**WEALTHWISE FINANCIAL ADVISING PLATFORM**

John Akujobi, Draix Wyatt, Norman Nguyen, Sawyer Theis

SE 305 – Foundations of Software Engineering

Dr. Zainab Albujasim

Oct 24, 2024

# Table of Contents

[I. Table of Contents 2](#_Toc180705764)

[II. Request of Proposal 5](#_Toc180705765)

[III. Introduction 7](#_Toc180705766)

[a. Purpose of Document 7](#_Toc180705767)

[b. Major Problems and Project Goals 7](#_Toc180705768)

[1. Major Problems 7](#_Toc180705769)

[2. Project Goals 9](#_Toc180705770)

[c. Proposed System Overview 11](#_Toc180705771)

[d. Context model 11](#_Toc180705772)

[1. Payment Gateway APIs 12](#_Toc180705773)

[2. Email Service APIs 13](#_Toc180705774)

[3. Video Conferencing APIs 14](#_Toc180705775)

[4. Financial Data APIs 15](#_Toc180705776)

[5. Authentication and Security APIs 15](#_Toc180705777)

[6. Calendar and Scheduling APIs 16](#_Toc180705778)

[8. APIs for Analytics and Reporting 16](#_Toc180705779)

[e. Definition, Acronyms, and Abbreviations 17](#_Toc180705780)

[f. References 18](#_Toc180705781)

[g. Overview of Document 19](#_Toc180705782)

[IV. General Description 20](#_Toc180705783)

[a. Product perspective 20](#_Toc180705784)

[System interfaces 20](#_Toc180705785)

[User Interfaces 21](#_Toc180705786)

[Hardware Interfaces 21](#_Toc180705787)

[Software Interfaces 21](#_Toc180705788)

[Memory and Storage Constraints 21](#_Toc180705789)

[Operations 22](#_Toc180705790)

[Site Adaptation Requirements 22](#_Toc180705791)

[b. Major functions and Features 22](#_Toc180705792)

[Event Scheduling 22](#_Toc180705793)

[Providing Accessibility to Certified Advisors 23](#_Toc180705794)

[Subscription 23](#_Toc180705795)

[Financial Calculators and Budget Planner 23](#_Toc180705796)

[Learning Hub 24](#_Toc180705797)

[System Security Monitoring 24](#_Toc180705798)

[c. User Characteristics 24](#_Toc180705799)

[d. General Constraints 25](#_Toc180705800)

[e. Assumptions and Dependencies 26](#_Toc180705801)

[Assumptions 26](#_Toc180705802)

[Dependencies 27](#_Toc180705803)

[V. Software requirements 28](#_Toc180705804)

[a. Functional requirements 28](#_Toc180705805)

[1. User Management System 28](#_Toc180705806)

[2. Financial Tools and Calculators 35](#_Toc180705807)

[3. Financial Advisory System 40](#_Toc180705808)

[4. Market News and Financial Education 41](#_Toc180705809)

[5. Event Management System 43](#_Toc180705810)

[6. Payment Processing System 44](#_Toc180705811)

[7. Reporting and Analytics 46](#_Toc180705812)

[b. Non-functional Requirements 47](#_Toc180705813)

[8. Security 47](#_Toc180705814)

[9. Accessibility 49](#_Toc180705815)

[10. Backup 50](#_Toc180705816)

[11. Performance and Scalability 50](#_Toc180705817)

[VI. Log of Meetings, Reviews, and Meetings[sic] 54](#_Toc180705818)

[VII. Change Control 57](#_Toc180705819)

[VIII. Appendix 57](#_Toc180705820)

[a. Definition, Acronyms, and Abbreviations 57](#_Toc180705821)

[b. Work Structure Breakdown 59](#_Toc180705822)

[c. ER Diagram 60](#_Toc180705823)

[d. Data Flow Diagram 61](#_Toc180705824)

[IX. Terms and Conditions 62](#_Toc180705825)

# Request of Proposal

WealthWise Financial Advising is a digital platform designed to help individuals make informed financial decisions through personalized services. The platform provides access to certified financial advisors, a range of financial tools, and the latest market news, allowing users to manage personal wealth and investments efficiently.

The core of our platform revolves around offering personalized financial advice through one-on-one consultations with certified advisors, covering areas such as retirement planning, budgeting, and investment strategies, etc. Users can easily schedule appointments through an integrated booking system that ensures hassle-free experience. In addition to personalized advisory services, WealthWise offers exclusive financial events where users can register and attend seminars, workshops, and webinars led by financial educators.

The platform also features a financial education resource, offering access to market news, stock analysis, and insights into current financial trends. This allows users to stay informed on the latest developments that might affect their investments and financial decisions.

Furthermore, we provide a variety of financial calculators, including those for loan interest, Mortgage calculator, budgeting, and retirement planning insurance coverage, etc. all designed to give users a hands-on approach to their financial management. Clients can input their financial data and receive tailored insights, helping them make smarter financial decisions.

The platform offers a range of subscription plans that give users access to premium features, including personalized financial advice from certified advisors, advanced tools for detailed portfolio analysis, and exclusive content designed to help users achieve their specific financial goals.

We are looking for a software company willing to build this system for us and your company highly recommended. We look forward to working with you on this system that will allow our financial Advising platform to run much more smoothly and efficiently in the future.

# Introduction

## Purpose of Document

This document defines all the requirements for the software we are creating for WealthWise before development. It will serve as a written contract between the client and our team that describes all the software we must implement. The requirements were obtained from the proposal request and various meetings with the client.

## Major Problems and Project Goals

### 1. Major Problems

#### Data Security and Privacy Risks

WealthWise deals with sensitive financial information and plans. This makes it a tempting target for cyberattacks. A data breach will lead to lawsuits, significant financial penalties, identity endangerment, compromised user safety, and loss of trust in our system.

To mitigate this risk, Arqer should enforce strong encryption protocols and comply with data protection regulations like GDPR and PCI DSS (Payment Card Industry Data Security Standard) and should use multi-factor authentication where feasible.

#### Implementation of Video Conferencing

To implement features such as advisor meetings, webinars, and other events, the website requires a video conferencing gateway. This will be external to our system and will be a challenge to implement. Additionally, disruptions to any of these components will cause downtime.

To implement this, Arqer shall implement flexible APIs to third-party video conferencing services (ex. Google Meets, Zoom). Arqer should, where feasible, build redundancy into the system by integrating dependable services via alternative third-party services. Any implemented services and APIs implemented shall be ones of good repute within the given industry.

#### Payment

To implement payment-based subscriptions, the website will require payment gateways. This will be a challenge as said gateways will be outside our system and will carry sensitive data. To implement this, Arqer shall integrate payment gateways, financial data providers, and other APIs into the website and resolve compatibility and dependency issues with said services. services implemented should include PayPal and credit cards (visa, Master Card, etc.).

#### Managing Real-Time Data Updates

Features like market news and stock data require real-time updates and can be resource intensive. Other systems, such as advisor scheduling and event scheduled will also require real-time data syncing.

To create the best user experience and have any data as up to date as possible. Arqer should cache data as necessary and feasible, redirect the users to the actual data source where applicable, and implement database management systems such as the ACID (atomic, consistency, isolation, and durability) model.

### 2. Project Goals

#### Centralize WealthWise’s Services

The primary goal of this website is to bring all WeathWise’s services to one central website. Arqer should bring all listed services (see Software Requirements section) to one central website.

#### Providing Accessibility to Certified Advisors

One major service, WeathWise, provides access to conferences with certified financial advisors. Arqer should allow users to hold video calls with certified financial advisors. Users shall be able to schedule a video meeting with a financial advisor through an event scheduler that allows for meeting scheduling, editing, and cancellation.

#### Provide Advanced Financial Tools (Calculators)

To allow users to make sound financial decisions, Arqer should implement a variety of calculators including a loan interest calculator, a mortgage calculator, a budgeting tool (by subscription), a retirement planning calculator, and an insurance coverage calculator.

#### Education and News Resources

To allow users to stay up to date on financial education and news, Arqer shall implement a learning hub with resources such as news articles, financial tutorials, webinars, and other educational resources as necessary. External resources shall be accessed via relevant APIs and gateways.

#### Secure Payment

To allow system monetization and subscriptions, Arqer shall implement systems to allow the user to make payments to WealthWise. Said payments shall be made secure via reputable third-party services such as PayPal and its 128-bit SSL protocol.

#### System Security Monitoring

Site administrators should be able to monitor the system and log events. Significant system events (logins, data transfers, API requests) shall be logged. An intruder detection system shall be used to identify unauthorized access to or tampering with the system.

#### Intuitive UI/UX

The website should be designed such that the target user shall be able to navigate to every feature and tool without significant delay or loss of user experience. This shall be achieved through intuitive web design that clearly indicates where tools are located throughout the site.

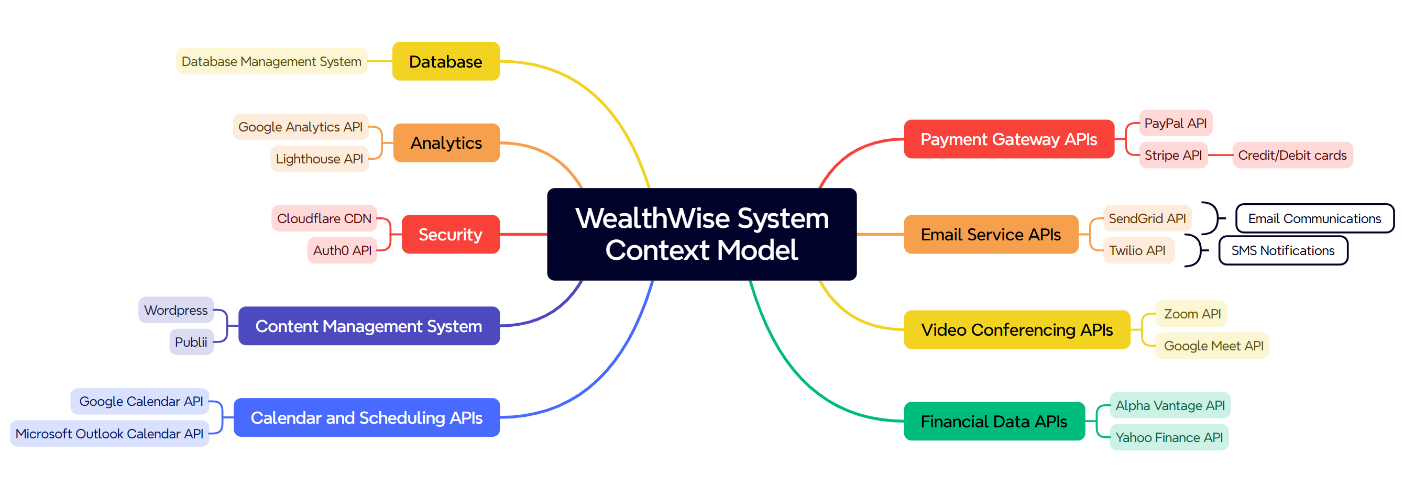
## Proposed System Overview

A chart with text on it

Description automatically generated

Figure 1. Proposed system overview

## Context model

Figure 2. Context model for the system.

### 1. Payment Gateway APIs

Payment gateways securely manage all financial transactions, from subscription purchases to event ticket sales and consultation fees.

**a. PayPal API**

This will let users pay with PayPal accounts and credit or debit cards. PayPal’s REST API will be used for initiating transactions, subscriptions management, refund processing, and transaction history. The transaction communications shall be properly secured using TLS encryption. Listen for status updates related to the payments in the form of webhooks provided by the PayPal integration for events like payment successful, failed, and refunded. The integration shall, in substance, follow the guidelines on PCI DSS by PayPal for secure transactions.

**b. Stripe API**

Stripe is to process subscriptions and event payments for flexibility in various payment types, including credit cards and ACH transfers. Used for dynamically building the payment form, securely capturing credit card information. Stripe tokens to represent details of the card, not actual details of it, so no sensitive information is passed around.

Payment processing, refunds, subscription setup. Stripe webhooks will report the platform about transaction statuses, subscription renewals, and failures of payments.

### 2. Email Service APIs

These APIs shall handle automated email communications for all account verifications, notifications, reminders, and marketing purposes.

**a. SendGrid API**

SendGrid shall be employed for the delivery of all transactional emails, which include verification of account, password reset, event reminders, notifications.

The platform will use SMTP Relay provided by SendGrid for sending emails and the Web API for more customized sending of emails-for example, bulk sends. The template setup via SendGrid will maintain a consistent look for verification emails, password reset emails, notification emails, etc SendGrid’s analytics API will be used to track delivery, opens, clicks, and errors for the emails.

**b. Twilio API - For SMS Notifications**

Twilio will be used to send out SMS notifications for critical alerts, such as appointment reminders or urgent security updates.

This will use Twilio’s RESTful API to send, receive, and track delivery statuses of SMSs. This allows automatic sending of SMS based on users’ preferences. Twilio can be used to introduce SMS-based 2FA.

### 3. Video Conferencing APIs

These APIs will allow virtual consultations between clients and their respective advisors to take place, thus enabling remote financial advisory services.

**a. Zoom API**

The application will use Zoom for secure video calls between users and advisors for virtual consultations. OAuth 2.0 will be used for authentication of Zoom accounts on the platform for scheduling meetings. The Meeting API will be responsible for exposing create, manage, and delete functionality from inside the platform for virtual meetings.

The Zoom Embedded SDK can be employed to offer in-browser video calls so that the users may not have to leave the comfort of the platform. Webhooks will notify Zoom on events like the start of a meeting and the end of a meeting. It provides a well-known, reliable, and secure virtual meeting solution that is necessary for remote financial consultations.

**b. Google Meet API**

It will be an alternative for users whose preferred video conferencing platform is Google Meet. OAuth 2.0 Authentication: OAuth will be used for the authorization of Google accounts and to manage meetings.

Google Meet will be integrated with the Google Calendar API to automatically add meetings to calendars and invite others. This provides functionality for creating, managing, and deleting meetings right within the platform.

### 4. Financial Data APIs

These APIs will deliver real-time financial information to the **Learning Hub**, calculators, and market news updates.

**a. Alpha Vantage API**

Alpha Vantage offers real-time and historical market data - stock prices, forex, and economic indicators. The application will make a call to the Alpha Vantage’s API and fetch financial data for real-time stock tickers, charts, and investment analysis tools. Rate limiting is used to manage API call limits and ensure stable performance.

b. **Yahoo Finance API**

The API will be utilized to retrieve the quotations of stocks, financials of companies, and trends of industries for **Learning Hub**. This API will be used to provide fresh data on stock, charts, and financial news to keep the users well informed. Integration with Alpha Vantage and data from Yahoo Finance will provide wider insights and comparisons.

### 5. Authentication and Security APIs

These APIs will manage authentication for users, access controls, and data security.

**a. Auth0 API**

It will be used by Auth0 for user authentication: secure login, registration, and session management. For single sign-on capability, user authentication is taken care of by Auth0 using these protocols. MFA can also be configured in Auth0 to allow for an extra layer of security. Auth0’s API will be used for managing user roles, permissions, and access controls within the system.

### 6. Calendar and Scheduling APIs

These are APIs that provide appointment management, event management, and user scheduling.

**a. Google Calendar API**

It will provide users and the advisors with the capability of creating, editing, and managing appointments and events within the platform. The events should be created, edited, and deleted while automatically synchronizing with the Google Calendars of users. OAuth authentication allows the users to securely grant access to their calendars. Improvement the user experience is brought about by it, thereby ensuring smooth event and appointment management.

**b. Microsoft Outlook Calendar API**

This API shall provide to Outlook users the same functionality for scheduling, as given to Google Calendar users. Sync with the users’ Outlook Calendars and thereby manage appointments and events. This user authentication should be done in a secure way so that they may provide access to their calendar.

### 8. APIs for Analytics and Reporting

These represent data visualization, reporting, and analytics about the users.

**a. Google Analytics API**

Google Analytics will be used in keeping track of user interactions, feature usage, as well as event attendance that shall help admins understand how users behave. It needs to be able to track user engagement metrics, conversion rates, and session data. The admin dashboard will have insights into user behavior and platform performance with the help of Google Analytics data.

## Definition, Acronyms, and Abbreviations

* + - * ACID: A database model designed for data synchronicity standing for atomic, consistency, isolation, and durability
      * ADA: Americans with Disabilities Act
      * AES: Advanced Encryption Standard
      * API: Application Programming Interface
      * ARIA: Accessible Rich Internet Applications
      * AWS: Amazon Web Services
      * DFD: Data Flow Diagram
      * GDPR: General Data Protection Regulation
      * IDS: Intruder Detection System
      * IP: Internet Protocol
      * JWS: JSON Web Tokens
      * PCI DSS: Payment Card Industry Data Security Standard
      * RD: Requirement Document
      * SFTP: Secure File Transfer Protocol
      * SQL: Structured Query Language
      * SSL: Internet communication security protocol: Secure Sockets Layer
      * REST: An architecture for distributed systems (REpresentational State Transfer)
      * TLS: Internet communication security protocol: Transport Layer Security
      * UI: User Interface
      * UX: User Experience
      * WCAG: Web Content Accessibility Guidelines
      * Web app: a website with application functionality
      * WSB: Work Structure Breakdown

## References

* + - * SendGrid Website: https://sendgrid.com/en-us

This website is the home page of SendGrid, our proposed emailing API our site will use to email users for various purposes.

* + - * WS Website: https://sendgrid.com/en-us

This website is the home page of JWS, our proposed system to manage web tokens.

* + - * Sharding reference: https://www.geeksforgeeks.org/database-sharding-a-system-design-concept/

This site overviews the sharding methodology we have proposed for our database.

* + - * AWS Website: https://aws.amazon.com/

This website is the home page of AWS, our proposed system for cloud infrastructure.

* + - * PCI DSS Website: https://www.pcisecuritystandards.org/

This website is the home page of PCI DSS, our proposed standard for payment security.

* + - * Security Onion Website: https://www.pcisecuritystandards.org/

This website is the home page of Security Onion, our proposed IDS.

* + - * Nessus Website: https://www.tenable.com/products/nessus

This website is the home page of Security Onion, our proposed vulnerability scanner.

## Overview of Document

This document provides the client with detailed information about every feature, component, and process related to the system. First, we will cover the general description of the system, including the product perspective, key functions and features, user characteristics, and general constraints such as design and process limitations, along with assumptions, dependencies, and modifiability.

Next, we will outline the software requirements, identifying the functional requirements like input/output, networking, and configuration, as well as non-functional requirements such as exception handling, testing, backup, and security. We will also briefly describe the design constraints to ensure the system aligns with the brand’s colors and preferences.

After detailing the system requirements, we will review meeting logs, meeting policies, and the review process. Following that, we will present the change control policy, which outlines the procedures for handling change requests. Finally, the document will conclude with an appendix and the terms and conditions.

# General Description

## Product perspective

The new digital platform, called WealthWise Financial Advising, unifies financial services of all kinds onto one platform, allowing non-professionals to manage their private wealth and investments more effectively. It will thus combine personalized financial advisory services, state-of-the-art financial tools, educational resources, event management, and secure payment processing in a single, user-friendly app.

### System interfaces

The system will require interfaces for payment gateways, financial data providers, video conferencing, and email notifications.

Payment Gateways APIs shall be used for processing the payments via Stripe (credit/debit cards), PayPal. Financial Data Providers API shall extend to news services, stock platforms, and education resource sites to receive current market news in real-time, current stock data, financial trends, and education resources such as videos, tutorials, and courses.

Video conferencing tool APIs shall extend to Google Meets, and Zoom as applicable to facilitate engaging with consultations, virtually, between the clients and their advisors. Email and Notification Services APIs shall extend to external sources to inform users of critical information as relevant (e.g. scheduled events, meeting starting soon). These shall act as a means of communication and notification. Services implemented should include SendGrid and/or Twilio.

### User Interfaces

The web application shall be accessible through the website using desktop browsers Chrome, Firefox, Edge, and Safari and be accessible through both Android and iPhones. The Web app shall adjust its size to accommodate differing screen sizes and shall follow all web accessibility norms to include color contrast and keyboard navigation.

### Hardware Interfaces

Server-side operations shall be able to be hosted on scalable cloud services, such as AWS or Azure, and implemented such that such horizontal scalability is possible. Client-side operations shall be accessible independently of hardware. Both Desktop and mobile access should be implemented.

### Software Interfaces

Server-side operations should be operating system-independent and require Python, SQL, and HTML support. Client-side operations should be accessible via Chrome, Firefox, Edge, or Safari browsers and be accessible through both Android and iPhone devices. Interaction between the client and server shall take place via RESTful APIs. Future enhancements may be integral with other third parties.

### Memory and Storage Constraints

Server-side resources shall be sufficient to store all relevant user information, backup, and application logs. This should be sufficient for the proposed peak user load of 25,000 concurrent peak users. Client-side resources should be extremely limited. Storage shall be allocated for storing cookies and local caching, to increase performance.

### Operations

The web app shall have three-nines of availability, or 99.9% availability (uptime) with less than or equal to .1% downtime. Scheduled maintenance shall take place during non-peak hours with prior user notification.

### Site Adaptation Requirements

The web app should provide translation into various languages and different regional settings, as needed.

## Major functions and Features

### Event Scheduling

Users and site administrators shall be able to schedule events via a planning calendar. This will allow the creation and scheduling of events. Furthermore, the creator of the event or the site administrators shall be able to modify, reschedule, or delete the event. Users shall be able to register for the event ahead of time to allow event participation.

### Providing Accessibility to Certified Advisors

The web app should allow users and advisors to hold video meetings following an appointment scheduling. Users shall be able to schedule a video meeting with a financial advisor through an event scheduler that allows for meeting scheduling, editing, and cancellation.

### Subscription

The web app shall have a free and subscription tier. The free tier will allow access to all website features, except for the budget planning feature. Users shall be able to make payments to WealthWise to gain subscription tier and access to the budget planning feature.

### Financial Calculators and Budget Planner

The web app shall include a loan interest calculator, a mortgage calculator, a retirement planning calculator, student loan, car loan, and an insurance coverage calculator.

The budget planner shall allow the user to input income sources and amounts, fixed expenses, variable expenses, and savings goals and return a comprehensive budget overview, breaking down percentages of income sources and total percent allocation to expenses; spending analysis, which will break down of expenses by percentage; and recommendations for savings, which suggests categories that the user may be overspending on by comparison against percentages determined by the site’s certified financial advisors.

### Learning Hub

The learning hub shall feature market news and stock data in real time. The learning hub shall also provide users with resources such as news articles, financial tutorials, webinars, and other educational resources as desired.

### System Security Monitoring

The web app should have sufficient cyber security incident monitoring. Significant system events (logins, data transfers, API requests) shall be logged. An intruder detection system, such as Security Onion, should be used to identify unauthorized access to or tamper with the system.

## User Characteristics

General users are people seeking financial advice. They can have a wide range of interests and technical proficiencies, from young professionals to seniors seeking retirement advice. Software should be intuitive for users since many may have novice technical expertise. Since these users will voluntarily use the software, it should focus on acceptability.

Financial advisors are another type of user who interacts with the software. These users can be assumed to have a higher degree of technical knowledge and some training in the software. They shall use the software repeatedly. We should focus on appealing to efficiency for these users, so it interrupts their work as little as possible.

## General Constraints

* + - * Server-side architecture shall support Python, SQL, and HTML, as they will be the primary languages used to build the system architecture. Servers should also be run on scalable hardware or user scalable cloud computing solutions such as AWS.
      * Integration: All features (learning hub, event planner, advisor meetings, etc.) should be integrated on the same website and allow the user to move between each.
      * Security: any sensitive data shall be end-to-end encrypted in transport using TLS or SSL as applicable and encrypted while at rest using 256-bit AES.
      * Performance: The website should not excessively waste system resources and feel responsive to the user having a maximum response time of 10 seconds.
      * Dependability: The website shall remain primarily in a state of up-time during its service period with 3 nines of availability (99.9%).
      * Compatibility: The website should work on current versions of Chrome, Firefox, Edge, and Safari and be accessible on iOS and Android.
      * Privacy: Data collected by Arqer should always remain within company control and not be shared with outside parties.
      * Availability: The website should remain available under a peak user load of 25,000 concurrent users.
      * Data storage: Data shall be stored in compliance with relevant laws and regulations, including GBPR and laws and regulations of the host country and region.
      * Localization: the website shall be accessible to all targeted locations and regions.
      * User Experience: Control of the website should be intuitive to the target user, allowing seamless navigation without degradation of user experience.

## Assumptions and Dependencies

### Assumptions

#### System assumptions:

The system should run on a server to provide services to the customer. Users can use their up-to-date web browser (Chrome, Edge, Firefox, or Safari) to connect to the server and perform actions on their devices. The system should be able to manage 25,000 simultaneously without issue. It should also perform flawlessly, efficiently, securely, and respond timely. The system should maintain an average response time of at most 10 seconds. Additionally, maintain high availability.

The system should handle most services, use APIs to fetch information from third-party or external service providers, and be able to store all encrypted data in a database. It should also be able to frequently back up the system without interrupting services for longer than 2 hours during non-peak hours. They also should include error correction codes via single-bit parity to maintain data integrity; and protection against bypasses such as injections or database bugs.

#### User assumptions:

There shall be three user types, customers, advisors, and system administrators. Each user group shall have a different system view based on their access level/permission and role. Customers who have subscriptions shall be able to use all the features. Advisors shall not be able to modify the system information/data directly as the system administrators. System administrators shall have sufficient permission to modify the system directly. Additionally, they shall be able to control every component in the system.

### Dependencies

* The website should be dependent on video hosting services (Google Meets and Zoom), APIs (news, education resources), and databases (PostgreSQL).
* The server system shall support Python, HTML, and SQL.
* The account management system should be dependent on the database system and payment process of third-party transaction services (PayPal, Stripe).
* System administration should depend on all other components to monitor the system (the monitoring system requires a system to monitor).

# Software requirements

## Functional requirements

### User Management System

The User Management System will be the backbone of the WealthWise website including user account creation, log in, profile management, and interaction with other components.

1. **User Registration**
2. Display of Form:

Users shall be able to create an account by clicking the "register" button on the login page, entering an email that will double as a username, and choosing a password. Fields allowing entering information such as financial goals shall be optional for clients.

Registration page should display a clean and user-friendly form with an input field for email, password, and basic user information including a phone number. - User shall be able to create an account by clicking the "register" button on the login page, entering an email that will double as a username, and choosing a password.

1. Data Validation:

The data from this interaction shall be encrypted using TLS to ensure the password remains secure. The website shall perform front end validation to include ensuring the email is syntactically valid (could feasibly be a real email address), the password meets complex requirements (see below), and all required fields are filled. The system shall the entered email is not a duplicate and display an error message when a user tries to create an account with a duplicate email. Passwords shall be at least 12 characters in length. They must have at least one capital letter, one lowercase letter, one number, and a special character. Stored passwords will be salted to protect against brute-force attacks.

1. Email verification:

Upon entering in and submitting all information, the system shall send a confirmation email to the provided email address using SendGrid. This confirmation email shall contain time-sensitive link. This link shall allow the user to validate that they own the email they entered. If the users fail to verify their account within 24 hours, their account information shall be deleted, and their account shall not be created. Emails containing verification links shall be encrypted using TLS to mitigate security risks.

#### **User login**

* + 1. Requirements

The user shall be able to access their account using the email and password they created the account with. The user shall enter their credentials on the login page and click login. The password shall be encrypted using TLS to ensure it remains secure. The user shall be authenticated and allowed to access the website if the credentials match. Should the user fail to log in 3 times, their account shall be locked out for 3 minutes. Once authenticated, the system should establish JWT with the client for easy authentication. This effectively allows “remember me.”

1. **Password Recovery**

1.3.1 Password Reset Request

If the user wishes to reset their password, whether it is forgotten or not, they shall be able to do so by clicking "reset password". SendGrid shall send a password reset link that expires in 10 minutes. Upon clicking the link, the user may then choose a new password using the same requirements seen in 1.1.

1.3.2 Token Validation

When the user clicks on the reset link, the system shall first verify the token has not expired. If it is still valid, the user shall be requested to create a new password using the same requirements seen in 1.1. When the password is reset, the system shall log the user out of all running sessions. Users will then need to log in again with their new credentials to access the site.

1. **Profile Management**
2. Profile Viewing and Editing

A logged in user shall be able to edit their personal information from their profile including their name, phone number, address, email, profile picture, financial goals, and privacy settings. For advisors, the system shall display ratings from previous clients.

The system shall verify the validity of address fields. Only letters shall appear in the city, state, county, etc. Phone numbers shall be in the format XXX-XX-XXX, and characters shall appear in the name field. If the account type is an advisor, the system shall allow the advisor to upload certification/licenses through SFTP, allow editing bio and area(s) of expertise.

1. Advisor-Specific Fields

The Advisors shall be able to edit additional fields to include certifications, licenses, bio, and areas of specialization. The system shall allow secure file uploads via SFTP for certifications.

1. Privacy Settings

Users shall be able to choose what information other users can see to include financial goals and contact information. Should the user keep any information private, only administrators shall be able to view the mentioned information.

1. **Subscription Management**
2. Subscription Status Viewing

The user shall be able to view subscription information from their profile including whether they have an active subscription, the date until subscription expiration, a summary of their plan, benefits, billing cycle, expiration date, and renewal status.

1. Transaction History

The user shall be able to view a complete history of past transactions, including the date of payment, amount paid, and method of payment. The downloadable receipts included in every transaction entry shall be available for personal record-keeping by the users or for expense tracking.

1. Subscription Modification

The user shall be able to view all subscription packages, pricing options, and their respective features. The user shall be able to cancel a subscription, renew an existing subscription, or buy a subscription. As of now, the subscription options include choosing between a monthly pricing of $10 per month or yearly pricing at $100 per year. With subscription, users get access to the budget planning tool and certain learning tools. Once the subscription is made, the user shall get immediate access to those features.

1. Payment Management

The user shall be able to subscribe by using either PayPal or Stripe (credit/debit card). Data shall be secured based on the vendor's security options: for example, PayPal uses 128-bit SSL.

Payment history collected shall include amount, date, and relevant payment information-such as credit card information or PayPal information and shall be recorded in the database.

The user shall be able to activate auto-renewal via a check box on the menu. In this case, the website would have permission to access stored payment information automatically at any point in time to purchase subscriptions. This shall be able to be turned off by de-selecting the same auto-renew box.

1. Payment Security

In addition to vendor specific security standards, transactions shall be secured as per the PCI-DSS standards.

1. **Interaction History**
2. Consultation History

The date and time, name of advisor, and rating of an advisor shall be stored in the database after a meeting for future reference in the profile of the user. The customer shall be able to rate the advisor on a scale of 1-10, which shall be visible in the advisor's profile and the particular user's profile which shall be visible only to the user or administrators. Any files the counselor creates and uploads via SFTP to the server shall be downloadable from the consultation history entry in the customer profile.

1. Event Participation History

The date, time, and name of the event shall remain in the database after the event attendance and will be displayed for later reference in the user's profile. Any documents posted by an event organizer via SFTP shall be downloadable from the event history entry in the user profile.

1. **Advisor Management**
2. Advisor Management System

The system shall allow advisors to view previous client interaction, including dates, name of client, and client rating. The system should allow the Advisor to block dates and times they will not be available for consultation.

Advisors shall be able to access the calendar (Google calendar or Outlook) which allows the advisors to select time slots when they are available for consultation. They can "Available", "Busy,” or "Out of Office" at a time with date and time. They shall be able to set up reoccurring availability slots (e.g., Every Monday from 10:00 AM to 2:00 PM). Any change in availability, like adding or removing slots, shall be replicated in real time to the client-side booking interface for the most up-to-date scheduling options.

1. Advisor Account Creation

In addition to the user account requirements, account creation for advisors shall only be available via an administrator. An administrator must approve an admin account before it is integration into the system. Once the necessary information is submitted, an advisor's profile is automatically set to "Pending Approval," wherein the admins verify the credentials provided.

1. Administrator Verification

In the dashboard, administrators shall have access to all advisor applications. Administrators shall be able to approve or reject an advisor account. After approval by the admin, the account shall be allowed to be created.

Applications rejected shall automatically trigger an email including the reasons for rejection, along with further action required.

1. **Messaging and Notifications**
2. In-app messaging

The system shall allow users to securely message other users and advisors within the website. Chat shall allow sharing of files and text. The messaging interface shall allow the user to send and receive messages in real-time, showing the typing indicator, reading receipt, and timestamp of messages.

1. Message Encryption

Messages shall be end-to-end encrypted to protect users' privacy using TLS. Messages shall be saved for 3 months before they are deleted automatically. Users shall be able to archive or delete conversations.

1. Notifications and Alerts

Events to be notified include appointment reminders, new messages, renewals, and upcoming events. Notifications come via in-app, email, or SMS; the user prefers whichever. In this respect, different options provide a guarantee that no user will miss any updates, since at least one of the means of communication will be preferred.

1. Message settings

The system shall allow users to personalize notification settings to include or exclude certain alerts. For instance, users may want to receive consultation reminders and not marketing emails.

As events approach, the system shall send reminders via SendGrid (email) reminding the user of the date and time the event will be held, whether virtually or physically.

### 2. Financial Tools and Calculators

The various financial calculators allow the user access to use tools to have informed financial decisions.

1. **Loan Interest Calculator**

The loan calculator is an easy way for users to become educated about what their loan will cost over time.

1. User input

User shall be able to provide a loan amount, interest rate, loan term in years or months, and frequency of payments. The system shall only take numeric values and shall require the user to fill out all fields before it begins calculation.

1. Calculation

The system shall utilize the input values to calculate the monthly payment, total interest paid, and amortization schedule. The amortization schedule shall also break down each payment into principal and interest, it would allow users to see how this repayment structure changes over time.

1. **Mortgage Calculator**

The Mortgage Calculator allows the user to calculate how much it is going to cost every month to buy a home.

1. User Input

The user shall be able to input home price, down payment amount or in percent, interest rate, and loan term in years. The system shall check for input validity (numbers only) and offers a choice between fixed-rate or adjustable-rate mortgages.

1. Calculations

The system shall calculate the monthly mortgage payment, total interest paid, and total cost of the loan, with consideration of a down payment.

The system shall also allow users to view how changing the down payment alters the monthly payment and total amount paid via an input value that will recalculate the total cost of the mortgage.

1. Detailed Outputs

The system shall display a payment schedule that itemizes the payment amount between principal and interest per payment.

The system shall, upon user request, show comparisons of how altering the percent down payment or interest rate for affects the users per monthly payment.

1. **Budgeting Tool**

The Budgeting Tool will allow the user to have full insight into his or her financial standing to allocate income accordingly and denote areas of possible savings.

1. User Input

Users shall be able to fill in their various sources of income, fixed and varied, their expenses fixed and varied expenses, and savings goals.

1. Calculation Process

The system shall categorize both income and expenses. The system shall calculate the percentage of income spent in each expense category and identify categories in which the user may be over-spending based on data values set by the site’s advisors.

Increase Savings Recommendations - Based on user input of cost saving goals, the system shall make recommendations to improve savings using benchmarks provided by advisors. For example, no more than 30% of income on housing.

1. **Retirement Planning Calculator**

The Retirement Planning Calculator forecasts the user's requirement for retirement and gives actionable insights based on the goals of personal savings.

1. User Input

The system shall allow the user to enter their present age, expected retirement age, present savings, monthly contributions, and the expected rate of return.

1. Calculation Process

The system shall compute total savings at retirement time, with compound interest factored in for the period under consideration.

1. Recommendations

The system should calculate that the users are not on target to achieve their retirement goals, the system shall recommend increasing monthly contributions or adjusting their retirement age.

1. **Insurance Coverage Calculator**

The Insurance Coverage Calculator enables users to analyze their insurance requirements and possible shortcomings in the prevailing coverage.

1. User Input

The system shall allow the user entered: age, number of dependents, current insurance coverage, and any other risks the user would want to cover.

1. Output Presentation

The system shall calculate the amount of coverage required for various types of insurance, including, life, health, and disability. The system shall also calculate gaps between the current and recommended coverage and suggest purchasing additional policies where required. The results should show recommended policy amounts, potential providers, and estimated premiums.

1. **Student Loan**

The student loan calculator functions to estimate monthly payments, total interest, and the overall cost of a student loan.

1. User input

The user shall be able to provide details such as loan amount, interest rate, and repayment term.

1. Output

The system shall provide an estimate of monthly payments, total interest, and overall cost, and allows for comparisons between different repayment plans.

Additionally, the calculator should factor in early repayment options to show how making extra payments can reduce the total interest and shorten the loan term.

1. **Car Payment**

The car payment calculator allows users to estimate their monthly car payments, help users understand the total cost of the loan, including interest, and provide a clear picture of what their monthly payments will be.

1. User input

The system shall allow the user to enter their loan amount, interest rate, loan term, and down payment.

1. Output

The system shall show monthly payment, total interest paid, total loan cost, amortization schedule, payoff date, and impact of extra payments.

It shall compare different loan options and adjusting factors like loan length or down payment to find a manageable payment plan.

### 3. Financial Advisory System

1. **Advisory Booking System**

An Advisory Booking System shall allow searching for advisors by availability and expertise and book consultations accordingly.

1. Search and Filter Options

Users shall be able to filter the advisors on specializations, including retirement planning, investment strategy; ratings; availability; language; and cost per session.

The results shall be sorted dynamically based on user preferences and advisor ratings, always reflecting the best matches for each user's needs.

1. Booking Process

Once the selection of an advisor is made, it shall provide users with the facility to look at available time slots and choose when best for them. The user shall be able to select a time slot from the implemented calendar system (Google Calendar or Outlook).

The appointment cost shall be clearly visible here, along with a cancellation policy and a brief on what one should expect during the session.

1. Confirmation and Reminders

Automated emails at the time of booking a slot shall be generated for both user and the advisor.

Appointment details along with the time shall be shared via e-mail. Automatic reminders via email or phone shall be sent 24 hours and 1 hour before consultations avoid no-shows.

1. **Consultation Sessions**

The consultation sessions may be in-person or virtual. Whichever mode of consultation is available depends upon the preference of the user and on the availability of the advisor.

1. In-person Consultations

The locations and directions shall be provided to the users in advance via Email (SendGrid) for sessions. The user check-in component shall also be available through the mobile application.

1. Virtual consultations

Links provided for virtual consultation are secured via TLS and only visible to the particular user and advisor. Virtual sessions are encrypted via TLS for security and privacy; screen sharing, document sharing, or chart sharing shall be integrated during sessions.

### 4. Market News and Financial Education

The web application keeps the user updated with relevant financial news and a knowledge base for teaching aids to keep them informed about making financially correct decisions.

1. **Market News**
2. Content Aggregation

The user shall be able to view news from various top sources, including financial news portals, stock exchanges, and leading economic indicators.

The system shall allow users to customize the news feed about desired topics: stock performance, to bond yields, new trends, and macroeconomics.

1. Real-time Updates

News feeds should be delivered in real-time so that users gain access to the most relevant current market information and analysis.

1. **Learning Tools**

Learning tools introduce users to financial acumen and provide a fundamental tool for their further development.

1. Type of Contents

Resources shall be sorted into budgeting, investing, retirement planning, and risk management categories. Each of these categories shall have articles to further an individual in knowledge, video tutorials, webinars, and interactive guides.

1. Interactive Learning

Quizzes, assessments, interactive activities, and practice scenarios shall allow users to assess their knowledge. Learning progress is tracked in the system database to include activity completed by ID and completed date. The user shall receive a certificate for certain learning pathways, like "Budgeting Basics."

### 5. Event Management System

The Event Management System provides financial seminars, workshops, and webinars that give users an opportunity to gain more experience of personal finance and network with professionals.

1. **Creating and Manage Events**
2. Creating an Event

The system shall allow organizers to create an event and input details including time, speaker, topic, and location-physical or virtual. The system will allow user registration, hence enabling the organizer to set limits on seats, ticket prices, and deadlines for registration.

The system shall implement a calendar view using Google Calendar or Outlook to allow event scheduling, modification, and deletion.

When the event registration deadline data passes, the system shall automatically “gray out” the event listing and close user registration.

1. Editing and Deleting Events

At any point, organizers shall be able to change the details of events as listed in 5.1.1, and changes will be in real time for registered users and the calendar on the events page. In turn, events can be cancelled - which entails automated notifications and processes for refunding paid attendees.

1. **User Event Interaction**

Users can search for, sign up to, and attend events both in-person and virtually.

1. Event Discovery

Users shall be able to filter the event calendar events by date, topic, speaker, or event type, for example, webinar, in person. Events should be suggested based on user interests, past attendance, and subscription status.

User registration should be intuitive. Users shall be sent a confirmation email via SendGrid after having signed up.

1. Event Engagement

Following events, attendees shall be able to view resources such as slides, recordings, and further readings by the speakers. Attendees shall be able to be sent feedback forms via email upon completion of the event for data collection and use in upcoming events.

### 6. Payment Processing System

The Payment Processing System manages all matters dealing with money transaction processing, subscription payments, buying event tickets, and refunds.

1. **Payment Gateway Integration**

Different integrated payment gateways provide the user with diverse options to make their payments.

1. Supported Gateways

Users shall be able to make payments using PayPal, Stripe (credit/debit cards).

1. Secure Transaction Handling

All transactions shall be encrypted following the standards of PCI-DSS compliance and any standards used by the vendor (ex. PayPal uses 128-bit SSL), ensuring protection of data at all levels.

1. Error Handling

Should a payment fail, or subscription allocation fail, any transaction will be fully cancelled (the resource not allocated, and any transferred money cancelled), an error message displayed, and a prompt to retry displayed. The user shall be able to retry the payment immediately after the transaction is fully cancelled.

1. **Transaction Management**

Transaction management ensures transparency in record keeping and offers access to full payment history.

1. Invoicing and Receipts

Automated invoices and receipts for every successful transaction shall be sent to users via email (SendGrid), which are archived in the user's profile via storage in the database, storing date of transaction, payment amount, vendor payment details, and item purchased. Said information shall be visible in the user’s profile and shall be downloadable at any time.

1. Refunds and Cancellations

Refunds shall be possible through the vendor’s system as applicable/possible by set policies. A successful refund shall be noted in the transaction history for the amount refunded and sending notifications via email (SendGrid) to the users.

### 7. Reporting and Analytics

The system shall implement a system for reporting user analytics to depict user activities and platform engagement for users and administrators as appropriate.

* + 1. **User Reports**

1. Financial Reports

Users shall be able to generate progress reports in the budget calculator, and use said reports to gauge progress towards their financial goals. Such reports include graphs that analyses user income vs expenses, home mortgage payment percentage overtime, and retirement investments over time.

1. Customization Options

Users shall be able to generate custom reports by selecting any particular period, financial category, or goal.

* + 1. **Administrative Analytics**

The system shall implement an administrator’s dashboard that shows the administrator the performance of the platform along with the behavior of the users.

1. Usage Metrics of the Platform

An administrator shall be able to monitor active user numbers, daily logins, consultation rates, and event attendance. Further growth metrics include new user registrations, subscriptions, and revenue trends to give a high-level view of the performance of the platform. Reports shall be generated by querying the database for relevant data and compiling the relevant statistics.

Passwords shall be hashed using MD5 and salted before storage. Even in the event of database breach, an attacker cannot use encrypted passwords. An encrypted data breach is not a data breach.

1. Tracking Revenue

An administrator shall be able to generate reports on income from subscriptions, event tickets sold, and transaction volume are available to administrators. Revenue data shall be further categorized by source, time period, and user segment, making it supportive for strategic decisions. Reports shall be generated by querying the database for relevant data and compiling the relevant statistics.

## Non-functional Requirements

### 8. Security

WealthWise operates or will operate under sensitive financial information. Thus, security forms the basis of this site. This section analyses how the system will protect data in order to gain and uphold user trust.

1. **Encryption**
2. Data at Rest

All sensitive data including but not limited to user credentials, personal information, financial data, and logs of communication shall be encrypted using state-of-the-art encryption algorithms, e.g., AES-256.

Said database encryption shall take place on the server to assure data security against unauthorized access.

1. Data in Transit

Any data exchanged between the client and the server shall be over SSL/TLS protocols. Thus, no useable interception of the data can occur during transmission. HTTPS will be used to impose secure communication among all pages.

Vulnerability scans using software such as NESSUS will make sure that periodic identification of any weakness within the data transmission process is identified and takes measures to remove them.

1. **Access Control**

The system shall be capable of enforcing most rigid access controls in order for sensitive information and functionalities to be provided only to users who are duly authorized for the same purpose.

1. Role-based Access Control (RBAC)

RBAC shall assign different designations to various roles, such as client, advisor, admin, and many more. The users shall only see features that are assigned to their role; therefore, controlling functions or unauthorized access.

1. Session Management

The session tokens shall time limited and will re-prompt authentication after long periods of inactivity. Session tracking shall be implemented, which checks concurrency of session from different IPs and asks confirmation from user.

1. Audit Logs

All user activities, particularly modifications of sensitive data, shall be logged in an event viewer, such as Security Onion. Logging shall monitor the following activities attempts at logging on, password changes, profile data changes, transactions. Logs will record the time, user, IP address, port numbers, action, and success or failure of said action.

Logs shall be stored on the server, susceptible to periodic inspection via a tool such as Security Onion in case of suspicion.

### 9. Accessibility

The platform shall be made available to a wide range of users.

1. **ADA compliance**
2. Requirements Standards

The platform shall be developed to meet the requirements for ADA and WCAG 2.1 compatibility. The system shall also be developed to meet or exceed the levels of compatibility required by WCAG 2.1 Level AA to ensure access without barriers in the platform by people with disabilities.

1. Screen Reader Support

All user-facing content including forms, navigation elements, images, and buttons shall be compatible with screen readers. Descriptive alt text for images and ARIA labels for interactive elements will be provided so that screen readers can correctly interpret content.

### 10. Backup

Backup mechanisms shall be implemented to ensure data integrity and platform availability and mitigated unexpected incidents.

1. **Backup Strategies**
2. Manual

The system shall provide manual backup options to allow system administrators to back up the system before a suspected incident, failed scheduled backup, or before a major system overhaul. The manual system backup shall be implemented with standard operating system commands, such as Window’s robust copy (robocopy) command.

1. Schedule

Daily incremental and weekly full backups shall be able to be scheduled at the site administrator’s discretion, which shall create a full snapshot of both the database and file storage. This will ensure device failure or cyberattacks do not result in total data loss.

### Performance and Scalability

The platform shall be built to accommodate a large number of users and transactions without impacting on the performance and responsiveness of the platform during peak conditions.

1. **Performance Optimization**

The system needs to optimize performance so that interactions are fast and responsive, hence minimum delay and resultant dissatisfaction among end-users.

1. Response Time

Maximum time elapsed before the system responds to any action made by the user shall not exceed more than 10 seconds under a normal load. On average, pages should be loaded in 3 seconds. Database queries and calculations should take less than 10 seconds.

Caching mechanisms: Data that is often accessed, such as news feeds, educational resources, and user profiles, should be cached (in server memory or in local browsers as applicable) to return the responses quickly.

Asynchronous processing should be implemented for longer operations, such as data import and report generation.

Database indexing and query optimization should be implemented to speed up data retrieval and result in faster access times for user profile information, financial tools, and the history of transactions.

1. **CPU and Memory Usage**
2. Objective

The system shall utilize no more than 5% of a 2.3 GHz CPU per user session and utilize less than 1 GB of RAM per session, keeping performance high even multiple client instances are running.

1. Load balancing

Server capacity should be distributed across multiple servers to prevent any server reaching its maximum capacity and bottle necking the system.

Memory-intensive operations, such as printing large reports, should be run on a dedicated server to lighten the load on the web servers.

1. **Concurrency**
2. Goal

The platform shall be able to support up to 25,000 concurrent users without noticeably compromising its performance.

1. Implementation

In real-time or near real-time services, such as instant messaging or event notification, WebSockets shall handle the connection to manage simultaneous connections properly. The application should use database connection pooling as a method to manage simultaneous requests to the database, reducing variability in performance due to high loads.

1. **Horizontal Scaling**
2. Objective

The system shall oversee up to 25,000 users by adding capacity through horizontal scaling. Separate servers should be implemented for independent components for user management, subscription management, and financial tools, all of which should be independently scalable.

1. Cloud infrastructure

Cloud infrastructure, such as AWS, should be utilized. New instances can easily be added if the system sees spikes in the load. Auto-scaling mechanisms shall, if could infrastructure is implemented, dynamically apportion resources based on the traffic patterns that develop.

1. **Data Scaling**
2. Requirement

The database shall scale with the ever-increasing data load from users, advisors, transactions, logs, and content.

1. Implementation

Large databases shall be partitioned using Sharding so the data will be distributed across several servers. Data archiving should be considered for data not utilized after 3 months to take the load off from the primary database, therefore allowing faster retrieval times of current data.

# Log of Meetings, Reviews, and Meetings[sic]

1. September 25th, 2024 - Virtual Team Meeting

Attendees: Draix, John, Norman, Sawyer

The team discussed the Introduction and Work Structure Breakdown sections of the Requirement Document following the reception of the Request of Proposal from the client. The team talked through initial thoughts and confusion about the requirements.

1. September 26th, 2024 - Client Meeting

Attendees: Draix, John, Norman, Sawyer, Dr. Albujasim

The team met with the client and discussed a fair portion of the requirements and initial insight into what types of diagrams were necessary and how the WSB should be developed.

1. October 7th, 2024 - In-Person Team Meeting

Attendees: Draix, John, Norman, Sawyer, Dr. Albujasim

The team met and started jointly filling out sections of the Introduction and started the design of the WSB. The team then assigned responsibility for the remaining sub-sections to its members.

1. October 8th, 2024 - Client Meeting

Attendees: Draix, John, Norman, Sawyer, Dr. Albujasim

The team met with the client and discussed the full feature list the website should implement. The team also gained additional insight into how the client wanted the Proposed System Overview and the Work Structure Breakdown to be formatted and developed. The team agreed to work asynchronously on the remaining sections and meet virtually before submitting the increment of the RD known as “Assignment 2.”

1. October 9th, 2024 - Virtual Team Meeting

Attendees: Draix, John, Norman, Sawyer

The team met to review the final changes to the remaining sections of the Introduction and WBS before the final submission the next day.

1. October 18th, 2024 - Virtual Team Meeting

Attendees: Draix, John, Norman, Sawyer

The team met to review the full requirements for the RD. Initial thoughts for how the requirements needed to be specified were considered and the next meeting time was set up.

1. October 21st, 2024 - In-Person Team Meeting

Attendees: Draix, John, Norman, Sawyer

The team met and began jointly filling out all sections of the RD. Most notably, initial versions of all requirements were filled out. All remaining sections were assigned to members to be worked on Asynchronously.

1. October 22nd, 2024 - Client Meeting

Attendees: Draix, John, Norman, Sawyer, Dr. Albujasim

The team met with the client to discuss the depth of specification for the final requirements. Questions concerning Change Control, Meeting Review, and Terms and Conditions were also discussed. The team decided to finish the RD asynchronously and meet virtually before submitting the “final” RD.

1. October 23rd, 2024 - Team Meeting

Attendees: Draix, John, Norman, Sawyer

The team met and reviewed the final changes to the RD before the RD was submitted to the client.

# Change Control

Change requests shall propagate from meetings with the client or upon written notification from the client. Potential changes to the Requirements Document shall be requested, negotiated, specified, and validated, as necessary. Once this is complete, the Requirements Document shall be updated to reflect the specified changes, and an entry in the “Log of Meetings, Reviews, and Meetings[sic]” section shall be made detailing the changes to the requirements, the date of the change, and any relevant data, such as revision of cost estimate.

# Appendix

## Definition, Acronyms, and Abbreviations

* + - * ACID: A database model designed for data synchronicity standing for atomic, consistency, isolation, and durability
      * ADA: Americans with Disabilities Act
      * AES: Advanced Encryption Standard
      * API: Application Programming Interface
      * ARIA: Accessible Rich Internet Applications
      * AWS: Amazon Web Services
      * DFD: Data Flow Diagram
      * GDPR: General Data Protection Regulation
      * IDS: Intruder Detection System
      * IP: Internet Protocol
      * JWS: JSON Web Tokens
      * PCI DSS: Payment Card Industry Data Security Standard
      * RD: Requirement Document
      * SFTP: Secure File Transfer Protocol
      * SQL: Structured Query Language
      * SSL: Internet communication security protocol: Secure Sockets Layer
      * REST: An architecture for distributed systems (REpresentational State Transfer)
      * TLS: Internet communication security protocol: Transport Layer Security
      * UI: User Interface
      * UX: User Experience
      * WCAG: Web Content Accessibility Guidelines
      * Web app: a website with application functionality
      * WSB: Work Structure Breakdown

## Work Structure Breakdown

Figure 3. Work structure breakdown diagram.

## ER Diagram

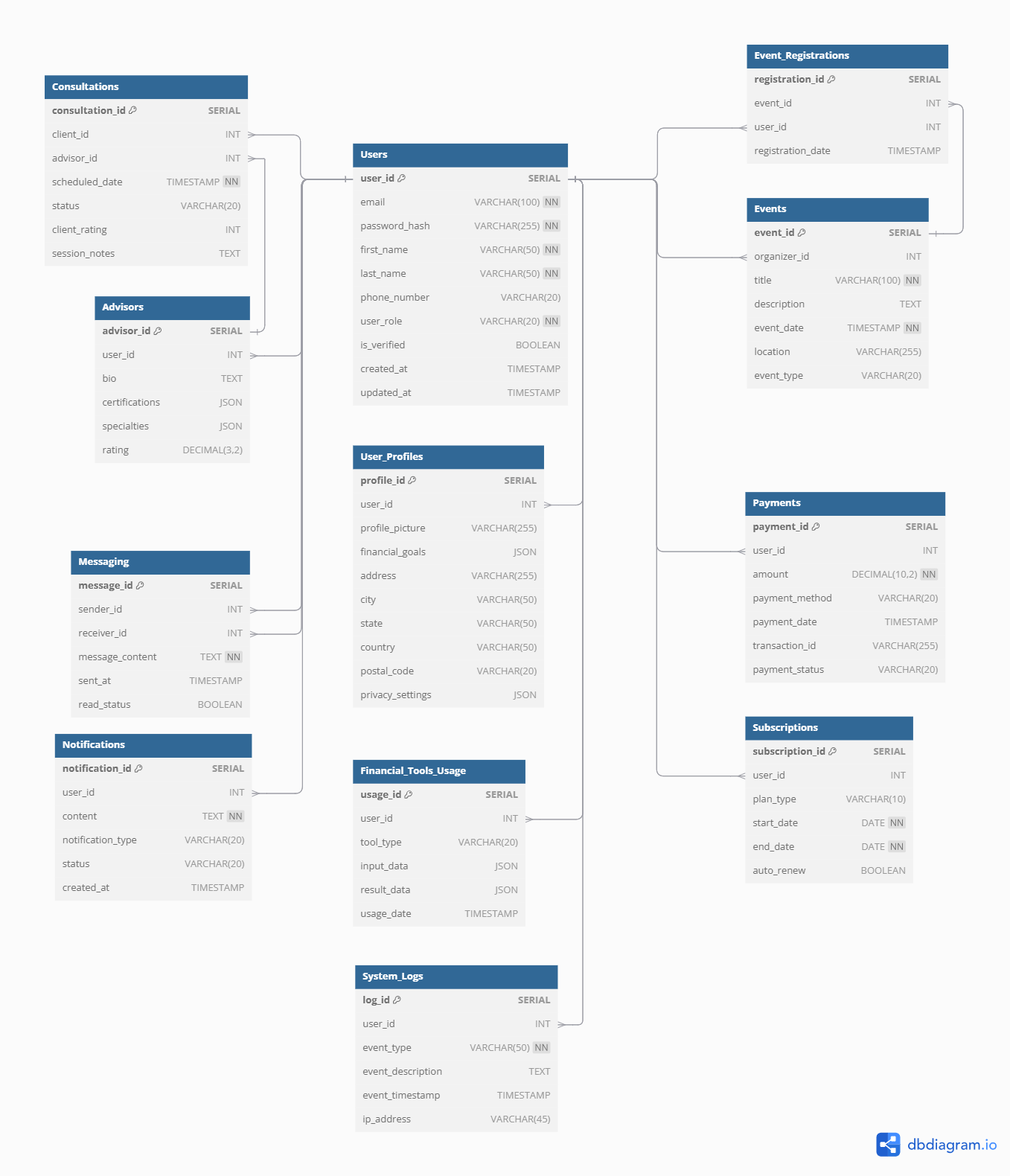


Figure 4. ER diagram for the system database.

## Data Flow Diagram

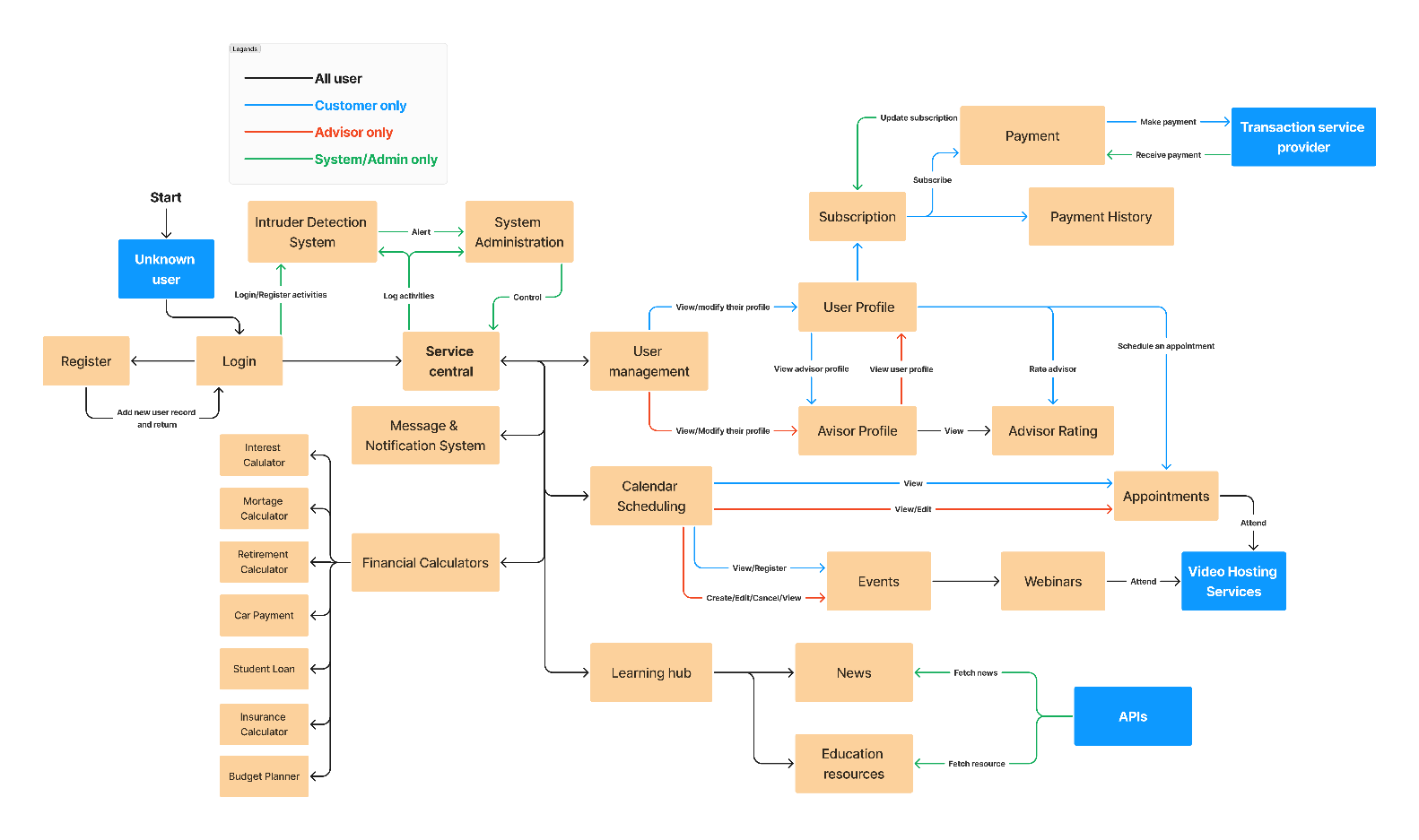


Figure 5. Data flow diagram of the system

# Terms and Conditions

These Terms and Conditions (“Agreement”) constitute a binding contract between the client and Arqer (“Service Provider”) for the use of our services to develop software as outlined in this Requirements Document.

Party A (Client):

Legal entity’s name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Party B (Service Provider):

Legal entity’s name: Arqer Software Development Agency

1. **Scope of Work**
2. Services Provided
   1. The developer team agrees to design, develop, assess, and deploy the software system as specified in the requirements document (RD)
   2. All work should be performed according to the specifications and timeliness mutually agreed upon by both parties.
3. Change Requests
   1. Any modification or additions to the project scope shall be documented and approved in writing by both parties.
   2. Additional work arising from change requests should result in adjustments to the project timeline and costs.
4. **Project Timeline**
   1. Milestones
      1. The project shall follow a schedule of milestones, each with specific deliverables and deadlines.
      2. Both parties agree to adhere to the project timeline unless modifications are mutually agreed upon.
   2. Delays
      1. The Developer shall promptly inform the Client of any circumstances that may delay the project.
      2. Delays caused by the Client, including late approvals or feedback, should result in timeline adjustments.
5. **Payment Terms**
   1. Fees and Invoicing
      1. The total project cost is $0, payable in installments as follows:
         1. 20% upon signing this Agreement.
         2. 30% upon completion of the initial prototype.
         3. 50% upon final delivery and acceptance.
   2. Payment Schedule
      1. Invoices shall be issued according to the milestone completions.
      2. Payments shall be paid within 7 days of invoice receipt.
   3. Additional Costs
      1. Any additional expenses incurred due to scope changes or Client requests shall be billed separately with prior approval.
6. **Intellectual Property Rights**
   1. Ownership
      1. Upon full payment, all intellectual property rights for the developed software, including source code and documentation, shall transfer to the Client.
   2. Developer’s Tools
      1. The Developer shall retain ownership of any pre-existing materials or generic modules used in the project, provided they do not contain the Client's proprietary information.
7. **Confidentiality**
   1. Non-Disclosure
      1. Both parties agree to keep all confidential information strictly confidential and not disclose it to any third parties without prior written consent.
   2. Data Protection
      1. The Developer must implement industry-standard security measures to protect the Client's data.
8. **Warranties and Liabilities**
   1. Warranty and Maintenance Period
      1. The Developer offers a warranty period of 1 month from the date of final acceptance, or until the end of the course, whichever comes first, during which any defects will be corrected at no additional charge.
   2. Limitation of Liability
      1. The Developer's total liability under this Agreement shall not exceed the total fees paid by the Client.
9. **Maintenance and Support**
   1. Post-Deployment Support
      1. Maintenance and support services beyond the warranty period should be provided under a separate agreement.
   2. Updates and Upgrades
      1. Any updates or upgrades requested by the Client shall be subject to additional fees and timelines.
10. **Termination**
    1. Termination by Client
       1. The Client may terminate this Agreement with a 30-day written notice. Fees for services rendered up to the termination date shall be paid immediately.
    2. Termination by developers
       1. The Developer may terminate this Agreement if the Client fails to make payments or breaches any terms, with a 30-day written notice.
11. **Dispute Resolution**
    1. Negotiation
       1. Both parties agree to attempt to resolve disputes through mutual negotiation.
    2. Arbitration
       1. If negotiation fails, disputes will be settled by arbitration in accordance with the rules of the state of South Dakota.
12. **Governing Law**

This Agreement shall be governed by and construed in accordance with the laws of the state of South Dakota.

1. **Force Majeure**

Neither party shall be liable for delays or failures due to causes beyond their reasonable control, such as natural disasters, war, or government actions.

1. **Entire Agreement**

This Agreement constitutes the entire understanding between both parties and supersedes all prior agreements, whether written or oral.

1. **Amendments**

Any amendments to this Agreement shall be made in writing and signed by both parties.

1. **Severability**

If any provision of this Agreement is found to be unenforceable, the remaining provisions shall remain in effect.

1. **Notices**

All notices shall be in writing and delivered to the addresses provided by both parties.

By signing below, both parties acknowledge that they have read, understood, and agree the terms and conditions outlined in this Agreement.

**Party A’s Representative**

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

Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**Party B’s Representative**

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

Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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