A Report On

Live Stockmarket Dashboard Application

Submitted by

Farris Suhale 16IT218

Vaishnavi Mishra 16IT145

in partial fulfillment for the award of the degree

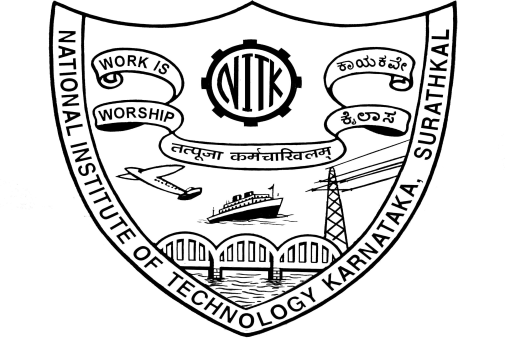
of

Bachelor of Technology

In

Information Technology

At



Department of Information Technology

National Institute of Technology Karnataka, Surathkal.

April 2019

Abstract

Our Live Stock Market dashboard shows real time stock market trends and provides pretty looking, sleek and lightweight graphical components that consist of filters allowing users to query for all sorts of trends in order to analyse the stock market. View Various Stocks in Real Time, Stock price history and trends, Stock News in real time and login to save your history. This application is aimed at amateur finance hobbyists as well as financial and stock experts.

Contents

1. Abstract…………………………………………………………………………........................1

2. Introduction……………………………………………………………………………….…….3

3. SRS …………….………………………………………………………………………………4

4. Manual Testing………………………………………………………………………………….6

5. Implementation…………………………………………………………………………………7

6. Results and Discussion…………………………………………………………………………9

7. Conclusion…………………………………………………………………………………….10

8. References……………………………………………………………………………………..11

I INTRODUCTION

Stocks and the stock market will never go out of style. But not everyone can keep track of and understand them easily. Our web application, through visual analysis of stock price history and by real-time (refreshable at a chosen interval) data on the stocks aids in the understanding of the same by amateur users and finance experts alike.

Stock Market Dashboard is a website with real time stock prices, graphical analysis, stock history of selected stocks and live news updates related to stocks and general things. Login capabilities enable personalization and saving of search history.

The purpose of this application is to present a detailed analysis of real time stocks.

This software system is a Web Application which is created for users to check real time stock prices along with stock history of selected stocks, live news updates related to stocks. This helps in better understand of the stock market and understanding what to invest in when. Visual aids simplify learning and improve interpretability.

The product originated from our interest in the stock market but lack of understanding of it. Using financial experts and graphical displays stock market trends can be analyzed more easily. This is our replacement for existing stock market dashboards as real-time news updates enable better up-to-date decisions.

This helps in better understand of the stock market and understanding what to invest in when.

The application aims to:

1. Populate local database with real time data fetched from APIs.
2. Fetch and send data to graphical components at regular time intervals.
3. Use relationships to generate interesting insights about the trend in stocks.

II Software Requirements Specification

# Introduction

## Purpose

Stock Market Dashboard is a website with real time stock prices, stock history of selected stocks, live news updates related to stocks and chatroom facilities for discussing with experts. Login capabilities enable personalization. The purpose of this document is to present a detailed description of the E-commerce web application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

Release 1, revision 1.

## Document Conventions

* The document is prepared in standard IEEE template using Google Docs .
* All the headings in the document are in 18 point Times New Roman with single spacing.
* The sub-headings are 14 point Times New Roman.
* The font of the corresponding body is 11 point Arial.

## Intended Audience and Reading Suggestions

The Software Requirements Specification document provided in intended for a plethora of

readers. These include and not limited to,Developers,Project Managers,Database

Managers,Marketing Staff,Quality Assurance testers,Interface Designers,Documentation Writers

and the Users.

The intended order of reading to be followed is to start from the beginning,followed by the

particular sections intended for each of the different types of readers.

For example,it is imperative for a Developer or Quality Assurance tester to read all the

sections:Introduction,Overall Description,External Interface Requirements,System

Features,Non-functional Requirements and Other Requirements along with the glossary,as they

need to understand the project thoroughly to implement it further or test the various features that

the project offers.

However,a User,Marketing Employee or Manager who will not directly be affecting the

development of the project need not have a technical understanding of the project.They will

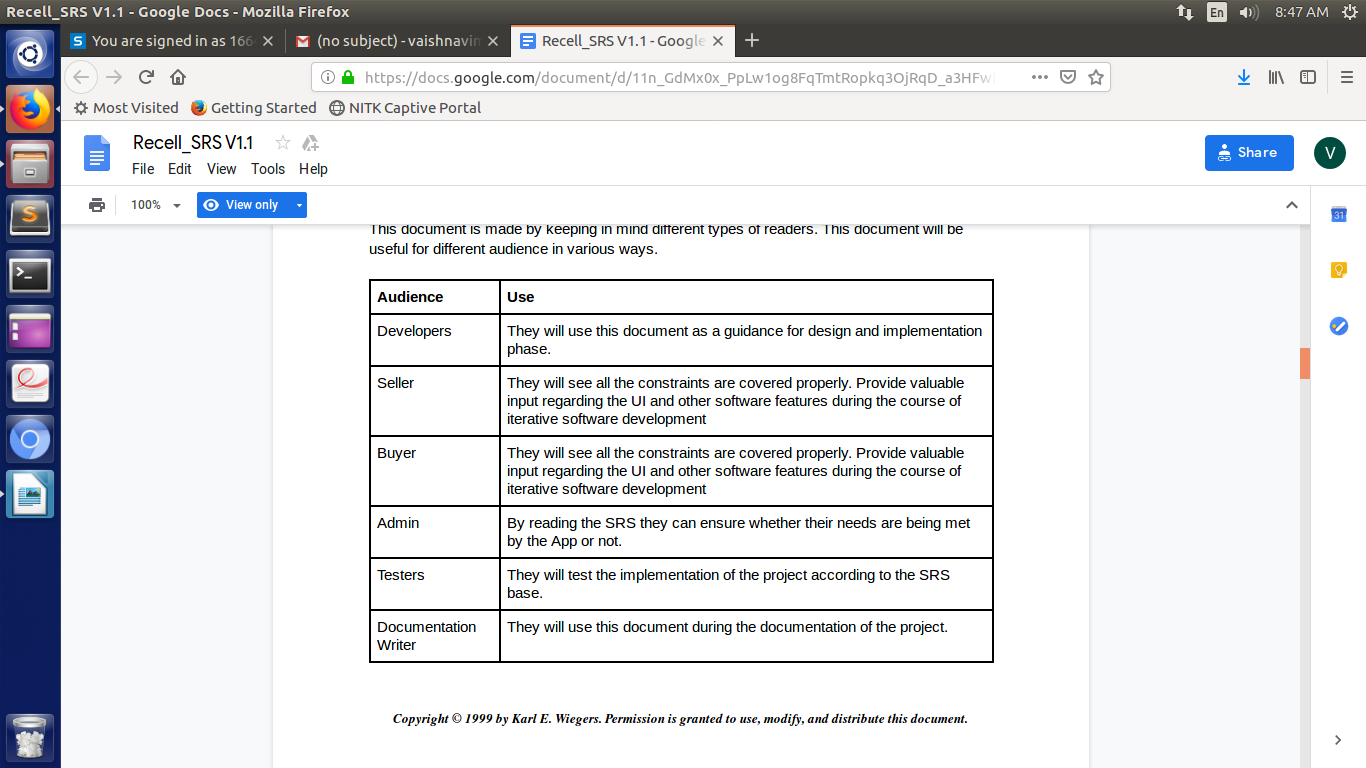
however be required to read the sections of:Introduction,Overall Description and System

Features to gain a good understanding of the workings of the project.

A Designer would have to understand the project in order to create the design language and

features in the User Interface,therefore the sections pertinent to him/her are:Introduction,Overall

Description,External Interface Requirements and System Features.



## Product Scope

This software system will be a Web Application which is created for users to check real time stock prices along with stock history of selected stocks, live news updates related to stocks and chatroom facilities. This helps in better understand of the stock market and understanding what to invest in when.

## References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements

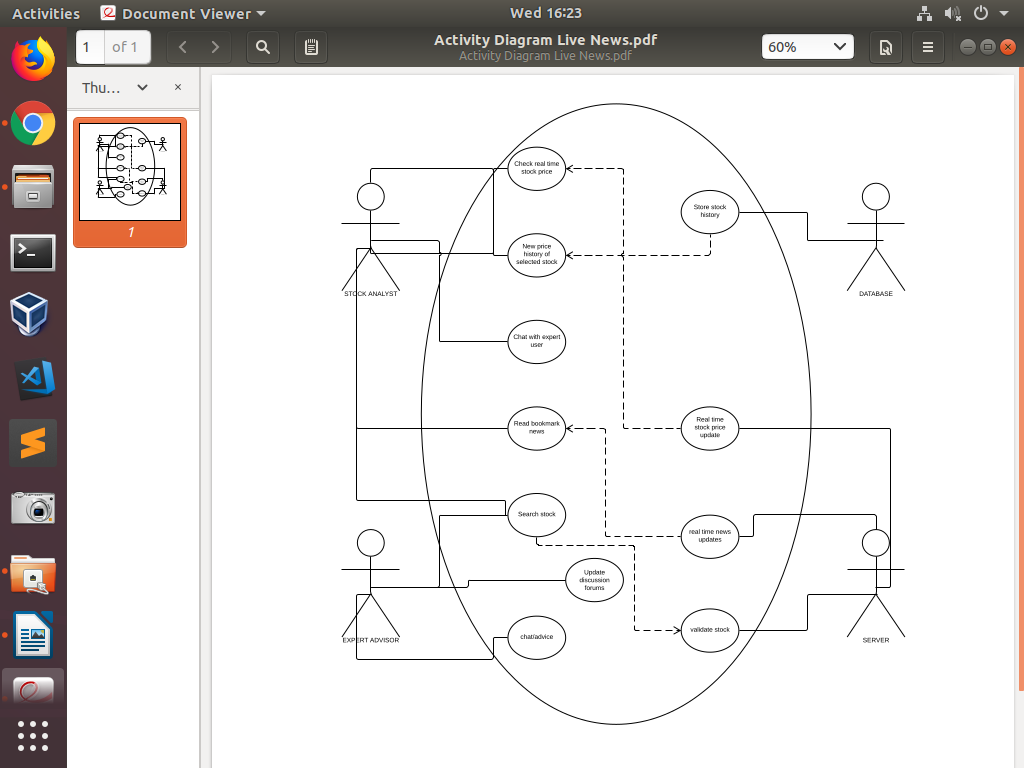
Specifications. IEEE Computer Society, 1998.

# Overall Description

## Product Perspective

The product originated from our interest in the stock market but lack of understanding of it. Using financial experts and graphical displays stock market trends can be analyzed more easily. This is our replacement for existing stock market dashboards as real-time news updates enable better up-to-date decisions.

This software system will be a Web Application which is created for users to check real time stock prices along with stock history of selected stocks, live news updates related to stocks and chatroom facilities. This helps in better understand of the stock market and understanding what to invest in when.



## Product Functions

Functions of our software include:

1. View live stock prices

2. View stock price history and trends

3. View real time news updates related to stocks

4. Chat with experts or other users

5. Login and personalize- bookmark stocks

## User Classes and Characteristics

User classes that would use this software:

1. Stock market analysts – login functions to save favourite stocks.

2. Financial experts – provide advise

3. Novice user requiring real time stock market updates- will require pictorial graphs for better understanding of financial trends. Require advise of financial experts

The most important user class of these is the Stock market analyst.

## Operating Environment

Regarding Hardware and Software environments,the application has been optimized to run on

all browsers. It has also been optimized to run on all devices,including Personal

Computers,Laptops,Tablets and Mobile Phones.

The only other software required to run the application is any Web Browser(Mozilla

Firefox,Google Chrome,Apple Safari,Opera etc).

The application is not resource- or graphics-intensive, so there are no practical hardware constraints. The application will rely on Django for the backend and Bootstrap for the frontend.

## Design and Implementation Constraints

Our web application, being easy to use and requiring no programming, designing or financial skills, is usable by all users.

No items or issues will limit the options available to the developers. These include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards.

Simply opening up the dashboard will get you started.

## User Documentation

SRS will be delivered along with the website.

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

We assume that the user is capable physically and mentally of accessing the internet and therefore our application.

# External Interface Requirements

## User Interfaces

The website is user friendly, easily navigable by novice users and intuitive. The colour schemes, positioning of text, images, news updates etc are all aesthetic, understandable and follows logical sequencing or positioning. Standard buttons and functions (e.g., help) will appear on every screen, keyboard shortcuts, error message display standards, and so on and any user familiarizing with this website will understand it easily and adapt to using it.

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

Our website deals with front end and database and therefore hardware interfaces are irrelevant here.

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

Software interfaces used:

* We propose to use some API for fetching the live stock price information.
* We will use an API for obtaining real time news updates
* We will also use APIs in chatroom facilities
* For designing Web App we will be using Django and as for mobile,the web app can be accessed using any web browser commonly available.
* Website Development will require framework like Django.
* Most of the softwares used will be Open Source.
* Communication between server and Database is done using MySQL

## Communications Interfaces

* Chatroom facilities require a communication-based API.
* APIs will inherently use various communication protocols like HTTP or database might employ FTP
* Communicating with the database of the server requires MySQL

# System Features

The multitude of usages our product provides are evident in its manyfold features which are listed below.

## Live Stock Prices

4.1.1 Description and Priority

This high priority feature is the focus of our website. Real-time stock prices are fetched by the server using an API and returned to the user at a time interval that is selectable by the user. These stock prices are constantly updated even without prompting from the user. The time period of updation is also selectabgle by the user.

4.1.2 Stimulus/Response Sequences

Opening up of the main page I.e the dashboard on the web browser will display a list of live stock prices to the side. This can either be viewed right there or can be enlarged to a full screen. In response to this, all live stock prices will show up on the screen and the user is free to scroll and browse.

4.1.3 Functional Requirements

* This feature uses a predefined API in order to fetch the live stock prices. The API in recurringly accessed in order to keep stock prices up-to-date.
* On the user end, only a web browser is needed and a good wifi connection.

## Stock Search and History

4.2.1 Description and Priority

This high priority feature is another focus of our website. By using the search bar to search for a particular stock of user interest, real-time stock prices are fetched by the server using an API and returned along with a stock history for a selectable time period. Stock history for selected stocks is stored in the database.

4.2.2 Stimulus/Response Sequences

Opening up of the main page I.e the dashboard on the web browser will display a search bar on top. The stock name can be entered in this search bar, and in case of a valid search, the said details of the particular stock will be returned.

4.2.3 Functional Requirements

* This feature uses a predefined API in order to fetch the live stock prices. The API in recurringly accessed in order to keep stock prices up-to-date.
* MySQL is required in order to fetch stock price history of the particular stored stocks from the database.
* On the user end, only a web browser is needed and a good wifi connection.

## Real-time News Updates

4.3.1 Description and Priority

This high priority feature is another focus of our website. Through an API,

4.3.2 Stimulus/Response Sequences

Opening up of the main page I.e the dashboard on the web browser will display a search bar on top. The stock name can be entered in this search bar, and in case of a valid search, the said details of the particular stock will be returned.

4.3.3 Functional Requirements

* This feature uses a predefined API in order to fetch the live news updates.
* On the user end, only a web browser is needed and a good wifi connection.

# Other Nonfunctional Requirements

## Performance Requirements

* Data from the server should be fetched within 2 seconds.
* Query results must return within 5 seconds
* News updates must be fetched within 2 seconds

## Safety Requirements

* Under failure, system should be able to come back at normal operation in one or two hours.
* Stock prices should be made available within 5 seconds of change in actual index

## Security Requirements

* All external communications between the data’s server and client must be encrypted.
* All data must be stored and protected.

## Software Quality Attributes

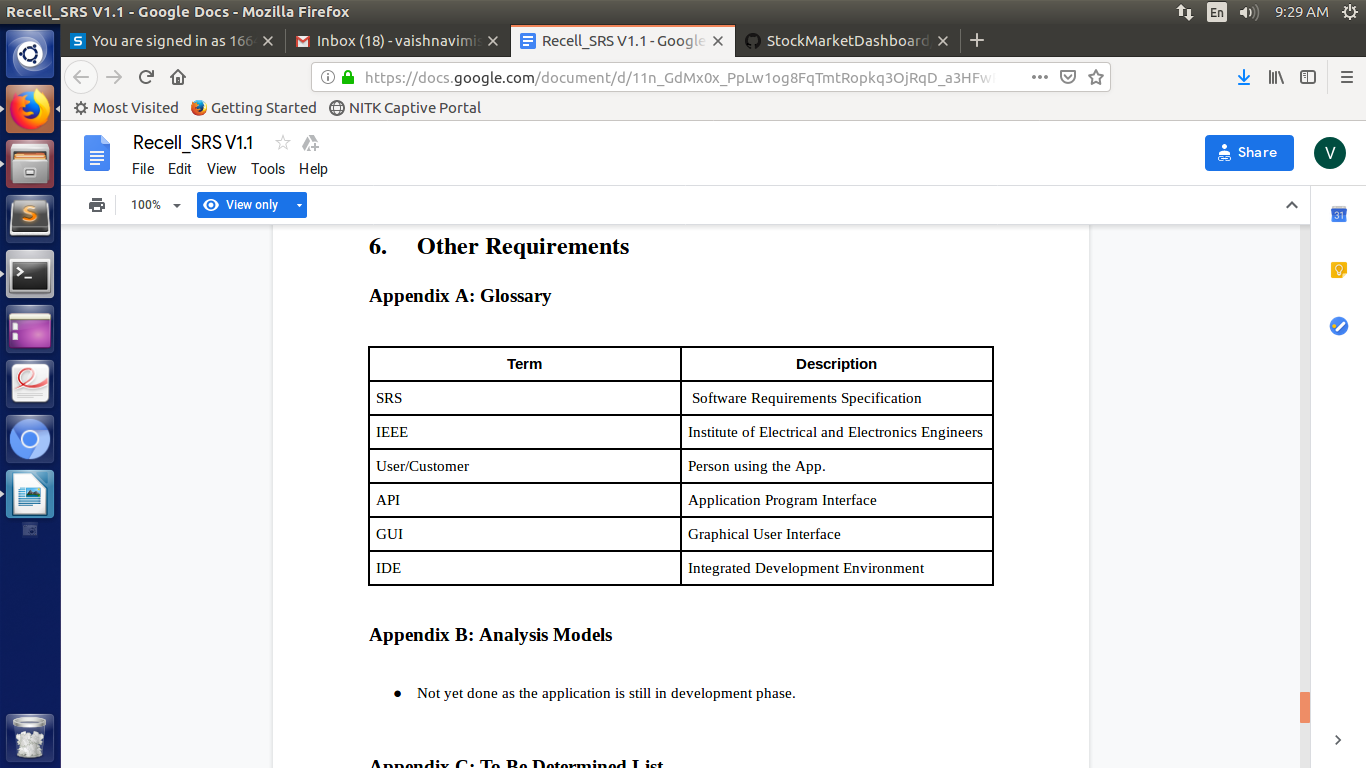
* The website is easy to use.
* All the features present in the website are easy to locate.
* The website uses simple English so that the user does not get confused with the terms

## Business Rules

* System Admin - It will be the system administrator. He will maintain the overall website.
* Stock analyst - It includes the people who will view stock-related information.
* Financial expert- Academia who gives advice about stocks and trends

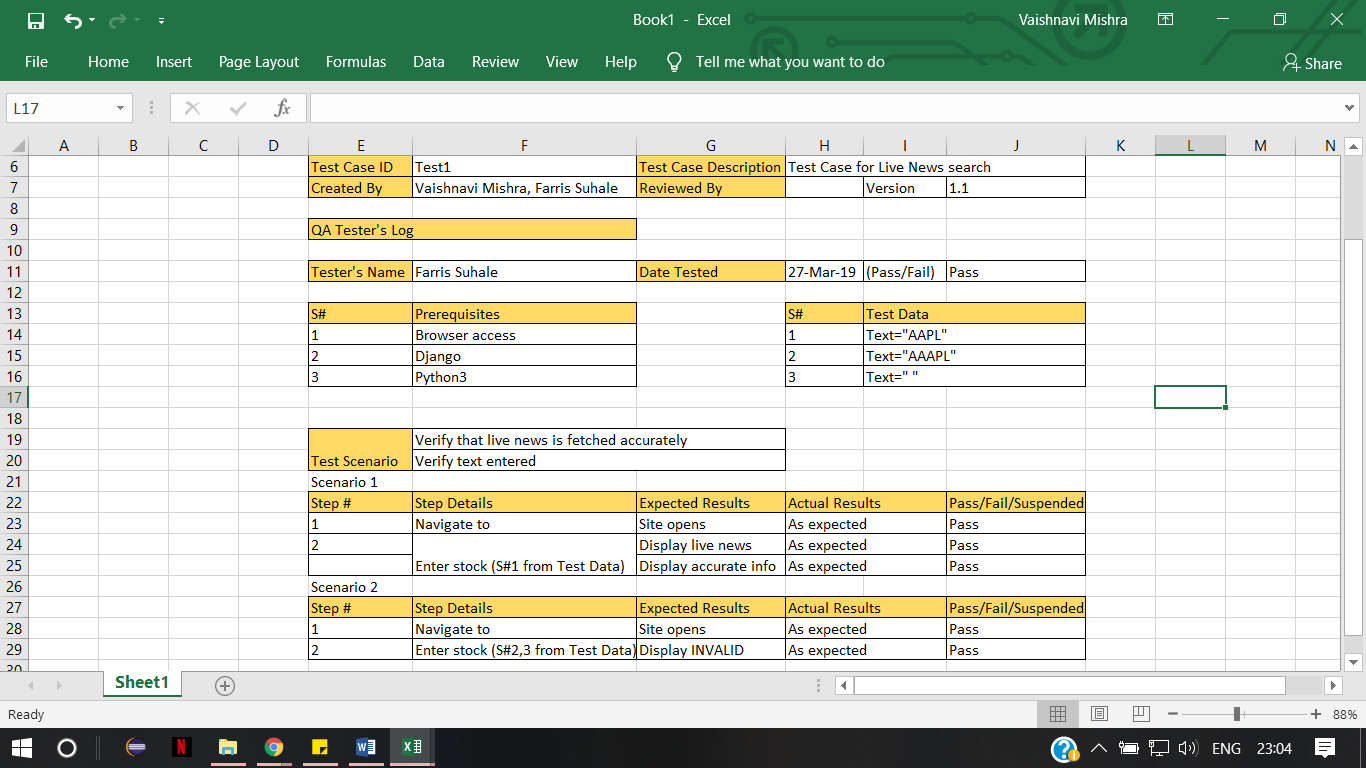
# Other Requirements

Appendix A: Glossary

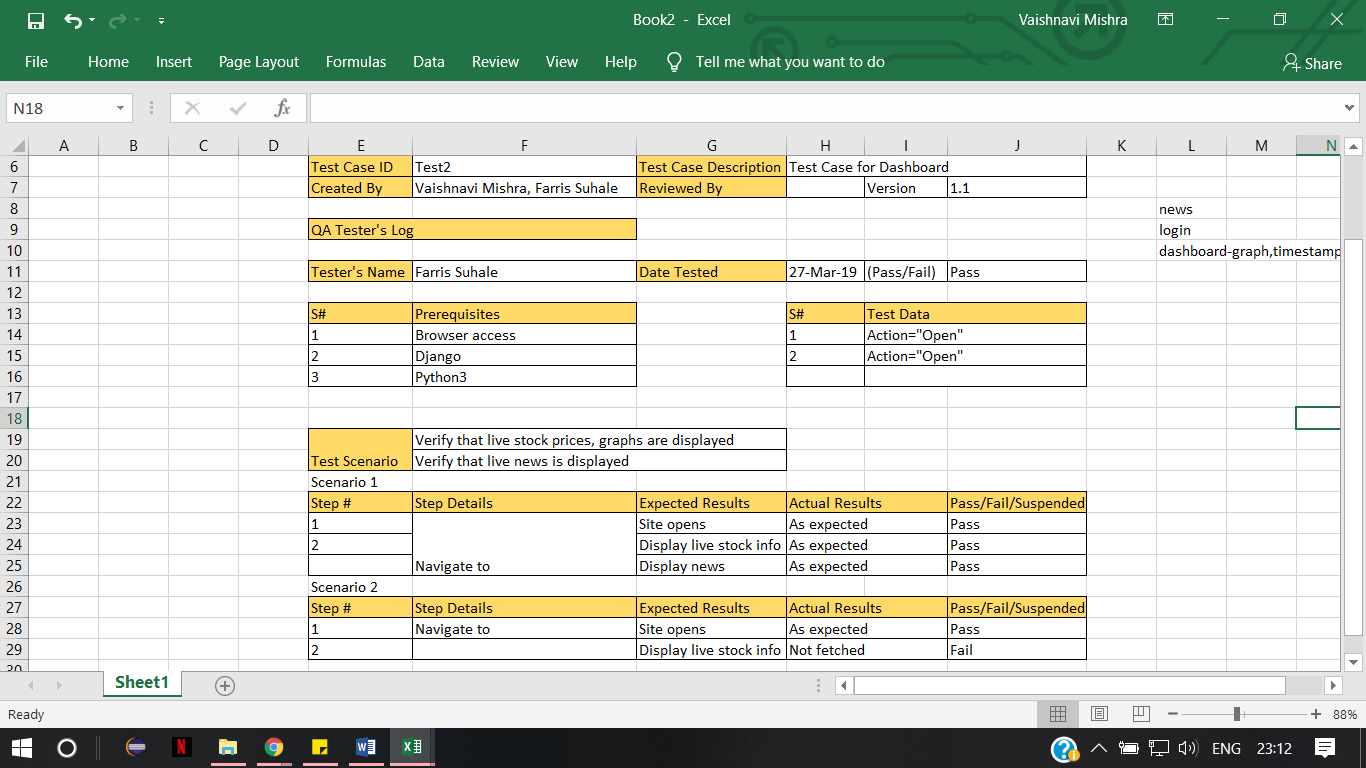


III MANUAL TEST CASE DESIGN

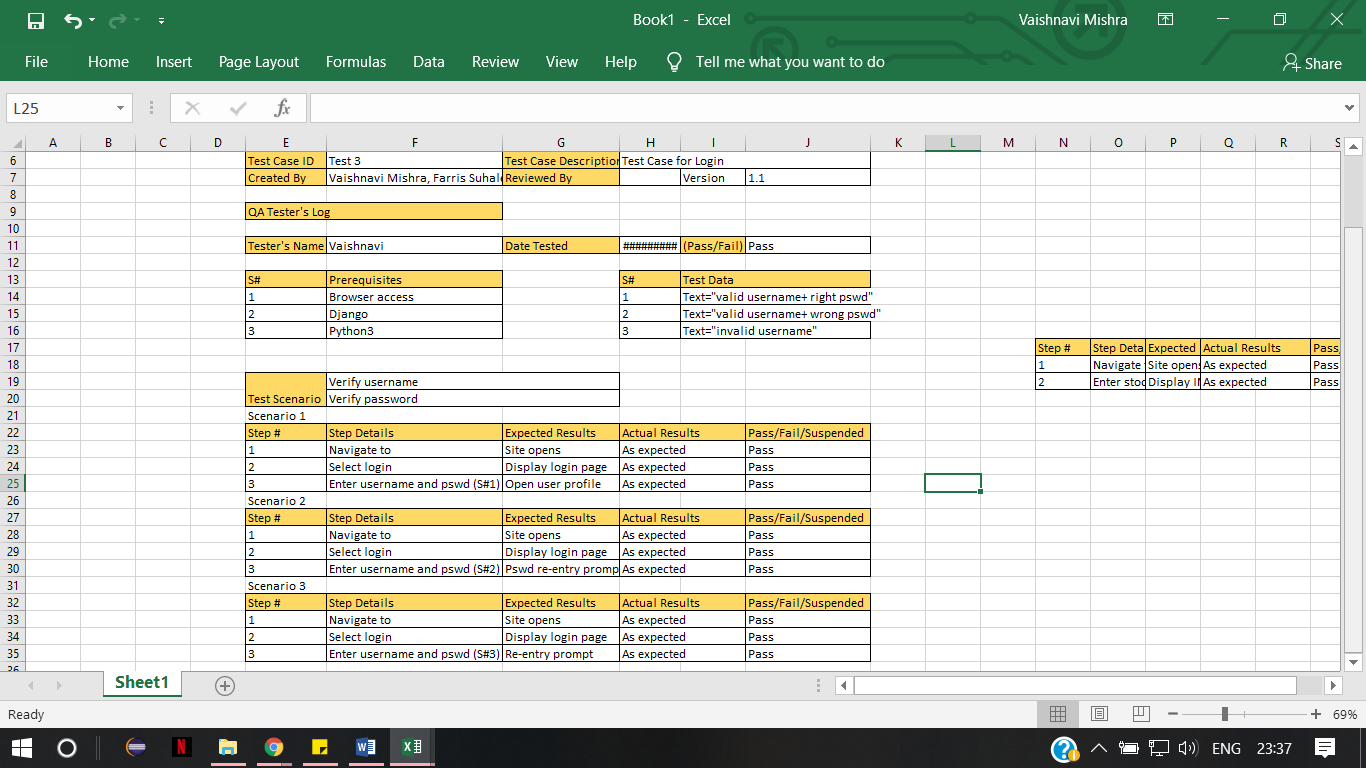
TEST 1: Live news search



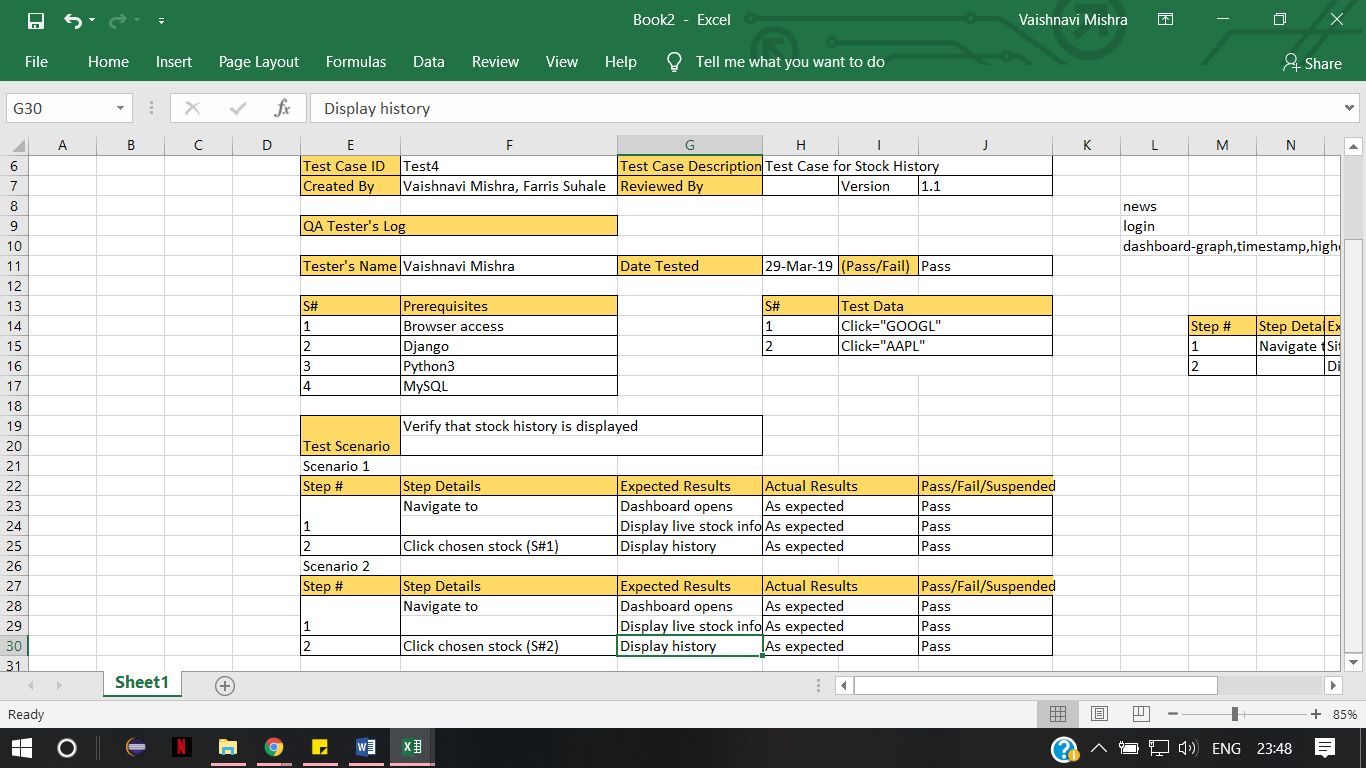
Test 2: Dashboard testing



Test 3 : User login



Test 4 : Stock History



IV TOOLS USED

1. Selenium

Selenium is a set of different software tools each with a different approach to supporting test automation. Most Selenium QA Engineers focus on the one or two tools that most meet the needs of their project, however learning all the tools will give you many different options for approaching different test automation problems. The entire suite of tools results in a rich set of testing functions specifically geared to the needs of testing of web applications of all types. These operations are highly flexible, allowing many options for locating UI elements and comparing expected test results against actual application behavior. One of Selenium’s key features is the support for executing one’s tests on multiple browser platforms.

Selenium 2 is the future direction of the project and the newest addition to the Selenium toolkit. This brand new automation tool provides all sorts of awesome features, including a more cohesive and object oriented API as well as an answer to the limitations of the old implementation.

Selenium was used by us to automate the working of our entire web applcation in order to ease testing.

1. Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django was designed to help developers take applications from concept to completion as quickly as possible. It is reassuringly secure. Django takes security seriously and helps developers avoid many common security mistakes. It is exceedingly scalable. Some of the busiest sites on the Web leverage Django’s ability to quickly and flexibly scale. Although you can use Django without a database, it comes with an object-relational mapper in which you describe your database layout in Python code. The data-model syntax offers many rich ways of representing your models – so far, it’s been solving many years’ worth of database-schema problems.

Django was used by us as to develop and design our website.

1. MySQL

MySQL is the world’s most popular open source database. With its proven performance, reliability, and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, and all five of the top five websites\*. Additionally, it is an extremely popular choice as embedded database, distributed by thousands of ISVs and OEMs.

MySQL is one of the best RDBMS being used for developing various web-based software applications.

MySQL was used by us to store stock price history.

Conclusion

Stocks and the stock market will never go out of style. But not everyone can keep track of and understand them easily. Our web application, through visual analysis of stock price history and by real-time (refreshable at a chosen interval) data on the stocks aids in the understanding of the same by amateur users and finance experts alike.

References

References:

1. <https://grid.cs.gsu.edu/~baydin2/proj/CSC8530/parallel_knn_survey.pdf>
2. <https://link.springer.com/chapter/10.1007/11751649_5>
3. <http://padas.ices.utexas.edu/static/papers/knn.pdf>
4. <https://ieeexplore.ieee.org/document/5607480>