Department of Computer Science North Carolina A&T State University

System Test Plan

COMP 496: Senior Design II

Spring 2020

Finance Goons

Invest the Best

Kyle Setzer

Kameron Slater

Malcomb Coley

James Adams

Revision History

Revision	Date	Author(s)	Description
1.0	04/21/2020	JA	document creation
1.1	04/21/2020	KS,KDS,MC,JA	initial release

Contents

List of Figures

1. Introduction

1.1 Product Concept

Our project, Invest the Best, is a website that we are creating that will allow people to view the prices of the stock market, so that investors have the ability to decide which stocks they want to invest in, so they have the potential to earn more money by owning a portion of the company. The service that we are planning to create will be able to separate company stocks based on the industry that they are a part of. The service will be simple enough so that investors who are experienced in buying stocks and investors who are just starting to invest in different companies. The service will also include a login system that will allow users to log in to the website and have the ability to save their favorite companies that they like to invest in.

1.2 Product Scope

The scope of our project is to use what we have learned in our classes to try and create a website that will be able to also allow users to invest in stocks. Some information that will be included in the system will be an API that will allow our system to update the information that the graphs for each of the companies that will be represented in the website. This is so that investors are able to see how the stock prices of a company changes each date and make decisions on which stocks to purchase as they change throughout the day. We also plan to use HTML and JavaScript techniques to create a website that will be used to display the information provided by the database and web server to the user. This will allow the users that see the stocks of the companies in the stock market so that they can become investors. With the knowledge of machine learning and the python language, these techniques will be used to create an AI algorithm that will be able to find the predictions in the stock market. These predictions will be able to figure out how the stock prices for specific companies will change throughout the day. These information and techniques will be used as the basis for creating the system that our project is based on.

1.3 Testing Scope

1.3.1 Test Inputs

Some tests that will be required when we test our system will be to make sure that our system has a working login feature that will use our database of users that will connect to the web server. The AI algorithm will have to find an accurate prediction on how the stock will change for the next day as the stock market is changing every day. We also need to test our website in great detail such as making sure we can search different companies with the search bar on the website.

1.3.2 Test Environment

For our project we will be testing the website on one of our group members that have a web server so that the website can run on the laptop used so present the data. The web server will be tested on the chrome web browser, the website will be tested using a Tiingo web server. The AI algorithm will be using tensorflow and Keras techniques to make sure that we find the accurate prediction for the stock market.

2. References

This section is about the System Requirement Specifications, the Architecture Design Specifications and the Detailed Design Specifications for our project to be completed. The System Requirement Specifications will contain a detailed explanation about the system requirements for our project and how they are related for the test plans.

2.1 System Requirement Specification

The purpose of the System Requirement Specification is to make sure that the list of requirements that we want out project to system. This is important to our test plan because these are some requirements that are needed for each of our systems that will be needed to make sure the product is working correctly.

Customer Requirements

SRS#	Requirement	Description	Priority
1	Design	User friendly and simple design for ease of use	Critical
2	Search Bar	Users will be able to browse through companies or search a specific company/industry	Critical

Packaging Requirements

SRS#	Requirement	Description	Priority
1	Web Browser	Users will a webpage that runs HTML, Javascript, and CSS	Critical
2	Python and Python Packages	Person hosting our project will need to download a few python modules	Critical

Safety and Security Requirements

SRS#	Requirement	Description	Priority
1	Login feature	Allows users to log in an see their protected and saved preferences for companies they invested in	Critical

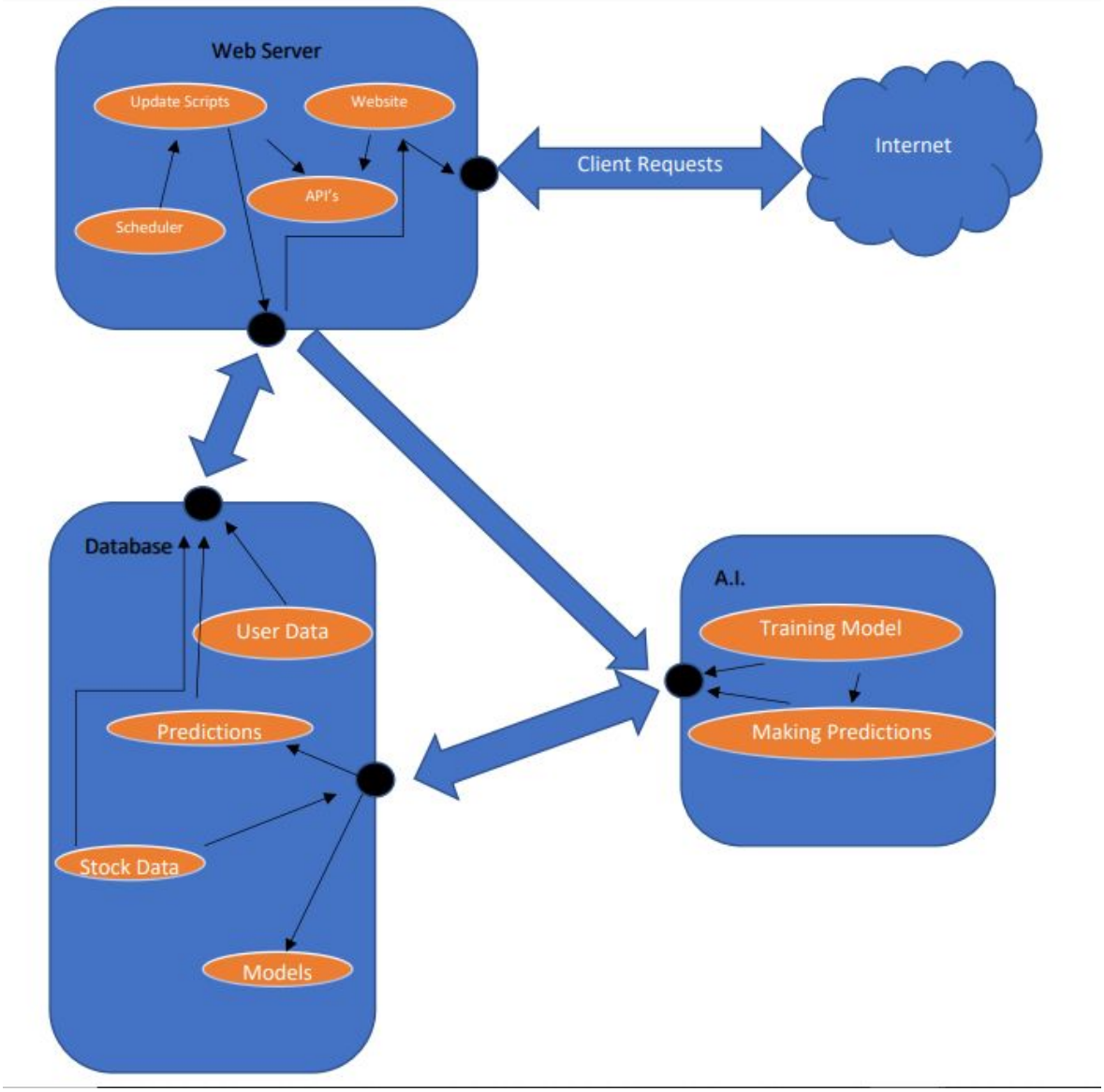
Maintenance & Support Requirements

SRS#	Requirement	Description	Priority
1	Updates	Weekly check-ins on the AI and website to make everything is working correctly	Moderate

2.2 Architecture Design Specification

The architecture design specification uses the details from each of the parts of the system to determine whether the parts of the system for our project will work depending on how each layer of the system is set up. Our project contains four layers to the system which include the website, database, AI algorithm, and the web server. These four layers are important to test because these layers need to be integrated with each other, for the system of our project.

2.2.1 Architecture Design Diagram



2.2.2 Data Flow Definitions

N/A

2.2.3 Design Layers

The layers contained within our architectural design include the website, database, AI algorithm, and the web server. The website is the physical system on which the information that we are getting from the database and the .The database is the system which contains the information of the users for our website and the information of the company's stock market that we plan to display on the website. The purpose of the AI algorithm is to take the data of different company

stock prices and use methods like Tensorflow and Keras to find the prediction of the stock market within the next day. The web server is used as a server to display the website and track package requests made to the made and update the data in the database.

2.3 Detailed Design Specification

The Detail Design Specification goes in a more detailed description of how each of the systems in our project will be used in the project and how each of the parts work. It relates to the testing of our project and it allows the different systems of our project. The plan of this part is to show that the detailed description

2.3.1 Module Descriptions

The website section of the system is to display the information provided by the web server and the database. This allows for many design implementations to be tested in the system such as the login system, the search bar, and the company display window. The database will display the information of the company stock market and the information of the users that will be entered in the . The A.I. Algorithms use the techniques within the code to find the prediction of how a company's stock market price will change in the next day. The web server will be used to test whether the website that is created will be able to run on its server.

2.3.2 Module Decomposition

N/A

2.3.3 Data Flows

Data flow included with the Architecture Design Diagram

3. Test Items

3.1 Hardware Tests

Hardware Test Table

Test ID	Hardware	Input	Output	Test	Priority
#01	Computer	Download website file	Project will run	Make sure that the website works on any device	Critical

3.2 Unit Tests

3.2.1 Website

Website Unit Test Table

Test ID	Module	Input	Output	Test	Priority
#02	Search Bar	Enter a company	Displays company information	Tests the information of each company is accurate	Critical
#03	Company Graph	Search a company	Displays a graph of its stock	Will display company stats	Critical
#04	Login System	Enter a name	User gains access to website	See if a user can login into website	Critical

3.2.2 Web Server

Webserver Unit Test Table

Test ID	Module	Input	Output	Test	Priority
#05	Website Connection	Start Website Application	Open Website	Checks to see if the website opens on the server	Critical

3.2.3 A.I. Algorithm

A.I. Algorithm Test Table

Test ID	Module	Input	Output	Test	Priority
#06	Model Creation	Stock Data	Prediction Model (.h5)	Training script runs to completion without fatal errors	Critical

3.3 Component Tests

3.3.1 Website

Website Component Test Table

Test	Subsystem	Input	Output	Test	Priority
#07	Website	User Information	Log into system and access database of companies	Enter a name from the database into the login	Critical

3.4 Integration Tests

Integration Test Table

Test ID	Layer	Input	Output	Test	Priority
#08	Website	Company Name	Company Stock Graph	Enter a company name	Critical
#09	Database	Enter a user's name	User will log into system	Enter a name from the database	Critical

#10	Web server	a) Successful GET request of /home.html from local network. b) Any company or symbol in search bar	Web page with team logo, company information, graph of stocks, and predictions	Internal connectivity of all systems	Critical

3.5 System Validation

System Validation Test Table

Test ID	Validation	Input	Output	Test	Priority
#11	WAN connectivity	GET request outside of the servers LAN	Web page with team logo, company information, graph of stocks, and prediction	Remote access to the web host	Critical

4 Risks

4.1 Risk Table

Risk Table

Test ID	Risk	Impact	Severity	Mitigation Strategy
#1	Website Failure	Users will not be able to access the industry or company webpage	Critical	Run many tests
#2	Web Server Failure	Users will not be able to access the website	Critical	Run many tests
#3	Login feature	People will not be able to log in	minor	Continue working on it
#4	AI	The predictions will be wrong	Critical	Run many test

5 Features to be Tested

5.1 Customer Requirements

5.1.1 Requirement ID: Data

Description: Test if the data update when new information comes in
Test Approach: Unittest the server

5.1.2 Requirement ID: Login

Description: Test the login and the database
Test Approach: Check if a new user is added to the database

5.2 Performance Requirements

5.2.1 Requirement ID: Website

Description: Make sure the website displays what is needed
Test Approach:

5.2.2 Requirement ID: Server

Description: Test all the server function when interacting with the website
Test Approach: Unittest the server

5.3 Safety and Security Requirements

5.3.1 Requirement ID: Server

Description: Testing how the client website securely connect to the server
Test Approach: Unittest the server

5.4 Maintenance and Support Requirements

5.4.1 Requirement ID: Server Down

Description: If the server is ever stopped
Test Approach: If stopped it should output the when it stopped and error and maybe restart

6 Features not to be Tested

6.1 Packaging Requirements

6.1.1 Requirement ID: Login

This requirement will not be tested because it is a future requirement, therefore testing procedures will be in place for this module but not implemented.

7 Approach

In this section we are describing our approach to successfully finish this project. WE will go through our strategy to test and how it affects the development process. The tools that we are using to test. Also, any core Functionalities that we will need to test.

7.1 Strategy

We have already tested this project as individual pieces. We are going to test the entirety of the project by running the website on the web server and having someone test it by going to the site. We will be checking for mistakes as well as looking to see what can be improved to get the results that we require.

7.2 Tools

We are mostly using python to test all of these features including the AI predictions and the website's page transfers.

7.3 Core Functionality

1. AI Predictions
2. Web server running
3. Website maneuverability
4. Database storage

7.4 Test Metrics

Table 12: Test Metrics

Priority	Description	Success Criteria	Fail Criteria
#1	The AI predictions are a major piece of the project.	100%	80%
#2	The web server needs to be running so the website can be accessed.	100%	100%
#3	The website needs to be able to move through it so it can show the results.	80%	20%
#4	The database has to be able to hold all of the information	90%	10%

8 Item Pass/Fail Criteria

All tests are considered to be successful if, and only if, output matches its respective input for each test. Otherwise it is considered to be a failed test.

8.1 Hardware Tests

Test ID	Hardware	Input	Output
#01	Computer running a web browser	/home.html	Formatted web page ##Logo is not required##

8.2 Unit Tests

Test ID	Module	Input	Output
#02	Search Bar	Company Name or Symbol on /home.html	/company.html web page with correct company
#03	Company Graph	Stock data	Line graph for company displayed on web page
#04	Login System	Google Sign-in data	Name and Email
#05 (web server)	Web connection	Request for /home.html	/home.html
#06	Model Creation	All stock data	.h5 file for each company. .png for each company. .csv for each company with predictions.

8.3 Component Tests

Test ID	Component	Input	Output
#07	Website Login	Google sign-in	User preferences

8.4 Integration Tests

Test ID	Integration Layer	Input	Output
#08	Website	Company Name	Web page with company information, graph of stocks, and prediction
#09	Database	· User data · Predictions · Models · Stock data	No fatal errors
#10	Web server	a) Successful GET request of /home.html from local network. b) Any company or symbol in search bar	Web page with team logo, company information, graph of stocks, and prediction

8.5 System Validation

Test ID	Validation	Input	Output
#11	WAN request for any company's stock prediction	a) Successful GET request of /home.html from outside local network b) Any company or symbol in search bar	Web page with team logo, company information, graph of stocks, and prediction

9 Test Deliverables

9.1 Test Plan Instructions:

To ensure all testing is done in the same manner, it is recommended that all documents be git cloned or downloaded in zip format before starting any tests, as working with modified documents may cause unforeseen errors. The tests are also done in a way that leads into the next for most tests.

1. Download zip file from: https://github.com/Adams408/Invest_the_Best
2. Unzip content into known location

* It is recommended that the documents be downloaded onto a machine capable of running a web server such as Microsoft IIS
** Troubleshooting is beyond the scope of this document

Test ID #01:

1. Navigate to the file directory where the files were unzipped
2. Go to: ../code/app/web/templates
3. Double click on “home.html” to launch into the default web browser
4. If home.html is displayed, proceed to the next test.

Test ID #02:

1. From home.html, type a company name into the search bar
2. Click search
3. If a page is displayed, proceed to the next test.

Test ID #03:

1. If the stock data for that company shows in the graph, proceed to the next test.

Test ID #04:

1. From any page, click the login button
2. If you are prompted to log into a google account, proceed with login
3. If login is successful, proceed to the next test.

Test ID #05:

*set up of the web server is beyond the scope of this document.

1. Close all web browser windows
2. Open new window, and navigate to address of web server
3. If home.html is displayed, proceed to the next test.

Test ID #06:

*Installation of python and required modules are beyond the scope of this document

1. Open a terminal, for windows - cmd.exe
2. From the home directory, navigate to ../code/app/ai directory
3. Run “train_works_multi_predictions.py”
    a. File name may change
4. If the script finishes without error, proceed to the next test.

Test ID #07:

*This test requires the use of the database.

1. Once, logged into your account, save a company as your favorite
2. Logout
3. Log back into that same account
4. If favorites are displayed, proceed to the next test.

Test ID #08:

1. Open a web browser window to the home page of the website
2. Type a company into the search bar
3. If all company data is shown, including prediction, proceed to the next test.

Test ID #09:

1. Open the prefered database server
2. Ensure User, prediction, model, and stock data is available
3. If all data is available, proceed to the next test.

Test ID #10:

1. Close all web browser windows
2. Open new window, and navigate to address of web server
3. If home.html is displayed, proceed to the next test.

Test ID #11:

1. From outside the web server’s local network, navigate to the website
2. If home.html is displayed, all tests have been successfully completed and the system is validated.

10 Test Schedule

Test ID	Current State	Expected time for test completion (hh:mm)	Expected time for correction if test fails* (hh:mm)
#01	**Pass**	00:01	00:32
#02	**Fail**	00:01	01:02
#03	**Pass**	00:01	01:02
#04	**Fail**	00:02	01:04
#05	**Pass**	00:01	01:02
#06	**Pass**	00:30	02:00
#07	**Fail**	00:01	01:02

#08	**Fail**	00:01	01:02
#09	**Pass**	00:10	01:20
#010	**Pass**	00:01	01:02
#011	**Fail**	00:02	01:04
Totals		00:51	12:14

*The expected time for correction (column 4) includes: the initial test, estimated time for debugging, and another test.