Application Design Document

Bike and Parts Management System (BPMS)

v1.0

Bhuvesh Gupta
bgupta1@umd.edu
UID: 117326611

Table of Contents

List of Figures

1.0 Introduction

This software design document (SDD) is going to provide an architectural blueprint for Bike and Parts Management System (BPMS). All the functionalities are discussed in detail.

1.1 Purpose

The purpose of this document is discuss the designs and standards of BPMS. This document is intended to help the developers, documentation personnel and the testers working on this project. The document also gives the security implementations in detail.

1.2 Scope

The BPMS is designed to ease the commute and help the students have better, healthier and environment friendly transportation options. The students at the university would have options to buy or rent bikes and buy parts/accessories. Bike shop supervisors can also view the sales data along with managing student accounts. Supervisors can also view and manage the inventory at the bike shop.

2.0 Architectural Design

2.1 System Design

This section is going to outline the logical organization of the software and interaction between each of the software components.

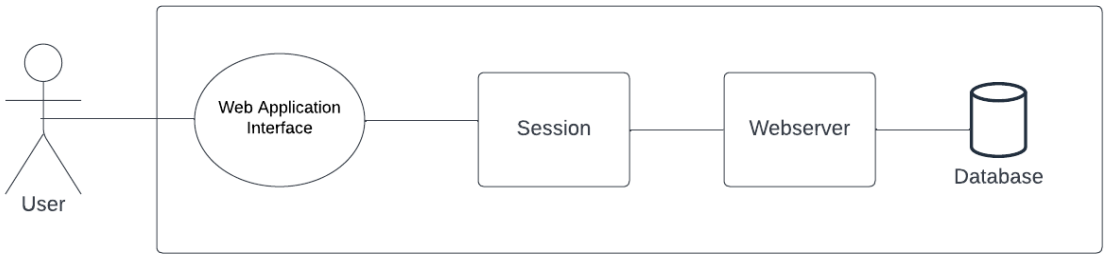

Figure 1:System Level Architecture

The user will interact with the web application to register or login to an account. After logging in, a session would be created with the webserver from where the purchases or rentals can be made. Different levels of users will have different access levels on the database.

Also, an unauthenticated user can only view the bikes and parts information.

2.2 System Level Architecture

The application will be using the MVC architecture which shows the flow as below:

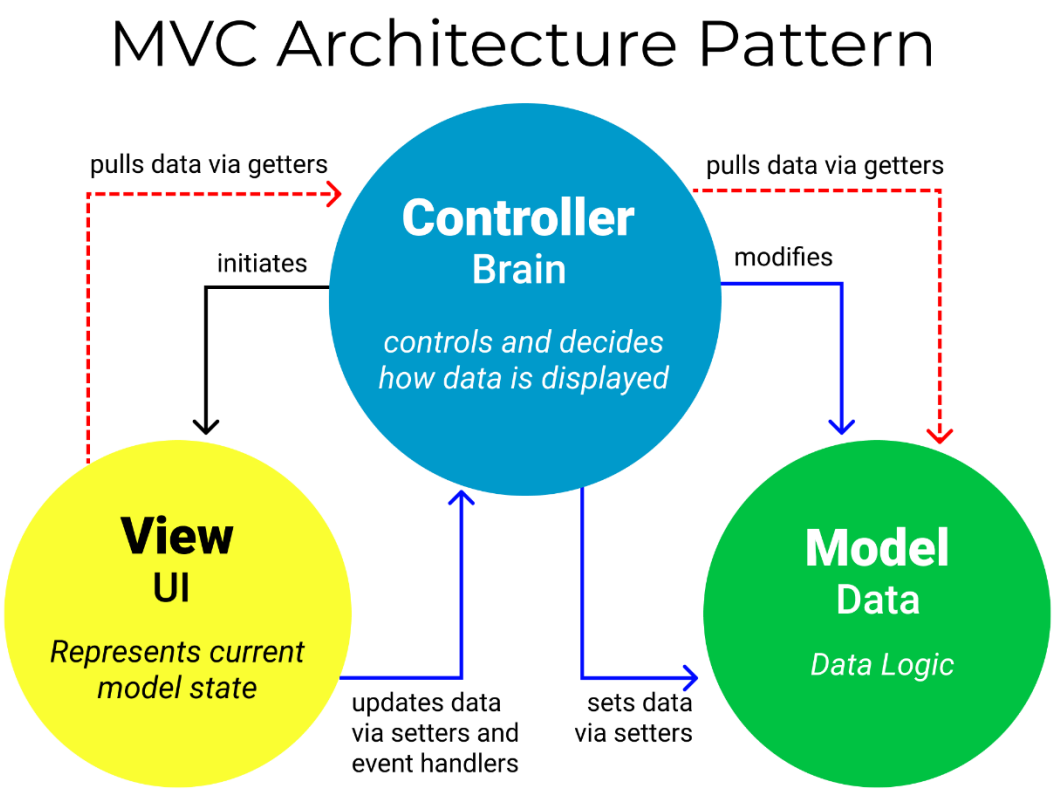

Figure 2: System Level Architecture

2.3 UI Design

The application would contain several pages and few pages would only be accessible to the supervisors.

Home Page

It will show basic information about the user such as Full Name, Amount Due, University ID, Account Expiry Date and Current Rentals and Due Dates and current Purchases.

It will also have options to visit the following pages:

Purchase Bikes, Bike Rentals, Return Bike, Parts and Accessories

Purchase Bikes

This page would display all the bikes available for purchase along with the inventory count of each bike as well. After selecting the bike and quantity, the user can click the Checkout button to visit the payments page for completing the transaction.

Bike Rentals

This page would display all the bikes available for rent along with the inventory count of each bike as well. After selecting the bike, the user can see the due date to return the bike and then click the Checkout button to visit the payments page for completing the transaction.

Return Bike

If the user has already taken a bike for rent then it can start the return process which would then calculate the fine if the user has passed the due date. After which the user would be redirected to the payments page for any payments. If the user is within the due date then it doesn't have to pay any fines.

Parts and Accessories

This page would display all the parts and accessories available for purchase along with the inventory count of each bike as well. After selecting the bike and quantity, the user can click the Checkout button to visit the payments page for completing the transaction.

Modify User Accounts

This page would be only visible to the supervisors and would allow them to Add, Modify or Delete user accounts.

Inventory Data

This page would be only visible to the supervisors and would allow them to Add, Modify or Delete inventory items.

Sales Page

This page would be only visible to the supervisors and would allow them to view the sales data and earnings of the shop.

2.4 Database Design

The database will have 5 tables which would be linked to each other to obtain useful information. Below are the table designs:

Users

Field Name	Field Type
UID	*Int*
First Name	*String*
Last Name	*String*
Email ID	*String*
Phone Number	*Int*
Account Created	*Date*
Account Expiry	*Date*
Graduation Month	*Int*

Graduation Year	*Int*
Password Hash	*String*

Roles

Field Name	Field Type
UID	*Int*
Role ID	*String*

Bikes Inventory

Field Name	Field Type
Bike Number	*Int*
Bike Model Name	*String*
Bike Size	*String*
Bike Count	*Int*

Parts and Accessories Inventory

Field Name	Field Type
Item Number	*Int*
Item Model Name	*String*
Item Count	*Int*

Purchase Record

Field Name	Field Type
UID	*Int*
Purchase Type (Rent/Buy)	*String*
Rent Date	*Date*

Due Date	*Date*
Bike Return Date	*Date*
Total Amount	*Int*
Product Purchased	*String*

Flow of data within the tables:

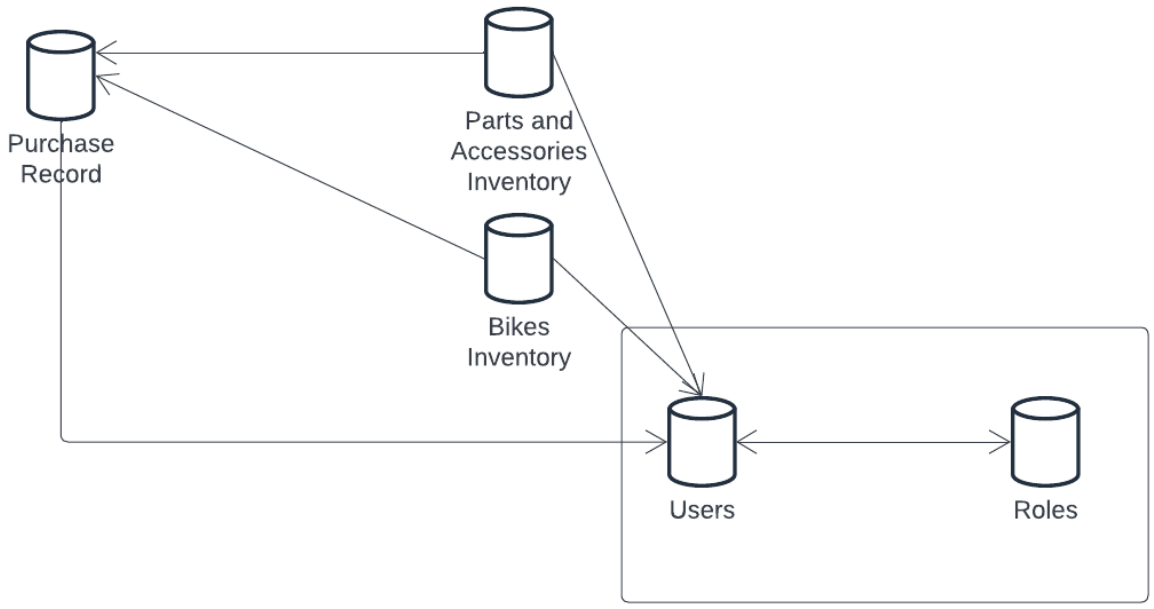

Figure 3:Data Flow Within Tables

2.5 Internal Component Design

All users can view the bikes and parts information. Valid users can buy or rent bikes and also buy parts or accessories. Shop supervisors can edit the user account data and can also view the sales page to estimate their earnings. Supervisors can also edit the inventory data of bike and parts/accessories.

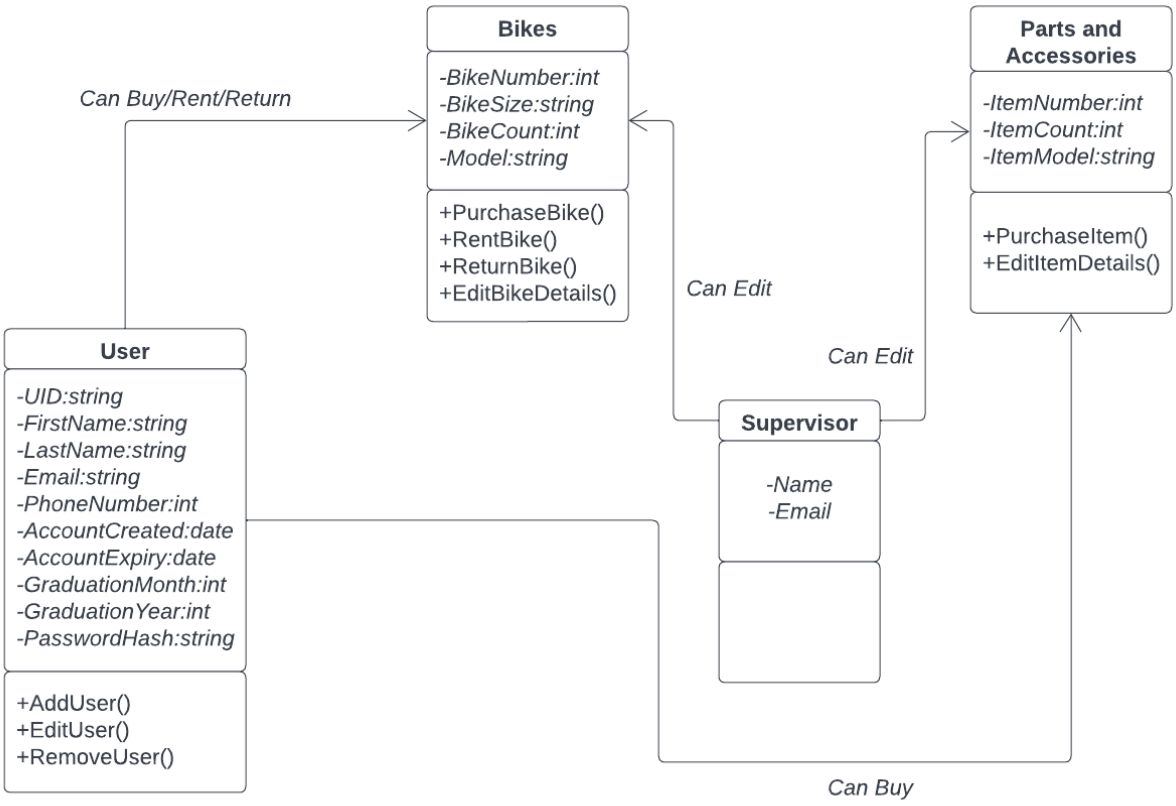

Figure 4: Class Diagram

2.6 Security Functionality

- Strong password policy would be implemented which can probably have a minimum length of password including numbers, lowercase, uppercase and special characters.
- User account will get locked for 2 hours if it enters incorrect password 10 times consecutively within 5 minutes. This activity can be monitored via checking the system logs for failed login attempt.
- Session Management can be implemented by deleting the session cookie whenever the user logs out or closes the browser tab

- Input filtering and sanitization will be implemented at all the places to avoid any kinds of XSS, SQL Injection or other attacks. We can implement a whitelist for a stronger protection.
- The database will store the password in an encrypted format (hashed with secure random salt) to prevent attackers from accessing the password.
- Appropriate error handling to not give out any useful information to the user while throwing out errors. The errors will give out a generic output message.
- Implementing regex timeout to prevent any kinds of regex DoS attack. The application will stop the regex parsing if it takes more than 2 minutes.
- The application will have a SSL/TLS certificate with a certified domain.
- The URL input would be filtered and sanitized for any kinds of directory traversal attack attempts.

3.0 Testing

- A valid user can login into the system with valid credentials.
- An invalid user is not able to login into the system with invalid credentials.
- A new user can sign up using the valid credentials.
- A new user can view the bikes and parts available page.
- A valid user can buy or rent the bikes by selecting the desired height and quantity.
- A valid user can buy the bike parts and accessories.
- Supervisors can edit the user accounts.
- Supervisors can edit the inventory items.
- Supervisors can view the sales data.
- A valid user can logout of the system successfully.

4.0 References

- https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/