CENG 407

Innovative System Design and Development I

2024-2025 Fall

e-TurFinSAS: Entity Based Turkish Financial Sentiment Analysis System

Team 20

Software Design Description

Names of members:

- Taner Onur UYAR Student Number: 202011014
- Ahmet Eren YAĞLI Student Number: 202011038
- Baha ÖÇALAN Student Number: 202011056
- Ahmet Gökay ÜRKMEZ Student Number: 202011057
- Mert Şerafettin KARGI Student Number: 202011061

27.12.2024

Table of Contents

List of Figures

1.INTRODUCTION

1.1. Purpose

The purpose of the e-TurFinSAS project is to create an entity-based Financial Sentiment Analysis system with Turkish texts from X, TradingView, KAP and Investing related to BIST100. System analyses Turkish content, and classify sentiment as positive, neutral and negative regarding BIST100 companies. The meaning of this project for users are assisting them in making better decisions about their investments, enhancing prediction of financial trends and stock market dynamics.

The goal of this Software Design Description (SDD) is to provide technical guidance for the development of the project. This document outlines the architectural and design details of the system. The document acts as reference for future development, ensuring alignment with the project's purpose and requirements, with the respect to information in the document.

1.2. Scope of the Project

This Software Design Description outlines the design and architecture of the "e-TurFinSAS: Entity-Based Turkish Financial Sentiment Analysis System." The document provides a detailed description of the system's objectives, functional and non-functional requirements, design decisions, and architectural components. It aims to serve as a comprehensive reference for stakeholders, including developers, project managers, and future maintainers, ensuring alignment on the technical aspects of the system.

The scope of the document includes:

- **System Overview:** Define the purpose of the system, which is to perform entity-based sentiment analysis on Turkish financial texts, particularly social media data, to classify sentiment as positive, negative, or neutral.
- **Functional Requirements:** Details the expected behavior of the system, including the collection, processing, and analysis of social media data for sentiment classification.

- **Design Considerations:** Discusses challenges such as the morphological complexity of Turkish, data preprocessing, and model training for the financial domain.
- **Design Approach:** Describe the high-level architecture of the system, including backend data collection, natural language processing (NLP) pipelines, and model integration for sentiment analysis. Focus on modularity and scalability to ensure adaptability to future requirements.
- **System Architecture:** Describes the software components, their interactions, and how they integrate to meet project goals.
- **Technical Constraints:** Identifies limitations such as dataset availability, processing power, and real-time analysis requirements.
- **Evaluation Metrics:** Defines the success criteria, including accuracy, precision, recall, and F1 score for sentiment analysis models.
- **Risk Assessment:** Addresses potential risks and mitigation strategies related to data privacy, ethical considerations, and model performance.

This document provides a foundation for developing a robust and efficient system that meets the technical and functional requirements of the project. It serves as a reference for the development team and ensures alignment across all phases of the project lifecycle.

1.3. Glossary

Term	Definition
Accuracy	Measure of the frequency of accurate sentiment analysis predictions.
Activity Diagram	Graphical representation of workflows and processes within the system.
Admin Page	An interface for the system that lets administrators change configurations, data, and methods.

API (Application Programming Interface)	Set of protocols for interacting with software applications or data sources.
BIST100	The top 100 companies listed on Borsa Istanbul, a major Turkish stock market index.
Class Diagram	A diagram illustrating the structure and relationships of classes in the system architecture.
Comparison Feature	Function that enables users to compare stock performance of different companies.
Dataset	Structured collection of the data used for training and testing sentiment analysis models.
e-TurFinSAS	Entity-Based Turkish Financial Sentiment Analysis System designed for BIST100 companies.
Entity-Based Sentiment Analysis	Analysing sentiment specific to entities such as companies or stocks within financial contexts.
F1 Score	The harmonic mean of precision and recall, used as an evaluation metric for sentiment analysis.
Firebase	A backend-as-a-service platform used for database and application management.
Investing	Financial website that provides market data, used as a data source in this project.
KAP	Public Disclosure Platform (Kamuyu Aydınlatma Platformu), a source for Turkish financial data.
Main Page	The central interface for users to access system functionalities like analysis and comparisons.

NLP (Natural Language Processing)	The field of AI focused on the interaction between computers and human language.
Precision	A metric to evaluate the relevance of positive sentiment predictions made by the system.
Preprocessing	Steps taken to clean and prepare raw data for analysis, such as tokenization or normalization.
Recall	A metric to assess the system's ability to identify all relevant sentiment instances.
SDD	Software Design Document.
Sentiment Analysis	Process to determine the emotional tone (positive, negative, or neutral) within text data.
TensorFlow	Open-source machine learning framework used for model development in this project.
TradingView	Platform that provides stock market analysis and data, utilized in this project.
X (formerly Twitter)	Social media platform used as a data source for gathering public sentiment about companies.

1.4. Motivation

We are senior Computer Engineering students who are interested in Data Analytics, NLP (Natural language processing) and Finance. To further develop our ideas on these topics, we are taking the NLP course and conducting research and studies in this field. Our goal is to contribute to the Turkish Stock Exchange and Turkish NLP technologies by combining our knowledge in these fields and providing accessible and useful analysis to users and companies.

In order to achieve our goal, we have chosen familiar platforms and tools such as **Python, TensorFlow and Firebase**. In finance, we have taken and continue to take Turkish data from reliable and popular sources such as KAP, X (formerly Twitter) and Investing, taking care to clean the data. In the sentiment analysis part, we are trying to carefully research and develop up-to-date methods for Turkish, which is a complex language. This project combines NLP technologies and the field of finance, while making a significant contribution to the literature for the Turkish language and the Turkish stock market.

1.5. Overview of Document

Here are the remaining chapters and what they include:

Section 2 is about Architectural Design. It explains how the project was developed. This part also has the class diagram of the system and the architecture design for the simulation. It talks about things like actors, exceptions, basic steps, priorities, and conditions before and after. There's also an activity diagram for the scenario generator in this section.

Section 3 is Use Case Realization. It shows and explains a block diagram of the system. This diagram is made based on the use cases in the SRS document.

Section 4 is about the Environment. Here, we show example frames of the environment from the prototype and describe the scenario.

2. ARCHITECTURE DESIGN

2.1. Simulation Design Approach

We decided to use the agile methodology while doing our project. Agile methodology is a flexible and collaborative approach used in software development and project management. In short, agility is a paradigm shift in the usual way of working. The word agile has passed into Turkish as "çeviklik". The concept of agile, which is the ability to adapt quickly to changing conditions; Although it is defined as method, methodology, method or project management in most sources, it is a way of thinking contrary to these. At the core of this methodology are the principles that individuals and interactions are more valuable than processes, a working product is more important than comprehensive documentation, customer collaboration is more important than contract negotiations and adapting to change is more valuable than following a plan. Agile aims to deliver projects faster, increase customer satisfaction, and adapt quickly to changing requirements through small, manageable work cycles (e.g., sprints). Although it was initially developed for software development, today it is widely used in different sectors such as marketing, finance and manufacturing. Adopting a continuous improvement and value-oriented approach, Agile enables teams to work more efficiently and effectively.

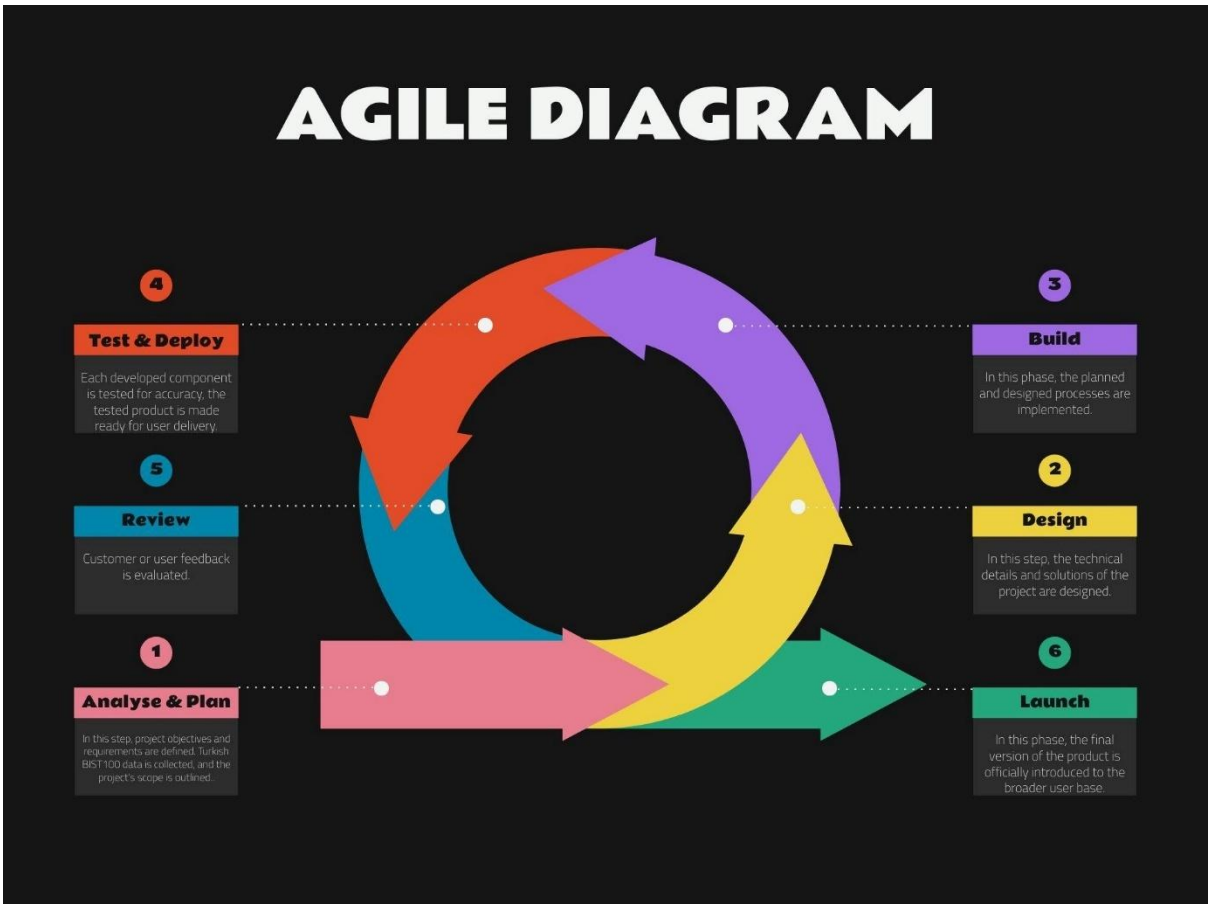

Figure 1. Agile Diagram

- In the **Analysis & Planning** phase, it is decided which data will be collected, from where, and under which topics. Along with task distribution for the team, deficiencies and requirements are identified, and a timeline is created.
- In the **Design** phase, decisions are made regarding which technologies and methods will be used for the project. Since a Turkish NLP model will be created at this stage, care is taken to select methods that are effective for Turkish, and answers are sought to the question of how these methods and the literature can be contributed to.
- In the **Build** phase, the project is implemented by following the previous two phases as closely as possible. This phase includes the NLP model, UI, backend, and all other remaining project components.
- In the **Test & Deploy** phase, all components of the final version of the project are tested, and it is prepared for release.
- In the **Review** phase, the final version of the project is shared with stakeholders and optionally with potential users, and feedback is collected. Stakeholders make new decisions regarding the project's shortcomings and additional requests, marking the end

of the cycle for the next iteration. After this step, the project can either be published or move into a new agile cycle, with all steps repeated.

- In the **Launch** phase, the final version of the completed project is published. After the launch, maintenance and updates are carried out regularly.

2.1.1. Class Diagram

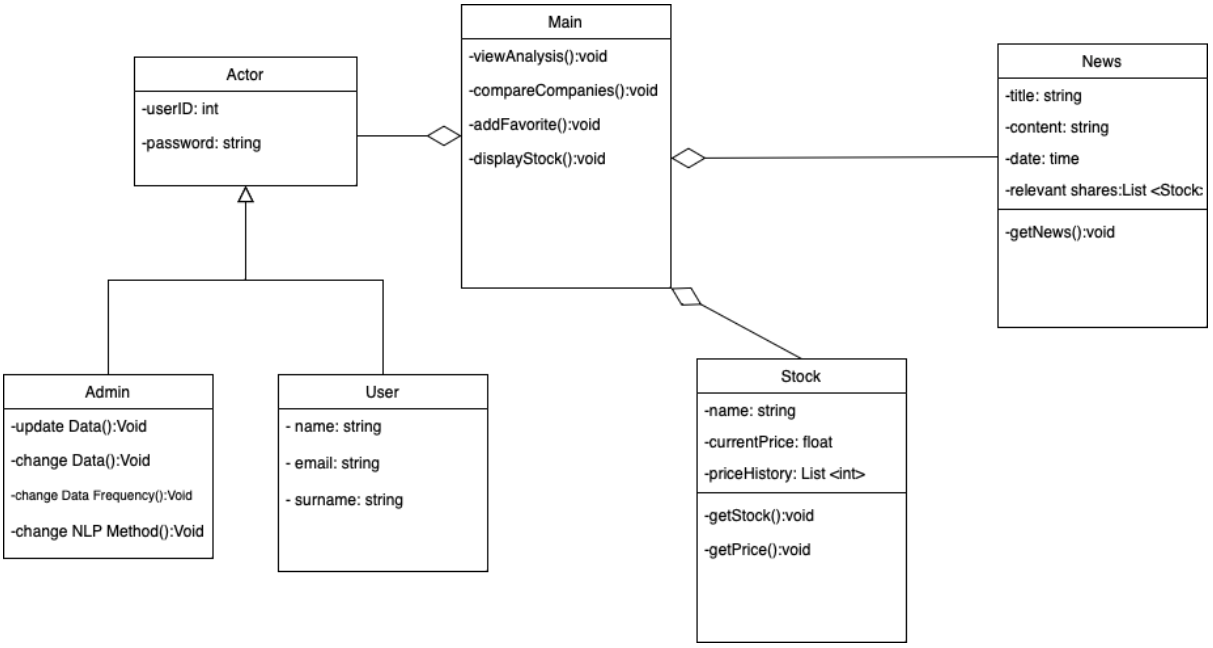

Figure 2. Class Diagram of e-TurFinSAS

The class diagram illustrates the structure and relationships of the e-TurFinSAS system components. The Main Class serves as the central system, managing the connections and interactions between the various components, such as Actor, Stock, and News. It facilitates the flow of data and ensures seamless operation across the system. The Actor Class represents all individuals interacting with the system. The Admin Class, a specialized Actor, is responsible for managing and maintaining the system, ensuring security, and overseeing user activities. The User Class represents regular users who can register, log in, and request sentiment analysis for stocks. Users can also view historical data and trends.

The Stock Class contains information about the stocks being analysed, including their names, symbols, and relevant metadata. This class interacts with the News Class for retrieving stock-specific content for sentiment analysis. The News Class represents text data sources like articles, tweets, or reports used for analysis. This class provides the input data needed for sentiment labelling and prediction in the backend.

Relationships between classes include the Main Class connecting Actors with Stock and News classes, ensuring that users and admins can interact with the system effectively. Admins manage Stocks and News, while Users request sentiment analysis and view results derived from these components. This structure ensures that all components work together cohesively, allowing users to seamlessly interact with the system while admins oversee operations and maintain its reliability.

2.2. Architecture Design of Simulation

2.2.1. Register Page

Summary: The system adds to new users into DB. Users need to input their email, password, name-surname and phone number information to register successfully.

Actor: User

Precondition: User must not already have an account.

Basic Sequence:

1. User opens the application and navigates to the Register page.
2. User enters their email, password, and other required details.
3. User clicks the "Register" button.
4. System validates the input and creates a new account if all information is valid.

Exception: If any input data is invalid program shows a relevant error message.

Post Condition: A new account is created; the user is redirected to the login page.

Priority: High

Activity Diagram:

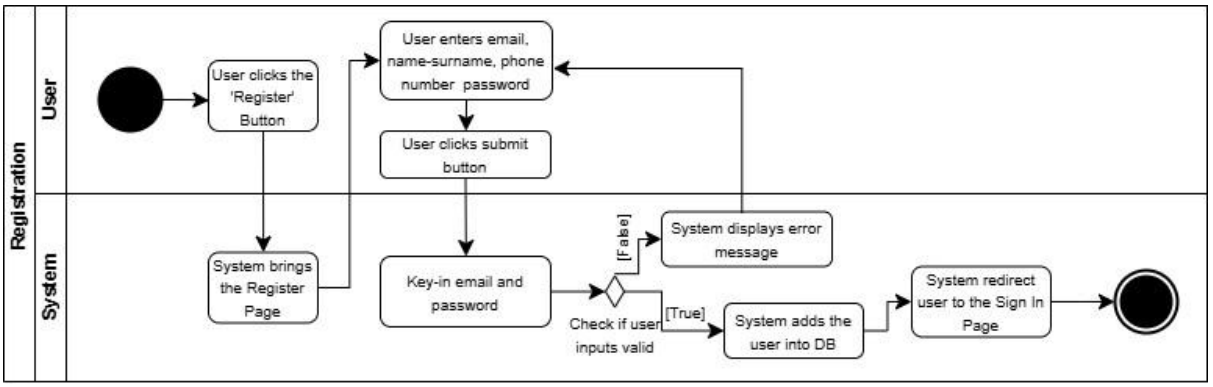

Figure 3. Activity Diagram of Register Page

2.2.2. Login Page

Summary: The system allows users or admin to enter their email and password to sign in the system.

Actor: User

Precondition: User must have an account.

Basic Sequence:

1. User opens the application and navigates to the Login page.
2. User enters their email and password.
3. User clicks the "Login" button.
4. System verifies the credentials and grants access if valid.

Exception: If any input data is invalid program shows a relevant error message.

Post Condition: Users log in and are redirected to the main page.

Priority: High

Activity Diagram:

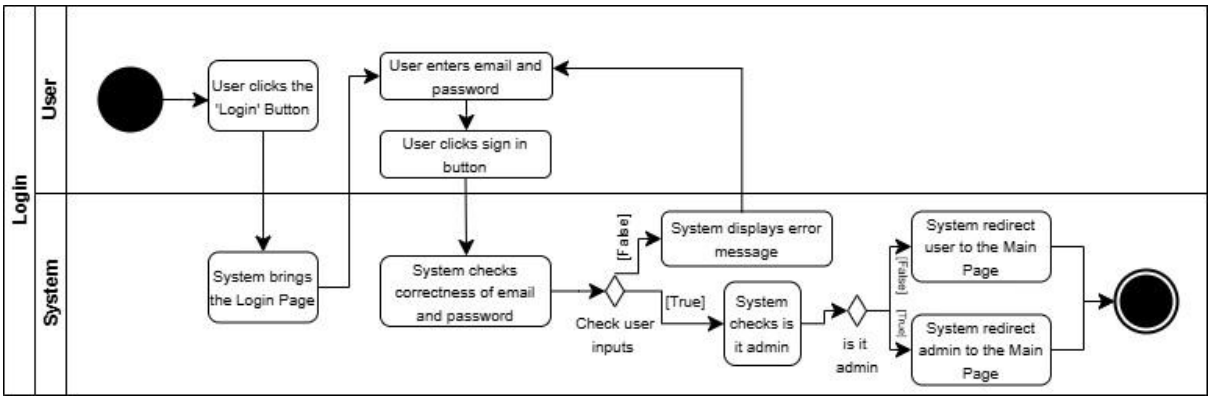

Figure 4. Activity Diagram of Login Page

2.2.3. Admin Page

Summary: The system is used by admin. Admin can perform operations such as update data, change data frequency, change data, change NLP method from this page.

Actor: Admin

Precondition: Administrator must run the program and log in as admin.

Basic Sequence:

1. Admin click to enter the Admin Page with the admin account.
2. After logging in, the admin selects one of the options: update data, change data frequency, change data, change NLP method.
3. After the admin makes the desired changes, admin approves the changes.
4. The system saves the changes made.

Exception: Problems may arise with changes made.

Post Condition: Changes made by admin will be saved.

Priority: Medium

Activity Diagram:

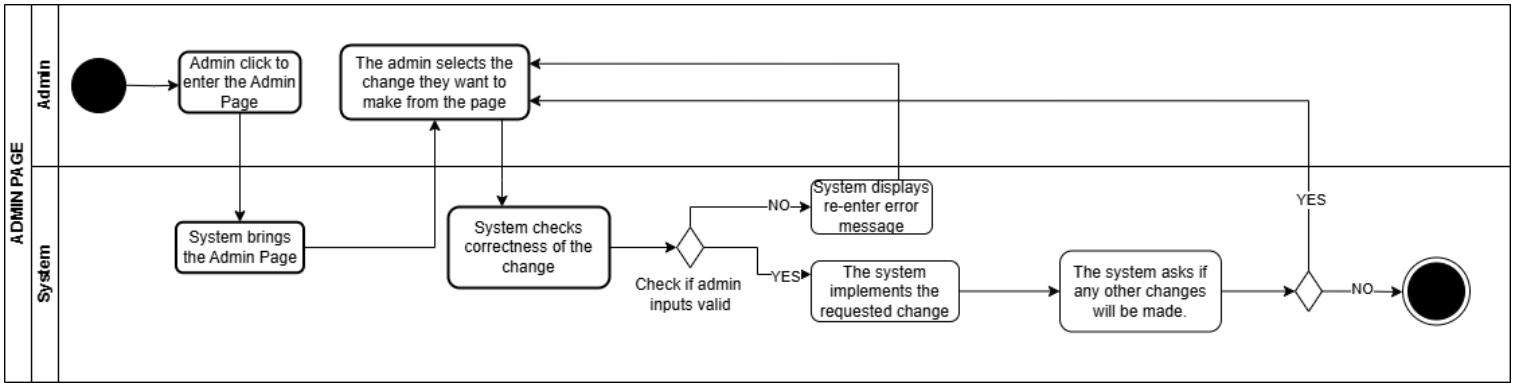

Figure 5. Activity Diagram of Admin Page

2.2.4. Compare Companies

Summary: This system is used by user and admin. User can compare stock of companies however he wants. Admin can change some features here such as supporting graphics.

Actor: User, Admin

Precondition: User must click the compare button to open the page.

Basic Sequence:

1. User must be on the Main Page.
2. User must click the button Compare.
3. User must choose the companies which he wants to compare their stocks.
4. Admin can do changes on the model, the data period which model takes as input.
5. User can exit from the system by selecting exit button.

Exception: Database connection, connection to the model can be failed due to API, internet or some backend failures.

Post Condition: None

Priority: Low

Activity Diagram:

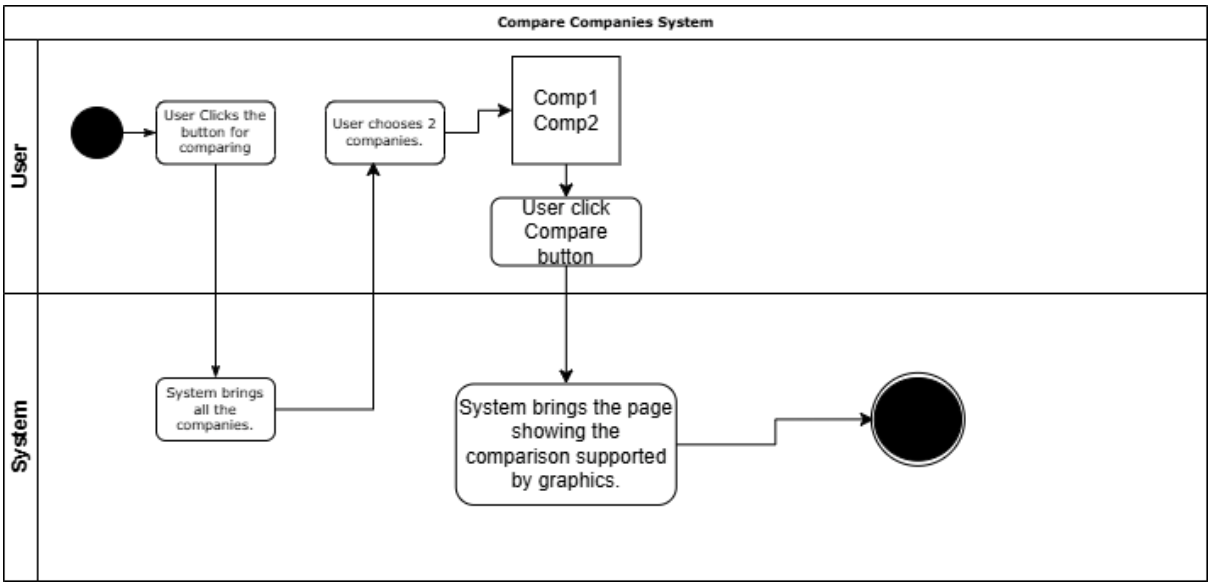

Figure 6. Activity Diagram of Compare Companies System

2.2.5. Main Page

Summary: Main functionalities of the system can be seen and accessed in this page. Users can view BIST100 Analysis, compare the analysis of two companies, add these companies to favourites and see the details of stock market shares.

Actors: User, Admin.

Precondition: User must login to the program.

Basic Sequence:

1. User must be logged on to the program in order to access the main page.

2. After login, system shows the choices the user can make.
3. User can see various functionalities and choose to view BIST100 Analysis, compare the analysis of two companies, add the desired companies to favourites or see the details of stock market shares.
4. After the choice of the user, the system directs the user to that page.
5. After the user completes his task, he can turn back to the main page.
6. User can choose another functionality or logout and close the program.

Exception: Network Problems and user session timeout.

Post Condition: None

Priority: High

Activity Diagram:

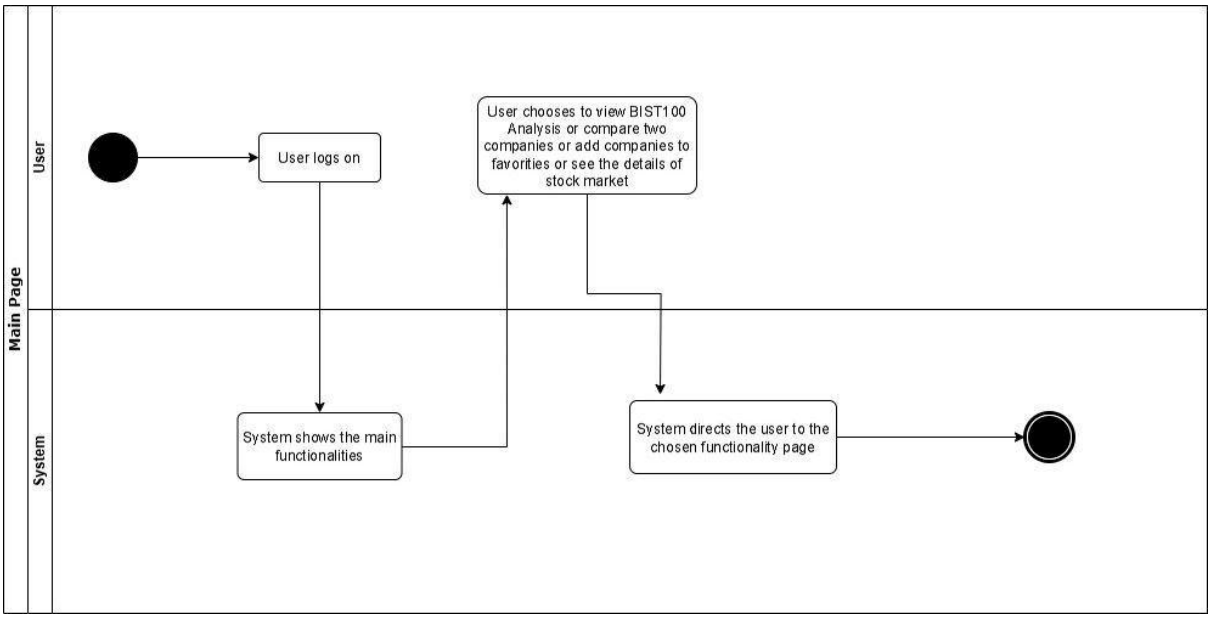

Figure 7. Activity Diagram of Main Page

3. USE CASE REALIZATIONS

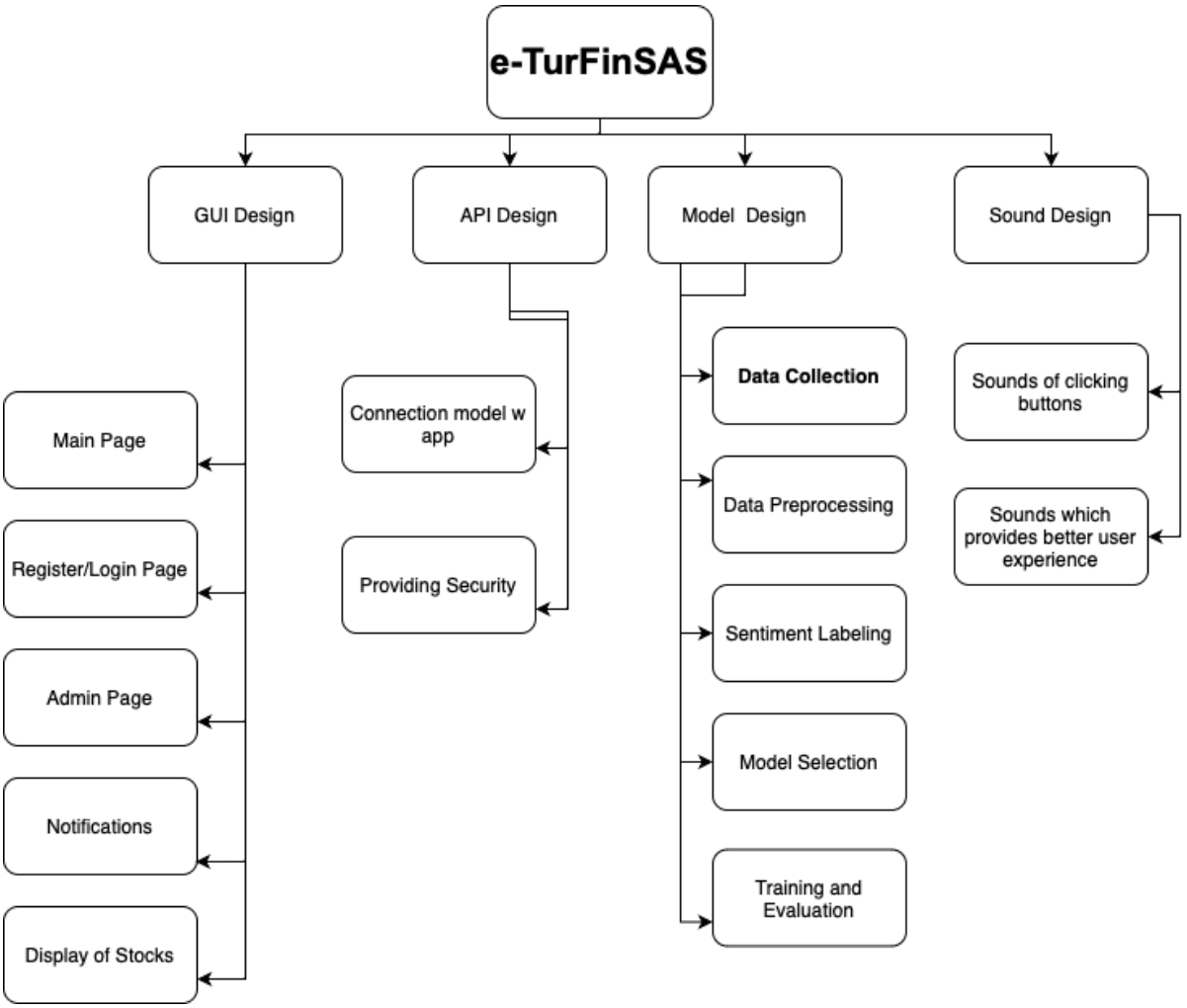

Figure 8. Project Components of e-TurFinSAS

3.1. Brief Description of the Project Components

3.1.1. GUI Design

GUI design is responsible for interaction between the actors and the system. There are 5 sub-systems in this design which are Main, Register/Login, Admin pages, Notification and display of stocks. Main Page is the actor can reach all activities s/he wants. Register/Login is where s/he can sign up or sign in. Admin Page is where the admin, which is determined by us, can do some important changes on the model, application and more. Notifications are for letting the user know about stocks, or application. Display of Stocks is the page where the user can view the analysis of the model of stocks which s/he wants.

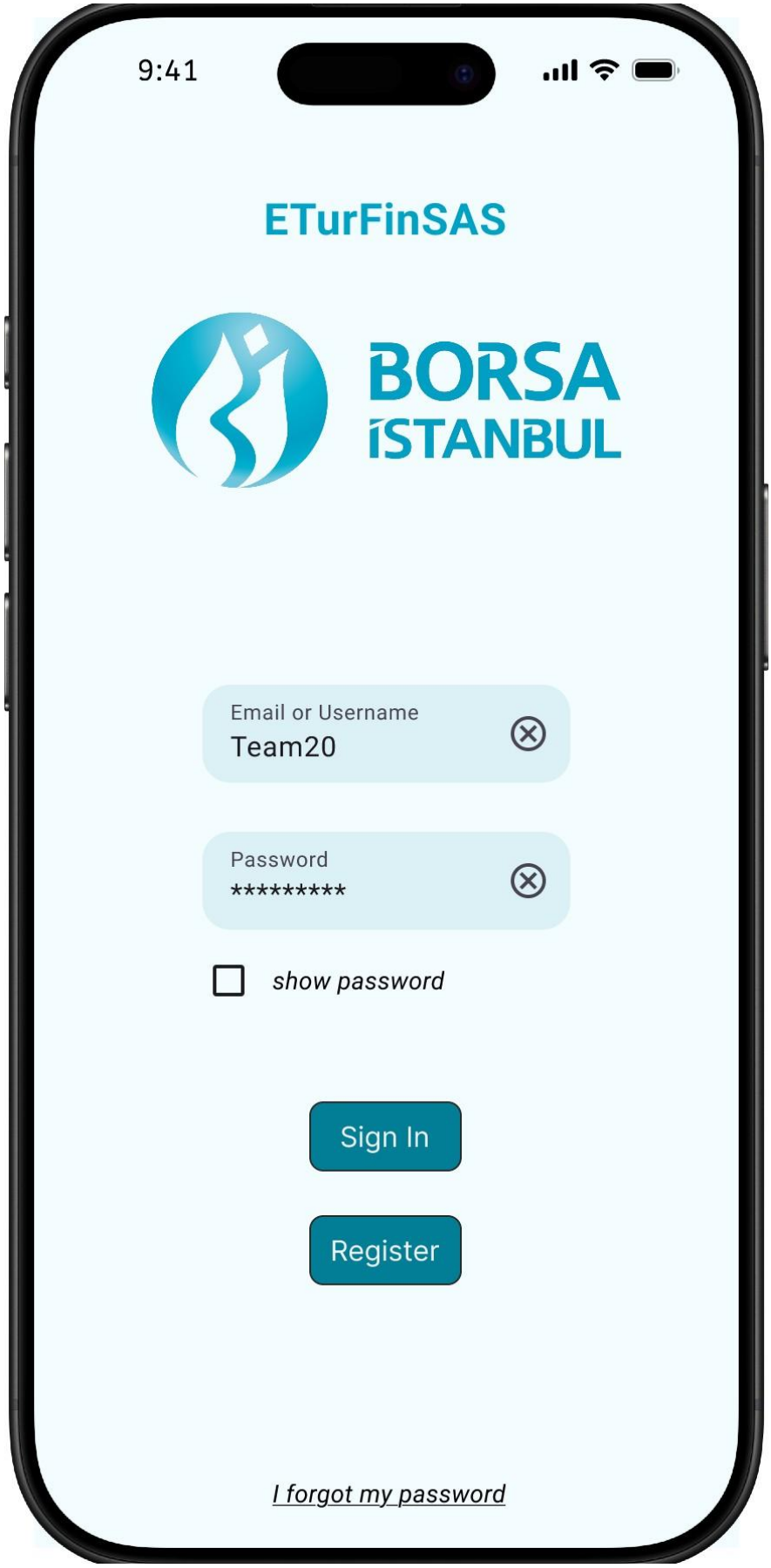

Figure 9. Prototype UI Design of Login Page

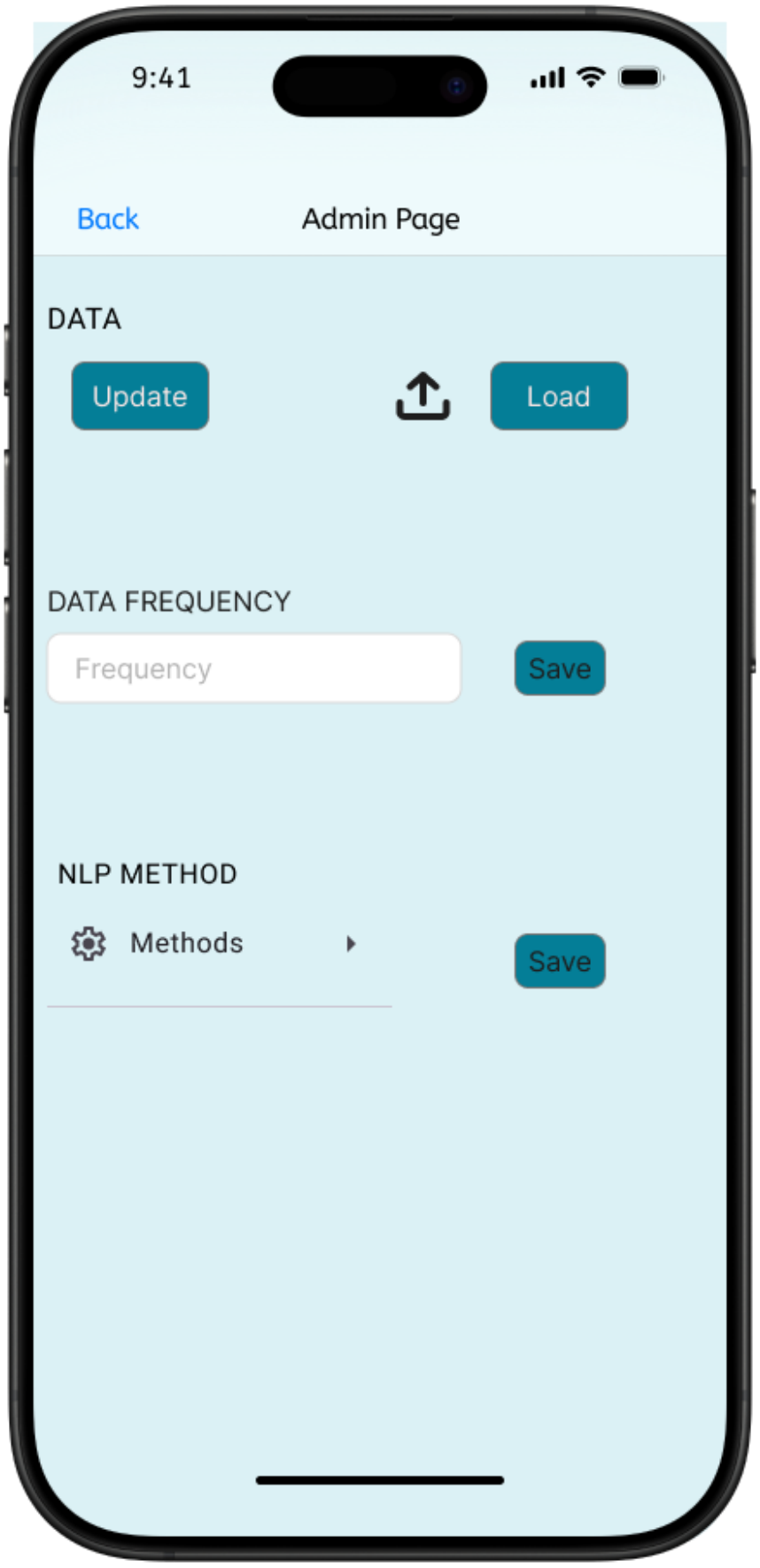

Figure 10. Prototype UI Design of Admin Page

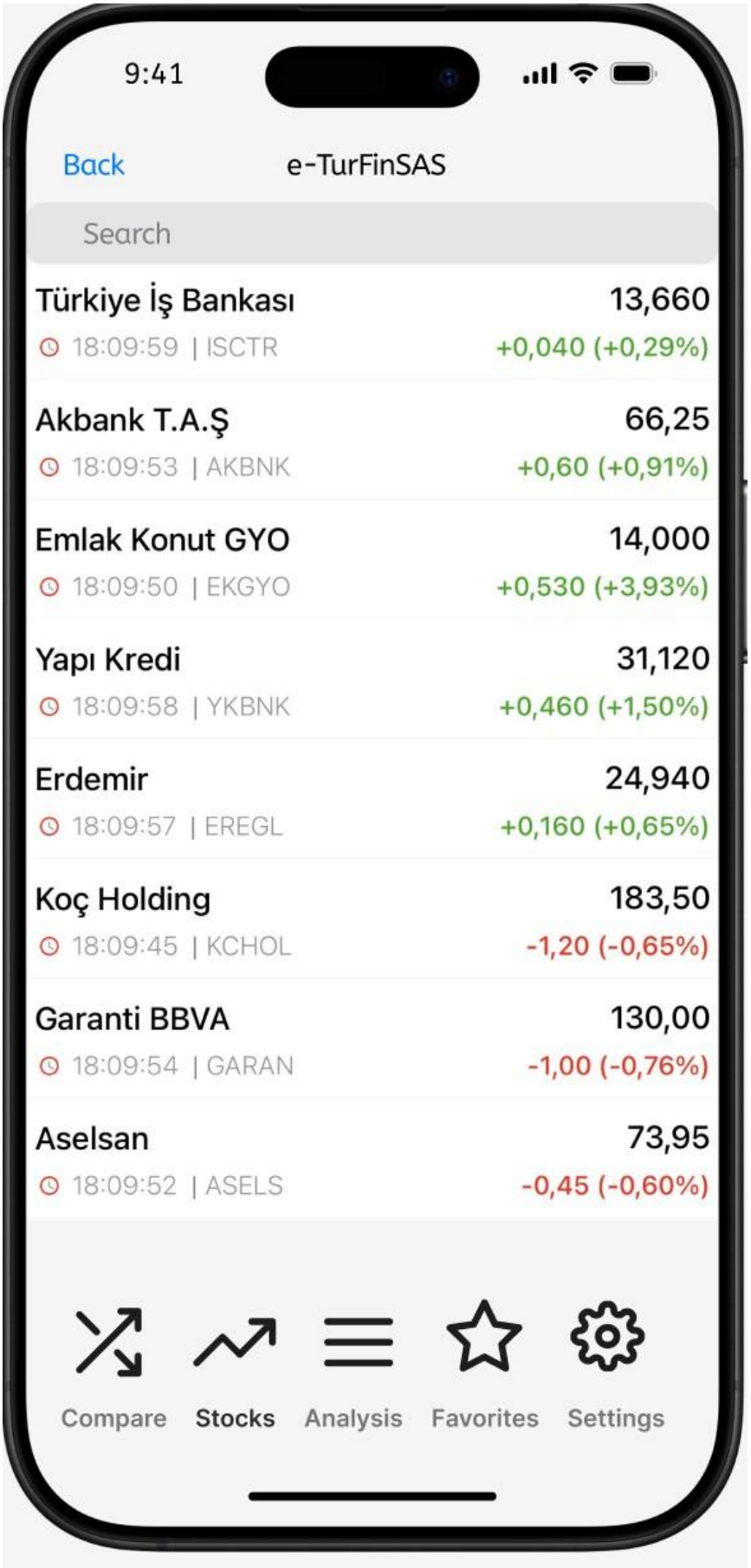

Figure 11. Prototype UI Design of Main Page

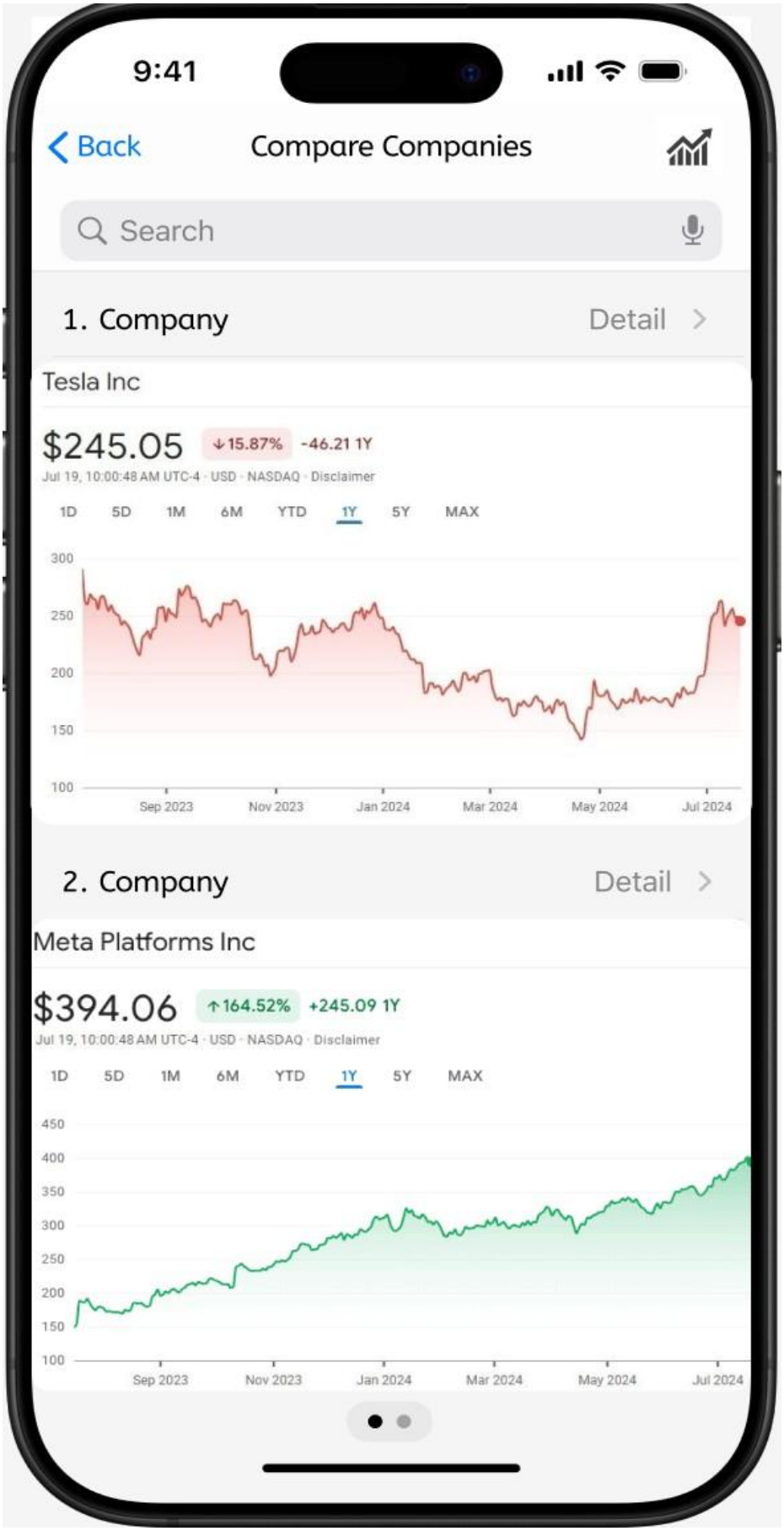

Figure 12. Prototype UI Design of Compare Companies Page

3.1.2. API Design

The API Design module is responsible for facilitating communication between the application and the backend servers where the sentiment analysis is performed. This system ensures efficient data transfer and accurate responses without burdening the user's device.

- **Request Module**: Accepts user input, such as the stock name and related text data, and forwards it to the backend for analysis.
- **Response Module**: Retrieves sentiment analysis results from the server, including sentiment classification (positive, negative, or neutral) and confidence scores, and delivers them to the application.

3.1.3. Model Design

The Model Design module is responsible for processing text data and performing sentiment analysis using advanced machine learning models. The system operates entirely on the backend, leveraging pre-trained transformer-based models for high accuracy.

3.1.4. Sound Design

The Sound Design module is responsible for enhancing the user experience through audio feedback, providing cues for various system interactions and processes.

4. REFERENCES

- For drawing Agile Diagram (Figure 1.): https://wepik.com/
- For drawing Use Case Diagrams (Figure 2-8.): https://draw.io/
- For drawing Prototype User Interfaces (Figure 9-12.): https://www.figma.com/
- For more information about Agile Methodology: https://www.geeksforgeeks.org/what-is-agile-methodology/
- For more information about Software Development Phases: https://www.harness.io/blog/software-development-life-cycle-phases
- For more information about UML Class Diagrams: https://www.lucidchart.com/pages/uml-class-diagram
- For more information about Activity Diagrams: https://www.smartdraw.com/activity-diagram/