School of Electrical Engineering, Belgrade
Principles of Software Engineering (SI3PSI)

GoldenView

Financial trading web application

Version 1.0

Team Terminal

Contents

History of changes

Version no.	Date of release	Description	Author
1.0	12.3.2023.	Initial version	Andrej Dujović, Aleksandar Radenković

1. Introduction

1.1 Summary

The "GoldenView" project is a part of the practical teaching in the course Principles of Software Engineering. The project is a web application for trading on the stock market. Users with their created account will be able to monitor and analyze stock prices in real time, as well as access other information that is important for trading.

1.2 Purpose of the document, and target group

The principle aim of this document is to present and explain the functionalities of the application, distinct user roles within the system, employed technologies, system limitations and future development of the application.

2. Description of the problem

Having reviewed the current stock exchange programs/applications we have identified a significant issue common to all of them - non user-friendly interface which results in a steep learning curve for beginners. The “GoldenView” would resolve this issue with its intuitive interface while preserving some of the most significant features in existing programs. This would enable users with little to no prior experience to enter the world of finance with ease.

3. Categories of users

The system provides different functionality to users according to their user role. There are four user roles: Guest, Basic user, Broker, and Administrator.

3.1 Guests

Guests are non-registered or not logged-in users that can only view the basic information on the home page such as the user manual, features previews, comparison between "GoldenView" and other already existing solutions.

3.2 Basic users

Basic users are users that made an account. After the registration they are presented with the main feature of our web application and that is the trading view. Users are then provided with real-time financial market data, news that can help them make informed trading decisions and options like buying/selling their stocks.

3.3 Brokers

Brokers are basic users that request to be brokers and are approved by Administrators as official brokers. Once their request is approved, besides the basic user features they also unlock the possibility of helping other basic users with their stock portfolio. They can buy and sell other users’ stocks, in exchange for a fee that they and basic users agreed upon.

3.4 Administrators

Administrators are registered users that have all the options to administrate the system. Some of their unique features are: Deleting user accounts, changing user profiles, administration of the system, updating the information within the webapp.

4. Product description

In this section of the document, we describe the main features and the architecture of our product.

4.1 Architecture of the system

Our web application consists of three parts: the client-side (the website), the server-side (communication between the website and the database), and the database for storing information.

Technologies that will be used for the client-side are: HTML, CSS, and JavaScript, AJAX. They will be used to build a user-friendly interface that anyone can easily understand. Such intuitive and easy to navigate interface represents the key feature to attract the users.

On the server-side we will be using Django.

There will also be a MySQL database that will contain account information for all the users, their login credentials, their portfolios etc.

4.2 Features overview

Benefit for the user	The feature that provides it
Access from any device	Since this is a web application, users can access it from any device, independent of the platform. The only requirement is that they have access to Internet
Best user experience for trading stocks	Our goal is to have the best user experience so that even novice users can understand how to trade stocks
Monitor and analyze asset values in real time	With the trading view page users can track and analyze changes in asset value in real time, this feature makes it easier to buy/sell assets at their current price
Assign brokers to users' portfolios	Brokers can make trades for basic users and charge a fee
Lates tips and news about stock market	A page with all the news and advice on what is happening on the stock market

5. Functional requirements

In this section of the document, we describe the main functional requirements that our product should have implemented.

5.1 User registration and log in

Our web application has a basic registration system, with signing up and logging into an account. Registration will compose of basic information that will be required to fill in (email, password etc.)

5.2 Password changing

Users can request to change their account password.

5.3 Requesting to become a broker

A registered basic user can request to become a Broker. When he sends the request through a dedicated page for that, he will then need to wait for the Administrator to accept or decline the request (based on the conclusion if that basic user is good enough to become a Broker). Administrator has an option to remove the status of a Broker from the user if needed.

5.4 Administration of the system

Administrators have access to modify everything about the system. That includes accepting/denying user requests to become a Broker, deleting user accounts, changing web application information etc.

5.5 Real time assets view

Every registered user can access the prices view that includes choosing the asset that will be shown, graph of changes in value over time and other basic information regarding that asset. Additionally, graphs could display GoldenView's predictions of future trends.

5.6 Buying and selling assets

Registered users can buy and sell assets, that will change their portfolio page where all their assets are listed. User can monitor their portfolio at any moment in any page via a convenient sidebar.

5.7 Brokers-Basic User contracts

Brokers are able to send a request with the entered fee that the Basic user will pay, in order to help the user manage and improve their portfolio.

5.8 Comparing companies

A streamlined webpage equipped with the capability to compare two user-selected companies, using a range of collected parameters, and then provide the user with a clear indication of the more favorable option.

6. Assumptions and restrictions

As this is a web application for real-time trading on the stock market, all users are advised rethink all their purchases since changes to their portfolios are irreversible. Prices of assets change at rapid rates and we can’t guarantee identical rates between the actual purchase and its preview.

Sensitive information and users' authorization credentials are stored permanently in the database and any kind of security breach could have fatal consequences such as overtaking users' access to financial assets purchase functionalities. Therefore, it is necessary for the users to have an option of updating their password.

7. Quality

All the afore mentioned critical functionalities should be thoroughly tested using unit and integration testing techniques. Since the system has a SQL database, the system should be tested for SQL injection attacks. The stock view should be fast, reliable and user friendly since our top priority is for users to have a great experience and profit from stocks.

8. Non-functional requirements

8.1 System requirements

Server must support Python and Django as well as have a fast internet connection that the quality of real-time transactions depends on. The frontend should be accessible on the most web browsers. It's crucial to ensure that the web pages' layouts remain consistent to the design, regardless of the user's browser.

8.2 Other requirements

The system should be capable of delivering prompt and accurate real-time pricing information, while also providing a visually dynamic user interface.

9. User documentation requirements

9.1 Instructions for using the web application

These instructions would be needed for Brokers, they would contain information such as: how to request authorization from basic users to trade on their behalf, how to set up transaction fees, use case examples and tiny guides for novice users, etc.

9.2 Labeling

Every page should display the GoldenView logo. All stock information is followed by a timestamp representing when such information was fetched.

10. Plan and priorities

The following is a prioritized list of features, in order of significance from highest to lowest.

1) Registration and log in of users
2) Trading view, charts, and real time updates
3) Buying/selling assets
4) Broker-Basic User contract system
5) Administrator privileges

Future versions will require the development of Android and iOS versions to enhance performance, and security by introducing 2 step verification before each purchase. Additionally, real payment options for purchasing should be integrated.