Software Requirements Specification

for

Portfolio   
Management  
 Solution (PMS)

Prepared by Alexander R. Ramsey

University of Maryland University College  
ITEC 630: Information Systems Analysis, Modeling, and Design  
Semester Individual Project

04/05/2017

Disclaimer: This document is generated for educational purposes only and does not officially represent any mentioned technologies, solution providers, or brands.

Table of Contents

Revision History 2

1. Introduction 3

1.1 Purpose 3

1.2 Problem Statement 3

1.3 Document Conventions 3

1.4 Intended Audience and Reading Suggestions 3

1.5 Product Scope 4

1.6 References 4

2. Overall Description 4

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Classes and Characteristics 5

2.4 Operating Environment 6

2.5 Design and Implementation Constraints 6

2.6 User Documentation 6

2.7 Assumptions and Dependencies 6

3. External Interface Requirements 7

3.1 User Interfaces 7

3.2 Hardware Interfaces 11

3.3 Software Interfaces 11

3.4 Communications Interfaces 12

4. System Features 12

4.1 Non-functional Requirements Definitions 12

4.2 Functional Requirements Definitions 15

5. Other Nonfunctional Requirements 21

5.1 Performance Requirements 21

5.2 Safety Requirements 21

5.3 Security Requirements 21

5.4 Software Quality Attributes 22

6. Database Requirements and Definitions 22

6.1 Data Flow Diagram Level 0 22

6.2 Data Dictionary 23

6.3 Entity Relationship Diagram 24

7. System Architecture 25

7.1 High Level Architecture Diagram 25

8. Project Plan 26

8.1 Task List for Gantt Chart 26

8.2 Gantt Chart 27

8.3 Estimated Cost 28

8.4 Estimated Work Hours 29

9. Key Terms 30

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Alexander R. Ramsey | 04/05/2017 | Inception, Addition of Section 1 | 1.0 |
| ^ | 04/06/2017 | Addition of Section 2 | 1.1 |
| ^ | 04/07/2017 | Addition of Sections 3 - 5 | 1.2 |
| ^ | 04/08/2017 | Revision of Section 1. Introduction  Addition of Sections 6 - 9 | 1.3 |

# 

# Introduction

## Purpose

This document defines the requirements, system design, database design, and project plan for the implementation of a v1.0 Portfolio Management Solution (PMS) designed for Investment Advisors. A Portfolio Management Solution is a financial software solution that manages investment account portfolios and associated investment holdings activity to better forecast and track financial performance. This solution will be designed as a web based solution, with the intended end user as a financial advisory firm and the end holding client. Further details regarding the user base are defined via a Use Case Model Survey.

## Problem Statement

Portfolio Holding Organizations and Financial Brokers provide limited insight into the strength, performance, and activity of client portfolio holdings. Investors often find the process for determining the performance of account holdings confusing. For example, an Investor with retirement holdings at several institutions may find it difficult to understand how their retirement account is performing in the market as a diversified set of accounts. This software will provide for them with a clearer picture of their account performance while consulting with their financial advisor, because it will 1) track historical contributions, 2) compare current performance with market rates, and 3) graph projections about future contributions.

## Document Conventions

This document is written with the intention to conform to the Institute of Electrical and Electronics Engineers (IEEE) 830 format and requirements analysis recommendation. This document contains language commonly found in the software, web application, and financial management fields. These terms are bolded at first occurrence to signify their industry specification. Industry Key Terms which are **bolded** are defined in the key terms section of this document.

## Intended Audience and Reading Suggestions

The intended audience for this document are project managers in the financial software development sector. It is recommended that former knowledge of web application development, computer programming, and the financial investment sector are known to fully understand and implement this software project. In addition, this document is intended to produce a first version release. This first release of the software will provide limited functionality and is only designed to deliver proof of concept level of functionality. Additional technical implementation, working financial details, and requirements gathering with key stakeholders would garner further business process details and system feature backlog.

## Product Scope

The intended objective and scope of this application can be categorized as:

* Providing deeper financial insights into the performance of financial holdings by
  + Tracking contribution activity over time
  + Graphing performance of holdings based on current market data
  + Graphing projections of holding value as affected by market performance and future contributions
* Improving business workflow and data management processes for financial advisory firms by
  + Providing client level access to account data via web portal
  + Facilitating secure and regular communication between client investor and financial advisor regarding investment decisions

## References

IEEE 830 Software Requirements Specification Recommended Practices. 1998. Accessed via https://learn.umuc.edu/d2l/le/content/202339/viewContent/7944885/View

Rational Unified Process Documentation. 1999-2007. Accessed via http://sce.uhcl.edu/helm/rationalunifiedprocess/index.htm

Standard ECMA-262. (2016). Retrieved March 25, 2017, from http://www.ecma-international.org/publications/standards/Ecma-262.htm

User interface mock ups produced utilizing Moqups.com service.

Google Material Design Specification. 2017. Accessed via https://material.io/

# Overall Description

## Product Perspective

PMS is a system designed to intake financial data from multiple institutions. It is intended to perform as an extension of existing financial aggregation solutions. These aggregation solutions will provide transaction and holdings data for the system to import and store. PMS will primarily be used by the Financial Advisory firms of which subscribe to the solution.

## Product Functions

Nonfunctional Requirements

* **Cloud** based Software as a service (**SaaS**) delivery model
* **Cloud** based Platform as a service (**PaaS**) hosted and developed
* **Cross platform** application accessibility via modern web browser
* **Asynchronous** user experience via a modern **API** client and server relationship

Functional Requirements

* Access via web browser
* Access secured area via login
  + Advisory level permissions permit advisor to view all clients
  + Investment Client level permissions permit viewing of owned accounts via Web Portal
* Create client account
* Import transaction data to client account
* Manually enter transaction data to client account
* View past contributions per account
* Graph performance of account holdings value
* Graph projection of account holdings value
* Advisor management of Client Web Portal
  + Messaging
  + Assignment of account data
  + Reset client login credentials
* Investment Client Web Portal access with messaging to advisor

## User Classes and Characteristics

|  |  |  |
| --- | --- | --- |
| **User Class** | **User Role** | **User Characteristics and Available Actions at v1.0** |
| Financial Advisor | Advises, tracks, manages Investment Client holdings data. Provides Investment Client with decision making to increase rate of return on investment holdings. Manages multiple clients. | * Access via web browser * Access secured area via login to view Client Account data * Create client account * Import transaction data to client account * Manually enter transaction data to client account * View past contributions per account * Graph performance of account holdings value * Graph projection of account holdings value * Advisor management of Client Web Portal   + Messaging to client   + Assignment of account data   + Reset client login credentials |
| Investment Client | Hires the Financial Advisor to manage, improve, and report on own investment holdings. May hold investments from multiple institutions. Has a vested interest in maintaining communication with the financial advisor. | * Access via web browser: Secured Client Web Portal   + Messaging to advisor   + Access Client Web Portal via web browser   + View past contributions per account   + Graph performance of account holdings value   + Graph projection of account holdings value |

## Operating Environment

OE 1 – The system shall be accessible via a modern web browser.

OE 2 – The system shall be hosted within a **cloud** Platform As A Service (**PaaS**) environment with a Unix based architecture.

OE 3 – The chosen **PaaS** system should provide scalable system resources to accommodate as many as 1,000 concurrent users.

## Design and Implementation Constraints

CO 1 – The System server shall be hosted in a Platform as a service **cloud** environment with virtual machine support.

CO 2 – The System’s front end design shall conform to the latest European Computer Manufacturers Association (ECMA) Script 6 development paradigm and shall use a modern JavaScript framework.

CO 3 – The system database shall be operated using the MySQL open source database platform.

CO 4 – Imported transaction and holdings related data shall conform to a common format.

## User Documentation

The system shall provide a user documentation area to instruct new and existing users on how to use functionality. This documentation area will be expandable for future development.

UD 1 – The system shall provide interactive tooltips throughout the process of user interaction.

UD 2 – The system shall present terms and conditions of usage during the user registration process.

## Assumptions and Dependencies

External financial data imported to the system will need to conform to a standard format to achieve the goal of providing accurate performance data. If the imported data is incomplete, inaccurate, or unavailable then the end user will not be able to benefit from the software solution. The system will depend on accurate data from external aggregation services. If these services provide incomplete data, the system shall alert the end user to provide more information, if available.

AS 1 – Users of the system shall have a basic understanding of computer and web technologies.

AS 2 – Users of the system shall have a valid and accessible email account for registration and notifications.

AS 3 – Imported financial data will conform to the specified system format.

DE 1 – External aggregation services will provide accurate, timely, and complete financial data to the system on a scheduled basis.

# External Interface Requirements

## User Interfaces

UI 1 – The system shall conform to the Google Material Design standard of user interface design accessible via material.io/guidelines

Google Material design states the following short goal statements:

“Create a visual language that synthesizes classic principles of good design with the innovation and possibility of technology and science.” And “Develop a single underlying system that allows for a unified experience across platforms and device sizes. Mobile precepts are fundamental, but touch, voice, mouse, and keyboard are all ﬁrst-class input methods” (Google, 2017).

*Google Material design is available as a CSS and JS library called Materialize CSS, located via* www.materializecss.com*. This framework provides similar functionality to Bootstrap via www.getbootstrap.com*

UI 2 – The system shall utilize **asynchronous** front end form and user response validation to eliminate any page reloading of the application.

UI 3 – The system shall utilize a responsive layout with a mobile first design concept to ensure compatibility with devices of all sizes.

The following user interface mockups provide a basic overview of layout and input/output design and procedures:

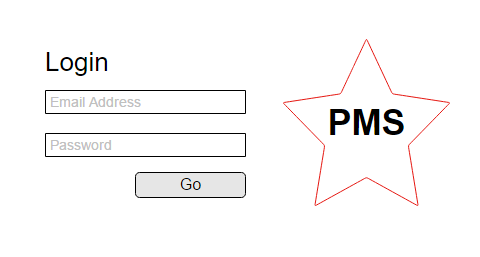
* Login View

Figure 1 Login Screen

* Advisor Views
  + Client Account Dashboard

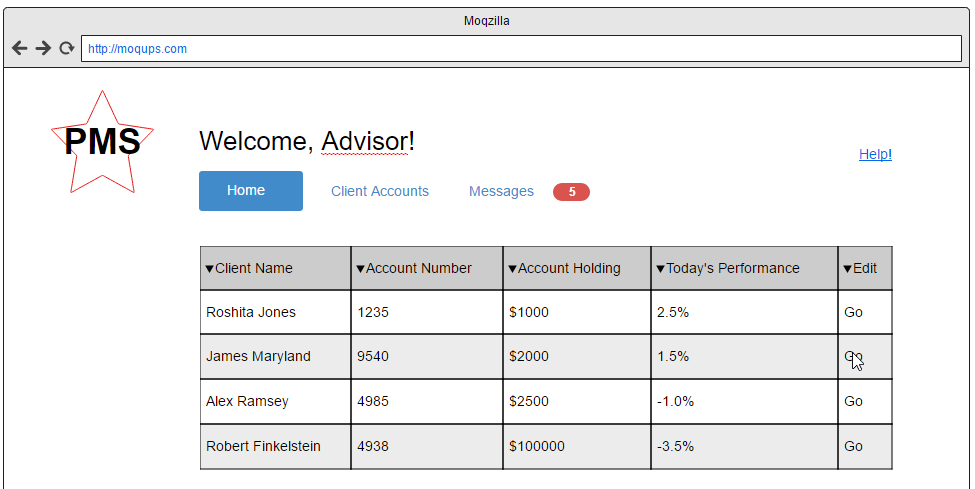


Figure 2 Advisor Dashboard

* + - Help Documentation Link
  + Create Client Account

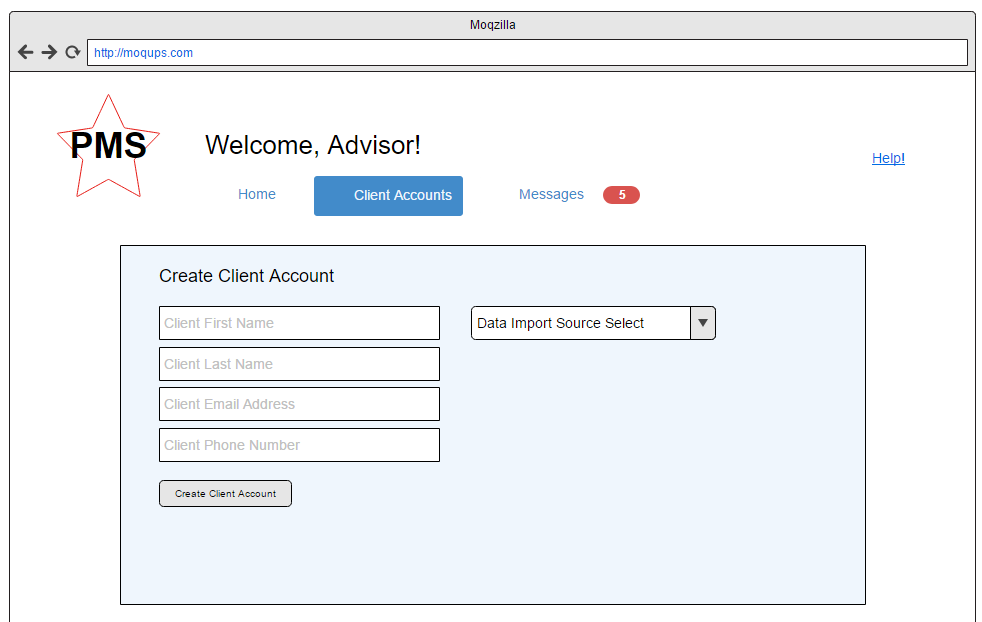


Figure 3 Create Client Account

* + View Client Account

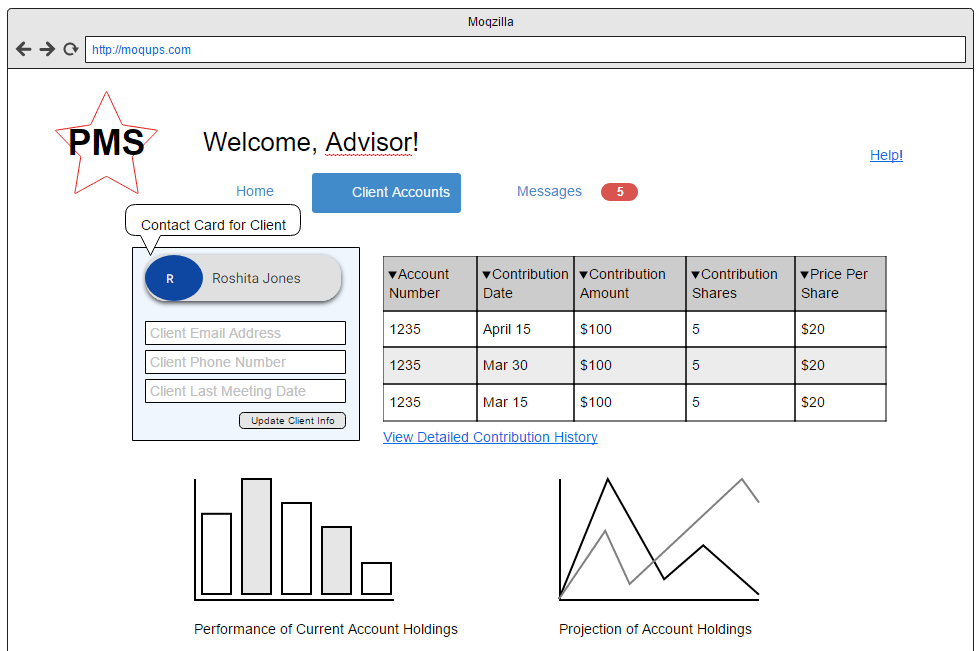


Figure 4 View Client Account

* + - View past contributions per account
    - Graph performance of account holdings value
    - Graph projection of account holdings value
  + Import Transaction Data to Client Account

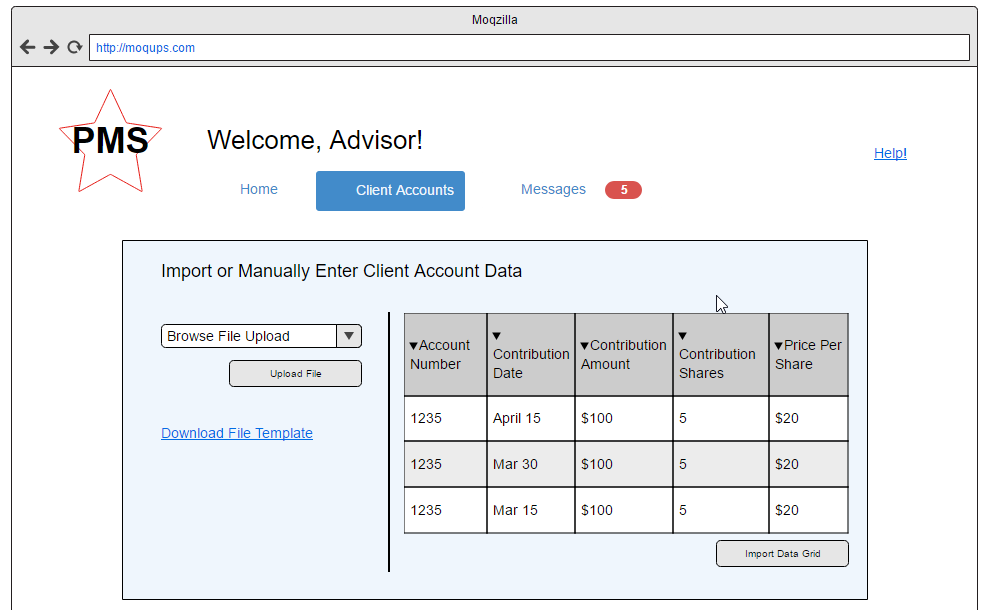


Figure 5 Import or Manually Add Client data via editable Data Grid

* + Client Portal Management via Client account edit view
    - Messaging
    - Assignment of Client Data
    - Reset Client Credentials
* Client Web Portal View
  + Messaging

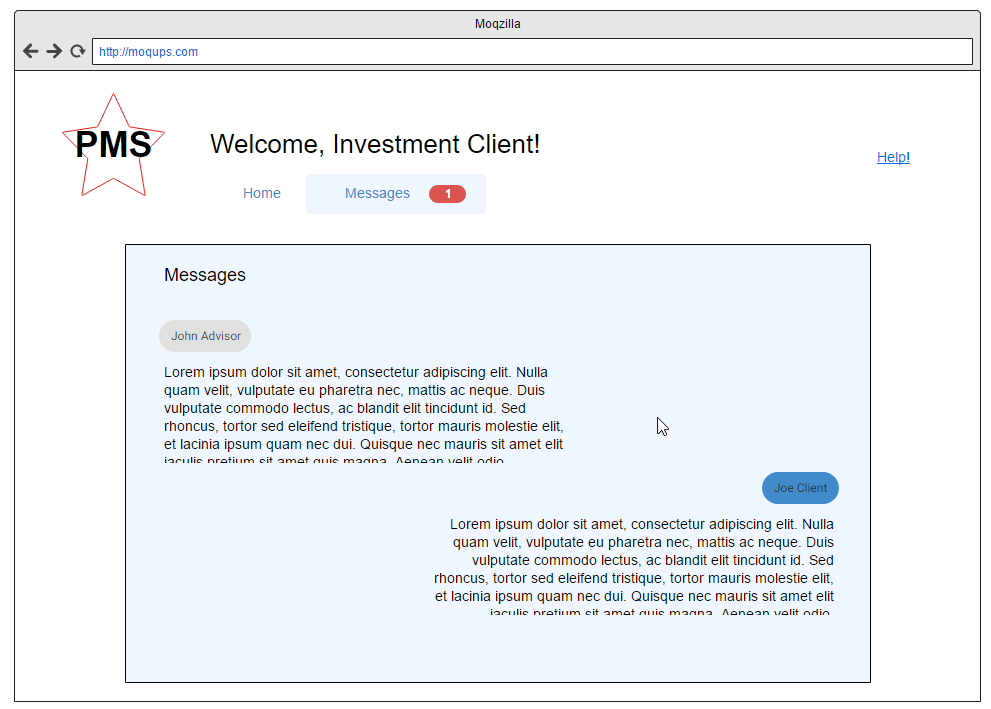


Figure 6 Client and Advisor Messaging View

* + View Client Account
    - View past contributions per account
    - Graph performance of account holdings value
    - Graph projection of account holdings value

## Hardware Interfaces

HI 1 – The system will utilize existing end user platform hardware to establish a network connection link via WiFI or Ethernet connection.

## Software Interfaces

SI 1 – The system will interface with external financial aggregators via automated data imports. These data imports will have failover to a manual data import if nonfunctional.

SI 2 – The system will utilize a front-end web development framework called AngularJS, which provides reliable HTTPS communication between the client side and the server side.

SI 3 – The system will utilize modern web browsers and must maintain compatibility with these browsers to remain functional.

SI 4 – The system will relay data from the server **API** endpoint to a MySQL database.

## Communications Interfaces

CI 1 – The system will utilize modern HTTPS communication via the Secure Sockets Layer protocol to ensure security of data transfer.

# System Features

The following requirements definition section will be divided into two (2) major parts: the first describes the non-functional requirements and their high-level implementation details. The second section provides descriptions of the functional requirements and their itemized definitions following the IEEE 830 recommended specification.

## Non-functional Requirements Definitions

### Cloud based Software as a service (SaaS) delivery model

#### Description and Priority

A Software as a service (**SaaS**) delivery model dictates that the system will host and maintain all aspects of the application for the end user. This allows for the system to maintain consistency over the user experience. This nonfunctional requirement is a high priority for the application and will help the application reach the market in the most efficient and controlled manner. This requirement has a high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User requests the system via URL. | System responds with authentication page hosted and managed by service provider. |
| User requests the system while offline. | System does not receive response and is unreachable by User. |

#### Nonfunctional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| NFR-1 | Access and Registration  The system will provide secure login access via a common URL to the public. |
| NFR-2 | Maintenance and Support  All required maintenance, support, and upgrades of the system will be provided to the advisory services and end client userbase as a paid service. |

### Cloud based Platform as a service (PaaS) hosted and developed

#### Description and Priority

A Platform as a service (**PaaS**) **cloud** model dictates that the system will be developed and hosted within a **cloud** service. This will provide for the system the flexibility, reliability, and speed to market needed to be competitive in the marketplace. A platform as a service **cloud** solution provides all of the development environment and underlying infrastructure required to host an application for the PMS organization ownership. This requirement has a high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| PMS project management team starts service for a new client. | Server instances are provisioned via **PaaS** environment. |
| PMS development team receives new requirements for implementation. | Development of new requirements are generated, tested, and deployed within the **PaaS** environment. |
| PMS support team receives maintenance and operations requests. | Remedy of client issues are performed within a **PaaS** environment. |

#### Nonfunctional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| NFR-3 | **PaaS** Hosted Front End Server  When a user requests the application via public URL, the **PaaS** web server will respond by displaying the front end files for the website including the login screen. |
| NFR-4 | **PaaS** Hosted Back End Server  When a user submits information in the front end, the front end server will communicate with the back end server with the platform environment. |
| NFR-5 | **PaaS** Hosted Database Server  When a user submits information that is to be persisted as directed by the back end server, the data will be stored within the platform database environment. |
| NFR-6 | **PaaS** Hosted Database Backup Solution  Periodic backups will be taken of the back end database server to ensure data integrity as an added service to the client base. |

### Cross platform application accessibility via modern web browser

#### Description and Priority

The system shall be accessible and compatible with modern web browsers which support the ECMA Script 6 or newer standard of web development. This will allow the application to be easily accessible and viewable by users independent of any specific operating system platform. This requirement has a high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User requests the system via URL using a modern web browser a Windows Platform computer. | System responds with authentication page hosted and managed by service provider. |
| User requests the system while offline. | System does not receive response and is unreachable by User. |

#### Nonfunctional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| NFR-7 | System shall be developed to support Modern Web browsers and their standards for development via ECMA Script 6 or greater. |

### Asynchronous user experience via a modern API client and server relationship

#### Description and Priority

The system shall be designed as a Single Page Application (SPA). A Single Page application utilizes an Application Platform Interface (**API**) which responds to user requests in an asynchronous manner. This will require that the system is able to actively engage a client session while listening for further requests, providing responses with data to the user, permitting the user to receive data without refreshing the browser page. This improves the user experience and allows for more flexibility in the user interface interactions. This requirement has a high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User performs valid login. | System initiates user session via encrypted token. System actively listens to HTTP requests from this user. System does not require additional browser page refresh. |
| User performs action which requires persistence of data. | System receives request and issues a response based upon success or failure of data retrieval and storage operation. System does not require additional browser page refresh. |
| User requests View | System responds locally by providing precached view to the user, while requesting updated persisted data from the back end web server. System does not require additional browser page refresh and dynamically loads in newly refresh data from back end server. |

#### Nonfunctional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| NFR-8 | Common Application Platform Interface (**API**)  The system shall be designed around a common **API** where the web server listens via URL endpoint for new user issued requests. These requests may include views, persisted data, encrypted tokens, or permissions. |
| NFR-9 | System issuing of user specific session encrypted tokens  To ensure security of the API endpoint, requests will be preapproved via an issued encrypted web token to the client after successful login. |

## Functional Requirements Definitions

### Access via web browser

#### Description and Priority

The system shall be accessible via a modern web browser. This requirement has a high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User requests the system via web browser at a public URL. | System responds by displaying login page. |

#### Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-1 | System shall provide a front end user interface for common login via publicly accessible URL. |

### Access secured area via login

#### Description and Priority

The system shall be accessible only to registered users with valid credentials. Those who attempt to access the system with invalid credentials shall be rejected from entrance until valid credentials are supplied. This requirement has a high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User requests the login form page. | System responds by displaying login form. |
| User enters invalid credentials into the login form. | System responds by displaying login page with error message, asking for correct credentials. |
| User enters valid credentials into the login form. | System responds by directing the user to their assigned dashboard. |

#### Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-2 | System shall store an encrypted, common repository of valid user credentials. |
| FR-3 | System shall be able to display only information assigned to each user. |
| FR-4 | System shall reject entrance to any user which supplies invalid credentials. |
| FR-5 | System shall provide an encrypted, timed, session specific token to each permitted user after successful login. |

### Advisory level permissions permit advisor to view all clients

#### Description and Priority

The system shall provide user based permissions to user accounts. Advisor level users shall be able to view all clients under their ownership and creation. This requirement has a high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| Advisor level user successfully accesses the system. | System responds by displaying dashboard. System displays only advisor owned client data. |

#### Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-6 | System shall be able to differentiate between advisor users and investment client users. |
| FR-7 | System shall be able to display client accounts assigned to the advisor level account. |

### Investment Client level permissions permit viewing of owned accounts via Web Portal

#### Description and Priority

The system shall provide a web portal for investment client users to check on their owned accounts.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| Investment client user successfully accesses the system. | System responds by displaying client portal dashboard. System displays only investment client owned client account data. |

#### Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-8 | System shall be able to display a client specific dashboard to investment clients. |

### Create Client Account

#### Description and Priority

The system shall provide for advisor level users with the ability to create a new client account. This requirement has a high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| Advisor level user clicks on Create New Client Account button. | System responds by displaying create client account page. |
| Advisor enters form valid information into the Client Account create form and clicks Create. | System persists client account data. System generates login credentials for client account. |

#### Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-9 | System shall provide an input form for advisor users to create a new client account. The input form shall target the system **API** endpoint. System shall validate entered client data based upon the following requirements:   * First Name (required) * Last Name (required) * Email Address (required) * Phone Number (required) * Data Import Selection (selected) |
| FR-10 | System shall persist client account data in a secured database, with personally identifiable information encrypted in the datastore. |

### Import transaction data to client account and/or Manually enter transaction data to client account

#### Description and Priority

The system shall provide for advisor level users with the ability to import financial data via a spreadsheet upload or automated system integration. This requirement has a high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| Advisor selects data upload. | System permits uploading of data view spreadsheet via data entry view. |
| Advisor uploads spreadsheet to client data upload view with valid data. | System responds by importing data to selected client account. |
| Advisor uploads spreadsheet to client data upload view with invalid data. | System responds by rejecting data import, notifying advisor to correct provided data. |
| Advisor selects manual data entry. | System provides data grid view for advisor to manually input data. |
| Advisor selects automatic aggregation in create account view. | System **API** accepts incoming data from permitted aggregation solution. |

#### Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-11 | The system shall provide a view for each client account permitting an advisor to provide client data via spreadsheet file upload or manual data grid entry based upon selection. |
| FR-12 | The system shall validate uploaded spreadsheet data and manually entered data to ensure correctness of transactions based upon the following requirements:   * Account Number (required) * Contribution Date (required) * Contribution Amount (required) * Contribution Shares (required) * Contribution Price per Share (required) |
| FR-13 | The system back end server shall accept incoming **API** PUT requests from known data aggregation services to automatically import transaction data into each client account. |

### View past contributions per account

#### Description and Priority

The system shall provide for advisor level users with the ability to view past contributions for all owned client accounts. The application shall provide for the client investor with the ability to view past contributions for their personally owned accounts. This requirement has a high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| Advisor selects a client account record. | System responds by displaying data grid view of past contributions. |

#### Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-14 | The system shall be able to store past contribution transaction data for each account. |
| FR-15 | The system shall provide a data grid view for advisors to be able to view their client specific data. |
| FR-16 | The system shall provide a data grid view for client accounts to be able to view their personally owned accounts contribution data. |

### Graph performance of account holdings value

#### Description and Priority

The system shall provide dynamic, interactive, graphs which display the contribution amount and performance amount over time. The graphs should display percentage gains of performance for the individual investment holding. The graphs should display percentage gains of performance for comparable market indicators like the S&P 500, Nasdaq, or other stock exchanges. This requirement has a medium-high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User opens client account. | System responds by displaying performance graph of client data with comparable market index. |

#### Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-17 | The system shall display graphical representations of client account performance over a period where data is available. |
| FR-18 | The system shall display comparable performance indexes as an overlay to compare to client account holdings. |
| FR-19 | Displayed graphics shall provide a level of interactivity and dynamic loading effect to engage the user. Such effects include a hover over animation to display labels and a load in effect when first opening the page. Differing colors should be used to differentiate between client account and performance indicators. |

### Graph projection of account holdings value

#### Description and Priority

The system shall provide dynamic, interactive, graphs which display a system estimated projection of performance over time. This projection of performance shall be based upon continued contributions at the current rate. In addition, the system should provide estimated performance gains for the investor based upon increased levels of contribution. This requirement has a medium-high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User opens client account. | System responds by displaying projection graph of client data with selectable increased contribution amount as comparison. |

#### Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-20 | The system shall display graphical representations of client account performance over a period of 5, 10, and 20 years ahead of the current period based upon currently known data. |
| FR-21 | The system shall provide the user with the ability to estimate performance of account holdings in the future based upon increased levels of contribution. |

### Advisor management of Client Web Portal

#### Description and Priority

The system shall provide the advisor with the ability to perform several management functions for client accounts. Client accounts are viewable by the client through the client web portal. Management functions of the client web portal include: 1) Messaging the client, 2) Assigning client accounts with credentials, and 3) Reset client account credentials. All management functions are performed through each Client account edit view. This requirement has a medium-high priority and will be included in the first release of the system.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| Advisor requests client account edit view. | System provides Client web portal management options via the selected client account edit view. |
| Advisor initiates contact with client via messaging solution. | System provides client with message via client specific message inbox in Client web portal. |
| Advisor assigns a client account to client credentials via the client account edit view. | System persists client account assignment |

#### Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-22 | The system shall provide an edit view for client web portal functions through each client account in the advisor section of the webpage. |
| FR-23 | The system shall provide the advisor with the ability to create client credentials. |
| FR-24 | The system shall provide the advisor with the ability to assign client accounts to client credentials. |
| FR-25 | The system shall provide the advisor with the ability to send messages to the client web portal for each client credential. |
| FR-26 | The system shall provide the advisor with the ability to reset client credentials. |
| FR-27 | The system shall provide a client facing view of assigned, personally owned accounts and messages from the advisor. |

# Other Nonfunctional Requirements

## Performance Requirements

PR 1 – The system **API** shall respond to any incoming request within 10ms of receipt.

PR 2 – The system front end shall be designed in such a manner to reduce load times due to framework size, graphics, or other data loading periods.

PR 3 – The system shall provide real time input and updating of all data related elements as new data is provided and persisted.

## Safety Requirements

SR 1 – System issued user session tokens shall persist for no more than 30 minutes from the time of login.

SR 2 – System **API** endpoints shall be hardened and shall only produce responses upon receipt of a valid, active, and known token

SR 3 – System **API** responses shall be encrypted to protect data integrity and to prevent data losses.

## Security Requirements

SER 1 – System issued advisor credentials must be provided only by the software as a service solution provider, via the support and maintenance team of the organization.

## Software Quality Attributes

SQA 1 – The system shall be designed in such a manner as to be extensible; made to be extended upon for future development with ease of addition to existing functionality and infrastructure.

SQA 2 – The system shall be easy to interoperate and share data with external integrations by utilizing interfaces and object oriented design principles.

# Database Requirements and Definitions

## C:\Users\Alex\AppData\Local\Microsoft\Windows\INetCache\Content.Word\SRS for PMS ITEC 630 DFD.JPGData Flow Diagram Level 0

## Data Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Attribute*** | ***Required*** | ***Type*** | ***Field Length*** | ***Default Value*** | ***Notes*** |
| **User** | | | | | |
| User.ID | Yes | Integer | 10 | Null | PK AI |
| User.Username | Yes | String | 30 | Null |  |
| User.Password | Yes | String | 100 | Null | Encrypted |
| User.Email | Yes | String | 30 | Null |  |
| User.Phone | Yes | String | 10 | Null |  |
| User.UserRole.ID | Yes | Integer | 10 | Null | FK UserRole.ID |
| **Account** | | | | | |
| Account.ID | Yes | Integer | 10 | Null | PK AI |
| Account.Advisor.User.ID | Yes | Integer | 10 | Null | FK User.ID |
| Account.Owner.User.ID | Yes | Integer | 10 | Null | FK User.ID |
| **UserRole** | | | | | |
| UserRole.ID | Yes | Integer | 10 | Null | PK AI |
| UserRole.Name | Yes | String | 30 | Null |  |
| **Transaction** | | | | | |
| Trans.ID | Yes | Integer | 10 | Null | PK AI |
| Trans.ContributionDate | Yes | DateTime | 10 | Null |  |
| Trans.ContributionAmount | Yes | Float | 20 | Null |  |
| Trans.ContributionShares | Yes | Integer | 20 | Null |  |
| Trans.ContributionPrice | Yes | Float | 20 | Null |  |
| Trans.Account.ID | Yes | Integer | 10 | Null | FK Account.ID |
| **Message** | | | | | |
| Message.ID | Yes | Integer | 10 | Null | PK AI |
| Message.User.ID | Yes | Integer | 10 | Null | FK User.ID |
| Message.Date | Yes | DateTime | 10 | Null |  |
| Message.Body | Yes | String | 100 | Null |  |
| Message.Recipient.User.ID | Yes | Integer | 10 | Null | FK User.ID |

## C:\Users\Alex\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Entity Relationship Diagram for ITEC 630 PMS.JPGEntity Relationship Diagram

# System Architecture

## C:\Users\Alex\AppData\Local\Microsoft\Windows\INetCache\Content.Word\System Architecture ITEC 630 PMS.JPGHigh Level Architecture Diagram

# Project Plan

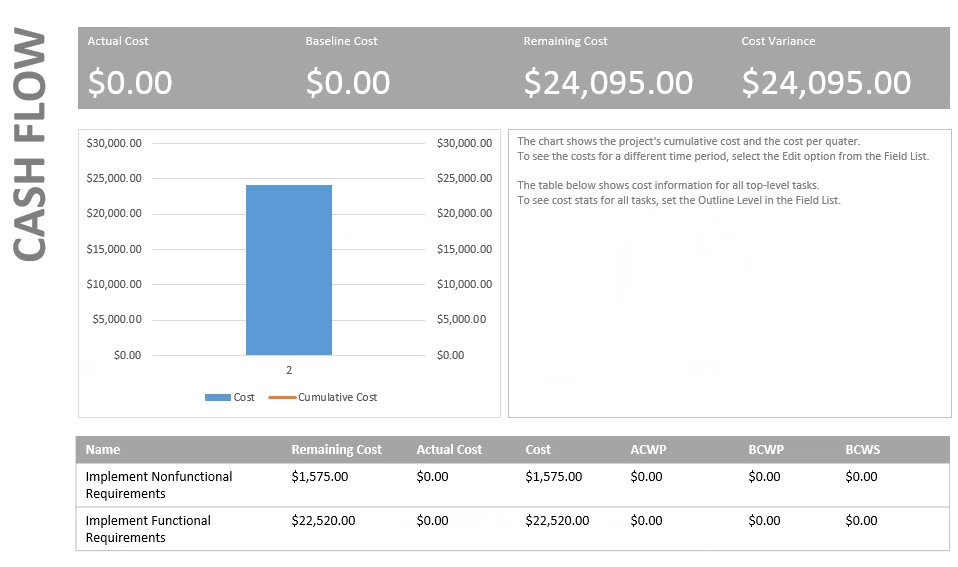
For detailed analysis of the project plan, refer to the Microsoft Project file included. The company is operating on a 5-day work week. The total estimated duration is 28.5 days.

## Task List for Gantt Chart

## Gantt Chart

## Estimated Cost

Cost estimations are based upon a flat employee rate of $45 per hour. The total estimated project cost is $24.095 for the proof of concept release of PMS.



## Estimated Work Hours

# Key Terms

* API – Application Platform Interface, a technical definition by which data is exchanged successfully by following a defined schema and access protocol among front end and back end web technologies.
* Asynchronous – term used to describe current web technologies wherein JavaScript calls are used to point to an API without browser page refreshes, thus improving the user experience.
* Back end – the server side components which listen for front end requests and supply an HTTP response
* Cloud based, Cloud – term used to describe how a web application is hosted and accessible to the public across the broader internet.
* Cross Platform – software technologies which operate seamlessly across multiple device form factors and operating systems.
* Front end – the client side scripting used to develop a user interface interpreted by web   
  browsers installed to an operating system, often communicates to the back end via HTTP protocol requests
* HTTP – Hypertext Transfer Protocol, a communications protocol used in the current World Wide Web ecosystem
* JavaScript – a currently popular, loosely typed, web language and technology ecosystem
* PaaS – Platform as a Service, a cloud delivery model in which infrastructure, development, environment maintenance are provided as a service to the development team.
* SaaS – Software as a Service, a cloud service model in which hosting, development, and support are provided for the end user.