

# INTERNSHIP REPORT



## FUND-TRANSFER

At  
SORAMITSU KHMER



ADVISOR: MR LAY VATHNA

National Institute of Posts, Telecoms & ICT

The Report in partial fulfilment of the requirement  
For the degree of Computer Science



By: PHIRAK TEPI SANDAP



ពិភាក្សាលេខាព័ត៌មាន និង កម្មសារ  
នគរបាល និង ក្រសួង ពេទ្យ

## មហាផិភាក្សាប័ណ្ណែងទំនាក់ទំនាក់ នគរបាល និង ក្រសួង ពេទ្យ

### នគរបាល និង ក្រសួង ពេទ្យ

នគរបាល និង ក្រសួង ពេទ្យ

គ្រប់គ្រង នគរបាល និង ក្រសួង ពេទ្យ ... មករា ២០២១

នគរបាល និង ក្រសួង ពេទ្យ

របាយការណ៍ នគរបាល និង ក្រសួង ពេទ្យ

ថ្ងៃទី ៩ មករា ឆ្នាំ ២០២១

របាយការណ៍

សម្រាប់

ប្រធូលីកសារនគរបាល

ឬអនុវត្តន៍យុទ្ធសាស្ត្រនគរបាល និង ក្រសួង ពេទ្យ

របាយការណ៍

ស្តីពី

នគរបាល និង ក្រសួង ពេទ្យ

ឬអនុវត្តន៍យុទ្ធសាស្ត្រនគរបាល និង ក្រសួង ពេទ្យ

របាយការណ៍



National Institute of Posts, Telecoms & ICT

## School of Computer Science

A Thesis submitted by:

**Ms. Phirak Tepisandap**

**Defense Date: ... January 2021**

**Thesis defense authorization**

**President:** \_\_\_\_\_

**January \_\_\_\_\_, 2021**

**Topic:** **Fund Transfer System**

**Enterprise:** **Soramitsu Khmer**

**Advisor:** **Mr. Lay Vathna**

**Supervisor:** **Mr. Vandy Sodanheang**

**Phnom Penh**



## ACKNOWLEDGE TO

First of all, I would like to dedicate my wholehearted thanks to Soramitsu Khmer for giving me such a great opportunity to learn and experience the working environment. Therefore, I consider myself as a very lucky person since I was provided an opportunity to be part of the company. I am also grateful to meet many amazing people with professional skills which led me through this internship program. In addition I would like to express my sincere gratitude to my principal, Dr. SENG Sopheap, President of National Institute of Posts, Telecoms and ICT, and all of my respectful lecturers for the patient guidance, encouragement, and advice throughout my academic years. I feel blessed that I have such an array of great lecturers helping me to become a good human being in society and a good software developer in the industry.

Secondly, I would like to specially thank Mr. LAY Vathna, my valuable internship advisor, for the advice and guidance he gave about materializing the final thesis of my senior year. Also, provided various guidance for this internship. Moreover, I was able to perform work smoothly in an organized way thanks to the knowledge that he imparted to me during class.

Last but not least, I would like to take this opportunity to express my deepest thanks to Mr. VANDY Sodanheang, senior Software developer at Soramitsu Khmer, for his effort in supervising and mentoring me on what it takes to be a professional software developer. He willingly guided and listened to my concerns so that I was able to complete the required work in time and experience knowledge that is useful for my future career.



# មុន្តនៃយោបាយ



# ABSTRACT OF

---

This thesis describes my work during the four-month internship at Soramitsu Khmer as a junior software developer. This internship is a crucial part of the Bachelor of Computer Science's curriculum at NIPTICT. The internship started from 15<sup>th</sup>-July to 15<sup>th</sup>-November 2020. The topic of the internship is Fund-transfer, a state-of-the-art system built for the National Clearing House focused on providing central clearing and settlement of non-urgent interbank payment transactions by checks, denominated in KHR and USD.

So, I joined Soramitsu Khmer as a full time junior developer in early July and I was immediately assigned with this project by my project manager, Ms. Prum Sophea, and provided with requirements for this project. My main responsibilities was backend development of the system which involves creating the necessary API using Spring boot, Kotlin runs on JVM and tests on the postman application as well as automate tests. Over the course of this project, I managed to handle the full power of Spring Ecosystem by diving deeper into advanced features such as Spring Data, JPA and Unit test integration and also improve exponentially on the backend skill that i previously lacked in such as Kotlin and new backend technology called Spring Boot which is great for building production-ready. On top of that, I experienced realistic Scrum Methodology being applied to development work and gained many useful soft skills which are vital to my professional growth. Indeed, I faced some challenges during the internship such as pressure and lack of technical knowledge but I successfully overcame them by consistently working hard to find creative solutions, learning new things and adapting myself to the work environment and diverse communication.

By the end of the internship program, I have completed all my assigned responsibility and I hope that this system will provide convenience to users everyday and make the National Clearing House ecosystem of Cambodia grow.



# TABLE OF CONTENTS

ACKNOWLEDGE TO .....	1
មុលន័យ សង្គប .....	2
ABSTRACT OF .....	3
TABLE OF CONTENTS .....	4
ABBREVIATION SYMBOLS .....	6
LIST OF FIGURES .....	7
LIST OF FIGURES .....	8
<b>I. INTRODUCTION .....</b>	<b>10</b>
1. Presentation of the Internship .....	10
1.1 Objective .....	10
1.2 Duration .....	10
2. Presentation of the Organization .....	10
2.1 Background .....	10
2.2 Service .....	11
2.3 Address and Contact .....	11
2.4 Organization Chart .....	11
<b>II. PRESENTATION OF THE PROJECT .....</b>	<b>13</b>
1. General Presentation of the project .....	13
2. Problematic .....	13
3. Objective .....	13
4. Development Methodology .....	14
5. Project Plan .....	15
<b>III. ANALYSIS &amp; GENERAL CONCEPTS .....</b>	<b>17</b>
1. Requirement .....	17
1.1 Functional Requirements .....	17
1.2 Non-functional Requirements .....	18

2. Analysis of Project .....	19
2.1 Use Case Diagram .....	19
2.2 Activity Diagram .....	20
a. Account Type activity .....	20
b. Account Class activity .....	21
c. Holiday activity .....	22
2.3 Database Design .....	23
<b>IV. DETAIL CONCEPTS .....</b>	<b>25</b>
1. Choice of Technologies .....	25
1.1 Spring Boot .....	25
1.2 Docker .....	25
1.3 Liquibase .....	26
2. Language .....	26
2.1 Kotlin .....	26
3. Tools .....	27
<b>V. IMPLEMENTATION .....</b>	<b>29</b>
1. Structure .....	29
2. Application Configuration .....	31
3. System Architecture .....	32
1.1 Physical Architecture .....	32
1.2 Logical Architecture .....	33
1.3 Feature Implementation .....	34
<b>VI. CONCLUSION .....</b>	<b>37</b>
1. Result .....	37
2. Strong points and Weak points .....	38
3. Difficulties .....	38
4. Experiences .....	38
5. Perspective .....	39
6. Conclusion .....	39
<b>LIST OF REFERENCES .....</b>	<b>40</b>
<b>ANNEX INDEX 01 &amp; 02 .....</b>	<b>41</b>
<b>ANNEX INDEX 03 &amp; 04 .....</b>	<b>42</b>
<b>ANNEX INDEX 05 &amp; 06 .....</b>	<b>43</b>



## ABBREVIATION SYMBOLS

- 
- API : Application Program Interface
  - NBC : National Bank of Cambodia
  - FI : Financial Institution
  - NCH : National Clearing House
  - KHR : Khmer Riel
  - USD : United States Dollar
  - NCS : National Clearing System
  - CEC : Check Exchange Center
  - SQL : Structured Query Language
  - IDE : Integrated Development Environment
  - MVC : Model, View, Controller



# LIST OF FIGURES

---

Figure 1: Soramitsu Khmer Co., Ltd Logo .....	10
Figure 2: Soramitsu Khmer Organization .....	11
Figure 3: Scrum Methodology .....	14
Figure 4: Use case diagram .....	19
Figure 5: Account Type activity .....	20
Figure 6: Account Class activity .....	21
Figure 7: Holiday activity .....	22
Figure 8: Database design .....	23
Figure 9: Spring Boot Logo .....	25
Figure 10: Docker's Logo .....	26
Figure 11: Liquibase's Logo .....	26
Figure 12: Kotlin's Logo .....	26
Figure 13: Spring Initializr .....	29
Figure 14: Project Structure .....	30
Figure 15: Project Dependencies .....	31
Figure 16: Physical Architecture .....	32
Figure 17: Spring boot layer .....	33
Figure 18: Spring boot logical architecture .....	34
Figure 19: Feature Holiday .....	35



# LIST OF TABLES

---

Table 1: Project Planning table .....	15
Table 2: Functional Requirements table .....	17
Table 3: Non-functional Requirements table .....	18
Table 4: Tools' table .....	27
Table 5: Results table .....	37
Table 6: Strong and Weak points table .....	38

# I. INTRODUCTION



## I. INTRODUCTION

### 1. Presentation of the Internship

#### 1.1. Objective

As a fourth-year student in the Computer Science Department at National Institute of Posts, Telecoms and ICT (NIPICT), students are required to do an internship for 4 months. The internship can be conducted in a public institution , organization or private company. The objective is to help students to apply their talent and to gain more experience in a real working environment. Moreover, they at least have to do one project in a company in order to defend their thesis. At the end of the internship, they need to submit a report about their project and prepare a presentation to defend their thesis.

#### 1.2. Duration

The duration of the internship at Soramitsu Khmer is about 4 months starting from 15th July to 15th November 2020, which suits the internship requirement.

### 2. Presentation of the Organization

#### 2.1. Background

Soramitsu Khmer Co., Ltd is a startup company in Cambodia in late 2018 to develop a technology related to many projects in the country and overseas. Soramitsu Khmer Co., Ltd is affiliated with those in Japan, USA and Eastern European companies. The aim of the company is to be a leader in technology innovation and development for enterprises and help build the Fintech and code writing community in Cambodia.



Figure 1: Soramitsu Khmer Co., Ltd Logo

## 2.2. Service

Soramitsu provides good focus mostly on mobile applications iOS and Android. Main customers such as National Bank of Cambodia, The University of Tokyo, Rakuten Securities Digital identity, National Settlement Depository, SOMPO Insurance, Indonesia Banking Group Digital identity.

## 2.3. Address and Contact

Soramitsu Khmer is reachable through contact information below:

**Address :** The Elysées Building, Street Kos Pich, Promenade Des Champs Elysées, No B2-011

**Tel** : +(855)-61 77 33 96

**Email** : [sotheavy@soramitsukhmer.com](mailto:sotheavy@soramitsukhmer.com)

**Website** : <https://soramitsu.co.jp/en/>

## 2.4. Organization Chart

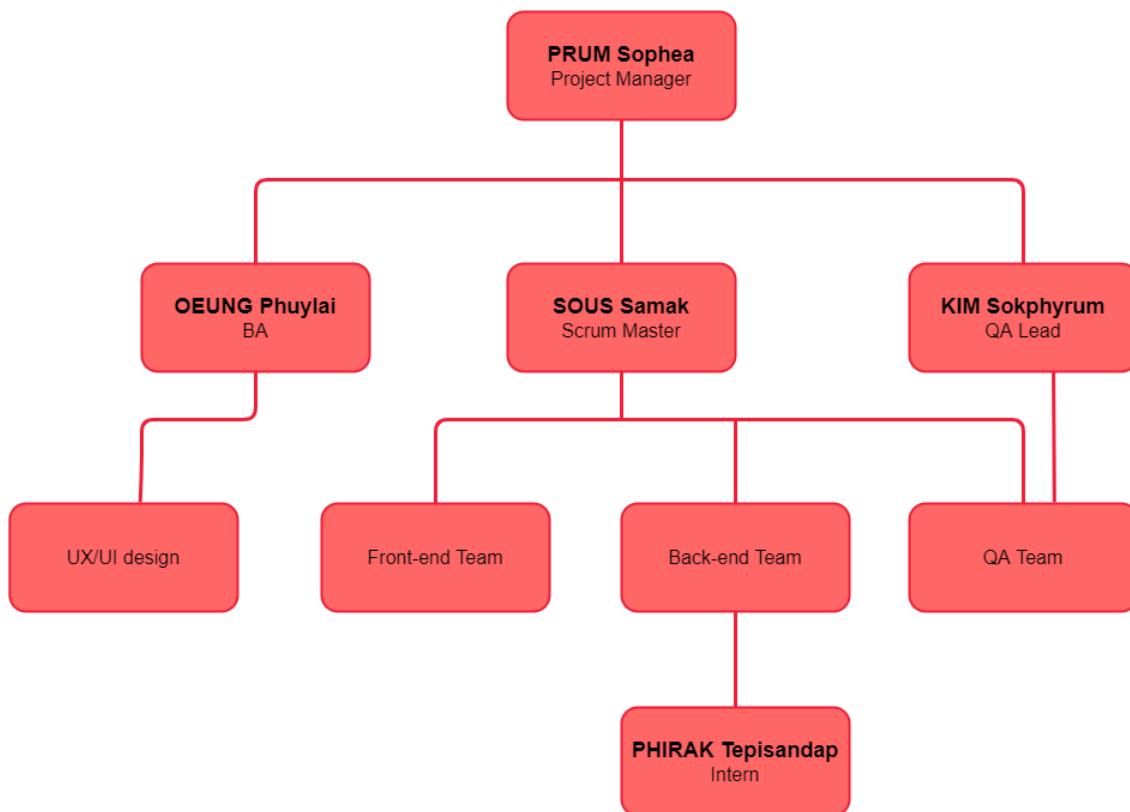


Figure 2: Soramitsu Khmer Organization

## **II. PRESENTATION OF THE PROJECT**



## II. PRESENTATION OF THE PROJECT

### 1. General Presentation of the project

The system is about check clearing which is a nationwide batch-oriented electronic funds transfer system operated by the National Clearing House (NCH), which is located on the premises of National Bank of Cambodia (NBC). The system provides central clearing and settlement of non-urgent interbank payment transactions by checks, denominated in KHR and USD. It is a batch-processing store and forward system. Rather than processing each transaction individually, transactions received by NCS members are stored and processed later in batch mode. The clearing and settlement process is done during a predetermined time period, which is set by NBC.

### 2. Problematic

There are a few problems that lead to the development of this project. First, in the case of check clearing, the exchange of physical checks and returned checks for the members is conducted at the check exchange center which is located in the National Clearing House, and the distances between financial institutions and National Clearing House are sometimes far. Without the system it can possibly cause traffic accidents along the way. Secondly, there is also another problem that is known as the failure of the process of the settlement checks in which it requires a huge amount of time to settle without proper scheduling. Thirdly, in terms of technology the old system is low performance which is hard to scale and maintenance. Last but not least, in terms of user experience, nowadays the system still lacks in design and that is what attracts and shapes better user journeys.

### 3. Objective

To solve those problems above, the client proposes the Fund transfer system with many useful functions including an auto settlement process which allows FI to submit checks through the system. In this case, it also prevents various disasters caused by the manual settlement process. Moreover, we develop a new system with microservice architecture which makes it

much easier to scale and maintain. So users can use the system with understandable interaction design based on the advantages of smooth client-side and server-side browsing that build to bridge the gap between great user experience and seamless browsing. Most importantly, we use the new technology which is the combination of Spring Boot and Microservice-Architecture design pattern. This leads the Fund transfer system to establish a high level security layer. Ultimately, we believe that the new system can fulfill what the old system has lacked over the years.

## 4. Development Methodology

To effectively initiate a project, it is essential to carefully choose a well-suited and effective type of methodology. Hence, Scrum methodology was specifically selected due to the nature of its flexibility and ease in adoption method as it supports a great variety of different types of project.

Agile Software Development with Scrum is usually considered as a methodology. The project manager plays a role as a Scrum Master. Each scrum lasts the period of one to two weeks and we carry out a small meeting about the process of each developer for approximately 10 minutes every morning.

Each task was assigned by the project manager and was split into sprints in total that one sprint took about 2 weeks. We always have our stand-up meeting every morning around 5-10 minutes to discuss our previous tasks, challenges, solutions, struggles and the planning for upcoming tasks.

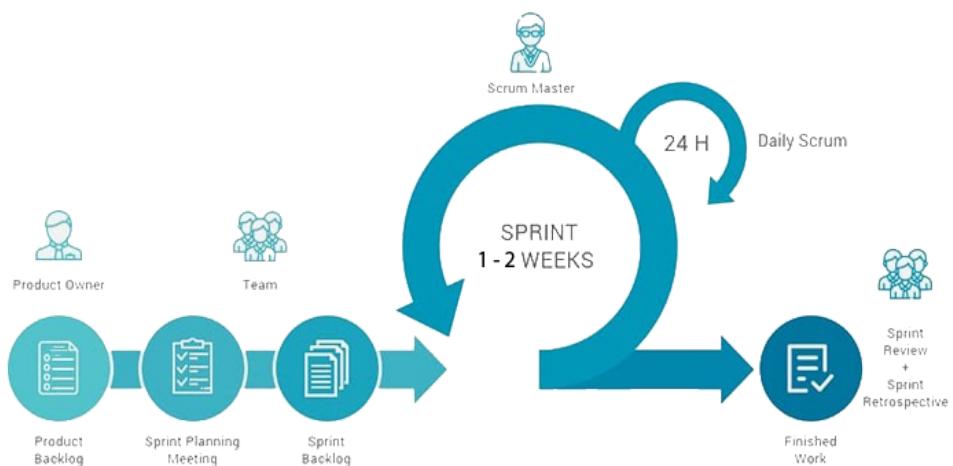


Figure 3: Scrum Methodology

## 5. Project Plan

The following table demonstrates my plan during the 16 weeks of the Fund Transfer project that uses Scrum methodology as I have described above.

Activity/Task	Sprint Break Down							
	1	2	3	4	5	6	7	8
Study Requirement	Red	Light Red	Light Red	Light Red	Light Red	Light Red	Light Red	Light Red
Study new technology	White	Red	Red	White	White	White	White	White
Implementation	White	White	Light Red	Red	Red	Red	Light Red	Light Red
Testing	White	White	White	White	White	Red	Red	White
Deployment	White	White	White	White	White	White	Light Red	Red

*Table 1: Project Planning table*

# III. ANALYSIS & GENERAL CONCEPTS



## III. ANALYSIS & GENERAL CONCEPTS

### 1. Requirement

In order to understand the objective of the project above, We need to list out and briefly describe each of all requirements of the project.

#### 1.1. Functional Requirements

The Functional requirements are the core functionalities needed to be implemented in order to make the system work. The details of these functionalities are below:

Functional	Requirement
Account Type	User can View, Add, Edit Deactivate, Activate
Account Class	User can View, Add, Edit, Deactivate, Activate, Select an Account class for settlement
Holiday Type	User can View list of holiday types such as Core Holiday, HQ Holiday and Branch Holiday
Holiday for core	User can View, Add, Edit, Deactivate, Activate a template for holiday of Core
Holiday for member	User can View, Add, Edit, Deactivate, Activate holiday of HQ
Holiday for branch	User can View, Add, Edit, Deactivate, Activate holiday of Branch

Table 2: Functional Requirements table

## 1.2. Non-functional Requirements

Apart from the functional requirements, it is crucial to consider as well about the non-functional requirements as it would contribute largely to the enhancement of the system quality and usability. The non-functional requirements include:

Non-functional	Requirement
UX/UI	Must be simple, intuitive and user-friendly so that users can navigate on our system smoothly.
Performance	All steps in developing new features must give the best performance because it is vital to make the system run quickly and efficiently without any errors.
Maintainability	The system must be programmed in an easy way and trouble-free for the next developers to maintain.
Scalability	The system architecture must be carefully designed in regard to the scalable nature so that new changes or adjustments can be performed in the future seamlessly and effectively.
Fault Tolerance	The system must have the ability to continue operating properly without interruption when one or more of each service fails.

*Table 3: Non-functional Requirements table*

## 2. Analysis of Project

### 2.1. Use Case Diagram

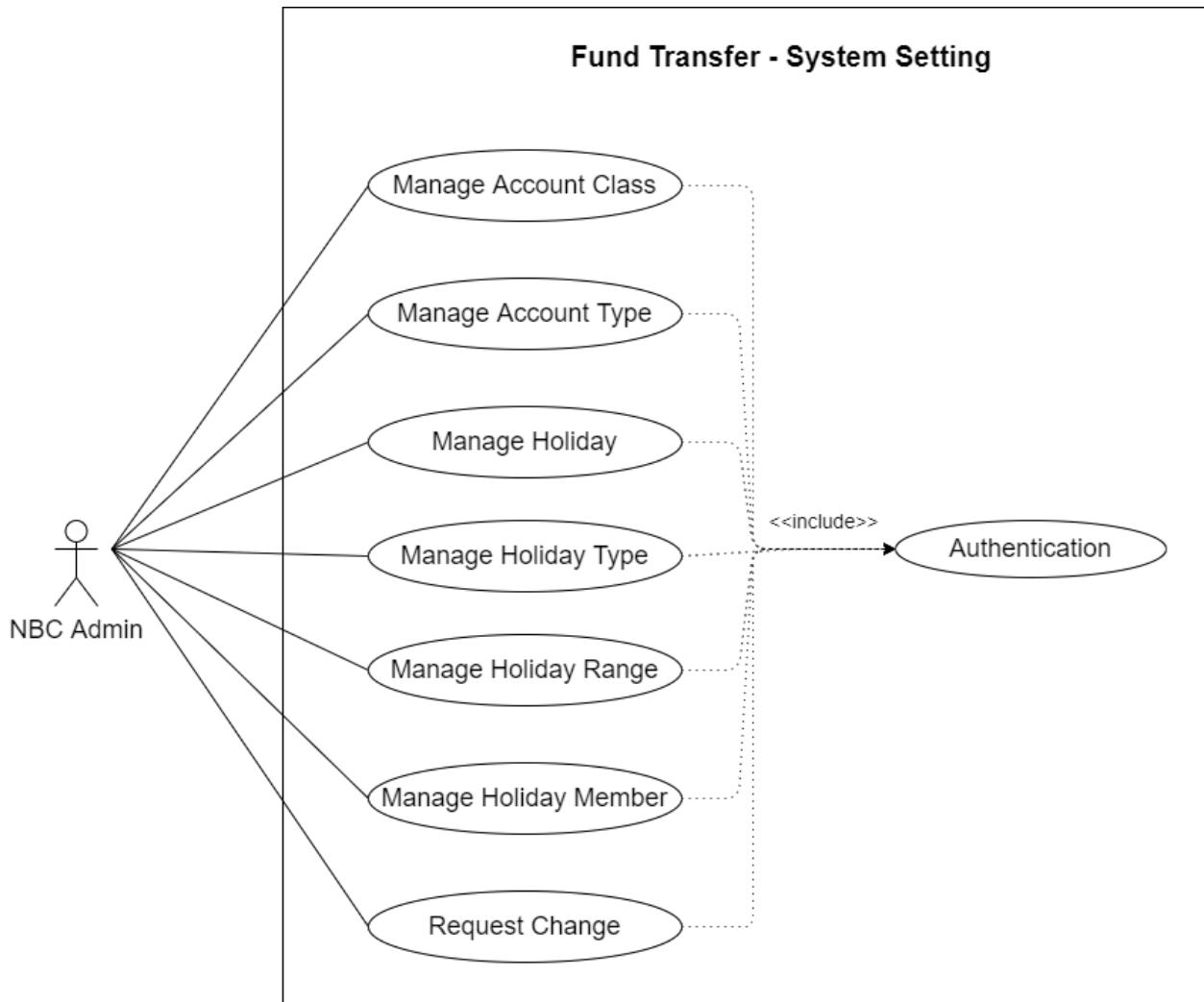


Figure 4: Use case diagram

The figure illustrates the use case diagram in our system, we are listing down the major use cases that the actor can perform. NBC Admin is the most basic type of user as NBC Admin can perform more core activities on top of other user's activities. The functions allow NBC Admin to manage account type, account class, holiday such as CRUD operation which are creating new data, get lists of each data and update existing data but for delete operation we use the soft delete method to protect data loss. To manage these functions, we need to input the valid information and complete all the required fields. They can also set settlement in account type. This functionality allows the user to set settlement scheduling for each currency in the system. Last but not least, the user must perform request changes in order to create and update data by

selecting one authorizer. But keep in mind, to perform all of these activities the user is first required to complete the authentication process as NBC Admin which is the fundamental step.

## 2.2. Activity Diagram

### a. Account Type activity

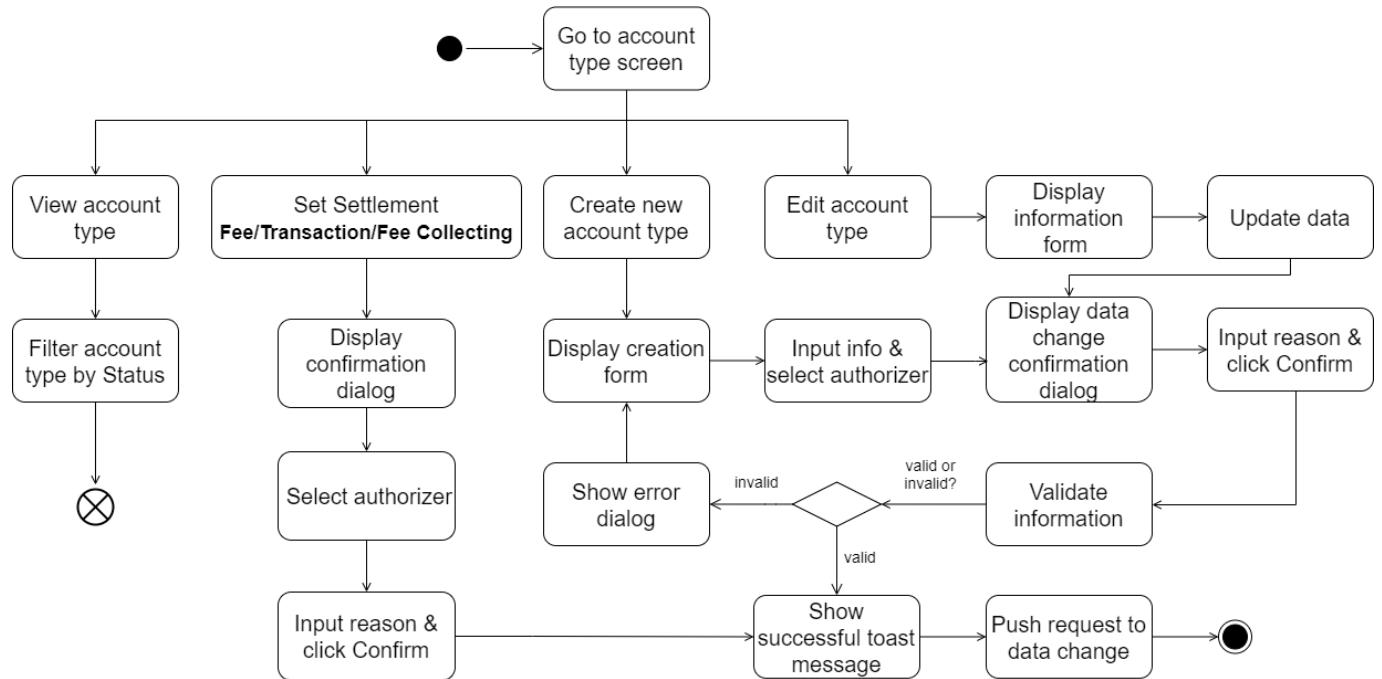


Figure 5: Account Type activity

A user can view the list of account types after sign-in as NBC admin. In this feature user can filter account type by status of active or inactive. And they can create a new account type by clicking the Create button and fill in the form, selecting one Authorizer then click Create. After that NCS displays the confirmation dialog form for the user to input the reason and confirm. If the information is valid the system will show a successful message and push requests to data change. If not, the user will need to input information again. They can also set any account type for settlement then fill in the reason for data change to complete action of this process. There are three types of settlement accounts. Firstly, it is the transaction settlement account. This account settlement is for NBC Admin to set which account type should be used to settle the transactions that were made before each settlement time. Then, the account type that has been set as account transaction settlement, will be updated accordingly to its net position. Secondly, it is the fee settlement account. This account settlement is used to notify banks about which account type will be active as receiver of the fee when the transaction was made by their customer. Lastly, it is the fee collecting which is used for NBC as the receiver of the fee from each bank that runs the transactions via the central bank. Furthermore, the user can edit an existing account type by

clicking the Edit icon then updating the data, selecting one Authorizer and clicking Confirm once the information form displays. The request will be sent to data change if the information is valid.

### b. Account Class activity

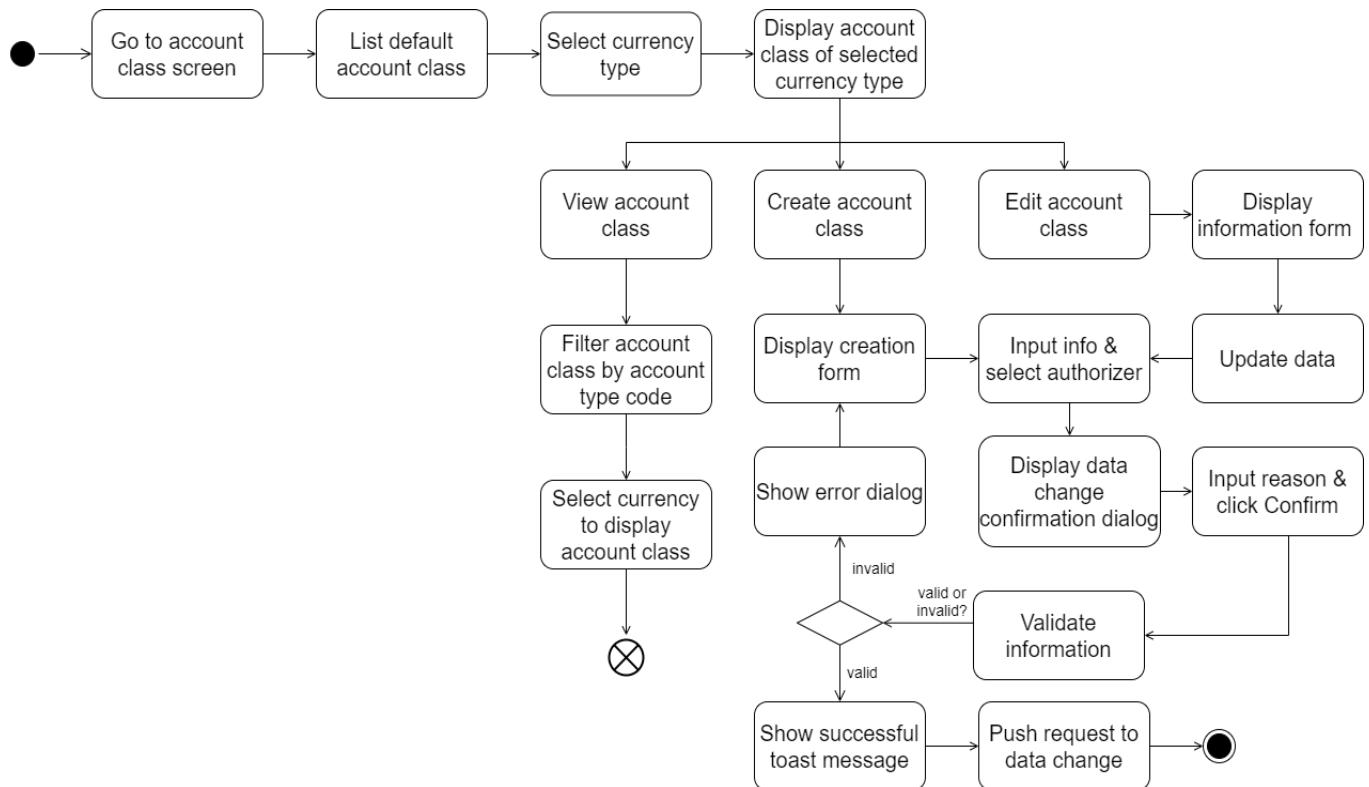


Figure 6: Account Class activity

A user goes to the Account Class screen and selects a currency type to display all account classes of selected currency type. In this feature user can filter account class by account type code. To create a new account class, NBC admin needs to click the Create button and fill in the form, selecting one Authorizer then clicking Create. After that the system displays the confirmation dialog form for a user to input the reason and click Confirm. If the information is valid the system will show a successful message and push requests to data change. If not, the user will need to input information again. NBC admin is able to edit an existing account class by clicking the Edit icon then updating the data, selecting one Authorizer and clicking Confirm once the information form displays. The request will be sent to data change if the information is valid.

### c. Holiday activity

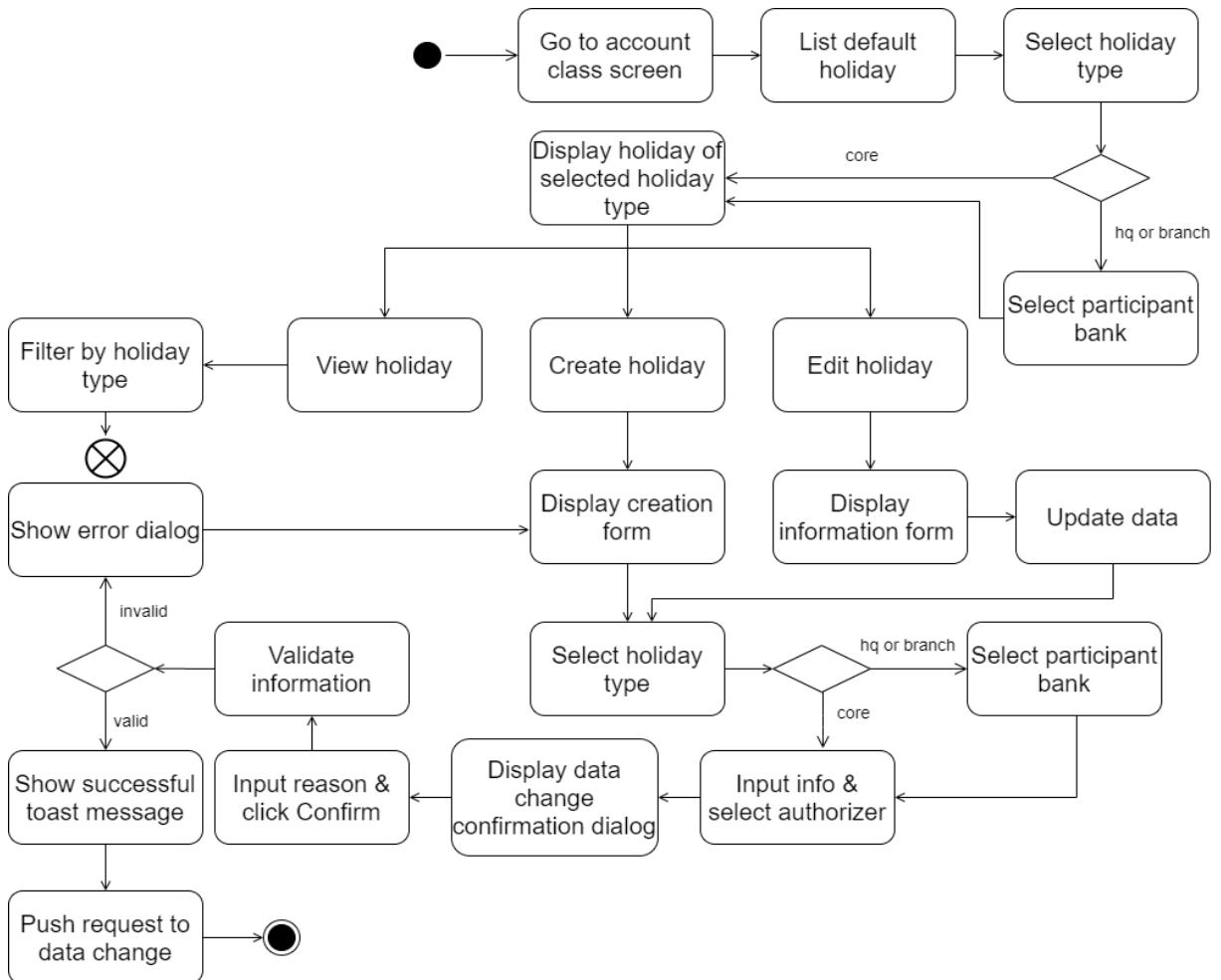


Figure 7: Holiday activity

There are three holiday types such as Core Holiday, HQ Holiday and Branch Holiday. NBC admin can filter holidays by selecting holiday types or participant banks. Holiday Member is the holiday that a user sets for HQ or Branch banks. In this feature user is able to create a new core holiday by selecting Core Holiday in holiday type then fills in information and select an Authorizer to perform data change process. In this case if the information is valid, the system will toast a successful message and push the request to data change. If not, it will toast the error message and redirect to the creation form dialog. To create a holiday for bank members, a user can just select holiday type to HQ or Branch holiday then choose one participant bank for that holiday and complete the data change request step. Moreover, NBC admin can also edit existing holidays by clicking on the edit button, update the fields as required and select one Authorizer. If the information is valid, it will push the request to data change and toast a successful message. If not, it requires the user to input information again. In addition, the holiday of a member refers to the holiday of the bank so there are no transactions allowed for this particular member for inward

and outward during holiday. And holiday of a branch means the bank does not participate in clearing for the particular CEC which has no transactions allowed for this particular CEC. On top of all holidays, core holiday refers to public holidays in the country so the NBC admin user will maintain the holiday in the system.

### 2.3. Database Design

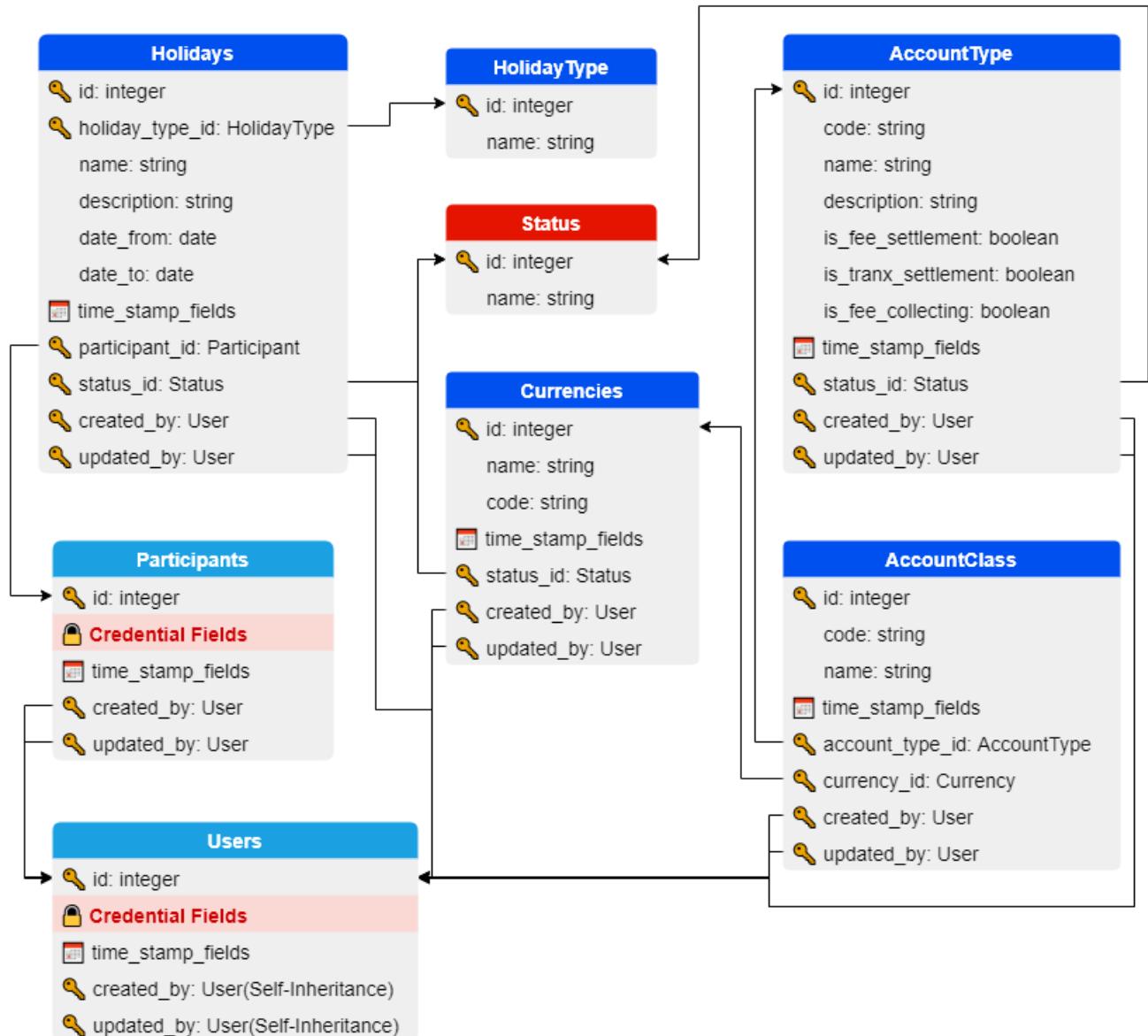


Figure 8: Database design

# IV. DETAIL CONCEPTS



## IV. DETAIL CONCEPTS

### 1. Choice of Technologies

#### 1.1. Spring Boot

First of all, a software product must meet all the requirements of the customer or end-user. The cost of developing and maintaining software should be reasonably priced and the development of software should be smooth. For this reason, **Spring Boot** is our solution for creating a Fund Transfer System that embedded web servers to make microservice development with responsive web applications and secure.



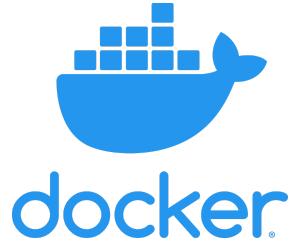
*Figure 9: Spring Boot Logo*

Spring Boot is an open source Java-based framework used to create Micro Services. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications. In the Fintech field, it is widely used the most since it has a proven track record of dealing with security issues quickly and responsibly. Third-party dependencies are also monitored closely, and regular updates are issued to help keep data and applications as safe as possible. In addition, Spring Security makes it easier to integrate with industry-standard security schemes and deliver trustworthy solutions that are secure by default.

#### 1.2. Docker

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Multiple languages, frameworks, architectures, and discontinuous interfaces between tools for each lifecycle stage creates enormous complexity. Docker simplifies and accelerates the

workflow, while giving developers the freedom to innovate with their choice of tools, application stacks, and deployment environments for each project.



*Figure 10: docker's Logo*

### 1.3. Liquibase

Liquibase is an open-source database schema management solution which enables developers to manage revisions of database changes easily. It controls when, where, and how database changes are deployed. In case there are some issues related to database changes, we can undo database changes, either automatically or via custom rollback SQL.



*Figure 11: Liquibase's Logo*

## 2. Language

### 2.1. Kotlin

Kotlin is a modern programming language which is a cross-platform, statically typed, general-purpose programming language with type inference. Kotlin is designed to interoperate fully with Java, and the JVM version of its standard library depends on the Java Class Library, but type inference allows its syntax to be more concise. Most importantly developers can avoid entire classes of errors such as null pointer exceptions and choose any Java IDE such as (IntelliJ idea, Android Studio, Eclipse) or build from the command line.



*Figure 12: Kotlin's Logo*

### 3. Tools

To facilitate the project development, we have used some other technology tools as follows:

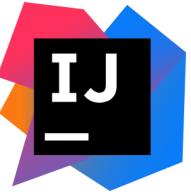
	IntelliJ IDEA is an integrated development environment written in Java for developing computer software. While IntelliJ IDEA is an IDE for Java, it also understands and provides intelligent coding assistance.
	DataGrip is a database management environment for developers. Databases can work locally, on a server, or in the cloud. Supports MySQL, PostgreSQL, Microsoft SQL Server, Oracle, and more.
 <b>POSTMAN</b>	Postman is a software development tool. It enables people to test calls to APIs. Postman users enter data. The data is sent to a special web server address.
 <b>Stoplight</b>	Stoplight is an all-in-one platform purpose-built for the design, management, and discovery of API designs and other related artifacts.
 <b>ClickUp</b>	ClickUp is a cloud-based collaboration and project management tool suitable for businesses of all sizes and industries.

Table 4: Tools' table

# V. IMPLEMENTATION



## V. IMPLEMENTATION

### 1. Structure

In order to build the project, we need to initialize the project using the Initializr website by visiting [start.spring.io](https://start.spring.io) and choose Kotlin language. Gradle is the most commonly used build tool in Kotlin, and it provides a Kotlin DSL which is used by default when generating a Kotlin project, so this is the recommended choice. But you can also use Maven if you are more comfortable with it. First, select “Gradle Project” or let the default “Maven Project” depending on which tool you want to use. Then enter the following artifact such as type of project, Language, Spring Boot version, Project Metadata, Packaging , Java version and lastly add dependencies field. Finally, your browser should now be in this state:

The screenshot shows the Spring Initializr web application at <https://start.spring.io>. The configuration is as follows:

- Project:** Gradle Project (selected)
- Language:** Kotlin (selected)
- Spring Boot:** 2.2.11 (selected)
- Project Metadata:**
  - Group: com.soramitsu
  - Artifact: system\_setting
  - Name: system\_setting
  - Description: System Setting Service for Fund Transfer
  - Package name: com.soramitsu.system\_setting
  - Packaging: Jar (selected)
  - Java version: 11 (selected)
- Dependencies:**
  - Spring Web**: WEB (selected)
  - Spring Data JPA**: SQL
  - Spring Security**: SECURITY
  - Eureka Discovery Client**: SPRING CLOUD DISCOVERY
  - H2 Database**: SQL
  - Oracle Driver**: SQL
- Buttons at the bottom:** GENERATE (CTRL + D), EXPLORE (CTRL + SPACE), SHARE...

Figure 13: Spring Initializr

After that click on GENERATE, this will download a zip file containing a Gradle project in the root directory. A typical project generated by Spring Initializr contains a Spring Boot application, a test and an empty configuration. If you run the main method of the application, you'll see an "empty" spring boot app starting on localhost:8080.

```
.gitignore  
.gradle  
gradle  
src  
|   └── main  
|       |   └── kotlin  
|       |       └── com.soramitsu  
|       |           └── system_setting  
|       |               └── SystemSettingService.kt  
|       └── resources  
|           └── application.yml  
└── test  
    └── kotlin  
        └── com.soramitsu  
            └── system_setting  
                └── SystemSettingServiceTests.kt  
  
build.gradle.kts  
gradle.properties  
gradlew  
gradlew.bat  
README.md  
setting.gradle.kts
```

*Figure 14: Project Structure*

Spring Initializr is also integrated in all major Java IDEs and allows you to create and import a new project without having to leave the IDE for the command-line or the web UI.

The following IDEs have dedicated support are:

- Eclipse/STS
- IntelliJ IDEA (Ultimate Edition)
- NetBeans (using the NB SpringBoot plugin)
- Microsoft VSCode

## 2. Application Configuration

Spring Boot is used to build the system so we have to make some configurations and installations as well. Fortunately, Spring Initializr provides us with a straightforward Dependencies field when initializing a project by input keyword of dependencies to see if there are any dependencies you need to install to complete the setup. After that, we will see a list of matching choices with that simple criteria since the Initializr offers a fast way to pull in all the dependencies you need for an application and does a lot of the setup for you.

The screenshot shows the Spring Initializr interface with the following details:

- Dependencies** button at the top left.
- ADD DEPENDENCIES... CTRL + B** button at the top right.
- Spring Web [WEB]**: Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
- Spring Data JPA [SQL]**: Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
- Spring Security [SECURITY]**: Highly customizable authentication and access-control framework for Spring applications.
- Eureka Discovery Client [SPRING CLOUD DISCOVERY]**: a REST based service for locating services for the purpose of load balancing and failover of middle-tier servers.
- H2 Database [SQL]**: Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.
- Oracle Driver [SQL]**: A JDBC driver that provides access to Oracle.

Figure 15: Project Dependencies

- Spring Web: For building web, including RESTful, applications using Spring MVC. It uses Apache Tomcat as the default embedded container.
- Spring Data JPA: For persisting data in SQL stores with Java Persistence API using Spring Data and Hibernate.

- Spring Security: For Highly customizable authentication and access-control framework for Spring applications.
- Eureka Discovery Client: is a REST based service for locating services for the purpose of load balancing and failover of middle-tier servers.
- H2 Database: It provided a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.
- Oracle Driver: is a JDBC driver that provides access to Oracle.

### 3. System Architecture

#### 3.1. Physical Architecture

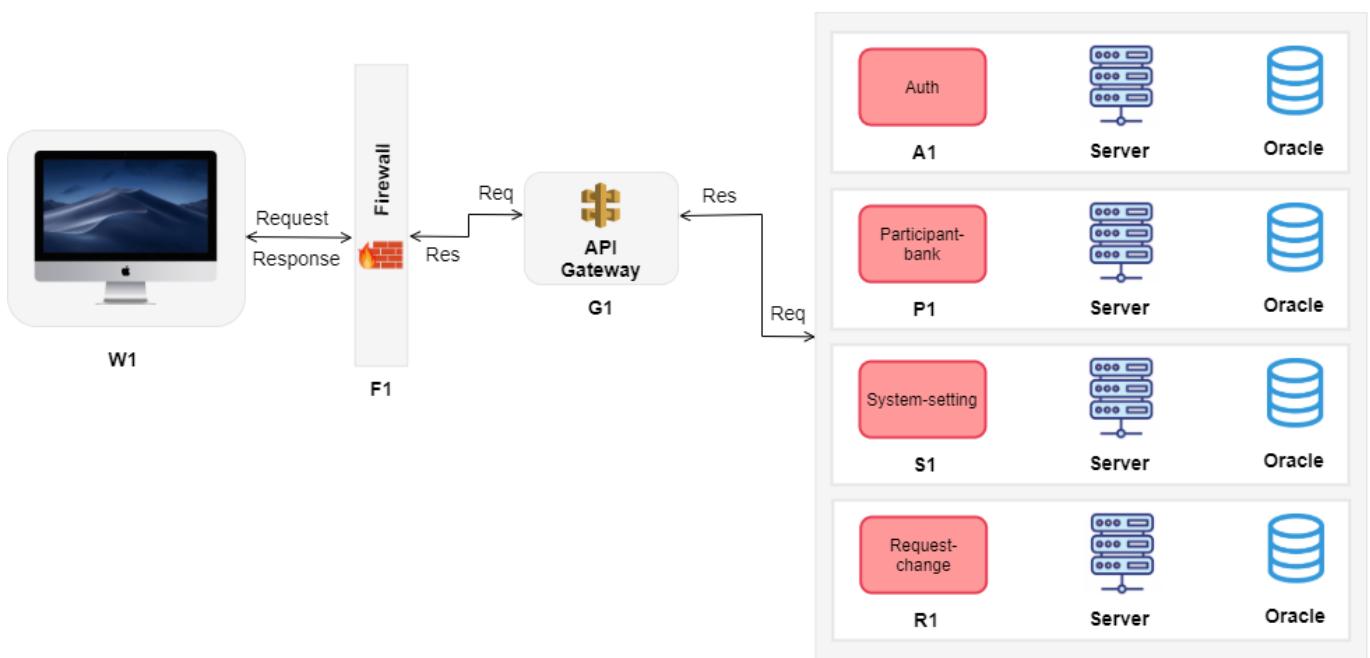


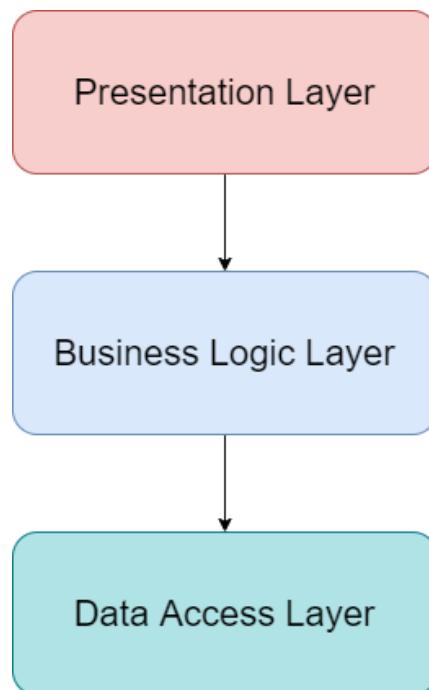
Figure 16: Physical Architecture

Based on figure 16, there are 4 major components of our physical architecture which are:

- **Web Client (W1):** represents the client web browser that retrieves and renders client and admin web such as Vue js and Vuetify with the data from server then server response back by JSON token.
- **Firewall (F1):** represents the tools that can be used to enhance the security of computers connected to a network such as a LAN or the internet. They are an integral part of a comprehensive security framework for your network.

- API Gateway (G1): represents the programming that sits in front of an application programming interface(API) and acts as a single point of entry for a defined group of microservices (is an approach to application development in which a large application is built as a suite of modular components or services). It includes functions such as:
  - Auth
  - Participant bank
  - System-setting
  - Request-change
- DB: database use Oracle that is a free and open-source relational database management system emphasizing extensibility and technical standards compliance. Here it is used for storing users information, activate and deactivate services, store settlement and request-change.

### 3.2. Logical Architecture

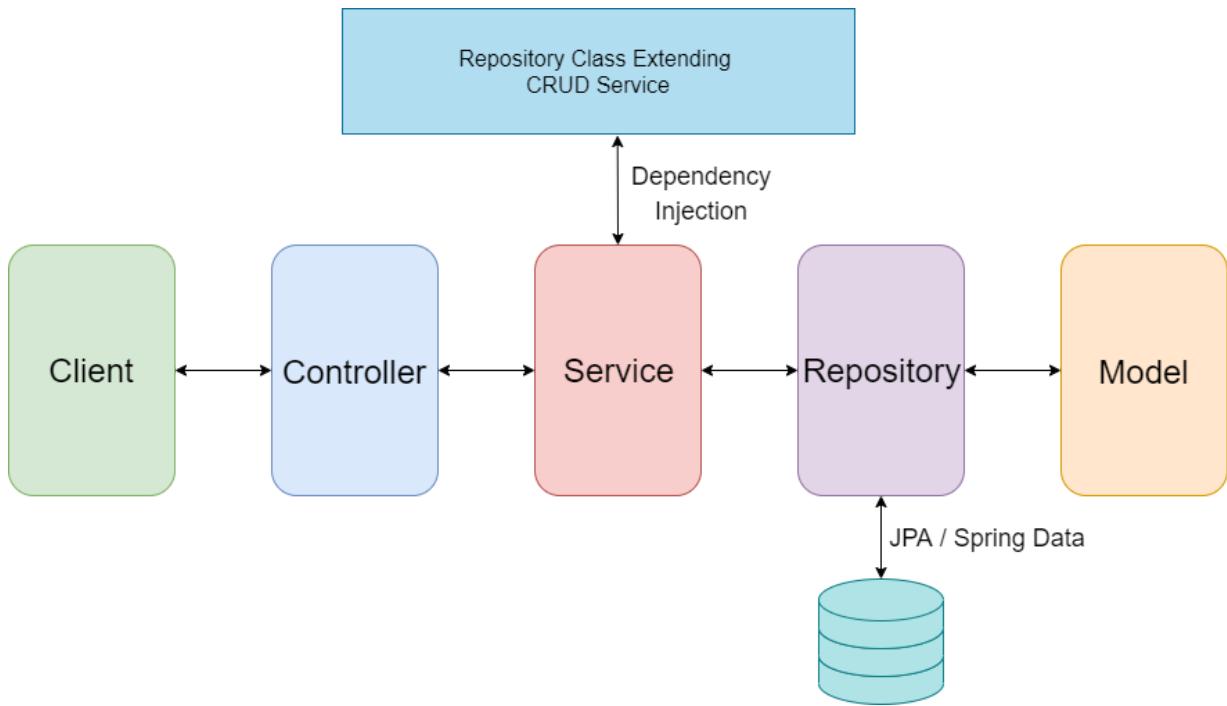


*Figure 17: Spring boot layers*

- **Presentation layer:** handles the HTTP requests, translates the JSON parameter to object, and authenticates the request then transfers it to the business layer. In short, it communicates with the frontend part.
- **Business Logic layer:** contains the business logic that drives the application's core functionalities. Like making decisions, calculations, evaluations, and processing the data passing between the other layers. It also performs validation.

- **Data Access layer:** responsible for interacting with databases to save and restore application data.

In a Spring MVC web application, the three layers of the architecture will manifest as follows:



*Figure 18: Spring boot logical architecture*

- **Client:** is the front end part which consists of views and user experience.
- **Controller as the Presentation layer:** This layer receiving and validating the inputs, manipulating the model object, returning the appropriate response objects, and so on. All the business-related operations should be done in service classes. Controller classes are usually put in a controller package.
- **Service as the Business Logic layer:** It performs the calculations, data transformations, data processes, and cross-record validations (business rules) are usually done at this layer. They get called by the controller classes and might call repositories or other services. Service classes are usually put in a service package.
- **Repository and Model as the Data Access layer:** This layer's responsibility is limited to Create, Retrieve, Update, and Delete (CRUD) operations on a data source, which is usually a relational or non-relational database. Repository classes are usually put in a repository package and Model classes are usually put in a Model Package.

### 3.3. Feature Implementation

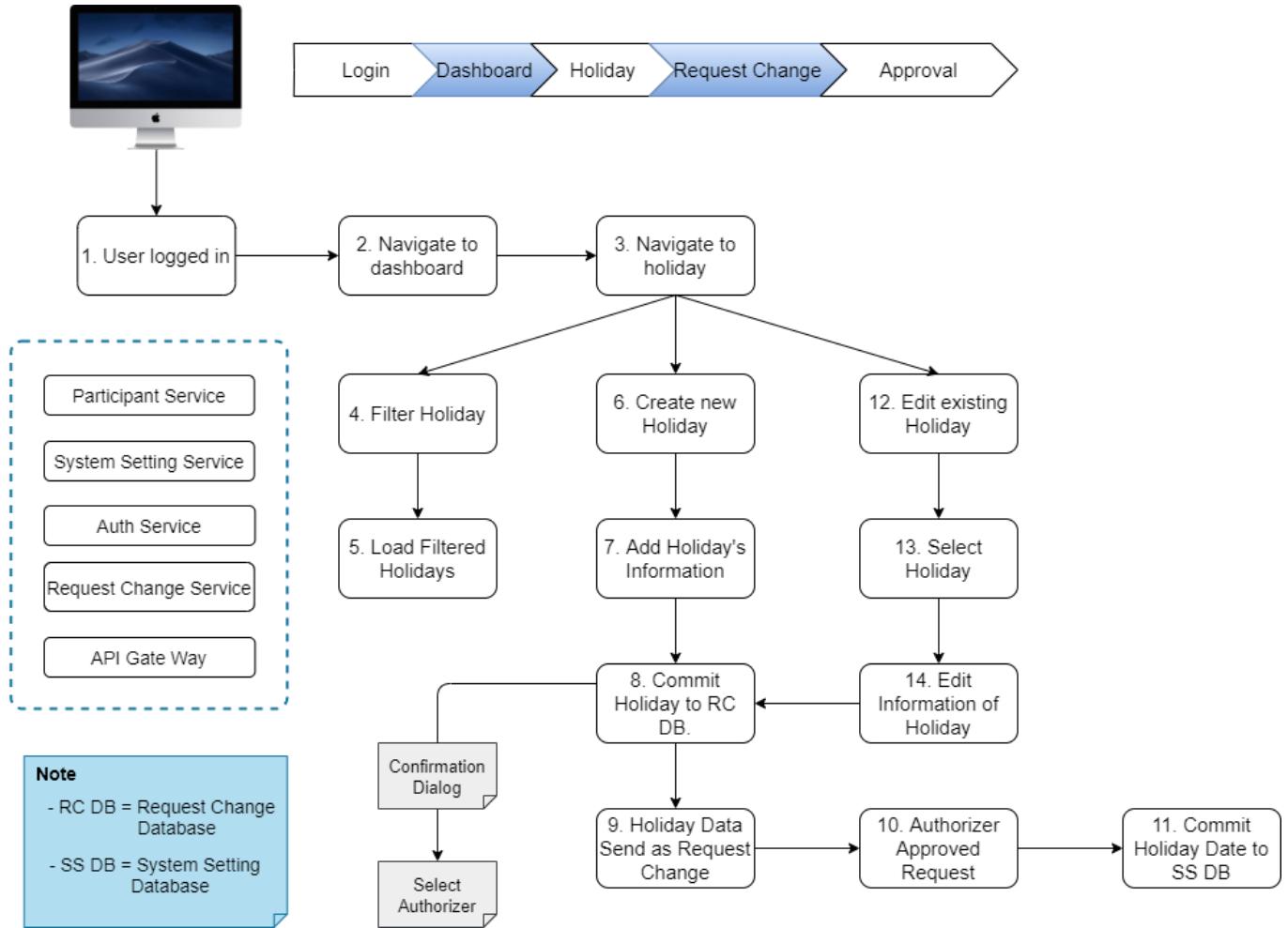


Figure 19: Feature Holiday

After a user completes the authentication process successfully, the system will navigate to the dashboard page.

- The user can process many core features as an NBC admin which contains the Holiday feature. To process this feature, the user selects on Holiday at sidebar navigation. System navigates to the Holiday page which lists all Holidays.
- They can filter holidays by selecting holiday types or year then data will be loaded.
- To create a new holiday user needs to input valid information and send the request to Request-change. After the authorizer approves the request, the data will be pushed to the system setting database.
- Also, they can edit the existing holiday by clicking on the “Pencil icon” of selected holiday and input new information then the request will be sent to Request-change service.

# VI. CONCLUSION



## VI. CONCLUSION

### 1. Result

After a four months internship, the Fund Transfer project is 95% complete with the remaining tasks left are test-call, automate testing, and planning future features. Moreover, the system has a good design and user-friendly interface which we improve upon day by day so users can enjoy using the system easily and effectively. As a result, I did my best and managed to contribute 50% of the whole project. Finally I completed all the important functions which is shown below :

Functionality	Status
<ul style="list-style-type: none"><li>• Manage Account Type</li><li>• Manage Account Class</li><li>• Manage Holiday Type</li><li>• Mange Holiday</li><li>• Manage Holiday Range</li><li>• Manage Holiday Member</li></ul>	Completed
• Test-call	In-progress

Table 5: Results table

The features I have completed smoothly to help users manage account type and set settlement for selected accounts with multiple currency. Furthermore, I successfully provided a way for users to manage account class by each currency in the system. Holiday assists users in managing holiday types such as Core holiday which is the public holiday, Holiday for HQ and Branch when they need to set settlement scheduling without worrying about picking the wrong dates. On the other hand, the remaining task is Test call. Apart from that, the project is still in the testing and debug stage. And possibly new features will be proposed and added soon.

## 2. Strong points and Weak points

No solution is perfect, so the same goes for my current web application project. It is uniquely designed and developed to fit specific purposes so it has its strong points and weak points.

Strong Points	<ul style="list-style-type: none"><li>• Users can easily use each function thanks to fast performance from Spring boot</li><li>• Super initiative and user-friendly interface making user be able to navigate across system smoothly without any hassling reload</li><li>• Most bugs in the system are handled</li><li>• With the increase in security threats nowadays, this factor is gaining importance. The Fund Transfer system uses only reliable resources like JVM, docker images etc</li></ul>
Weak Points	<ul style="list-style-type: none"><li>• Unable to perform sort operation in cross services</li></ul>

Table 6: Strong and Weak points table

## 3. Difficulties

Life is a struggle and full of difficulties, but that only makes us stronger and keep going forward without giving up. Throughout my period of the internship, I faced several challenges both technically and personality. When I first joined Soramitsu Khmer, I severely lacked in Kotlin and Spring Boot which is the main technology used there. I managed to overcome this obstacle by starting to learn and ask from both Google and my supervisor, eventually I got quite proficient at it. and I can safely say I am proud of my knowledge and skill in the Spring boot ecosystem now.

## 4. Experiences

After completing this project, I have gained a lot of experiences related to web-backend development and working environment. The Fund Transfer project has taught me so much about

good code practices and collaboration with designing and development. Unlike standalone projects I used to do at school, every function and view written does not have to be reusable later in other parts of the system. In terms of work environment, Soramitsu Khmer is a relaxing and flexible company because it consists of a small team. The most benefit of this is I can learn directly from seniors who work in the same room and can communicate and comment on my work frequently which is great for my self-development. It teaches me to love constructive criticism and not take things personally because criticizing is focused on the code not the person. Last but not least, besides work, our team has fun chatting about nerd stuff which is useful for teamwork spirit and we often have team celebrations in case there is a farewell party or exciting birthday party.

## 5. Perspective

Like I mentioned in weak points, the system is unable to perform the sort operation in cross services. We also have an eye on future features including AI integration, integrating machine learning in other parts of the check clearing site such as detecting fraud information and possibly enabling check clearing crypto currency. Moreover, I would try my best to solve those problems and help in implementing more functions into the system for a more effective business solution. I would write code more professionally and reduce bugs in the program so that it will run smoothly. Last but not least, I also hope that after the release, the system would help facilitate the business well and enhance productivity of the company's client.

## 6. Conclusion

I have no doubt the complete Fund Transfer project is going to launch smoothly and attract a lot of users due to its performance and flexibility thanks to the parallel combination of Spring Boot, Docker and Server on Local Authorization. Big thanks to Soramitsu Khmer who has contributed a lot to my growth both professionally and technically. I have nothing but marvelous gratitude to my supportive team for the last four months. Looking back, it has been a great journey studying at NIPTICT until now. Technically, with the experience from Soramitsu Khmer, I have become a capable junior developer in Spring Boot of Kotlin world. Practically, I have learnt a bunch of knowledge from backend development and now I am capable of independently developing a web application using that technology on my own. I am who I am today thanks to NIPTICT. I owe it to this school and all the amazing and supportive lecturers who have committed so far to teach me and other students everyday tirelessly. Since not all the

knowledge get into my little brain due to my laziness and carelessness back then, but it definitely acts as a great inspiration that drives me to find my passion in developer career and I will keep fighting to advance myself further, going forward and hopefully will be able to make a contribution back to NIPTICT one day.



## LIST OF REFERENCES

---

- <https://spring.io/projects/spring-boot>
- <https://www.docker.com/>
- <https://kotlinlang.org/>
- <https://www.liquibase.org/>
- <https://www.jetbrains.com/idea/>
- <https://microservices.io/>
- <https://soramitsu.co.jp/kh>

# ANNEX INDEX 01 & 02



NCS

Current business day & time: 2020-12-02, 2:39:35 PM

NBC nbc-admin

**Account Type**

Code	Name	Created by	Created at ↑	Transaction Settlement	Fee Settlement	Fee Collecting	Status	Action
CUR	Current Account	nbc-admin	2020-11-23 11:21:44	○	○	○	Active	edit
SET	Settle Account	nbc-admin	2020-11-21 12:23:07	○	○	○	Active	edit
TRD	Trade Account	nbc-admin	2020-11-20 15:59:36	○	○	○	Inactive	edit
INT	International Account	nbc-admin	2020-10-27 05:01:12	○	○	○	Inactive	edit

Account Type list

NCS

Current business day & time: 2020-11-29, 6:43:31 PM

NBC nbc-admin

**Create New Account Type**

Code*	LOA	Description
Name*	Loan Account	
Status*	Active	
Choose Authorizer(s)	nbc-authorizer-001	
Please select an authorizer.		

**Account Type**

10	1-4 of 4	<	>
Status	Action		
Active	edit		
Active	edit		
Inactive	edit		
Inactive	edit		

Account Type Creation form

# ANNEX INDEX 03 & 04



NCS

Current business day & time: 2020-11-29, 6:44:05 PM

NBC nbc-admin

Dashboard

Participant Manage...

Participant

Manage CEC

User Management

Role

User

System Management

Institution Type

Account Type

Currency Type

Error Type

Account Class

Billing Schedule

Set Threshold

Mocks

Log

Account Type

Update Account Type

Code: CUR

Description: Current Account

Name: Current Account

Status: Active

Choose Authorizer(s): NBC Authorizer

CANCEL UPDATE

10 1-4 of 4

Status Action

Active

Active

Inactive

Inactive

10 1-4 of 4

Rows per page: 10 1-4 of 4

Account Type Update dialog

NCS

Current business day & time: 2020-12-02, 2:40:41 PM

NBC nbc-admin

Dashboard

Participant Manage...

Participant

Manage CEC

User Management

Role

User

System Management

Institution Type

Account Type

Currency Type

Error Type

Account Class

Billing Schedule

Set Threshold

Mocks

Account Class

USD KHR

Account Type C... Search... + CREATE

Code	Name	Account Type Code	Created by	Created at ↑	Action
SETTLEU	Settle Account USD	SET	nbc-admin	2020-11-23 09:35:11	
SETTLEUI	Settle Account INT USD	SET	nbc-admin	2020-11-23 09:35:11	
CURRENTU	Current Account USD	CUR	nbc-admin	2020-11-23 09:35:11	

Rows per page: 10 1-3 of 3

Rows per page: 10 1-3 of 3

Account Class List

# ANNEX INDEX 05 & 06



NCS

Current business day & time: 2020-11-29, 6:38:26 PM NBC nbc-admin

**Holiday**

ID	Name	From	To	Description	Type	Status	Action
961	Khmer New year	2021-04-14	2021-04-16	Khmer New Year Holiday	Core Holiday	Active	
13	Water Festival	2020-11-06	2020-11-06	Water Festival Holiday	Core Holiday	Active	
11	Pchum Ben	2020-11-05	2020-11-05	Pchum Ben Holiday	Core Holiday	Active	

Holiday List

NCS

Current business day & time: 2020-11-29, 6:41:20 PM NBC nbc-admin

**Create New Holiday**

Type*	Branch Holiday	Status*	Active
Participant(Code and Name)*	ABA002-ABA Krong Battambang	Description	
Name*	ABA Branch BTB 10th Anniversary	It is the 10th anniversary of ABA Branch.	
Date(From To)*	2020-11-04 - 2020-11-05		
Choose Authorizer(s)	NBC Authorizer nbc-authorizer-001	Status	
Please select an authorizer.			

**CANCEL** **CREATE**

Holiday Creation Form