Software Requirements Specification

Banking Automation System

TABLE OF CONTENTS

Section 1. INTRODUCTION

Purpose

The purpose of this document is to give a detailed description of the requirements for the "Banking Automation System" (BAS) software. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

Scope

The "Banking Automation System" (BAS) is an application which helps customers manage their money. It also helps the staff to provide services to the customers as well as manage their accounts. The application will be free to download from the app store and will also have a web interface. Staff will access the system using the staff-portal. A technician also uses the staff-portal in order to administer the system and keep the information accurate. The technician can, for instance, verify staff accounts, and manage their information.

BAS needs the Internet to fetch and display results. Currently, the information is not persisted and therefore you will lose the information at the end of a session.

Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
Customer	Someone who interacts with the application.
Staff	Someone who manages a customer's account.
Staff-Portal	A web application which presents facilities for the staff to perform their duties.
Technician	System admin who has the permission for managing and controlling the system.

Overview

The remainder of this document includes two sections and appendices. The first section provides an overview of the system functionality and system interaction. This section also introduces different types of stakeholders and their interaction with the system. Further, the section also mentions the system constraints and assumptions about the product.

The second section provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

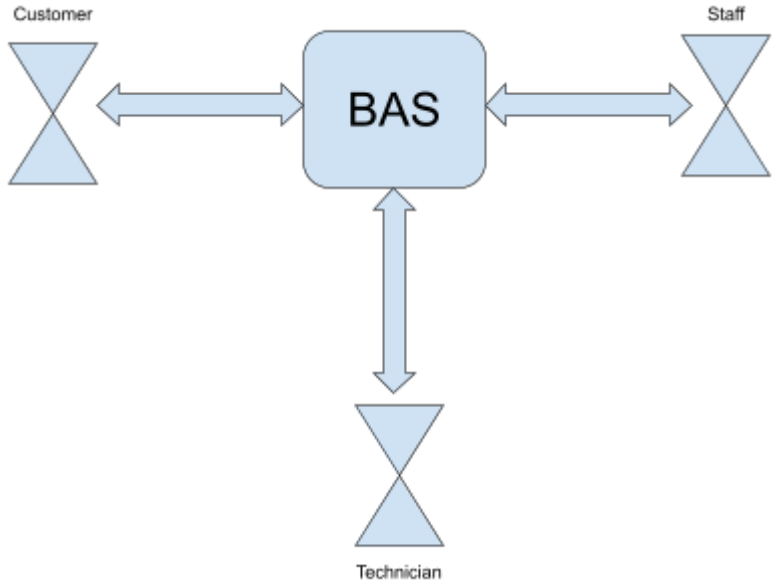
Section 2. General Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with its users and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

Product Perspective

This system will consist of one part: one web application. The application will be used by both the customers and the staff. The staff will have added privileges for managing customer accounts. As well, technicians will manage the access levels to the system. Since BAS is a simple system, we are not storing any customer data in a database. If needed, the user will be able to download information on their local machine. The data will, in no way, persist.

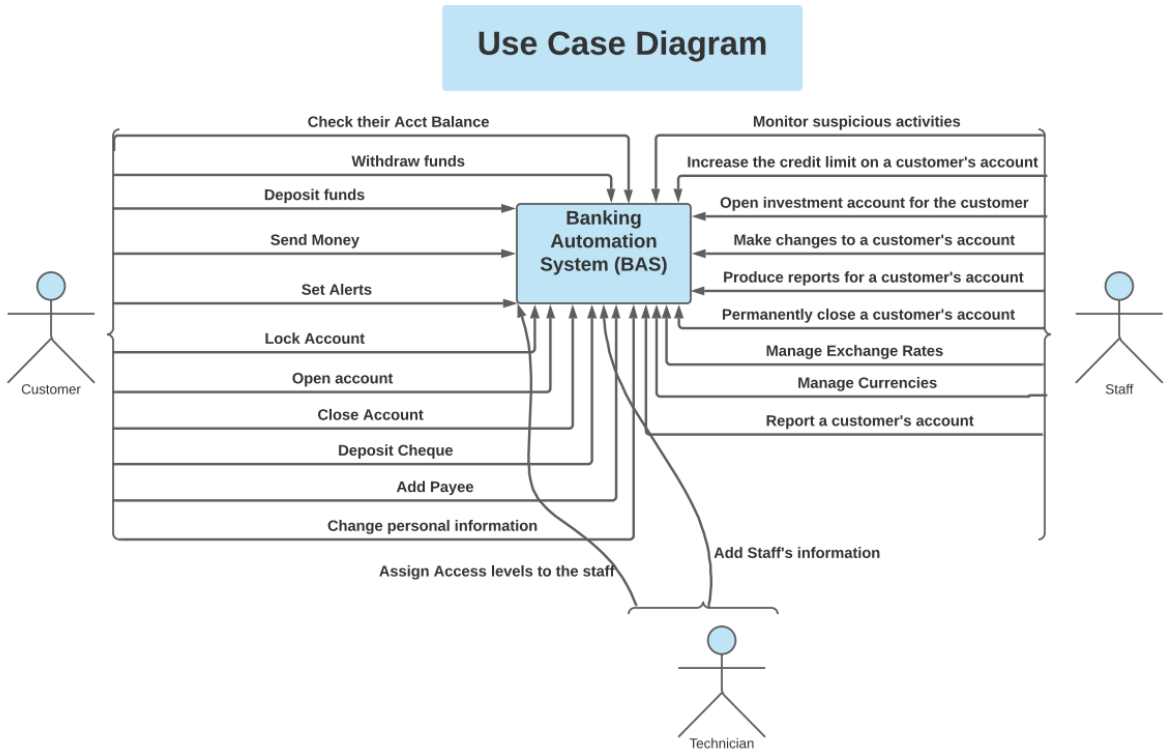
On a high level, the diagram below shows the major components and the interactions.



Product Functions

With the application, the users will be able to search information about their accounts. For a customer, this will include functions such as editing their personal information, opening and closing accounts, depositing and withdrawing money, etc. For the staff, they will have privileges where they can close an account, report an account, create reports, etc. Lastly, the technician will act as an administrator for the entire system.

See below for the use-case diagram of the system:



User Characteristics

There are three types of users that interact with the system: customers, staff and technicians. Each of these three types of users has different use of the system so each of them has their own requirements.

The customer can only use the system to check their own personal account. This means they will not be able to access anything else in the system.

The staff will use the system to monitor customer accounts based on their own access level. For instance, the manager will have higher privilege than a new hire.

The technician will ensure that the system is up to date and also manage the access levels for the staff.

General Constraints

The system will not be persisting any data in a database. Therefore, the user must download any information they want to save on their local machine. The system will also ensure that the same customer is not connected more than once at any given time

Assumptions and Dependencies

One assumption is that the system will always be used on a desktop application that has enough performance. Another assumption is that all desktops work in the same way.

Section 3. Requirements

This section of the SRS should contain all of the software requirements to a level of detail suficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input (stimulus) into the system, every output (response) from the system, and all functions performed by the system in response to an input or in support of an output. As this is often the largest and most important part of the SRS, the following principles apply:

a) Specific requirements should be stated in conformance with all the characteristics described in Annex 1

b) Specific requirements should be cross-referenced to earlier documents that relate.

c) All requirements should be uniquely identifiable.

d) Careful attention should be given to organizing the requirements to maximize readability.

Functional Requirements

Customer

1. *A customer should be able to see their account balance.*
2. *A customer should be able to withdraw funds from their account.*
3. *A customer should be able to add funds to their account..*
4. *A customer should be able to send money to other accounts.*
5. *A customer should be able to set alerts on their accounts.*
6. *A customer should be able to lock their account.*
7. *A customer should be able to open different types of accounts.*
8. *A customer should be able to close accounts.*
9. *A customer should be able to deposit cheque (online) to their account.*
10. *A customer should be able to add a payee to their account.*
11. *A customer should be able to change their personal information (address, etc.)*

Staff

1. *A staff should be able to monitor suspicious activity on a customer's account.*
2. *A staff should be able to increase the credit limit on the customer's account.*
3. *A staff should be able to open an investment account (RRSP, TFSA, etc.) for the customer.*
4. *A staff should be able to make changes to a customer's account.*
5. *A staff should be able to produce reports for a customer's account.*
6. *A staff should be able to permanently close a customer's account.*
7. *A staff should be able to manage exchange rates.*
8. *A staff should be able to manage currencies (if the customer wants physical money)*
9. *A staff member should be able to report a customer's account.*

Technician

1. *A technician should be able to monitor the status of the system.*
2. *A technician should be able to add staff's information (for eg: new hire)*
3. *A technician should be able to assign access levels to the staff depending on their*

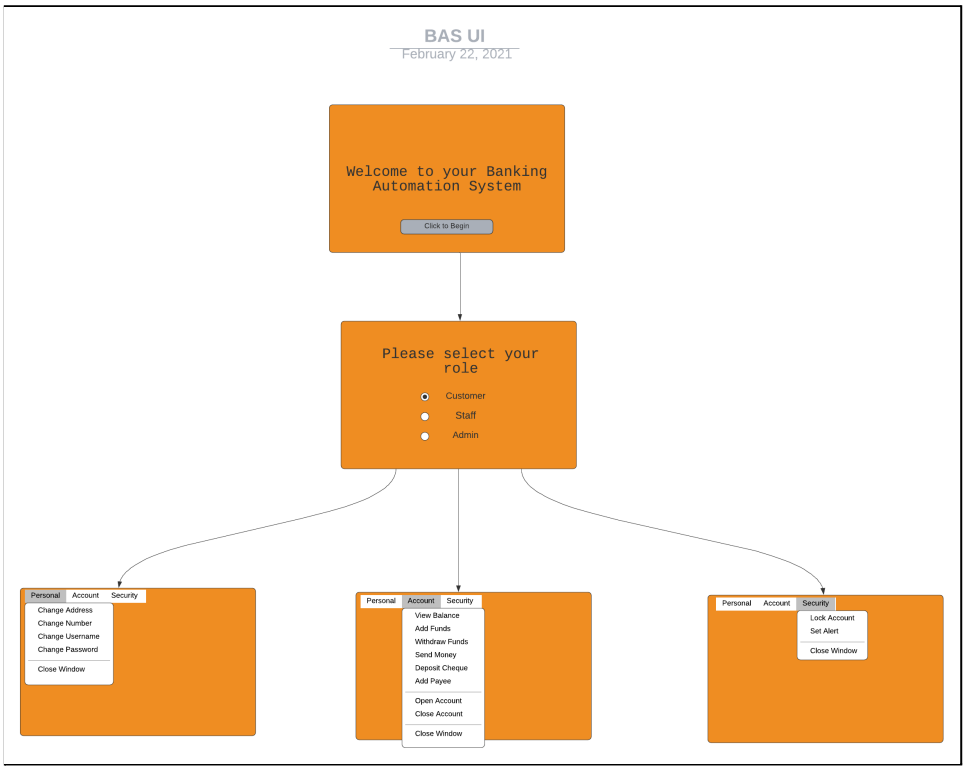
position.

External Interface Requirements

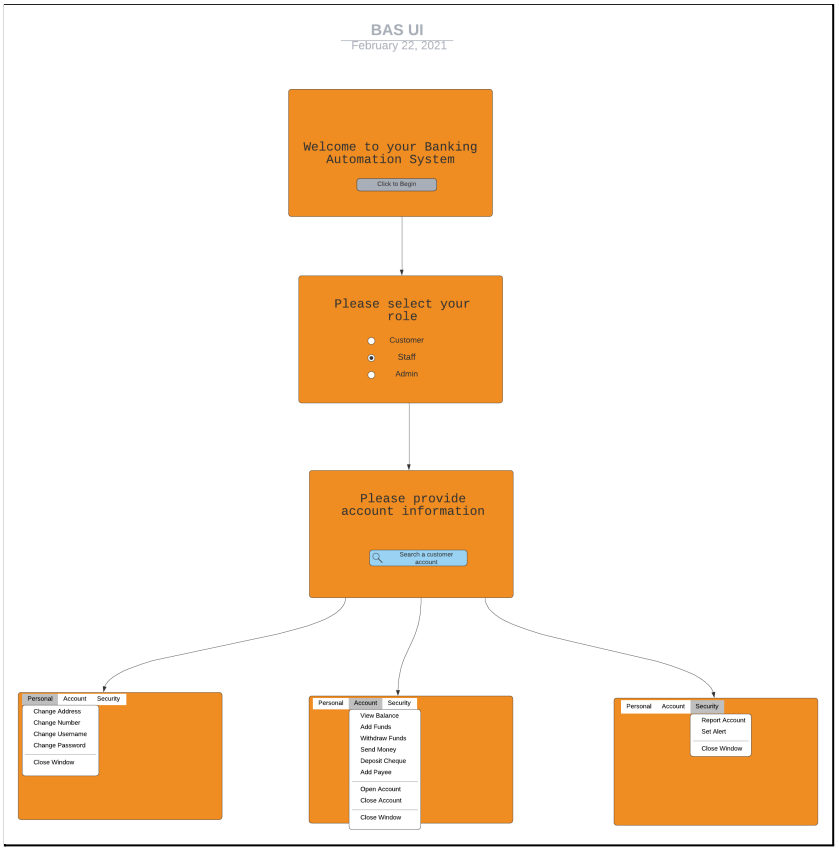
This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

User Interfaces

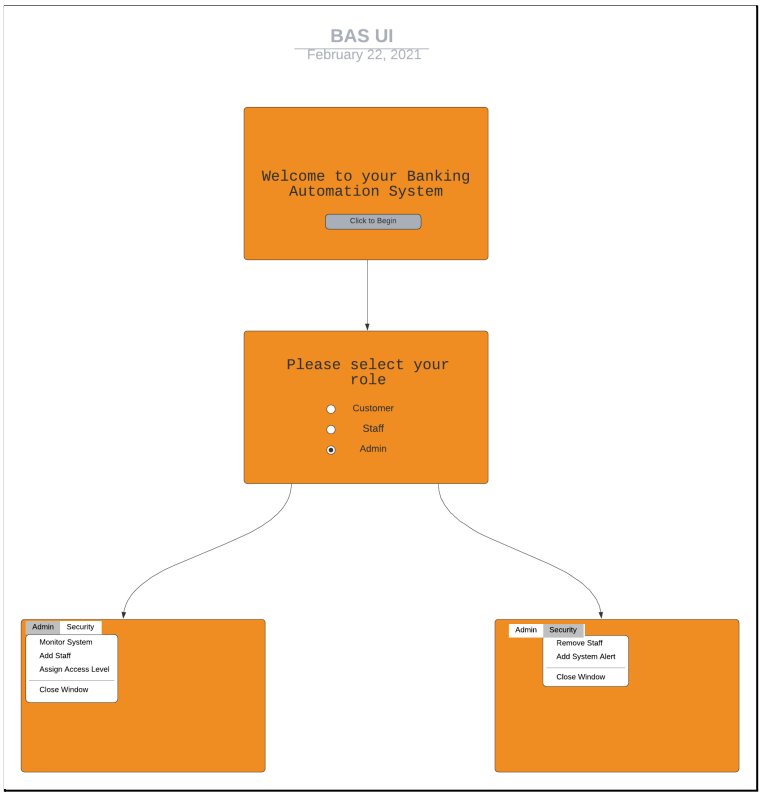
Customer Screen



Staff Screen



Technician Screen



Hardware Interfaces

Since neither the mobile application nor the web portal have any designated hardware, it does not have any direct hardware interfaces.

Software Interfaces

Not Applicable

Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems.

Performance Requirements

Since the system is not going to support simultaneous users, performance is not important.

Design Constraints

This section includes the design constraints on the software caused by the hardware.

Standards Compliance

Not Applicable

Hardware Limitations

Not Applicable

Attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes be specified so that their achievement can be objectively verified.

Availability

This should specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart.

Security

This should specify the factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to:

a) Utilize certain cryptographical techniques;

b) Keep specific log or history data sets;

c) Assign certain functions to different modules;

d) Restrict communications between some areas of the program;

e) Check data integrity for critical variables.

Maintainability

This should specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc. Requirements should not be placed here just because they are thought to be good design practices.

Transferability/Conversion

This should specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include the following:

a) Percentage of components with host-dependent code;

b) Percentage of code that is host dependent;

c) Use of a proven portable language;

d) Use of a particular compiler or language subset;

e) Use of a particular operating system.

Other Requirements

1. The BAS shall be accessible at all times i.e., no down time for the customer, staff, technician.

2. The BAS shall be protected against any physical threat.

3. The BAS shall have backup data accessible at all times.

4. The BAS shall be compliant to all the applicable laws in the relevant jurisdiction.

5. The BAS shall be protected against any cyber threat.

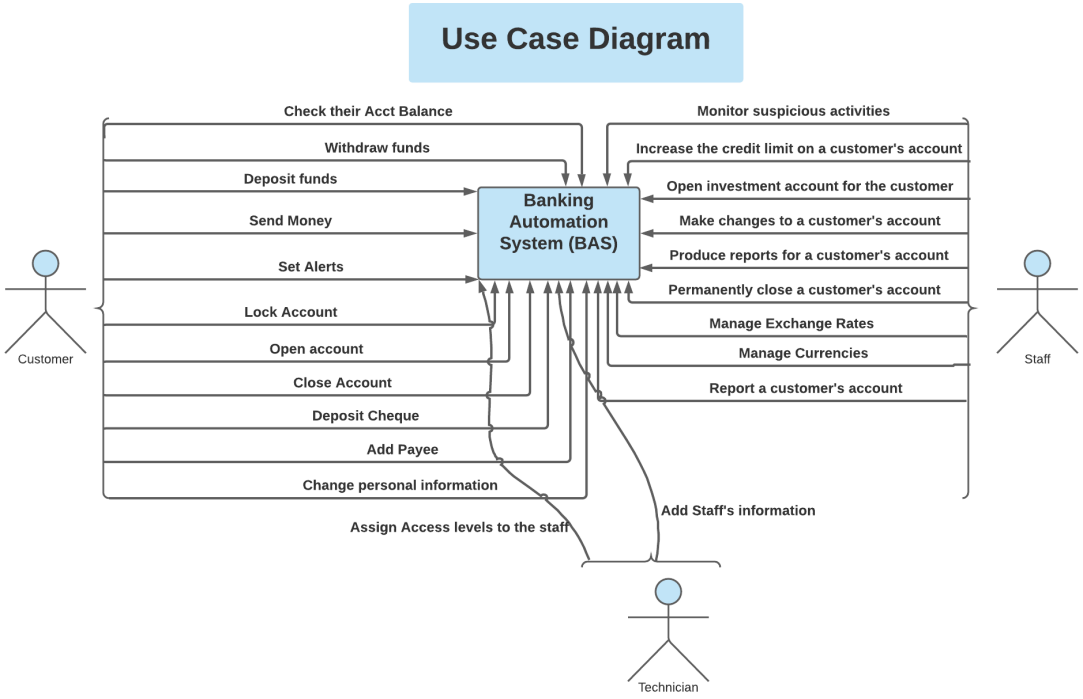
6. The BAS shall provide consistent response times 24x7.

7. The BAS shall be easy to maintain and upgrade.

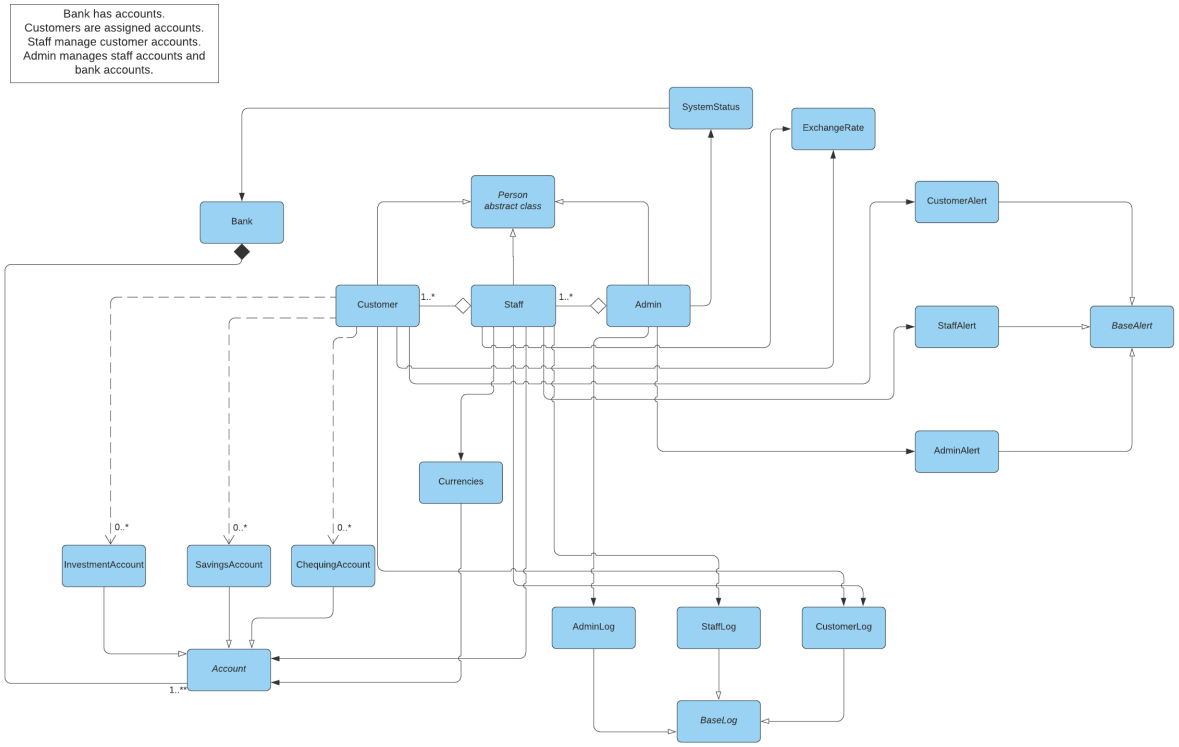
8. The BAS shall be documented thoroughly at all times.

Appendixes

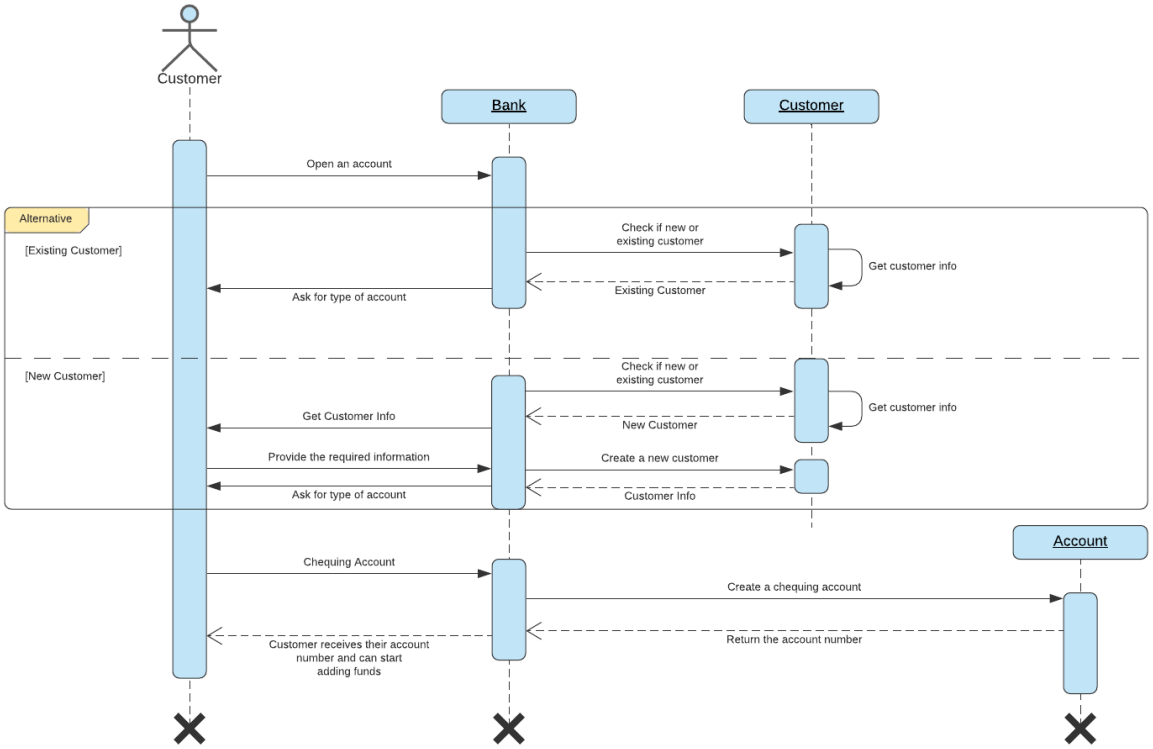
Appendix 1: Use Case Diagram

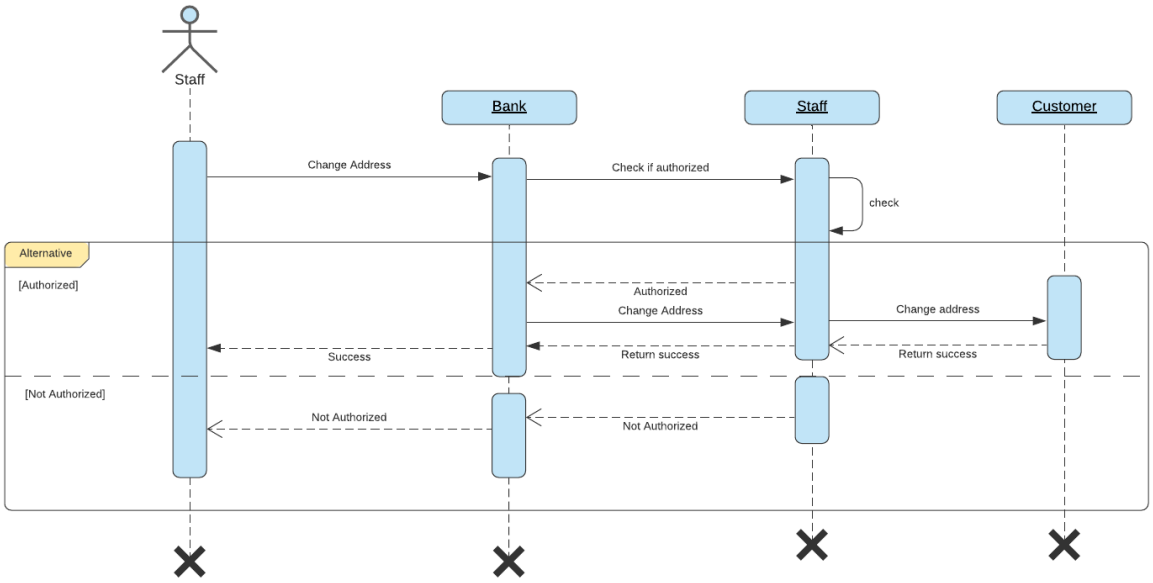


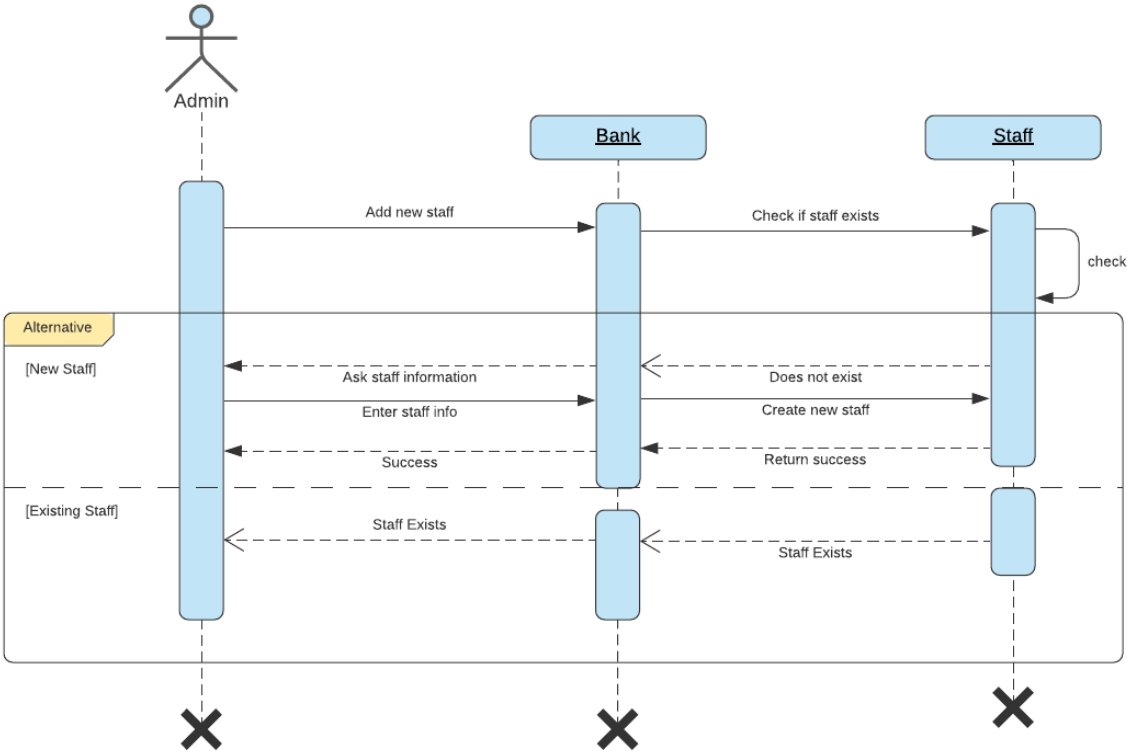
Appendix 2: Class Diagrams



Appendix 3: Sequence Diagrams






Admin
Bank
Staff
Add new staff
Check if staff exists
check
Alternative
[New Staff]
Ask staff information
Does not exist
Enter staff info
Create new staff
Success
Return success
[Existing Staff]
Staff Exists
Staff Exists