Software requirement specification (SRS) document template

Project name:	Team software engineering banking system
Date:	22/02/24
Version:	1
By:	Lewis, Joe, John and Sam

Revision history

Version	Author	Verson description	Date completed

Review history

Approving party	Version approved	Signature	Date

Approval history

Reviewer	Version reviewed	Signature	Date

Table of contents

1 Introduction

Describe the purpose of the document.

The purpose of this document is to provide a technical outline of the requirements of this project. It will be used to guide us on the technical requirements of the project as well as outlining what we want to achieve and how we will achieve it.

1.1 Product scope

List the benefits, objectives, and goals of the product.

The goal of this project is at the very least, to have a working banking system that allows users to create a current and a savings account, and be able to deposit, withdraw and transfer money between accounts real time. As well as a transaction history for each account that is timestamped, as well as robust security. If time permits, we also want to include a AI powered financial advice help desk which would recommend personalised investment advice based on the users account.

1.2 Product value

Describe how the audience will find value in the product.

The target audience will find value in the product because it will offer a straightforward and easy to use application for storing and managing their money. The aim is to have a basic but functional GUI that allows people of all technical skill sets to be able to easily use the application interface.

1.3 Intended audience

Write who the product is intended to serve.

The product is aiming to serve consumers that want a secure and easy to use application to store and transfer their money, whether that be in a savings account or a debit account.

1.4 Intended use

Describe how will the intended audience use this product.

The user will use the application to deposit, withdraw and transfer their money between the different types of accounts that they have open. They can then also view the transaction history of the account and be able to filter the transaction history for their account.

1.5 General description

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

The banking application would have a number of different functions. Most of these would be associated with the transaction element of the application. These would be transferring, depositing and withdrawing funds from the associated accounts at real time. Multiple accounts per user would be another feature of the application, allowing users to have a savings and a debit account for example, offering the user a way to store their savings separately.

2 Functional requirements

List the design requirements, graphics requirements, operating system requirements, and constraints of the product.

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).

The user interface for this application will be as previously mentioned, relatively straightforward. The design will be sleek and straightforward, with emphasis on not overcomplicating and cluttering the interface so everyone can use it with ease. The interface will have limited buttons and will follow a very similar style across all pages of the application.

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.

To reduce the hardware required for the application, we will be using docker combined with fast API.

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

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

4 Non-functional requirements

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

For security, we will be following data protection laws such as GDPR. We will achieve this by firstly limiting the amount of personal information required, as well as encrypting any stored personal information of users. This is so that even in the event of a security breach, any personal data acquired will be useless to the attacker.

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

The storage required for the application will change depending on the amount of customer accounts that are created, as the data for each account is stored in a database. Initially we will only need a small database, however we will need to make sure this can be upscaled in the event of increased customers.

4.3 Compatibility	List the minimum hardware requirements for your software.

In order to run the application, the end user will need a device that is capable of running docker.

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

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

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

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

It should be very easy for end-users to use the software. The design of the interface will be designed with that in mind,by having a limited amount of buttons and those buttons being very clear in what they are used for.

4.8 Other	List any additional non-functional requirements.

5 Definitions and acronyms
