CS Capstone Problem Statement

October 18, 2017

CS461 Fall 2017

Investment Performance Mobile App

Prepared for

HedgeServ

Brice Lemke

Prepared by

Group 68

Aviral Sinha, Tyler Jones, Samuel Cooney

Abstract

As one of the industry leaders in independent fund administration, HedgeServ has clients who hold assets of all types across the globe. This document outlines the full scope of a cross-platform mobile application that will allow users to track their investment performance on a portfolio level. The application will be user friendly, have quick response time, and be compatible with both iOS and Android devices. The portfolio data will be provided by HedgeServ, and the application will obtain real-time stock prices and other asset baselines from trusted sources, like Yahoo Finance. The portfolio will allow users to track assets and investments of many different types including but not limited to: stocks, properties, currency, and savings bonds.The application will be developed using C# for the front end experience and an SQL database on the back end. In addition to these tools, Xamarin will be used to ensure the app is compatible with both Android and iOS devices.

Contents

1 Definition and Description of Problem

HedgeServ is interested in the creation of a mobile application that lets users monitor the performance of their investments at a portfolio level and view specific investments in more detail. HedgeServ currently has relationships with hundreds of clients including hedge funds, private equity funds, and other institutional investment managers. The company also supports various strategies used in the industry that include but arent limited to long/short equity, fixed income, distressed mergers and acquisitions, and emerging markets. While the company is located in New York, the team that operates out of Portland will be the primary point of contact.

This project will entail the design and development of a cross-platform mobile application that allows users to view their portfolio information as well as the status of their investments. While collaborating with Brice Lemke and other HedgeServ team members, Xamarin and C# will be used to develop the financial application that can be used on Android and iOS. Using simulated portfolio and client data provided by HedgeServ, the application will contain, at a minimum, holdings, gains and losses, and transaction history, all in user-friendly charts and figures. Additionally, benchmark performances, which will be obtained from trusted sources online, for investments will be included when appropriate. This app intends to supplement users investment experience by allowing them unique insight into their unique assets. Property investments, cryptocurrency, futures, and more will be supported.

2 Proposed Solution

The proposed solution for this project is to use C# on the front end, and SQL Server for the back end. The front end will display different charts, facts, and figures regarding the investors various assets. The back end will utilize databases that are custom built, rather than existing databases that are used by HedgeServ. The application will utilize the Oregon State University provided SQL database. However, HedgeServ does not provide real-time pricing inside its portfolio data. In order to effectively show the user the performance of their investment, pricing from other accurate sources will be retrieved and displayed alongside the investors portfolio. The specific calculations and data to be displayed is still being decided at the time of this proposal, and will likely be subject to change as the project moves forward. While we will be updating the portfolio positions using a form, we will be updating the portfolios holding by reading a file of trades or using an API of some sort.

This project will culminate into a mobile app that is compatible with both iOS and Android. In order for the application to be successful, the use of a tool that permits the effective migration and testing of both operating systems is necessary. To achieve this, tool called Xamarin will be used. Xamarin is a piece of software that allows for cross-platform development of mobile applications. It operates on a shared C# codebase, and allows for the use of the same APIs, IDE, and language during development.

In order to address incorporating investments of many different types, the plan is to allow the user to fill out a blank form which specifies the type, name, and amount invested for their entry. Depending on the investment that was entered by the user, the data that is relevant will change. Therefore, it will be necessary to incorporate and provide information about the industry benchmarks for that given investment. For example, the benchmark for property investment and the benchmark for stocks are very different, and thus will both need to have different benchmarks displayed. The ability to enter any type of specific investment will allow the user to diversify their portfolio as they see fit.

This application will be developed over the course of the 7 months, October through April and thus, a timeline and clear goals must be set in order to ensure its success. There will be 3 major cycles that the development timeline will be segmented into. The first chunk will span October through December, where all of the design decisions will be made. This document falls under the design cycle of the overall timeline. The second will be January through March, which will be dedicated to development. The last chunk will be April through May, which will entail the final testing, fine tuning, and presentation of the final product.

3 Performance Metrics

Performance metrics are from the perspective of an end-user. This means that the metrics that are planned here are going to be analyzing the finished mobile app performance.

There are currently 5 metrics that will be used to grade the application:

App Crashes: The application must crash less than 2% of loads, anything greater will result in a lasting, negative impression on the user.

API/Backend Latency: Any use of APIs, or any calls to the Oregon State University database must take no more than 1 second from initial request to final response

Ease of Navigation: Attention span of the user and their desire to go digging for any given piece of data is quite low. Any subpage from the initial main page must be no more than 2 clicks/taps away

Scalability: For the back end, we must be able to handle a number of users that's on the scale of thousands. The application needs to remain quick with 1 second response time even with multiple concurrent users.

Effectiveness: The user is able to clearly see and find any data/charts or graphs, and the data must be correct. Any data or calculations for the information and charts must be the most recent data from the source of choice.