DolFin: Software Requirements Specification

BY: ARMAAN CHETAL

1. Introduction

1.1 Product Scope

- DolFin is a web-based financial wellbeing application designed to empower users to manage their finances effectively and achieve their financial goals. It aims to provide a comprehensive and user-friendly platform that goes beyond basic transactional data to offer:
- Financial health assessments: Personalized insights and scores based on user data using AI and machine learning.
- Goal-oriented budgeting: Tools to help users create, track, and adjust budgets aligned with their financial goals.
- Investment tracking: A consolidated view of investment performance and portfolio allocation.
- Educational resources: Interactive learning modules and personalized recommendations tailored to users' financial needs.
- Community features: A platform for users to connect, share experiences, and learn from each other.

1.2 Product Value

- DolFin offers several key benefits to users:
- Improved financial literacy: Provides educational resources and personalized guidance to enhance financial knowledge and decision-making.
- Enhanced financial planning: Tools and insights empower users to create realistic budgets, track progress towards goals, and manage debt effectively.
- Increased financial confidence: Regular financial health assessments and personalized guidance build trust and encourage positive financial behaviors.
- Simplified financial management: A consolidated platform streamlines financial activities, offering a holistic view of income, expenses, investments, and debts.

1.3 Intended Audience

The primary target audience for DolFin is:

- Millennials (born 1981-1996): This digitally native generation actively seeks financial management solutions through technology.
- Generation Z (born 1997-2012): As they enter the workforce, Gen Z needs tools to manage finances, plan for the future, and make informed financial decisions.
- The long-term vision is to expand the user base to encompass:
- Generation X (born 1965-1980): This demographic increasingly adopts digital tools for financial management, with needs around retirement planning and wealth management.
- Baby Boomers (born 1946-1964): As they approach retirement, this generation requires tools to manage their assets and ensure financial security.

1.4 Intended Use

- DolFin is intended for individual users to manage their personal finances. It caters to a wide range of financial needs, from budgeting and savings to investment tracking and financial goal setting. Users can access DolFin through a web browser on various devices, including laptops, tablets, and smartphones.

1.5 General Description

- This document outlines the Software Requirements Specification (SRS) for the DolFin application. It defines the functionalities, technical specifications, and non-functional requirements for the development process.

2. Functional Requirements

2.1 User Interface (UI) Requirements

- User-friendly interface: The UI should be intuitive, clean, and visually appealing, promoting ease of use and navigation.
- Responsive design: The application should adapt seamlessly to different screen sizes and devices (desktop, tablet, mobile).
- Accessibility: The UI should comply with accessibility standards to ensure inclusivity for users with disabilities.
- Customizable dashboard: Users should personalize the dashboard to prioritize and display the financial information most relevant to them.
- Interactive visualizations: Utilize charts, graphs, and progress bars to present financial data in an understandable and actionable format.

2.2 Core Functionalities

- Account aggregation: Securely connect to various financial accounts (bank accounts, investment accounts, credit cards) for a comprehensive overview.
- Budgeting: Create and manage budgets with income and expense categorization.
- Goal setting: Define financial goals (e.g., saving for a down payment, retirement planning) and track progress.
- Transaction tracking: Automatically categorize transactions and offer insights into spending patterns.
- Financial health score: Generate a personalized score based on AI analysis of user data, providing a snapshot of financial wellbeing.
- Educational resources: Offer a library of articles, videos, and interactive modules on various financial topics based on user needs.
- Community features: Facilitate user interaction through forums, discussion boards, or chat functionalities (consider implementing moderation tools).

2.3 Security Requirements

- Data encryption: Implement robust encryption for all user data at rest and in transit.

- Authentication and authorization: Secure user logins with multi-factor authentication and implement role-based access control.
- Regular security audits: Conduct periodic security assessments and penetration testing to identify and address vulnerabilities.
- Compliance with data privacy regulations: Adhere to relevant data privacy regulations (e.g., GDPR, CCPA) regarding user data collection, storage, and usage.

2.4 Performance Requirements

Fast loading times: Ensure the application responds quickly to user interactions

3. External Interface Requirements

3.1 User Interface (UI) Frameworks and Libraries

- Consider using a front-end framework like ReactJS to build a dynamic and responsive user interface with reusable components.
- Explore UI libraries like Material-UI or Ant Design for pre-built components that adhere to design principles and promote consistency.

3.2 Hardware Interface Requirements

- DolFin should be compatible with a wide range of internet-connected devices, including laptops, desktops, tablets, and smartphones.
- Minimum hardware specifications (e.g., processor speed, RAM) might be defined based on the complexity of functionalities and data processing needs.

3.3 Software Interface Requirements

- Financial data APIs: Securely integrate with APIs provided by banks, investment institutions, and credit card companies to facilitate account aggregation.
- Payment processing APIs (optional): If the application allows bill payments or investment transactions within the platform, integrate with secure payment processing APIs.
- AI/Machine Learning libraries: Utilize libraries like TensorFlow.js or scikit-learn (Python) to develop AI-powered features like personalized financial health scores and predictive budgeting.

DolFin's functionalities heavily rely on secure and efficient communication with external systems. This section delves into the technical details of Software Interface Requirements (SWIRs) related to:

Financial Data APIs

Payment Processing APIs (Optional)

AI/Machine Learning Libraries

3.3.1 Financial Data APIs: Account Aggregation

To empower users with a holistic view of their finances, DolFin needs to securely connect to various financial accounts. This necessitates integration with Financial Data APIs offered by banks, investment institutions, and credit card companies. Here's a breakdown of the technical considerations:

API Selection: Popular choices include:

- Plaid (US): https://plaid.com/ (Offers connections to a vast network of US financial institutions)
- Finicity (US): https://www.finicity.com/ (Provides account aggregation and financial data services)
- TrueLayer (Europe): https://truelayer.com/ (Focuses on open banking solutions in Europe)
- MX (Various Regions): https://www.mx.com/ (Offers account aggregation solutions globally)

Using Financial Data APIs to Empower DolFin Users

Financial Data APIs play a crucial role in DolFin's ability to provide users with a holistic view of their finances. Here's a detailed breakdown of how the mentioned APIs (Plaid, Finicity, TrueLayer, MX) can be leveraged in the DolFin project:

1. Account Aggregation and Data Retrieval:

Functionality: All these APIs offer secure connections to a vast network of financial institutions. DolFin can integrate with the chosen API to initiate a secure connection process with the user's bank, investment account, or credit card provider.

Implementation:

- The DolFin application will guide users through an authorization flow provided by the chosen API. This typically involves logging into the user's financial institution account and granting DolFin permission to access specific data categories (transactions, balances, holdings).
- Once authorized, the API facilitates the secure exchange of financial data between the financial institution and DolFin's servers.

Benefits for DolFin:

- Enables automatic import of transaction data for budgeting, categorization, and spending analysis.
- Aggregates account balances from various sources to provide a comprehensive view of a user's net worth.
- Retrieves investment performance data to track portfolio health within DolFin.

2. Specific API Features and Considerations:

Plaid (US):

- Offers a vast network of US financial institutions, making it a strong choice for US-based users.
- Provides real-time balance updates and supports investment account data aggregation.
- Consider potential limitations on data access based on Plaid's pricing tiers.
- Finicity (US):
- Offers a suite of financial data services beyond account aggregation, including budgeting tools and financial wellness insights (potential integration opportunities).
- Provides real-time balance updates and supports various account types.
- Evaluate pricing structures and ensure alignment with DolFin's business model.

TrueLayer (Europe):

- Focuses on open banking solutions in Europe, making it ideal for DolFin if targeting European users.
- Provides strong security features and real-time data access.
- Research regulatory compliance requirements for open banking solutions in specific European regions.

MX (Various Regions):

- Offers account aggregation solutions globally, catering to a wider user base.
- Provides real-time data access and supports various account types.
- Investigate MX's coverage in specific regions where DolFin plans to operate.
- Security Considerations:
- Regardless of the chosen API, prioritize robust security measures:
- Implement OAuth for secure user authentication and authorization.
- Utilize HTTPS for all communication with financial data APIs.
- Adhere to the API provider's security best practices and guidelines.

Security Protocols:

- Implement OAuth (Open Authorization) for secure user authentication and authorization when connecting to financial institutions through their APIs. OAuth ensures users grant DolFin access to specific account data without revealing login credentials.
- Utilize HTTPS (Hypertext Transfer Protocol Secure) for all communication with financial data APIs, encrypting data transmission and protecting sensitive financial information.

Data Parsing and Aggregation:

- Financial data APIs typically return data in a structured format like JSON (JavaScript Object Notation) or XML (Extensible Markup Language). DolFin needs to parse this data and convert it into a standardized format for internal storage and analysis.
- Utilize libraries like Python's json or xml modules for efficient data parsing based on the chosen API's response format.

Correlation to DolFin Project:

- Financial Data APIs are instrumental in achieving DolFin's core functionalities. By securely connecting to user accounts, DolFin can automatically:
- Import transactions for budgeting, categorization, and spending analysis.
- Track investment performance across various accounts.
- Calculate net worth based on account balances and liabilities.

3.3.2 Payment Processing APIs (Optional)

- If DolFin expands its scope to allow bill payments or investment transactions within the platform, integration with secure payment processing APIs becomes crucial. Here's what to consider:
- API Selection: Popular payment processing options include:
- Stripe: https://stripe.com/ (Offers a wide range of payment functionalities)
- PayPal: https://www.paypal.com/ (A well-established online payment processing service)
- Braintree (Owned by PayPal): https://www.braintreepayments.com/resources (Provides merchant account solutions)

Security Protocols:

Adhere to Payment Card Industry Data Security Standard (PCI DSS) compliance guidelines to ensure the highest level of security for cardholder data. PCI DSS mandates maintaining a secure environment for cardholder information throughout the payment process.

Implement strong encryption protocols to protect sensitive financial data during transactions.

Utilize tokenization, where sensitive data like credit card numbers are replaced with unique tokens during processing, minimizing the risk of data breaches.

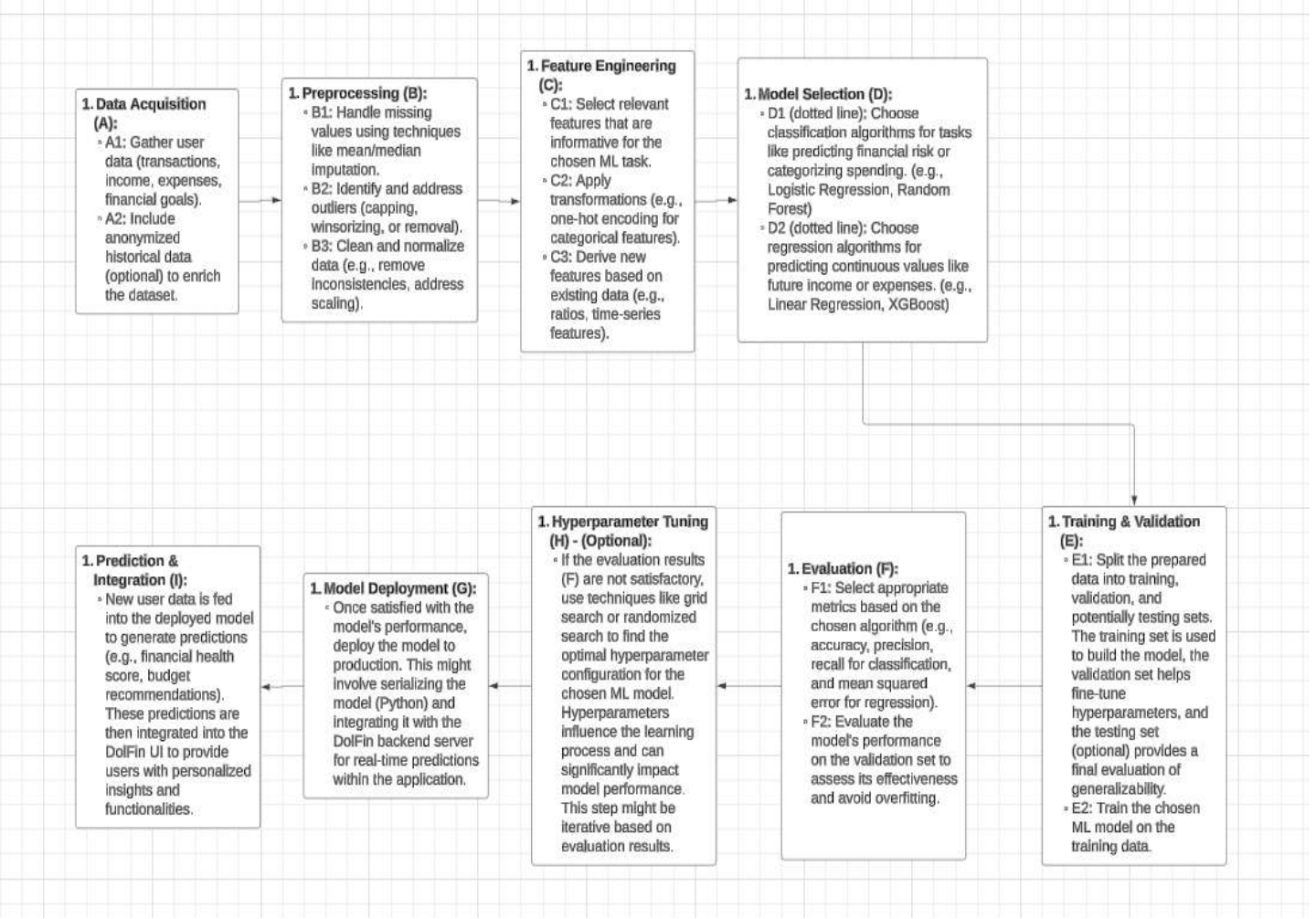
Transaction Management:

- Integrate with the chosen payment processor's API to initiate and manage user-authorized payments (bill payments, investment purchases).
- Ensure proper handling of successful and failed transactions, providing clear feedback to users.
- Correlation to DolFin Project:
- Payment processing APIs can enhance DolFin's value proposition by allowing users to:
- Conveniently pay bills directly within the platform.
- Seamlessly invest in various financial instruments without switching between platforms.

3.3.3 AI/Machine Learning Libraries: Powering Financial Insights

- DolFin aims to leverage AI and Machine Learning (ML) to deliver personalized financial insights and functionalities. Here's how AI/ML libraries come into play:
- Library Selection: Popular choices for Javascript-based projects include:
- TensorFlow.js: https://www.tensorflow.org/js (A Javascript library for deploying pre-trained machine learning models in web applications)
- Keras.js: https://keras.io/ (A high-level API for building and deploying neural networks)
- Python Libraries (For Backend Development):
- scikit-learn: https://scikit-learn.org/ (A comprehensive library for machine learning tasks in Python)
- TensorFlow (Python): https://www.tensorflow.org/ (The core library for building and training machine learning models)

DolFin's goal of using AI and Machine Learning (ML) opens doors for exciting functionalities like personalized financial health scores and predictive budgeting. Here's a detailed breakdown of the implementation process, technical considerations, and potential ML algorithms:



1. Data Acquisition and Preprocessing:

Data Sources: Leverage user-provided data (transactions, income, expenses, financial goals) and potentially anonymized historical data to train and evaluate ML models.

Data Cleaning and Feature Engineering:

Cleanse data by handling missing values, outliers, and inconsistencies.

Engineer new features from existing data that might be more informative for ML models. For instance, categorize transactions, calculate ratios (debt-to-income), and create time-series features for income and spending patterns.

Data Splitting: Divide the prepared data into training, validation, and testing sets. The training set is used to build the model, the validation set helps fine-tune hyperparameters, and the testing set evaluates the model's generalizability on unseen data.

2. Model Selection and Training:

Candidate ML Algorithms: Here are some potential algorithms suitable for DolFin's functionalities:

Classification Algorithms:

- Logistic Regression: For binary classification tasks, like predicting whether a user is likely to achieve a financial goal based on their spending habits and income.
- Random Forest: A robust classification algorithm for predicting financial risk or categorizing spending based on past transactions.

Regression Algorithms:

- Linear Regression: For predicting continuous values, such as future income or expenses based on historical data and financial trends.
- XGBoost: A powerful gradient boosting algorithm for making accurate predictions about future financial health or budget needs.
- TensorFlow.js or Keras.js (Frontend): If deploying pre-trained models in the web application, leverage these libraries to load and integrate the chosen models for real-time predictions within the DolFin UI.
- Scikit-learn or TensorFlow (Backend): For model development and training on the backend server, utilize these Python libraries to build, train, and evaluate the chosen ML models.

3. Model Evaluation and Deployment:

- Evaluation Metrics: Choose appropriate metrics based on the chosen algorithms (e.g., accuracy, precision, recall for classification, and mean squared error for regression). Evaluate the model's performance on the testing set to assess its generalizability.
- Hyperparameter Tuning: Use techniques like grid search or randomized search to find the optimal hyperparameter configuration for the chosen ML model, which can significantly improve its performance.
- Model Deployment: Once satisfied with the model's performance, deploy the model to production. If using pre-trained models, integrate them with the DolFin frontend using TensorFlow.js or Keras.js. For custom-trained models, consider serializing them using libraries like Pickle (Python) and deploying them on the backend server for real-time predictions within the application.

4. Examples of AI/ML Functionalities in DolFin:

- Personalized Financial Health Score: Based on a user's financial data and machine learning models, DolFin can generate a score that reflects their overall financial wellbeing. This score can be used to provide personalized recommendations and encourage positive financial behaviors.
- Predictive Budgeting: Machine learning models can analyze past spending patterns and income trends to predict future expenses. This allows Dolfin to suggest personalized budget categories and amounts, making budgeting more proactive and realistic.
- Goal Setting and Tracking: Machine learning can identify potential roadblocks to achieving financial goals based on historical data and user behavior. DolFin can then provide tailored advice and nudges to keep users on track.

3.4 Communication Interface Requirements

- DolFin should utilize secure communication protocols (HTTPS) for all data transmission between the application and servers.
- Implement real-time communication protocols (e.g., WebSockets) for features like live chat functionalities within the community section (if applicable).

4. Non-Functional Requirements

4.1 Security

As emphasized earlier, data security is paramount. Implement robust security measures as outlined in the functional requirements (Section 2.3).

4.2 Capacity

The system should handle a growing user base and increasing data volume without compromising performance. Consider scalability options (discussed in Section 4.5).

4.3 Compatibility

DolFin should be compatible with major web browsers (Chrome, Firefox, Safari, Edge) on different operating systems (Windows, macOS, Linux).

4.4 Reliability

The application should be highly reliable with minimal downtime and ensure data integrity. Implement measures for error handling, data backup, and disaster recovery.

4.5 Scalability

The system architecture should be designed to scale horizontally to accommodate an increasing number of users and data. This might involve using cloud-based infrastructure and containerization technologies.

4.6 Maintainability

The codebase should be well-documented, modular, and follow coding best practices to facilitate future maintenance and updates.

4.7 Usability

Usability is a core focus. The UI should be intuitive, user-friendly, and cater to a diverse range of users with varying technical expertise.

4.8 Other Non-Functional Requirements

Localization (optional): Consider the potential for future localization to support different languages and regional financial practices.

Offline functionality (optional): Explore the possibility of offering limited offline functionality for specific features, such as budget management or expense tracking.

5. Definitions and Acronyms

This section provides definitions for key terms and acronyms used throughout the DolFin Software Requirements Specification (SRS).

- API (Application Programming Interface): A set of protocols and tools that define how applications interact with external systems. APIs allow applications to access functionality and data provided by external services in a secure and standardized manner.
- CCPA (California Consumer Privacy Act): A law enacted by the state of California that regulates the collection, use, and disclosure of personal information by businesses operating in California. The CCPA grants consumers specific rights regarding their data, including the right to access, delete, and opt-out of the sale of their personal information.
- GDPR (General Data Protection Regulation): A regulation in EU law on data protection and privacy. The GDPR aims to give control to individuals over their personal data and simplify the regulatory environment for international business by unifying the regulation within the EU.
- AI (Artificial Intelligence): A branch of computer science that deals with the creation of intelligent agents, which are systems that can reason, learn, and act autonomously. AI research has been highly successful in developing effective techniques for solving a wide range of problems, from game playing to medical diagnosis.
- Machine Learning (ML): A subfield of Artificial Intelligence (AI) that focuses on algorithms and techniques that can learn from data without being explicitly programmed. ML algorithms can automatically improve their performance over time as they are exposed to more data.
- OAuth (Open Authorization): An open standard authorization framework that allows users to grant third-party applications access to their accounts without sharing their credentials directly. This ensures a more secure and privacy-preserving approach to user authentication.
- Open Banking: A system that provides third-party financial service providers with open access to consumer banking data, with the consent of the consumer. This allows for the development of innovative financial products and services.
- PCI DSS (Payment Card Industry Data Security Standard): A global standard that defines requirements for organizations that store, process, or transmit cardholder data. The PCI DSS is intended to ensure the safe handling of sensitive financial information and reduce the risk of credit card fraud.
- Scikit-learn: A popular open-source Python library that provides a comprehensive set of tools and algorithms for machine learning tasks.
- TensorFlow: An open-source software library for numerical computation using data flow graphs. TensorFlow is a popular choice for building and deploying machine learning models.
- TensorFlow.js: A JavaScript library that enables developers to deploy pre-trained TensorFlow models in web browsers. This allows for real-time machine learning predictions within web applications.
- UI (User Interface): The graphical elements of a software application that users interact with. The UI should be designed to be intuitive, user-friendly, and visually appealing to promote ease of use and user satisfaction.
- UX (User Experience): The overall experience a user has when interacting with a product or service. UX design focuses on creating a positive and seamless user experience by considering all aspects of the user's interaction with the product.

.

6. Revision History

Include a versioning system to track changes made to the SRS document throughout the development process.