Bilkent University

Department of Computer Engineering

Senior Design Project

Project Name: Stock Vision

Project Specifications Report

Remzi Tepe 21802713

Ekrem Polat 21802228

Abdulkadir Erol 21703049

Mert Atakan Onrat 21802520

Nihat Bartu Serttaş 21702301

Supervisor: Shervin Rahimzadeh Arashloo

Innovation Expert: Cem Çimenbiçer

http://rtepe.github.io/

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1. Introduction

The cryptocurrency market has evolved significantly over the last decade. As of 2022, the market cap has reached over $1.00 trillion, and over 300+ million people use/own cryptocurrencies [1]. This huge demand led to much research regarding the stock market for users to learn about it. Even so, reaching for trustworthy information about the market and analyzing the stock market have always been struggles for users. Many so-called economists have emerged and tried to direct people with some controversial or even incorrect concepts about the market. It takes a lot of time for beginners to choose which cryptocurrencies to use and how to invest while there are limitless ideas. Moreover, many cryptocurrencies arise and fail due to a lack of security, weak teams, scamming, etc.. This thriving industry has become very risky, especially for beginners and the ones who do not know what and how to invest their money.

Therefore, in the project, we aim to design a hands-on experience simulation app where users can invest the application's fake currency (given in any real currency such as Bitcoin or even dollars) and get AI-based results and tips in real-time. This way, users can learn the fundamentals of investing in cryptocurrencies without losing money or relying on anyone.

In this specifications report, we provide a description of our project, the constraints, the professional and ethical issues, and the functional and non-functional requirements of our project Stock Vision.

1.1 Description

Stock vision is a mobile application that works in iOS and Android. It provides a real-time stock/ETF simulation using past stock/ETF data and allows users to make investments using our application’s fake currency and get real-time results/tips about their investments. At first, the user will sign up, if not signed yet, and then log in to the system. After logging in, users can see and search real-time stock/ETF values. Stock market graphs of each cryptocurrency can be seen daily, weekly, monthly, or even yearly. Users can go through

these cryptocurrencies and see the data provided and read the tips given by our AI algorithms. More importantly, each user will start with a specific amount of fake currency and be able to use this fake currency to invest in stocks/ETFs. When users make investments, they can see tips from our AI on how to invest their money, the probability of values increase/decrease, and receive real-time results when gaining or losing money. With these tips, users can practice their graph reading abilities and see the results in real time. If the users run out of the application's fake currency, they can start over with the initial fake currency.

Furthermore, Stock Vision has graph reading algorithms using reliable stock graph patterns and providing a tipping mechanism for the graphs in real time. Analyzing stock values will be done with image processing, and a wide range of datasets of stock graph patterns will be used. Moreover, users' investments will be stored in our database, and the analysis of the user data will be done. Some tips and results will be provided after these analyses.

1.2 Constraints

1.2.1 Implementation Constraints

- The front-end software will be written in Dart/flutter [2] to make the application available for IOs and Android.
- A server will be used for authentication and data storage about the investments.
- GitHub will be used to collaborate while implementing.
- AI training will be done in our back-end application, and no calculations will be done on client-side devices.
- The back-end software will be written in Python.
- Object Oriented Programming will be used.

1.2.2 Economic Constraints

- The application will use the freemium package as a business model, a basic product or service that is provided free of charge, but money (a premium) is charged for additional features and services that expand the functionality of the free version of the software [3].
- Our application requires a server system like Firebase, which will be free for our application.
- Our website is on GitHub Pages, which is free to use.

1.1.3 Sustainability Constraints

- Both front-end and back-end applications can be updated in order to add new features according to user feedback.

1.1.4 Technical Constraints

- The front-end application will use limited internet when the client-side device is not connected to a Wi-Fi connection.
- The front-End application should not have an internet usage limit when the client-side device is connected to a Wi-Fi connection.
- The front-end application will be efficient in CPU and memory usage in order to save battery life on client devices.
- The back-end application should not have CPU and memory limitations for continuously training the AI and keeping the login/register services running.

1.1.5 Language Constraints

- English is the global standard, and thus the default language of both front and back-end applications will be English.
- Other languages for the front-end application may be added later.

1.1.6 Ethical Constraints

- Since users can create an account with usernames and passwords, they will be securely stored in our back-end application.
- Since users can decide their usernames, inappropriate language use should be detected and removed on the front-end application.

1.1.7 Legal Constraints

- Since our application will use user information and all their investment movements, our AI will analyze all the data. The data should be securely contained and not shared with anyone. However, later the data might be used for a profit model later on with the permission of the users.

1.3 Professional and Ethical Issues

1.3.1 Professional Issues

- During every project step, the workload will be shared equally among all group members.
- Each team member will work professionally.
- Online communication tools such as Discord and Microsoft Teams will be used for group communication.

1.3.2 Ethical Issues

- The application will have an accounting system. Therefore, the user's information, such as email address, password, username, etc., will be protected and secured.
- Since the personal data of the users will be kept, the terms and conditions of service must be provided. Users will be informed about the data they share and by whom they will be used.

- Since our AI's sole purpose is to give tips for users on how to invest, it should be known to the users and told beforehand that the AI also has a mistake percentage and thus should not be taken as real investment tips. If the users decide to follow tips from AI and invest real money outside of our application according to our AI tips and end up losing money, we are not to be held accountable. User discretion is advised.

2. Requirements

2.1 Functional Requirements

- Users must log in or register to the system before using the application.
- The system should define some amount of fake coins to users for investing in the market..
- The system must give advice to users by evaluating users' investments/transactions with AI.
- The system must give tips to users by evaluating the previous data about the currencies.
- The system must have two types of accounts: Free and Premium. Premium users can get more detailed tips and fake coins to invest in the market.
- The system must show users extended graphs related to predictions on currencies.
- The system must save all information about the currencies' progress, predictions, and graphs in the database so other users can benefit from it.
- Users can save their favorite currency pairs to see their progress.
- Users can change/delete their profile and related data in the database.

2.2 Non-Functional Requirements

2.2.1 Accessibility

- Android Jelly Bean, v16, 4.1, and iOS 8 or newer versions are required in order for users to use the system.

2.2.2 Availability

- The application will use past and present cryptocurrency data from reliable stock market sites. Unless these websites are down (it is very rare), the application can be used properly.

2.2.3 Performance

- Displaying the homepage to users should be under 5 seconds.
- Updating users' data on investments should take under 1 second.

2.2.4 Reliability

- Authentication is required in order to join the system.
- A server crash or power outage should not result in data loss.

2.2.5 Scalability

- The servers can be extended easily.
- Many other features can be added easily without losing performance.

2.2.6 Security

- A unique sign-up/log-in process will be done.
- The data will be put in reliable servers like Firebase, so data loss will not be encountered.

2.2.7 Usability

- The GUI of the application will be very user-friendly since the users are not expected to be an expert in using mobile applications.
- Any problems encountered by users can be reported so that sustainability can be managed and the bugs can be fixed.

2.2.8 Portability

- If the newer Android or IOS is installed, the system should not cause any errors.

3. References

[1]: “63+ cryptocurrency statistics, Facts & Trends (2022),” *(2022)*. [Online]. Available: https://buybitcoinworldwide.com/cryptocurrency-statistics/. [Accessed: 17-Oct-2022].

[2]: “Build apps for any screen,” *Flutter*. [Online]. Available: https://flutter.dev/. [Accessed: 17-Oct-2022].

[3]: J. L. Marín de la Iglesia and J. E. Labra Gayo, “Doing business by selling free services,” *Web 2.0*, pp. 1–14, 2008.