COMP 3059 – Capstone Project I

Software Requirements Analysis and Design Assignment

This assignment is an overview to gather the software needs with requirements analysis and help to proceed with the design.

The requirements analysis helps to break down functional and non-functional requirements to a basic design view to provide a clear system development process framework. It involves various entities, including business, stakeholders and technology requirements.

The design is the activity following requirements specification and before programming. Software design usually involves problem solving and planning a software solution.

To work on this assignment you could use the references and a sample template given below. The sample template can be customised to suit the nature of your project.

Reference Readings/Example:
http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf

www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc

Source for this template:
www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc

1.0 Introduction

1.1 Purpose

This document provides a high-level overview of the software requirements for the Ultimate Currency and Unit of Measure Converter project. It outlines the functional and non-functional requirements to guide the development team in building a system that meets the needs of end users and stakeholders. The purpose is to establish a foundation for the design and development stages by clarifying what the system should achieve rather than detailing how it will do so. This document will serve as a reference point for developers, stakeholders, and future contributors to understand the scope and objectives of the project.

1.2 Scope

The Ultimate Currency and Unit of Measure Converter system is designed to simplify currency and measurement conversions for users, focusing on converting to Canadian values. The system will enable users to enter or project exchange rates and convert various currencies and units to their Canadian equivalents. The project scope includes:

1. **Paying customer registration** for creating an account.
2. **Paying customer login** functionality to access the system.
3. **Conversion** of the price per unit of measure in a foreign currency to the target unit of measure in Canadian currency.
4. Integration with an **external API for real-time exchange rate retrieval**, which is a confirmed requirement and not subject to time constraints.

Initially, the system will be a browser-based application, with Canadian currency as the default target currency. The system aims to improve user efficiency by offering quick, reliable conversions. Out-of-scope items include mobile application versions and currency conversions to currencies other than the Canadian dollar.

2.0 System Overview

2.1 Project Perspective

The Ultimate Currency and Unit of Measure Converter is a new, self-contained system developed to address the need for accurate, rapid conversions for users dealing with regional currencies and units of measurement. The system does not follow a pre-existing family of applications but aims to establish itself as a standalone solution for users requiring Canadian conversions.

2.2 System Context

The system is intended to provide a reliable tool for individuals and businesses needing fast and accurate currency and measurement conversions to Canadian units. This application targets users involved in global trade, travelers, and professionals who regularly deal with foreign currencies and units. By simplifying the conversion process, the system aims to streamline users' tasks, saving time and reducing potential errors in calculations. The strategic goal of the project is to establish an initial user base, focusing on customer satisfaction, ease of use, and potentially expanding with additional features or integrations in future versions.

2.3 General Constraints

Key constraints for the system include:

- The system will be limited to browser-based functionality with no mobile app version.
- All conversions are limited to the Canadian dollar as the target currency.
- The application relies on a third-party currency API for exchange rates, contingent on the API's availability and reliability.
- The back-end must be developed and operational before the front-end can interact with the API.

2.4 Assumptions and Dependencies

Assumptions:

- The project scope remains stable throughout the development lifecycle.
- All team members possess the required skills for their assigned tasks.
- The application will be tested by end users according to an agreed timeline.
- The users will have access to the internet.

Dependencies:

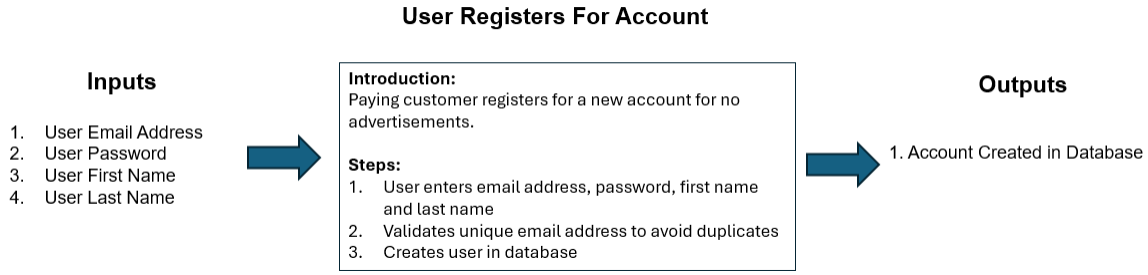
- The system's currency conversion functionality depends on the integration of a third-party currency API.
- Front-end development will depend on back-end progress, specifically the availability of APIs to handle user input and conversions.
- Approval of the graphical interface and user experience design will be necessary before front-end coding begins.

3.0 Functional Requirements

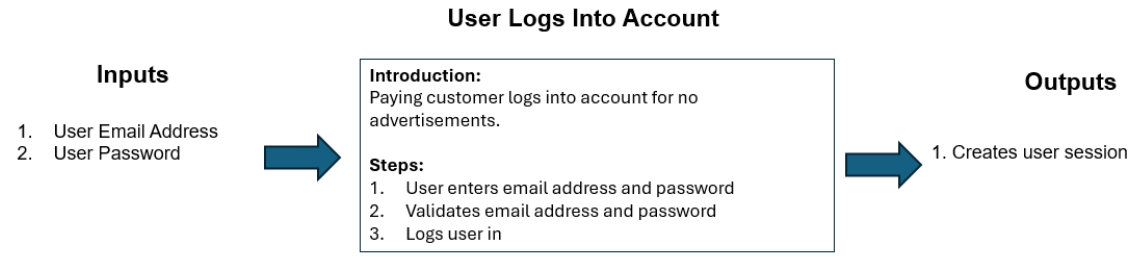
This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.1 <Functional Requirements/Features>

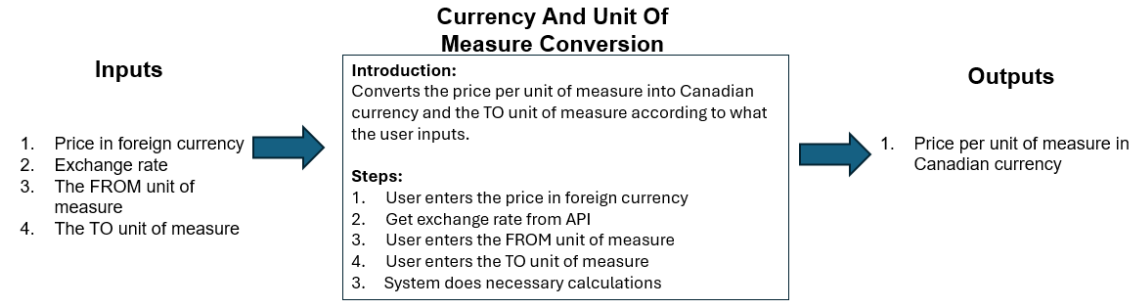
1. User registers for account (no ads website for paying customers)



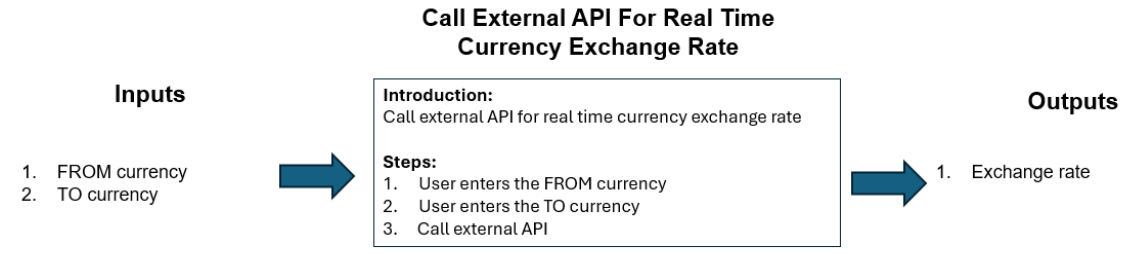
2. User logs into account (no ads website for paying customers)



3. Converts the price per unit of measure in a foreign currency to the target unit of measure in Canadian currency.

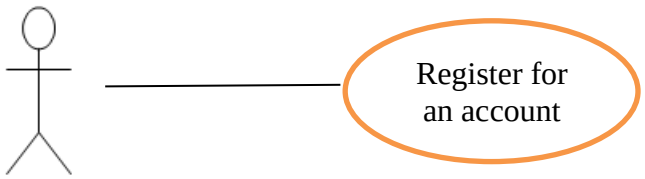


4. Call external API to get real time currency exchange rate



3.2 Use Cases

3.2.1 Use Case #1 Register for an Account

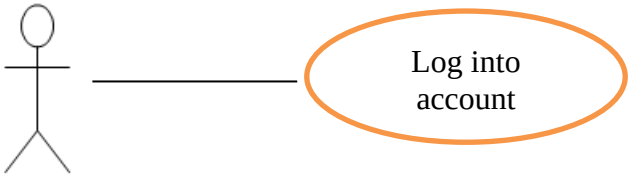


Brief Description
The paying customer registers for an account so that they will not see ads on the website.

Initial Step-By-Step Description
Before this use case can be initiated, the paying customer needs to have access to the internet and be on the Currency And Unit Of Measure Conversion website.

1. The paying customer clicks on the register button on the website.
2. The paying customer enters their email address, password, first name and last name.
3. The system validates unique email address to avoid duplicates.
4. The system creates user in database.
5. The system displays “Account Created” message.

3.2.2 Use Case #2 Log into Account



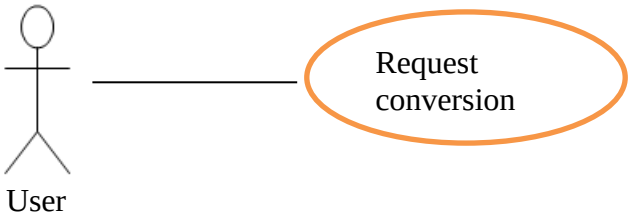
Brief Description
The paying customer logs into their account so that they will not see ads on the website.

Initial Step-By-Step Description
Before this use case can be initiated, the paying customer needs to have internet access, have registered for an account and be on the Currency And Unit Of Measure Conversion website.

1. The paying customer clicks on the Login button on the website.
2. The paying customer enters their email address and password.

3. The system validates email address and password.
4. The system logs user in.

3.2.3 Use Case #3 Request Conversion

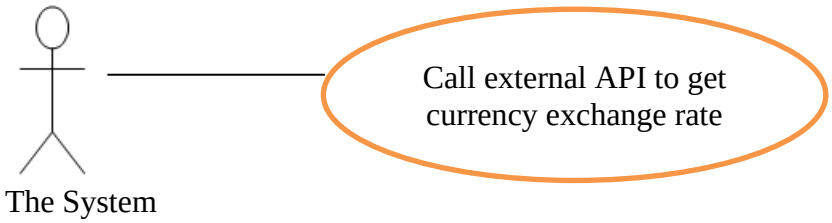


Brief Description
The user requests currency and unit of measure conversion.

Initial Step-By-Step Description
Before this use case can be initiated, the user needs to have access to the internet and be on the Currency And Unit Of Measure Conversion website.

1. The user enters the price in foreign currency.
2. The system gets the exchange rate from the external API.
3. The user enters the FROM unit of measure.
4. The user enters the TO unit of measure.
5. The system does the necessary calculations.
6. The system displays the results for the user.

3.2.4 Use Case #4 Call External API for Real Time Currency Exchange Rate



Brief Description
The system calls the external API to get the real time currency exchange rate.

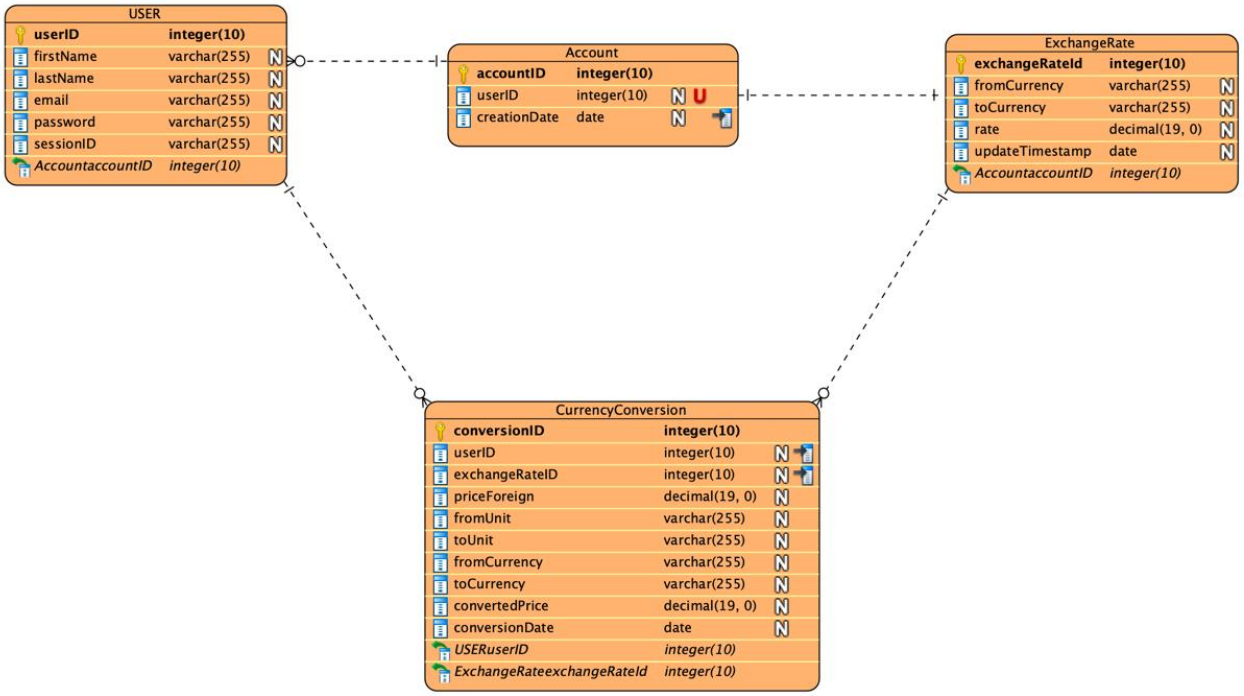
Initial Step-By-Step Description
Before this use case can be initiated, we assume that the external API will be available for use and its specifications have not changed since our development has started.

1. The user enters the FROM currency.

2. The user enters the TO currency.
3. The system calls the external API.
4. The external API returns the exchange rate.

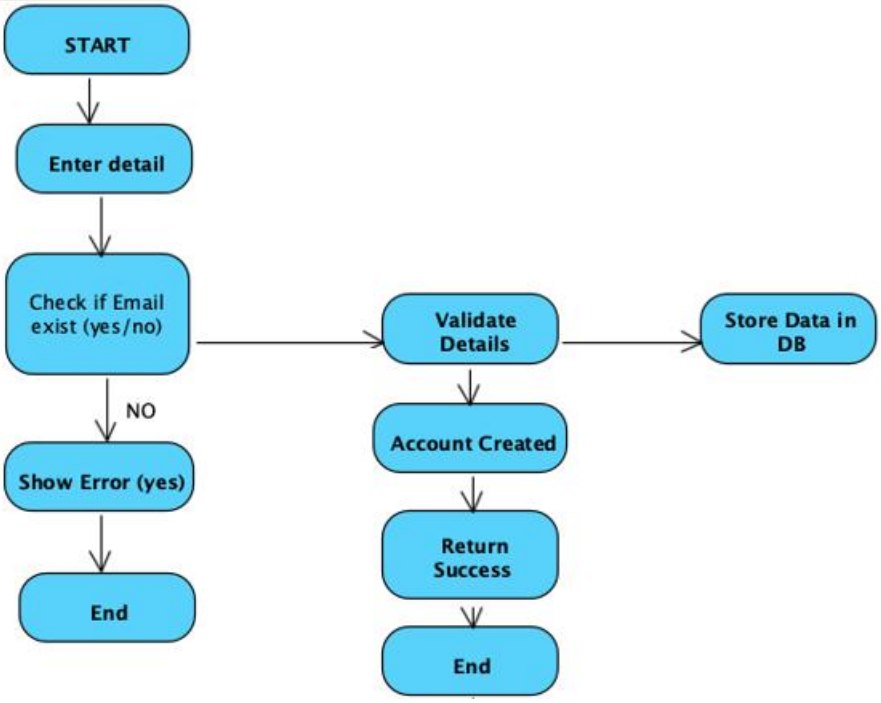
3.3 Data Modelling and Analysis

- Normalized Data Model Diagram

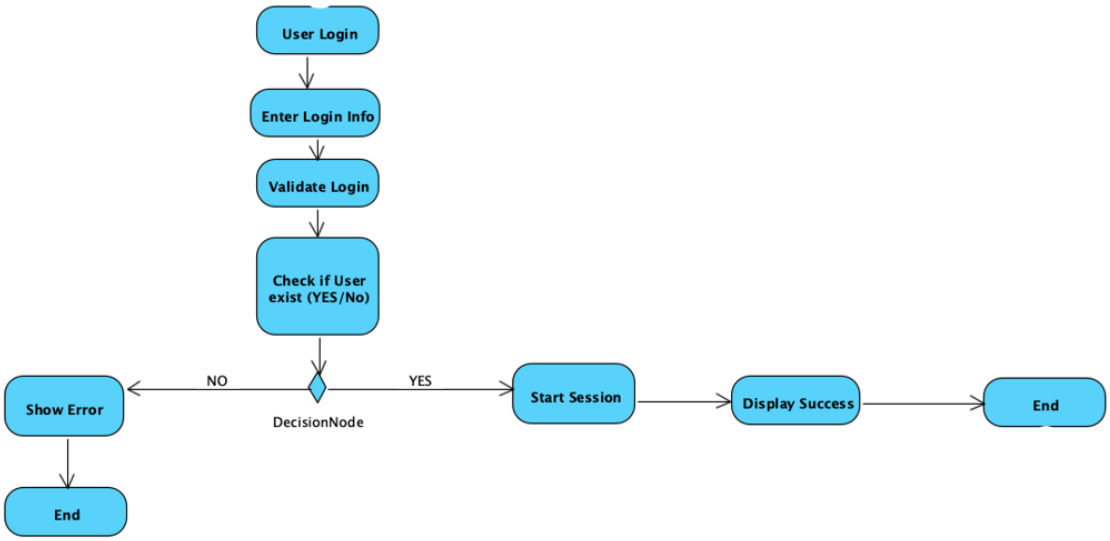


- Activity Diagrams

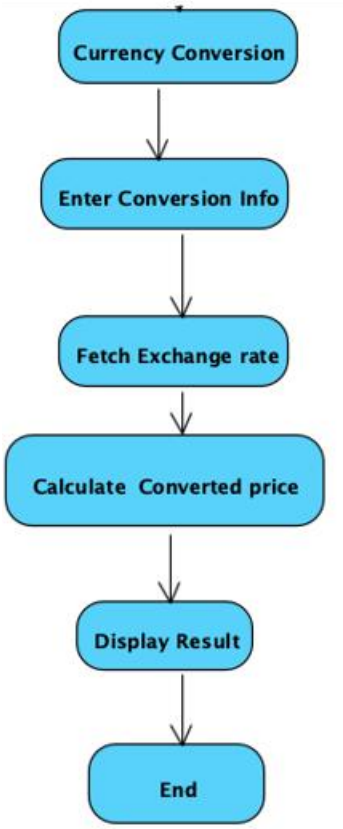
1. Register For An Account



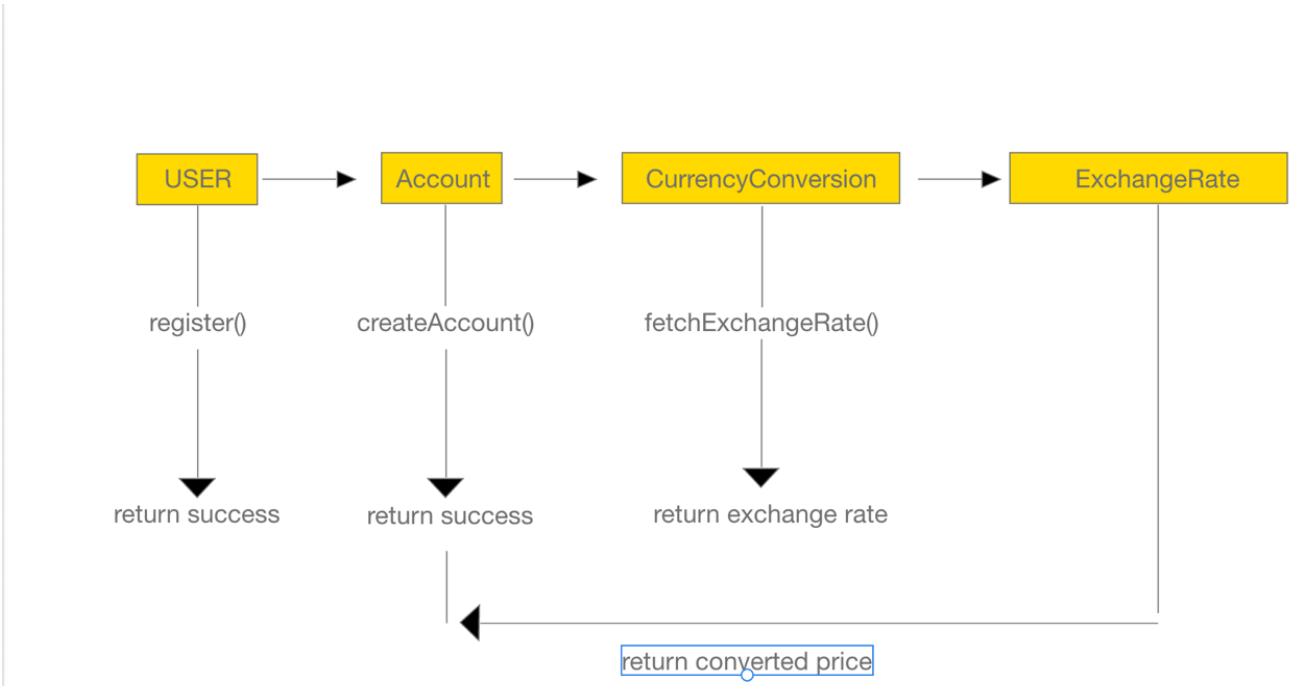
2. Log Into Account



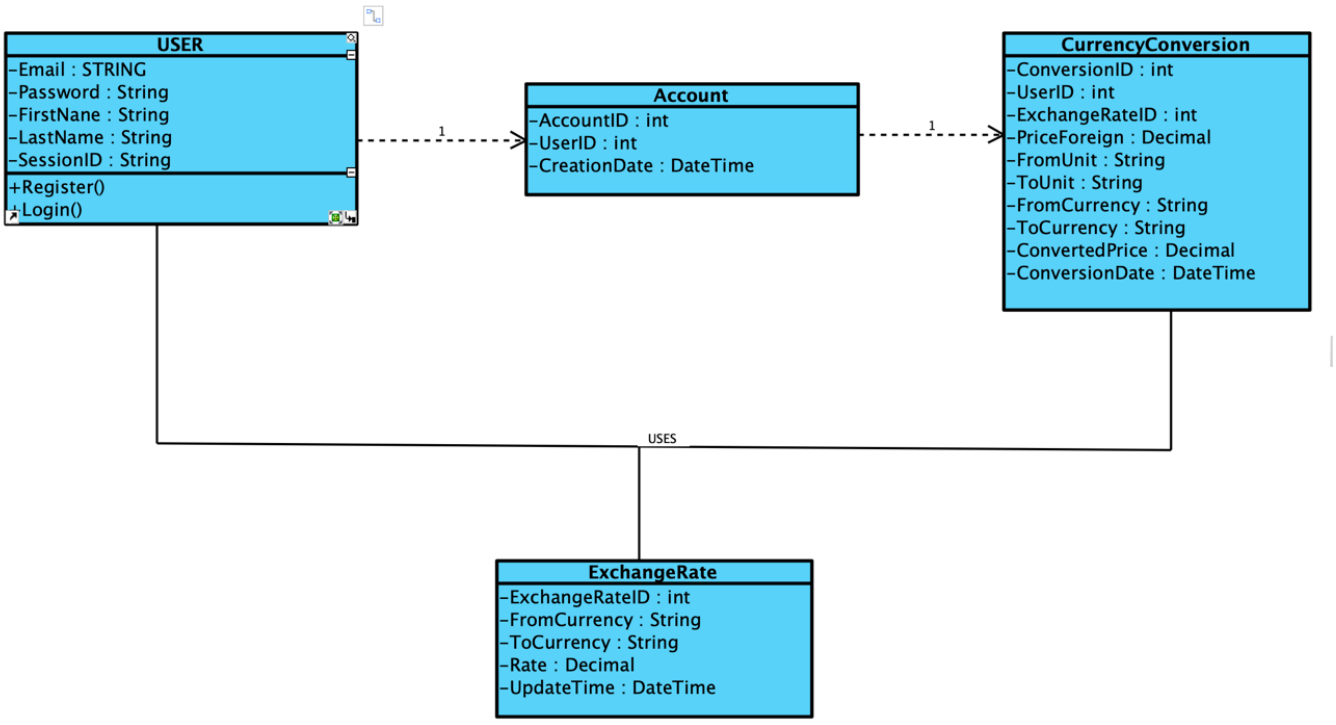
3. Request Conversion And Call External API For Real Time Currency Exchange Rate



- Sequence Diagram



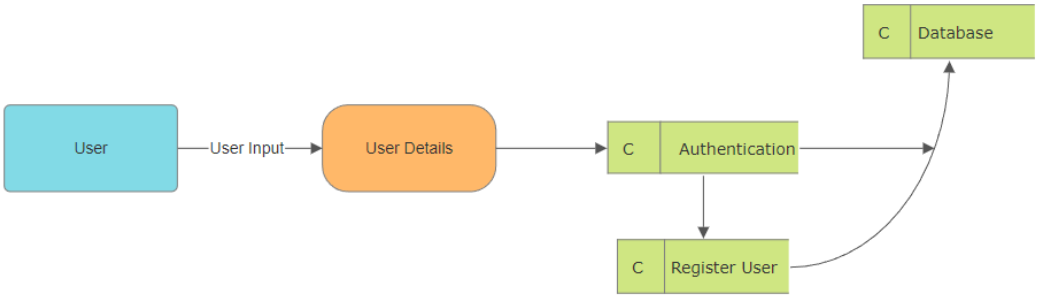
- UML Class Diagram



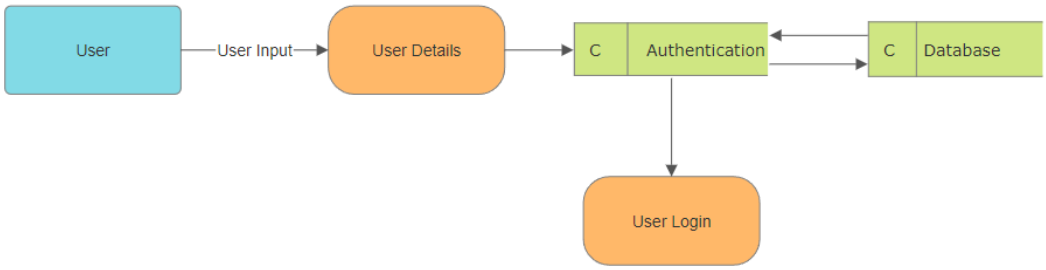
3.4 Process Modelling

- Data Flow Diagrams

1. Register User

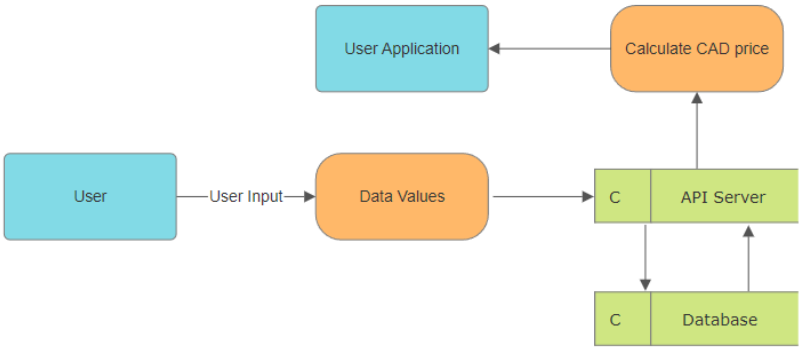


2. User Login



3. Measurement Conversion

4. Call to API Server for Currency Exchange Rate



4.0 Non-Functional Requirements

Performance – The system must respond to user input with requested information from server API within 1 to 2 seconds.

Reliability – The system must not crash due to programming errors and bugs for 99% of the time.

Availability – The system must be available to request and receive data 99% of the time.

Security – The system must protect 100% of users' data with encryption when information is passed between servers.

Maintainability – The system should facilitate easy updates and modularity.

Portability – The development team must ensure the software can run on the most current operating systems and browsers.

5.0 Logical Database Requirements

Logical Database Requirements	Description
Storage Capabilities	- Storage Capacity – how much data the infrastructure can contain.

	- Storage Hardware – HDD, SSD or hybrid storage will have some determinant in the performance of the infrastructure - Compression of data – reduce size of storage requirements
Data Retention	- Store user registration data - Data analytics and reporting tools - monitor data usage and retention. - Update database retention policies to ensure compliance with regulations.
Data Integrity	- Error checking and validation, - Entity integrity - Make primary keys for classifying data.
Data Formats	- Data serialization – converting structured data into a format that allows storage and recovery of its original structure.
Data Security	- Encryption of data – hashing of sensitive data, encryption of data in transit. - Role-Based Access Control of database - Firewall Configuration – Monitor, with real-time alters in case of unauthorized access. - Offline Database backups – in case of unauthorized access, power outage, etc. - Network segmentation – Isolate the database

6.0 Other Requirements

Additional requirements, if any.

7.0 Approval

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
Front-End	Justin Yeh	Justin Yeh	Nov 06, 2024
Front-End	Kevin Lapointe	Kevin Lapointe	Nov 06, 2024
Back-End	Fab Pisco	Fab Pisco	Nov 06, 2024
Back-End	Luilson Sousa	Luilson Sousa	Nov 06, 2024