Software requirement specification (SRS) document

Project:

Banking

Application

name:

Bank

Date:

21/02/24

Version: 1.0

By: Curtis Wallace

Revision history			
Version	Author	Version description	Date completed
1.0	Curtis Wallace	First Revision	21/02/2024

Approving party	Version approved	Signature	Date
Bank Owner	1.0		22/02/2024

Table of contents

1 Introduction

Describe the purpose of the document.

The purpose of this document is to create a Banking web application to allow users to create and manage a banking account, view transaction history and transfer funds.

1.1 Product scope	List the benefits, objectives, and goals of the product.

The Banking software will allow users to easily create and manage their own accounts. Users will be able to transfer funds between accounts. Users will be able to see a comprehensive list of their transaction history. We will have a relational database containing user and account information. We will focus on ease of use and allowing access from anywhere by implementing a front-end web application.

1.2 Product value	Describe how the audience will find value in the product.

Our product is important in providing banking services with a user-friendly interface. We will create a robust backend with high security implemented to keep user data secure. Our banking services will work on any device and any platform thanks to the implementation of using a progressive web application.

1.3 Intended audience	Write who the product is intended to serve.

Our intended audience is targeted at anyone of age who wants to open their own bank account. We will implement a minimum age requirement of 16. Our users will need to have proficiency in using websites, however we will try to create a user-friendly interface.

1.4 Intended use	Describe how will the intended audience use this product.

Our intended audience will use our product by first creating a personal or business bank account. Users will be able to manage their account information and close their account if wanted. Our users can view their transaction history and transfer funds between accounts. The following use cases illustrate this further:

1.5 General description	Give a summary of the functions the software would perform and the features to be included.

Functional Requirements

Account Management:

Title: Account Management	**Priority:** Must-have	**Estimate: 8**

User story

As a end-user

I want to create an account

so that I can login to my banking account and edit my details

Acceptance criteria

Given that I enter my Name, email address and a suitable password

when I press register on the form, my details are added to the accounts database

then I should be logged in, be able to edit my account details and be able to close my account if I want to.

Financial Transactions:

Title: Financial Transactions	**Priority:** Must-have	**Estimate: 5**

User story

As a end-user

I want to see my transaction history with filters.

so that I can deposit, withdraw and transfer funds between my accounts.

Acceptance criteria

Given that I can see my balance in accounts and transaction history

when I press the deposit, withdraw or transfer button

then I should be able to see my funds move between accounts and the balances be updated immediately

Account Services:

Title: Account Services	**Priority:** Must-have	**Estimate: 5**

User story

As a end-user

I want to see my current balance, transaction history and view my account information

so that I can set up direct debits and request new bank cards

Acceptance criteria

Given that I can see my account information, account type, balance

when I set up direct debits or press the button to request a new card

then I should be able to see the direct debits in transaction history or my new card details on the card tab.

Security and Compliance:

Title: Security and Compliance	**Priority:** Must-have	**Estimate:** 3

User story

As a end-user

I want to have two-factor authentication and have my personal data encrypted

so that I can achieve a high level of security and have confidence in the system

Acceptance criteria

Given that I have an account on the database

when I click login on the banking app

then a 2FA code should be requested by the system and sent as a text message or email. The data should be held and processed in line with my countries data regulations

Customer Support and Maintenance:

Title:	Priority:	Estimate:
Customer Support and Maintenance	Should have	3

User story

As a end-user

I want to be able to get support within the app from customer support

so that I can get help with my account and report fraudulent activities

Acceptance criteria

Given that I have an issue with my account that I want to report

when I click on the help button within the app

then a chat box should open and ask me for information. The message will be sent to the help desk and I will receive an email response from the customer support

Investment and Loan Services:

Title:	Priority:	Estimate:
Investment and Loan Services	Should have	3

User story

As a end-user

I want to be able to invest my money or get a loan

so that I can earn money on my savings or get extra money for purposes such as starting a business or buying a car

Acceptance criteria

Given that I have a personal or business account

when I click on invest or loan functions within the app

then I should be able to add an amount of money into an investment savings account or receive more money from the bank into my account as a loan with a given interest rate

AI-Powered Financial Advice:

Title:	Priority:	Estimate:
AI-Powered Financial Advice	Could have	5

User story

As a end-user

I want to be able to get financial advice based on my spending habits

so that I can learn where I can make changes to my habits to save more money and work on savings goals

Acceptance criteria

Given that I have an account and want to improve my savings

when I click on the AI help button within the app

then I should be able to get recommendations from an AI API that analyzes where I can make changes to my spending habits in order to save more money

3 External interface requirements

3.1	User interface requirements	Describe the logic behind the interactions between the users and the software (screen layouts, style guides, etc).

Our users will interact with the front-end application. Any inputs will trigger a backend process that will first authenticate any user communication with the database and external two-factor authentication service for sign on. When a user transfers money, this will appear to them instantly through the frontend application, but in the backend, the transaction will be secured, authenticated and processed on the database.

3.2	Hardware interface requirements	List the supported devices the software is intended to run on, the network requirements, and the communication protocols to be used.

Our software will be supported by any internet-enabled devices. The software will be delivered in the form of a web application. The software may perform better on devices with certain aspect ratios; however, it should be mostly universally accessible.

3.3	Software interface requirements	Include the connections between your product and other software components, including frontend/backend framework, libraries, etc.

The frontend and backend will be created in Mendix. No external libraries will be required for creation of the application. Backend database connections will be handled directly in Mendix. Our login procedure will use an external connection to a two-factor authentication API. This will require us to get a token from the API to authorize our users access to the application.

3.4	Communication interface requirements	List any requirements for the communication programs your product will use, like emails or embedded forms.

There will be a requirement to access an email provider's service. This will be needed when initially signing up to the Bank to verify the user's email address.
Every time the user logs in, they will be required to access their email to retrieve their two-factor authentication code.

4 Non-functional requirements

4.1	Security	Include any privacy and data protection regulations that should be adhered to.

Any financial application should not allow multiple sessions. Meaning when our user is signed in, it can only be on one device at a time, we will verify all communication from a single signed in session.
We will secure all communication over the secure trasnport layer, meaning our application can only be accessed by HTTPS and not HTTP. This stops injection hacking and man-in-the-middle attacks.
All data stored will be encrypted when being stored and decrypted for the user to view on client-side.

4.2	Capacity	Describe the current and future storage needs of your software.

Our storage needs all relate to the database side of our application. Initially our application will need to be stored, which will be as a cloud application to be accessed from anywhere with an internet connection. As demand increases or decreases the compute resources required will be automatically scaled with the cloud vendor.
Our database stores user details and sensitive information. All this information is structured text data and will be extremely small, because of this it is unlikely that we will need to make changes to capacity in future.

4.3	Compatibility	List the minimum hardware requirements for your software.

Minimum requirements:
Any internet enabled smart device with the ability to use an internet browser application.

4.4	Reliability	Calculate what the critical failure time of your product would be under normal usage.

Mean time to failure is calculated from total operating time divided by number of failures. With our application hosted in the cloud, it should have high availability. Cloud vendors usually provide a minimum uptime percentage guarantee of at least 99.9%. This is ideal for an essential banking application where reliability is extremely important.

4.5	Scalability	Calculate the highest workloads under which your software will still perform as expected.

Application will work under any workload. Application will scale automatically and whenever required automatically.

4.6	Maintainability	Describe how continuous integration should be used to deploy features and bug fixes quickly.

Any time we release an update, it will trigger an automatic build and test of the system. All the code will be built automatically, tested and then released.

4.7 Usability	Describe how easy it should be for end-users to use your software.

We will have a high level of usability. Our software can be accessed anywhere by anyone. Basic software proficiency should be expected for using a web-based application, being able to use online forms and remember personal security information.

5 Definitions and acronyms

Backend	The data and infrastructure of our application
Frontend	The visual elements that users interact with for our application
Progressive Web App (PWA)	Application will perform like a standalone application, but be fully supported through a web platform
Compute	Resources such as network, processing, memory and more required for computation of the program