**School of Electrical Engineering, Belgrade**

**Principles of Software Engineering (SI3PSI)**



GoldenView

Financial trading web application

Version 1.0

Team Terminal

Contents

History of changes 3

1. Introduction 3

1.1 Summary 3

1.2 Purpose of the document, and target group 4

2. Description of the problem 4

3. Categories of users 4

3.1 Guests 5

3.2 Basic users 5

3.3 Brokers 5

3.4 Administrators 5

4. Product description 5

4.1 Architecture of the system 6

4.2 Features overview 6

5. Functional requirements 7

5.1 User registration and log in 7

5.2 Requesting to become a broker 7

5.3 Administration of the system 8

5.4 Real time stock prices view 8

5.5 Buying and selling stocks 8

6. Assumptions and restrictions 8

7. Quality 9

8. Non-functional requirements 9

8.1 System requirements 9

8.2 Other requirements 9

9. User documentation requirements 9

9.1 Instructions for using the web application 9

9.2 Labeling 9

10. Plan and priorities 10

# History of changes

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

# Introduction

## 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.

## 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.

# 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.

# 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.

# Product description

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

## 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.

## 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 from 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 stock prices in real time | With the trading view page users can watch and analyze stock price changes in real time, this feature makes it easier to buy/sell stocks at their current price |
| Get advice about your portfolio | Brokers can help you with buying or selling stocks and in that way improve your portfolio |
| Lates tips and news about stock market | A page with all the news and advice on what is happening on the stock market |

# Functional requirements

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

## User registration and log in

Our web application has a basic registration system, if the user didn’t already register, and the log in if he already has an account. Registration will compose of basic information that will be required to fill in (email, password etc.)

## Requesting to become a broker

A registered basic user can request to become a Broker. When he sends the request through a specific page made 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.

## 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.

## Real time stock prices view

Every registered user can access the stock prices view that includes choosing the stock that will be shown, graph of changes in price of that stock and other basic information about that stock.

## Buying and selling stocks

Registered users can buy and sell stocks, that will change their portfolio page where all their stocks are listed. This feature is integrated on the same page as stock prices view, just a little right of it so that users can easily monitor, analyze, and decide when to buy or sell stocks.

//////////// ADD MORE STUFF IF NEEDED ///////////

# Assumptions and restrictions

Since this is a web application for trading on the stock market, all users are advised to think before doing anything since the changes to their portfolio can’t be undone. Prices of stocks are changing so we can’t guarantee that the user will be able to sell or buy stocks at the same time even if the time between transactions is very little.

All the user information is kept in the database so the security of the web application and the database should be a priority. Breach of any account could be a big problem since accounts are tied to the bought stocks.

# Quality

All of the already mentioned functions should be tested using black box methods. 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.

# Non-functional requirements

## System requirements

The front-end should be accessible on the majority of web browsers. It's crucial to ensure that the display of the pages remains faithful to the design, regardless of the user interface employed, such as Mozilla Firefox, Internet Explorer, and others.

## Other requirements

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

# User documentation requirements

## Instructions for using the web application

These instructions would be needed for Brokers, they would contain information such as: how to initiate a request to help other users, how to choose a fee for that help, how to change other users’ portfolio etc.

## Labeling

Every page should display the GoldenView logo.

# Plan and priorities

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

1. Registration and log in of users
2. Trading view, charts and real time view of price changes
3. Buying/selling feature for users
4. Broker system for helping Basic users
5. Administrator functions

Future versions will require the development of Android and iOS versions to enhance performance. Additionally, real payment options for purchasing should be integrated.