PAYMENT APPLICATION

BSC (HONS) COMPUTER SCIENCE

ORE BENSON

CANDIDATE NUMBER: 219685

WEB APPS AND SERVICES

DEPARTMENT OF INFORMATICS

ACADEMIC YEAR 2022/23

# Layers

The sections below describe the extent to which requirements were achieved, explaining the implementation for each.

## Presentation

The presentation layer consists of a set of templates through which users and administrators interact with the web application.

Users should be able to:

* View all their transactions
* Make direct payments to other registered users
* Request payments from registered users

Administrators should be able to see:

* user accounts
* all payment transactions
* register new administrators

## Business Logic

The business logic layer consists of views containing the logic that accesses the model(s) and defers to the appropriate template(s).

* Views should support transactions so that data integrity is preserved.
* You only need to annotate your views with the appropriate transaction attributes (or leave the default behaviour, if appropriate).
* It is also necessary to guarantee the ACID properties.

Users should be able to (15%):

* View all their transactions
* Make direct payments to other registered users
* Request payments from registered users

Administrators should be able to (5%):

* View all user accounts and balances
* View all payment transactions
* Register more administrators

## Data Access

The data access layer consists of models containing anything and everything about the data.

* To simplify deployment and configuration, you must use SQLite as your Relational Database Management System (RDBMS).
* Your model should contain the essential fields and behaviours of the data being stored.
* After installing the app, Django can create a database schema and a Python database-access API for accessing your objects.
* Full marks will be given if all access to application data is handled through models and properly relating tables to each other.

## Security

* Full marks will be given if users can register, login and logout. An admin must be registered in the system upon activating the models
* Access to templates must be restricted to authorised actors (users and admin).
* Communication with the web application using HTTPS
* Protection against cross-site scripting (XSS), cross-site request forgery (CSRF), SQL injection, and clickjacking attacks needs to be in place
* Upon activating the models, a single administrator account (username: admin1, password: admin1) must be present. Only an administrator can register more administrators through the restricted admin pages.

## Web Services

* You must implement a REST Service to be accessed by the business logic layer. The service will be deployed on the same server but accessed from the business logic layer in the standard way, i.e., through HTTP.
* A currency conversion RESTful web service that responds only to GET requests. The exported resource should be named conversion in a path such as the following:

baseURL/conversion/{currency1}/{currency2}/{amount\_of\_currency1}

* The RESTful web service should return an HTTP response with the conversion rate (currency1 to currency2) or the appropriate HTTP status code if one or both of the provided currencies are not supported.

# User Manual

This section shows a series of screenshots for the user, and the admin. These show all of the features of the website.

## User

## Admin