Project: TraderInvader

Date: 10/24/2019

Version 1.0

Emails: levi.leuthold@oit.edu, tristen.mengis@oit.edu, jesse.whiting@oit.edu, andrew.young@oit.edu

# Legal Notice

Levi Leuthold, Tristen Mengis, Jesse Whiting, and Andrew Young make no warranties of representation of any kind regarding the information presented in this document for any purpose. Under no circumstances will the document owners or other representatives be liable for direct, indirect, special, incidental, consequential, punitive, or aggravated damages. No responsibility is assumed for any consequence relating directly or indirectly to any action or inaction taken based on the information provided in this document.

# Copyright Notice

©2019-2020 Copyright Levi Leuthold, Tristen Mengis, Jesse Whiting, Andrew Young, All Rights Reserved. This document is protected by U.S. and international copyright laws. Reproduction and distribution of the document without written permission of the sponsors is prohibited.

# Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Authors | Company | Version | Date | Filename | Comments |
| Levi Leuthold, Tristen Mengis, Jesse Whiting, Andrew Young | N/A | 0.1 | 10/24/2019 | Proposal.docx | Starting draft |
| Levi Leuthold, Tristen Mengis, Jesse Whiting, Andrew Young | N/A | 0.25 | 10/27/2019 | Proposal.docx | Finished starting sections through 6.1 |
| Levi Leuthold, Tristen Mengis, Jesse Whiting, Andrew Young | N/A | 1.0 | 10/27/2019 | Proposal.docx | Finished all sections |

# Signatory Page

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author: Levi Leuthold Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author: Tristen Mengis Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author: Jesse Whiting Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author: Andrew Young Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Reviewer: Todd Breedlove Date

Contents

[Legal Notice 2](#_Toc23509188)

[Copyright Notice 2](#_Toc23509189)

[Revision History 2](#_Toc23509190)

[Signatory Page 3](#_Toc23509191)

[1. Introduction 5](#_Toc23509192)

[1.1 Purpose 5](#_Toc23509193)

[1.2. Scope 5](#_Toc23509194)

[1.3. Intended Audience 5](#_Toc23509195)

[2. Project Management 5](#_Toc23509196)

[2.1. Change Management Procedures 5](#_Toc23509197)

[2.2. Software Delivery, Installation, and Acceptance Criteria 5](#_Toc23509198)

[2.3. Documentation and Online Help 5](#_Toc23509199)

[2.4. Project Risks 5](#_Toc23509200)

[2.5. Customer Responsibilities 6](#_Toc23509201)

[2.6. Status Reporting 6](#_Toc23509202)

[3. System General Description 6](#_Toc23509203)

[3.1. Problem Statement 6](#_Toc23509204)

[3.2. Perspective 6](#_Toc23509205)

[3.2.1. History / Background / Prior Releases 6](#_Toc23509206)

[3.2.2. Major Subsystems 6](#_Toc23509207)

[3.2.3. Relation of Development of Existing Systems 6](#_Toc23509208)

[3.2.4. Hardware Platform Description 7](#_Toc23509209)

[3.2.5. Software Platform Description 7](#_Toc23509210)

[4. Product Requirements 7](#_Toc23509211)

[4.1. Functional Requirements 7](#_Toc23509212)

[4.2. Performance 10](#_Toc23509213)

[4.3. Reliability 11](#_Toc23509214)

[4.4. Data Description 11](#_Toc23509215)

[4.5. Security and Safety 11](#_Toc23509216)

[5. Appendix 12](#_Toc23509217)

[5.1. Change Request Form 12](#_Toc23509218)

# 1. Introduction

## 1.1 Purpose

This document will propose the design and implementation of the TraderInvader project. The following content will define the formatting of this document and describe the design plan. If this proposal is accepted, then it will serve as a guide for implementation of the project and as an agreement between the authors and reviewer. The following sections are devoted to defining the format of the document, the project management process, and the design and implementation of the project.

## 1.2. Scope

The project scope is constrained to general system discussion, legality, project management, perspective, as well as product design and requirements. The product design and requirements are meant to be general and will only be described in limited detail. Also included are potential features and functionalities that may later be added to the project.

## 1.3. Intended Audience

The intended audience of this document includes authors, reviewer Todd Breedlove, faculty of the OIT CSET department, and any third party interested in the project.

# 2. Project Management

## 2.1. Change Management Procedures

If any person desires change to the project, a change request form is provided in Appendix A of this document. Upon request, a team consisting of Levi Leuthold, Tristen Mengis, Jesse Whiting, Andrew Young, and Todd Breedlove will consider the changes. This team will make a decision concerning the change based on the impact it will have on the project.

## 2.2. Software Delivery, Installation, and Acceptance Criteria

All versions of the software for this project will be available on GitHub (can be found at github.com/TraderInvaderJP/TraderInvader) and will be made available for anybody who requests viewing. Upon completion of the project, the application will be hosted and made available through major web browsers (e.g. Google Chrome, Mozilla Firefox).

## 2.3. Documentation and Online Help

Software documentation will be provided on GitHub. The documentation will detail software requirements, installation process, and general operation of the software.

## 2.4. Project Risks

The stock information used in this application will be acquired from an external finance API. This could possibly incur costs or apply limits to the amount of data that can be used. As well, there may be other costs incurred depending on the hosting platform.

## 2.5. Customer Responsibilities

The customer will be responsible for providing a device capable of running a web browser with JavaScript enabled. In order to provide a complete user experience, the customer should use a web browser that supports service workers and the web app manifest (Google Chrome, Mozilla Firefox, etc.).

## 2.6. Status Reporting

Status reporting will happen weekly. The report will include the following:

* + Work completed during the current week
  + Work to be completed during the next week

# 3. System General Description

## 3.1. Problem Statement

Create a web application where the users can buy and sell stocks, simulating real world stock market of the United States stock market. The users will be able to create games and join games with their friends, being able to customize the win conditions and rules of each game. The users will be able to keep track of their current and past games played through a user statistics page. TraderInvader will notify the user when the market is open and when it will be closing. TraderInvader will be using the actual prices stocks of the United States stock market and business hours. The user will be able to create buy orders and sell order while the market is closed that will go into effect when the market opens the following day. The profits or losses the user makes in TraderInvader would be roughly the same if the user were to buy and sell stocks in the real-world stock market.

## 3.2. Perspective

### 3.2.1. History / Background / Prior Releases

TraderInvader has been an idea that was generated during the first few weeks of out junior project team meetings. Additional features and options are still be added and will continue until the project takes its finished form.

### 3.2.2. Major Subsystems

This system is comprised of two major subsystems. The user interface and the database. The user interface will consist of many different pages, each with different uses and options. The web application will consist of a main/home page and all other pages within the application will be accessible from that page.

The database will be used to hold the users current and past game information. It will hold the user's usernames, friend lists and statistics of their game history.

### 3.2.3. Relation of Development of Existing Systems

The United States stock market information that will be used in TraderInvader will be generated from a finance API.

### 3.2.4. Hardware Platform Description

* + - Minimum of 4 GB ram.
    - Internet access.

### 3.2.5. Software Platform Description

TraderInvader is intended to run in any web browser, including Internet Explorer, Google Chrome, Mozilla Firefox and Microsoft Edge and will operate regardless of the operating system.

# 4. Product Requirements

## 4.1. Functional Requirements

1. Login page
   1. The user will be able to create a new account or login to an existing account - H
      1. The user will be able to use Google sign in – M
   2. The system will use the google login API to confirm credentials. - M
2. Main page
   1. Accounts tab
      1. The user will be able to view his or her username and email - L
      2. The user will be able to edit his or her username, email, or password - L
      3. The user will be able to view their own friend code - H
         1. When the user creates an account a unique friend code will be generated for the user – H
         2. The system will generate a new unique random number for a user during account creation – H
   2. Games tab
      1. The user will be able to view current ongoing games or completed games – H
         1. The system will query the database and display the data when the user wants to view their ongoing or completed games – H
      2. The user will be able to create a new game – H
         1. The system will store the new game in the database – H
      3. Ongoing Game
         1. The user will be able to search for stocks by symbol to purchase -H
            1. The system will query a finance API to retrieve up to date information on the stock symbol the user searches for – H
         2. The user will be able to purchase stocks – H
         3. The user will be able to short stocks - S
         4. The user will be able to sell stocks from their portfolio - H
         5. The user will be able to view their portfolio for that specific game - H
         6. The user will be able to view current players and player rankings – H
            1. The system will store and track current player and player ranking information in a database – M
         7. The user will be able to see how much money they must spend and/or how much time is left - M
      4. Creating a Game
         1. The user will be able to give the new game a name - H
         2. The user will be able to select the win condition (i.e. most stocks bought/sold, most money made etc.) the win conditions are decided by the user - H
         3. The user will be able to select the time length of the new game (example: 1 day, 1 week, 1 month etc.) - H
         4. The user will be able to select the starting amount available (example: $10,000, $100,000, $1,000,000, etc.) - H
         5. The user will be able to invite other users to their game by using their friends unique friend code - H
         6. The user will be able to adjust buy / selling rules - M
         7. The user can set a limit for how many unique stocks can be purchased - M
         8. The user can specify the price range of stocks that can be purchased - M
         9. The user will be able to select other markets, i.e. crypto currencies, International currency trades – S
         10. The system will generate the game based on the specified rules and store that information into a database – H
      5. Daily Challenges
         1. The user will be able to select from a variety of challenges to complete over a specified length of time - M
            1. Examples could include being limited to a certain sector, reach a certain amount of money, limited to penny stocks, start with a very small amount, can only buy a certain number of stocks – M
         2. The system will store custom daily challenges and start a new one every day for users to participate in – M
   3. Stats tab
      1. The user will be able to view the global stats of top players who have the highest win rate of daily challenges by month. - M
      2. The user will be able to view the stats of users who have the highest win rate of the daily challenges by month in his/her friends list. - M
      3. The user will be able to view their own personal stats. - H
      4. The user will be able to view their achievements / badges – L
      5. The system will store stats and achievement information into the database – H
   4. Social tab
      1. The user can view their list of friends - H
      2. The user can add a friend using a friend code – H
      3. The user can view a friend’s profile - L
   5. Stock market status indicator
      1. Tells the user whether the United States stock market is open, closed, closing soon, or open soon - H
   6. Help tab
      1. The user will be able to view the help tab for a basic overview of how the application works - M
   7. Settings tab
      1. The user can change appearance settings of the application - L
      2. The user can change notification settings - L
      3. The user can change language settings - S
3. Progressive Web App (PWA)
   1. App Manifest
      1. The app will display an “add to home screen” prompt on supported browsers (Google Chrome, Mozilla Firefox, etc.) - M
         1. The app will run on Apple devices - S
      2. The app will have an icon to display on a user’s device - M
   2. Service Worker
      1. The app will cache an application shell to allow for instant loading on devices - M
      2. The app will cache pages to allow for some usability while the user is offline – M
4. Database
   1. Player table
      1. A new row is created whenever a new user account is created - H
      2. Store each player’s unique id - H
      3. Store each unique username related to each player id - H
   2. Game table
      1. A new row is created whenever a new game is created, generating a unique game id - H
      2. Store each unique game id - H
      3. Store the start time of game - H
      4. Store the end time of game - H
      5. Store the rules / win condition for the selected game - H
   3. Portfolio table
      1. Every game joined / started by the user creates a new portfolio row for that user storing relevant information related to the new game - H
      2. Stores the player id - H
      3. Stores game id - H
      4. Stores all stock symbol id - H
      5. Stores shares owned - H
      6. Stores original purchase price of each share - H
   4. Money table
      1. Stores player id - H
      2. Stores game id - H
      3. Stores amount of funds related to a game id for that user - H
   5. Friend list table
      1. Whenever a user adds a friend into their friend list, a new row containing their user id and their new friend’s user id is created - H
      2. Store player id - H
      3. Store friend id’s correlated to the user’s player id - H
   6. Stock table
      1. A database that is automatically updated from the API with current shares prices every 15 seconds. This table is queried by the user to see current prices of shares - M
      2. Store stock symbol id’s - H
      3. Store stock names - H
      4. Store current share price – H
   7. Daily Challenges table
      1. Store daily challenge id – M
      2. Store daily challenge name - M
   8. Achievements/Badges table
      1. Store achievement/badge id – L
      2. Store achievement/badge name – L
      3. Store badge requirements - L
   9. Player Achievement/Badge Table
      1. Store achievement/badge id - L
      2. Store player id - L
      3. Store id of game achievement/badge completed in - S
5. Backend
   1. The backend will allow the frontend to retrieve Player ID and username from the database - H
   2. The backend will allow the frontend to retrieve Game ID, start time, end time, rules, and win condition from the database - H
   3. The backend will allow the frontend to retrieve Portfolio stock symbol, shares, and original price for a Player in a Game from the database - H
   4. The backend will allow the frontend to retrieve the amount of money a player has in a game from the database - H
   5. The backend will allow the frontend to retrieve a Player’s list of Friends from the database - H
   6. The backend will allow the frontend to retrieve the Stock symbol and current price of a specific stock from the database - H
   7. The backend will allow the frontend to retrieve the current day’s daily challenge from the database - H
   8. The backend will allow the frontend to retrieve Badge IDs, names and requirements from the database - H
   9. The backend will allow the frontend to retrieve the list of badges a given player has completed from the database - H

## 4.2. Performance

This application will be able to perform on any major web browsers (e.g. Google Chrome, Mozilla Firefox). Due to the nature of progressive web apps, the service worker will cache the pages of the app to allow for instant loading like what is seen in native apps.

## 4.3. Reliability

Our application will maintain 98% reliability, if the user is connected to an internet source. We plan on hosting the application on a reliable cloud hosting service, so the reliability will be consistent.

## 4.4. Data Description

Stock data will be retrieved from an external API. Player and game specific data will be persisted in a database.

## 4.5. Security and Safety

This application will have sign-ins handled by Google meaning the program will not directly store passwords of any users.

# 5. Appendix

## 5.1. Change Request Form

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date Requested: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Description of Proposed Change:

Reason For Change:

Comments: