CS Capstone Problem Statement

October 27, 2017

Investment Performance Mobile App

Prepared for

HedgeServ

Brice Lemke _____ _____
Signature *Date*

Prepared by

Group 68

The Cleverly Named Team

Tyler Jones _____ _____
Signature *Date*

Samuel Cooney _____ _____
Signature *Date*

Avi Sinha _____ _____
Signature *Date*

Abstract

(https://tobi.oetiker.ch/lshort/lshort.pdf)

CONTENTS

1 Introduction

1.1 Purpose

The purpose of this mobile application will be to provide it's users insight into the quality of their investments from a portfolio level. By using this application, a user will be able to track and gain valuable insight that wouldn't be possible otherwise.

1.2 Scope

The scope of this project is to provide users with a method to track investments of a variety of different asset classes. Users will be able to either add investments in bulk or one investment at a time through the user interface. Upon entering an investment, the user will have a variety of options at their disposal to learn more about the quality of their investment.

1.3 Definitions, acronyms, and abbreviations

At this point in time, there are no relevant abbreviations or definitions to elaborate on.

1.4 References

NULL

1.5 Overview

The project will be written in C# on the front end, and use SQL on the backend.

2 Overall Description

2.1 Product perspective

2.2 Product Functions

This mobile app will provide users investment performance from a portfolio level. The user will be able to enter investments, either in bulk or individually, and then be able to drill down in order to get further data on individual investments, or on the entire portfolio.

General Requirements: This app must have the following general requirements: Firstly, the app will be written in C# for the front end, and SQL for the backend. There must be no longer than a 3 second response time on any given query to the database. If queries fail, the data should be blank or missing, rather than inaccurate. Users should be able to rely on 100% accurate pricing data, even if not 100% up to date; inaccurate pricing is not an option. Investment types, or asset classes, must not be hard coded - this means that the user must be able to enter any investment into the app, even if there is no automated pricing data for it. The app must support bulk data entry, or individual data entry - the user should be able to add their whole portfolio at once, or add just one new investment. Investments must be able to be removed from the portfolio, as the user should have complete control over what they track. In terms of design, each page in the app should require no more than 2 taps to access from the home screen.

Portfolio and Individual requirements: From a portfolio level, the user will be able to see a pie chart with their portfolio split up into percentages. For instance, if a user owns $1000 worth of assets, these may be across various

different asset classes. A user may have $500 invested in the stock market, and $500 in cryptocurrency. The pie chart would reflect this division. Additionally, the user will be able to retrieve benchmarks on their entire portfolio. The smp500, for example, would provide the user with a benchmark by which they could compare the performance of their entire portfolio at once. Morover, the user will be able to disect the peformance of their entire portfolio based off a variety of time intervals. The user will see the aggregate performance of their investments over either the short or long term. Lastly, the user will be able to see their Sharpe ratio and volatility of their portfolio. Each of these requirements must not only be provided from a portfolio level, but also an individual investment level. This means that the user should be able to retrieve time data, Sharpe ratio, volatility, and benchmarks of their investments, when relevant/available for any specific investment.

3 Specific requirements