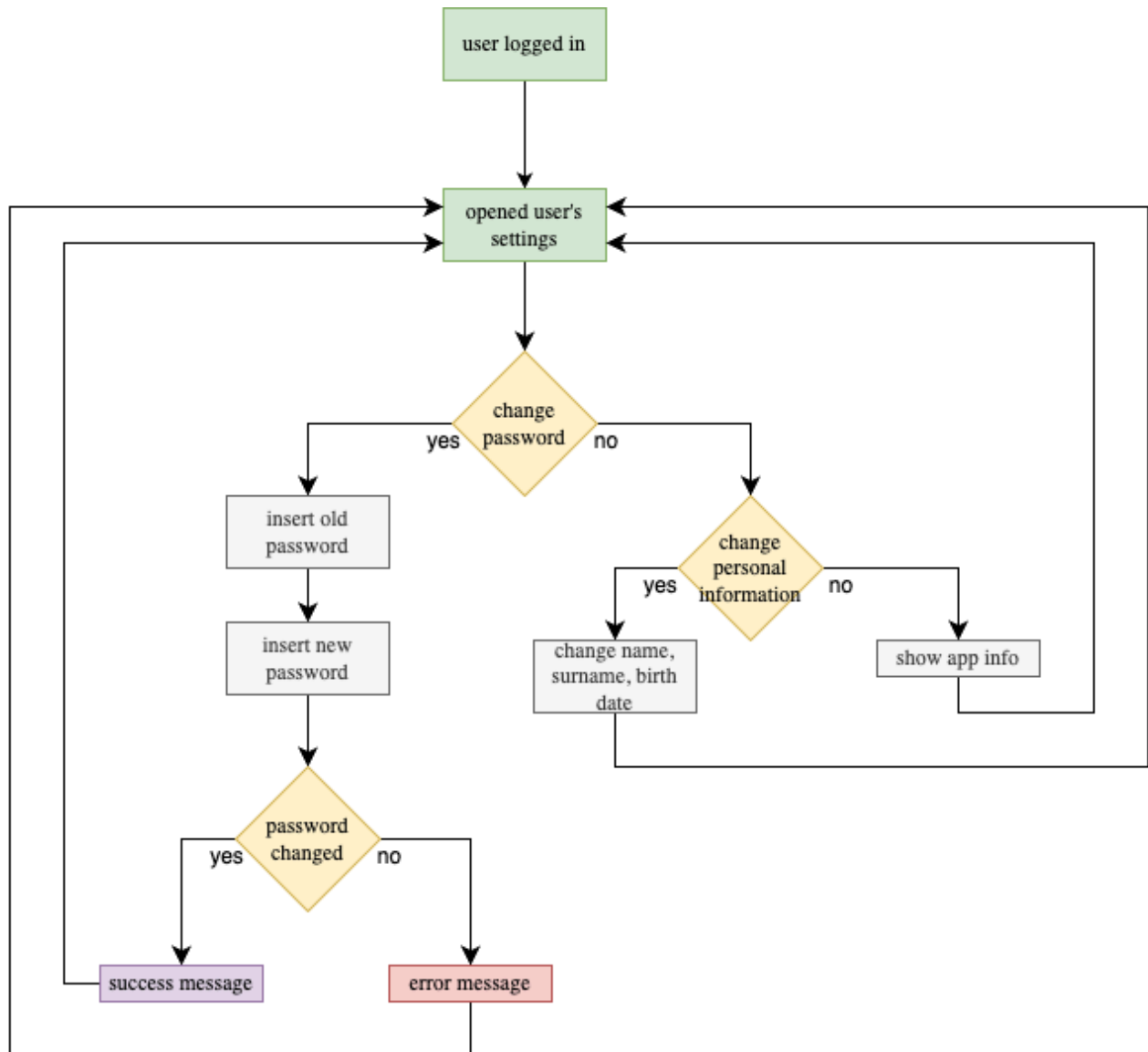


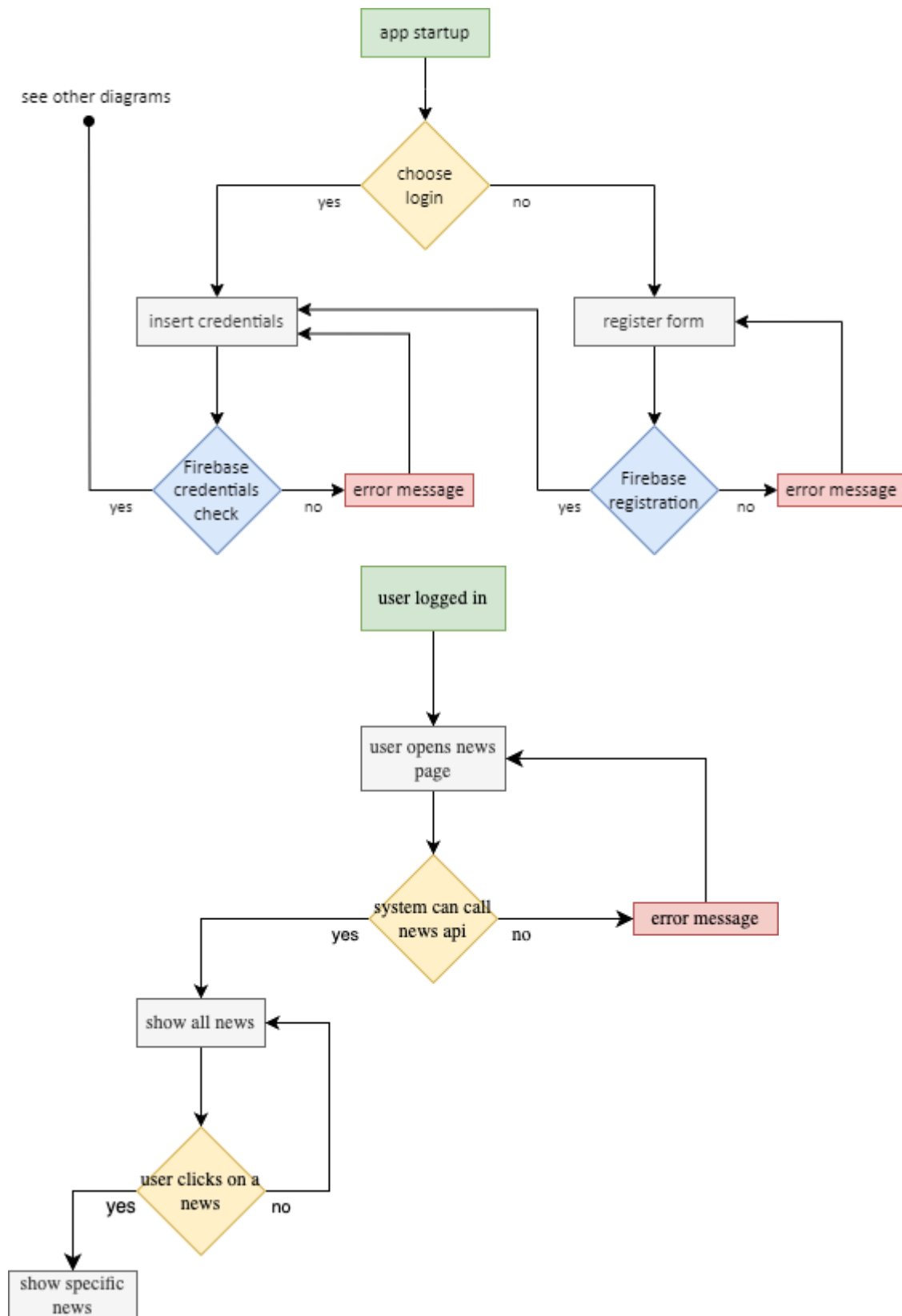


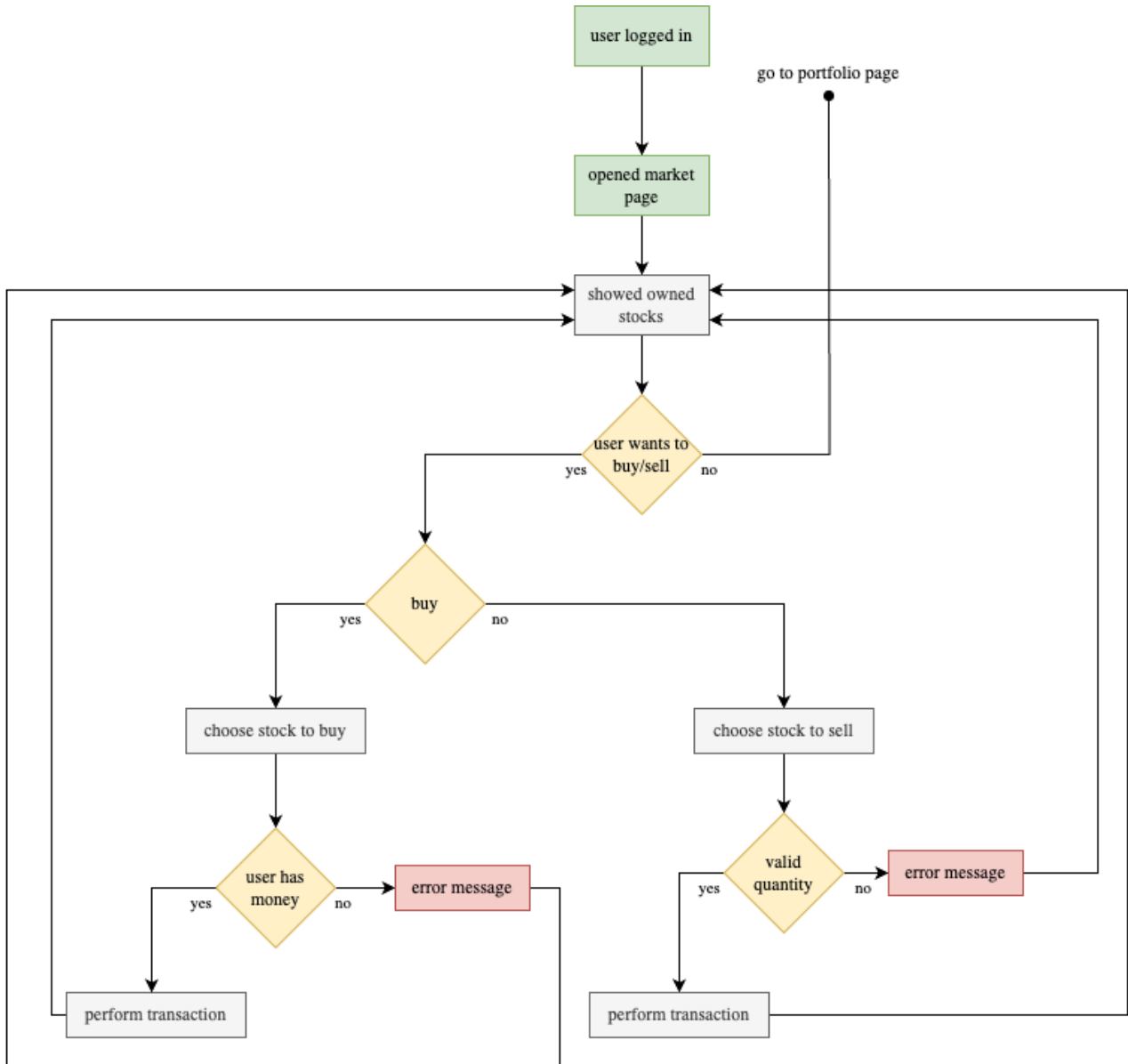


## 4.2. Activities Diagram

The activity diagram makes it possible to describe the process taking place within the system. It is a single diagram that describes the entire operation of the application. For pure illustrative simplicity they will be divided, although in reality they represent a single diagram.



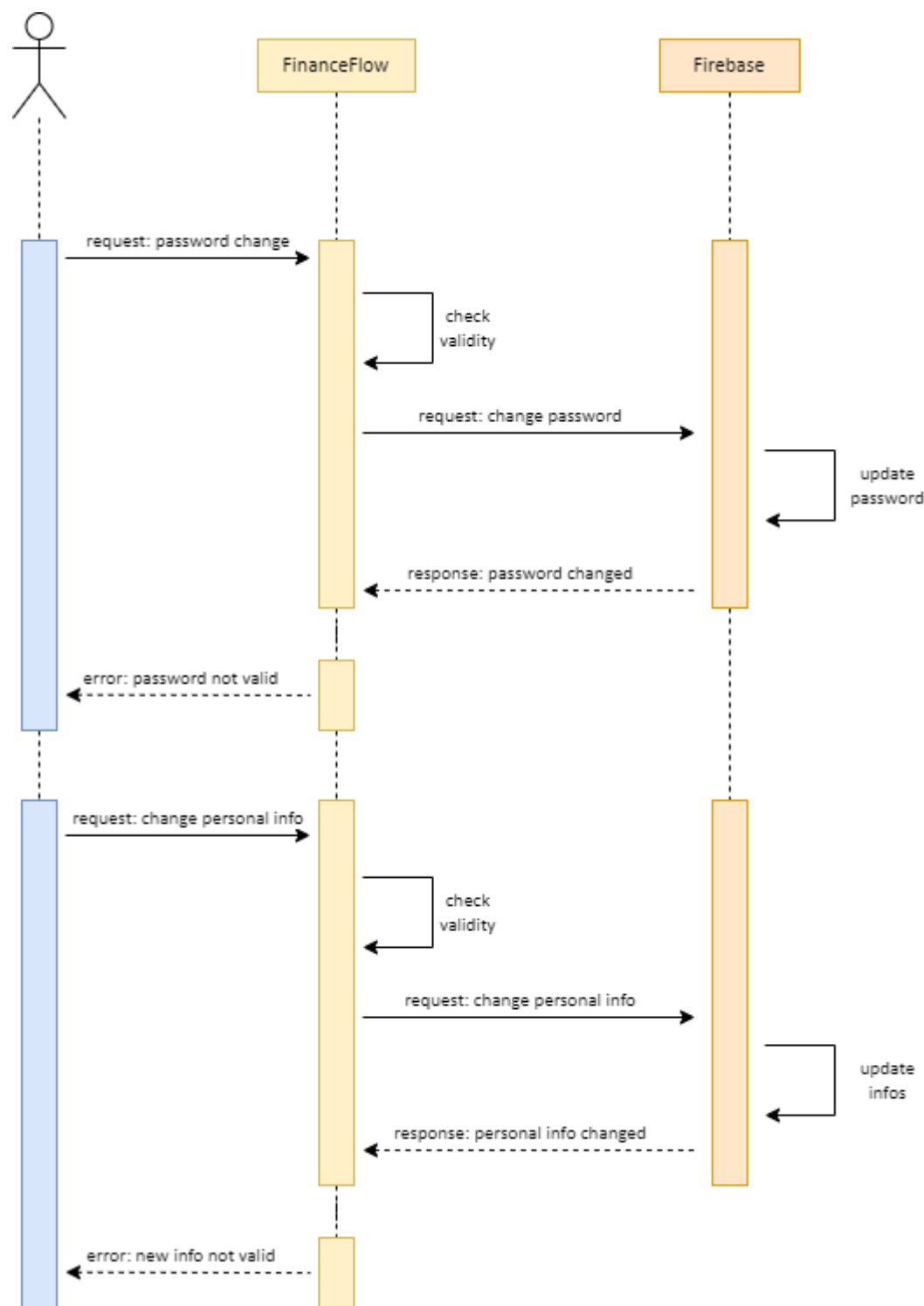






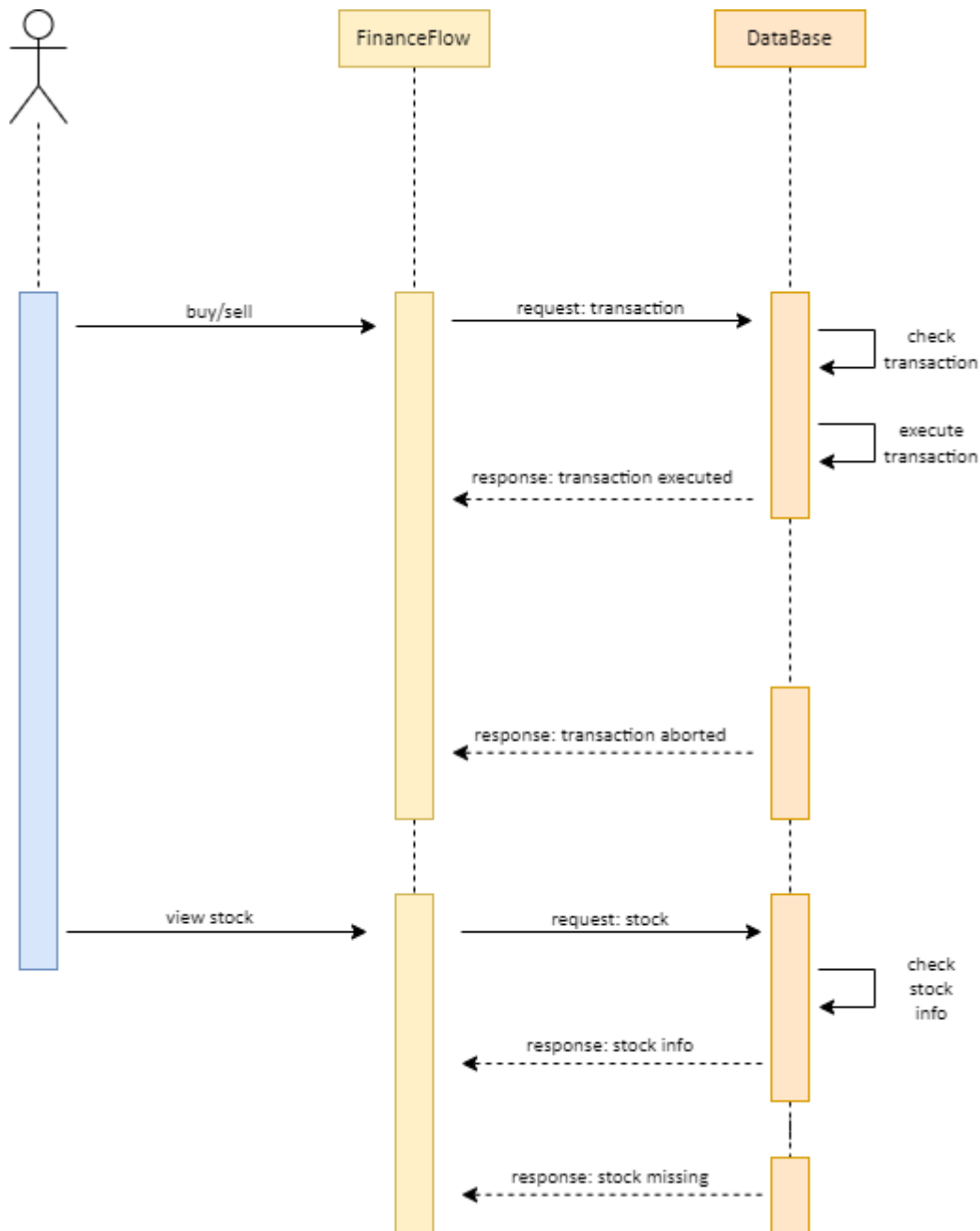


Account

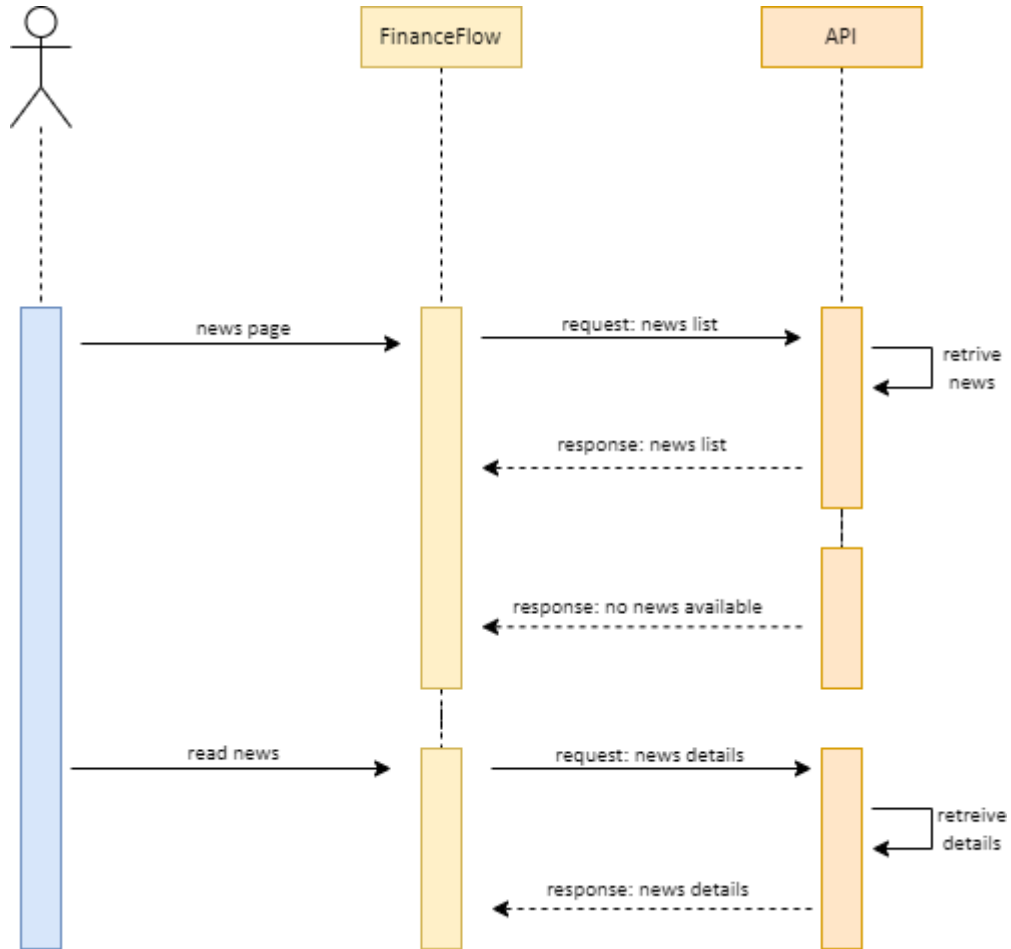




## Market



## News







## 5. User interface design

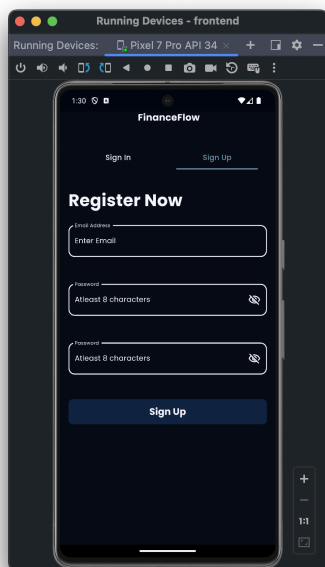
This section will list the screens of which the application will consist. For each of them, since they are not interactive prototypes, multiple images will be displayed in order to describe all the features.

### 5.1. First Access

During the initial visit, users are prompted to undergo a straightforward registration process on FinanceFlow. This involves providing necessary details, and completing the sign-in procedure. This sequential process ensures a secure and verified entry into the platform, safeguarding user accounts and fostering a trusted environment for engaging with the diverse educational resources offered by FinanceFlow.

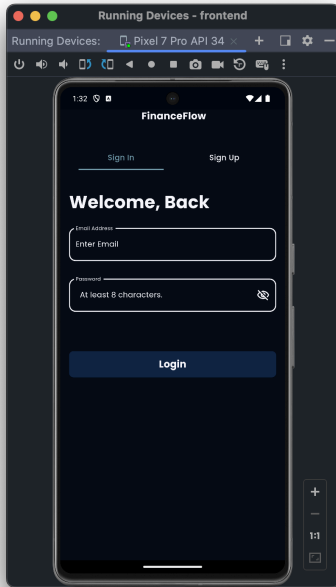
#### 5.1.1. Registration

The user registration is open to all ages, requiring only essential information—email, password, name, and surname. This streamlined process ensures accessibility for users at any stage of life, aligning with our commitment to universal financial education.



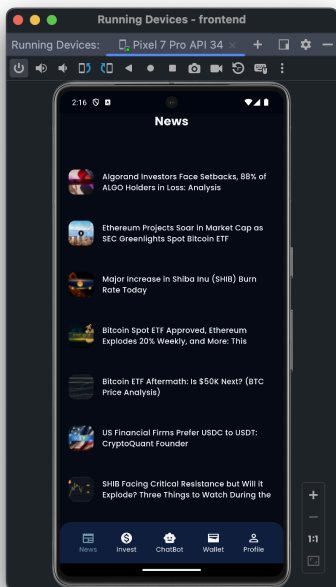
#### 5.1.2. Sign-In

Access serves as the concluding step for users in the registration process, marking the successful completion of the account creation. However, for users who already possess an account, access becomes the initial step as they don't need to create a new account.



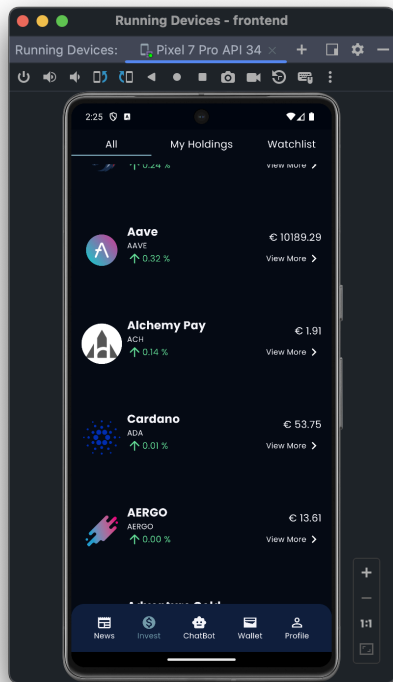
## 5.2. Bottom Bar

The bottom bar provides a convenient navigation tool, allowing users to effortlessly switch between four key pages: Home, Invest, Wallet, Profile.



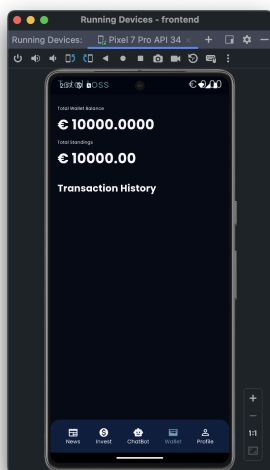
## 5.3. Feed Page

The Feed page stands as the focal point of our application, serving as the initial interactive interface post sign-in. This central hub provides a comprehensive overview, showcasing your account balance. Additionally, users can effortlessly view and manage their investments in stocks or cryptocurrencies, offering a user-friendly experience at the core of FinanceFlow's functionality.



### 5.3.1. Account Balance

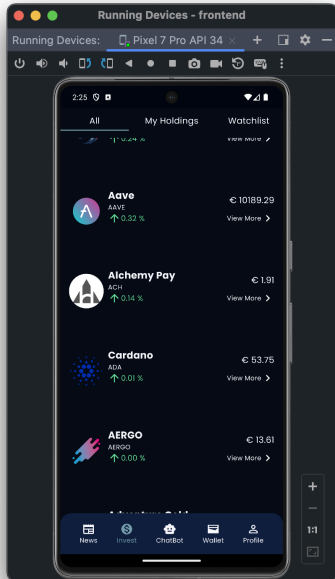
Within the Feed page, the Account Balance section offers an interactive feature—clicking it opens the dedicated Account Balance subpage. Here, users can delve into a visual representation of their account balance graph, gaining insights into its fluctuations over time. Additionally, this subpage provides a detailed overview of the associated stocks or cryptocurrencies that have been purchased.





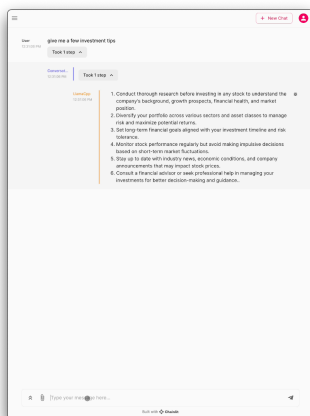
## 5.4. Stocks Page

The Stocks page, positioned second in the bottom menu, serves as a dynamic hub. Here, users can seamlessly search for specific stocks, stay updated on trending options, and execute buying or selling transactions.



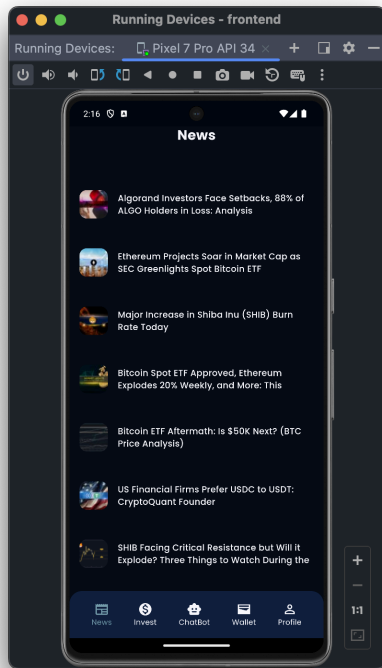
### 5.4.1. ChatBot

Users have the flexibility to initiate a new chat or join an existing one to engage with the ChatBot. This feature enhances user interaction by providing a personalized and conversational avenue for addressing queries, receiving assistance, or gaining insights on financial matters.



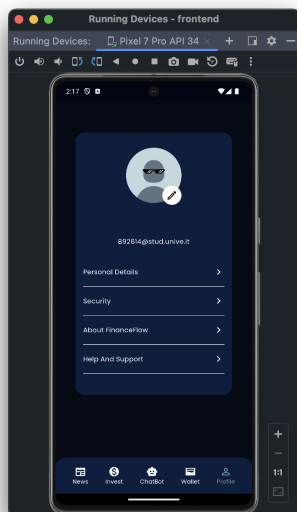
## 5.5. News Page

The News page on FinanceFlow offers a user-friendly, scrollable interface where news is categorized for easy navigation. Users can explore various categories such as Most Recent, Tech, Finance and more.



## 5.6. Profile Page

The Profile page is a straightforward yet comprehensive space that showcases user information. Here, users can easily manage their profile picture and update personal details. Additionally, the page offers convenient access to the user manual guide.





## 6. References

The following documents were taken into consideration for the preparation of the following paper:

- Analysis and specification documents related to previous years' projects.
- Course slides provided by the teacher.