Application Design Document

Bike and Parts Management System (BPMS)

v1.1

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 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.

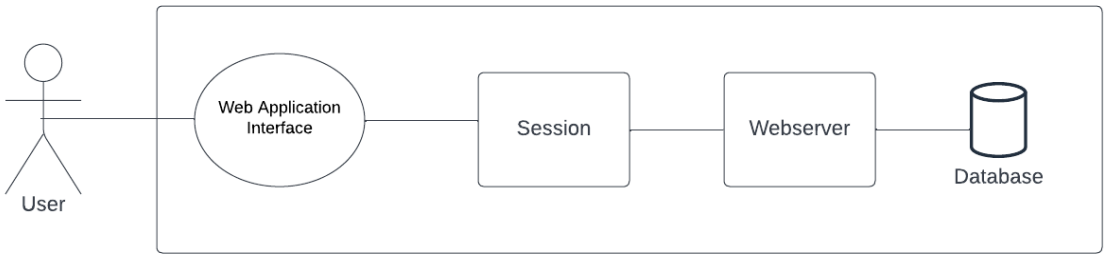

Figure 1:System Level Architecture

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:

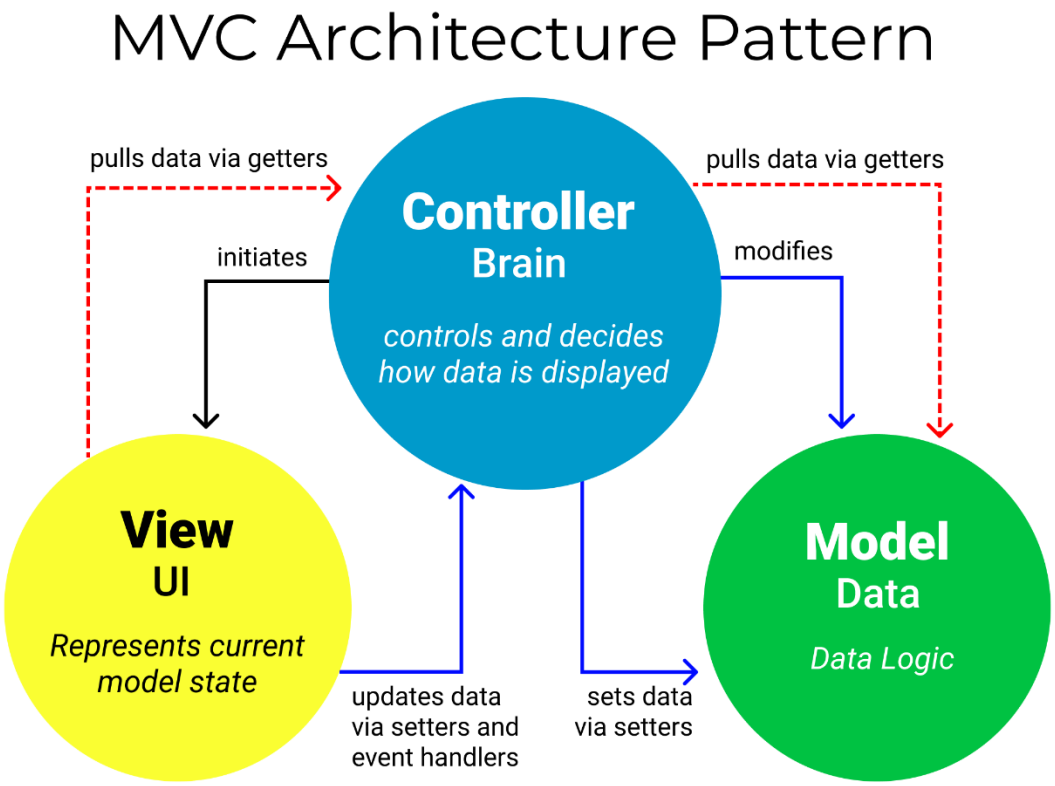

Figure 2: System Level Architecture

2.3 UI Design

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

Home

It will show basic information about the bike shop. User can see different options on this page to browse the application. It will have options to visit the following pages:

Buy Bikes and Parts, Rent Bike, Return Bike

Order History

It will show basic information about the user such as Order Date, Sub-Total, University ID and Return/Due Dates for its history of purchases and rentals.

Buy Bikes and Parts

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

> ****Quantity***- It is the field where the user can enter the number of parts or bikes it needs to purchase.
>
> ***-Keep Shopping***
> After selecting the bike or part, the user can click this button to keep browsing more bikes and parts before placing an order. The user can buy as many products as it wants but the number should be less than the inventory count.
>
> ***-Checkout***
> Finalizes and executed the order.

-Remove From Cart

Removes the product from the cart and lets user update its product choice and quantity.

Rent Bike

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 then click the Checkout button to complete the transaction. The Due Date of the bike can be seen in the Order History page and only 1 bike can be rented at a time.

-Keep Shopping

After selecting the bike, the user can click this button to keep browsing more bikes before placing an order. Only 1 bike would be allowed to rent at a time. The user needs to return the previously rented bike to rent a new bike.

-Checkout

Finalizes and executed the order.

-Remove From Cart

Removes the product from the cart and lets user update its product choice.

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. If the user is within the due date then it doesn't have to pay any fines.

Change Password

It allows the user to change their password.

Register

Every student can register for an account with their umd.edu email address.
Upon clicking Register, an email would be sent to the student to verify and activate their account. They cannot login without the verification.

Login

Students can login to their verified account.

Logout

Logs out of the account.

{Admin Functionalities}

Manage Products

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

Manage Rental Bikes

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

Sales

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 7 tables which would be linked to each other to obtain useful information. Below are the table designs:

AspNetUsers

Field Name	Field Type
ID	*Nvarchar*
UserName	*Nvarchar*
Normalized UserName	*Nvarchar*
Email	*Nvarchar*
Normalized Email	*Nvarchar*
Email Confirmed	*Bit*
Password Hash	*Nvarchar*
Security Stamp	*Nvarchar*
Concurrency Stamp	*Nvarchar*
Phone Number	*Nvarchar*
Phone Number Confirmed	*Bit*
Two Factor Enabled	*Bit*
Lockout End	*Datetime*
Lockout Enabled	*Bit*
Access Failed Count	*Int*

AspNetRoles

Field Name	Field Type
ID	*Nvarchar*
Name	*Nvarchar*

Normalized Name	*Nvarchar*
Concurrency Stamp	*Nvarchar*

AspNetUserRoles

Field Name	Field Type
User ID	*Nvarchar*
Role ID	*Nvarchar*

Products Model

Field Name	Field Type
Product ID	*Guid*
Product Category	*Nvarchar*
Product Description	*Nvarchar*
Product Size	*Int*
Inventory Count	*Int*
Product Prize	*Decimal*
Bike Part Image	*Nvarchar*

Cart Model

Field Name	Field Type
Order ID	*Guid*
Order Date	*Datetime*
UID	*Nvarchar*
Sub Total	*Decimal*

Return Date	*Datetime*

Order Details Model

Field Name	Field Type
Product ID	*Guid*
Product Price	*Decimal*
Quantity	*Int*
Cart Model Order ID	*Guid*
Table ID	*Int*
Product Category	*Nvarchar*
Returned	*Bit*
UID	*Nvarchar*
Order Date	*Datetime*
Return Date	*Datetime*
Bikes Parts Image	*Nvarchar*

Rent Bikes Model

Field Name	Field Type
Product ID	*Guid*
Product Description	*Nvarchar*
Product Size	*Int*
Inventory Count	*Int*
Product Prize	*Decimal*
Rental Bike Image	*Nvarchar*

Flow of data within the tables:

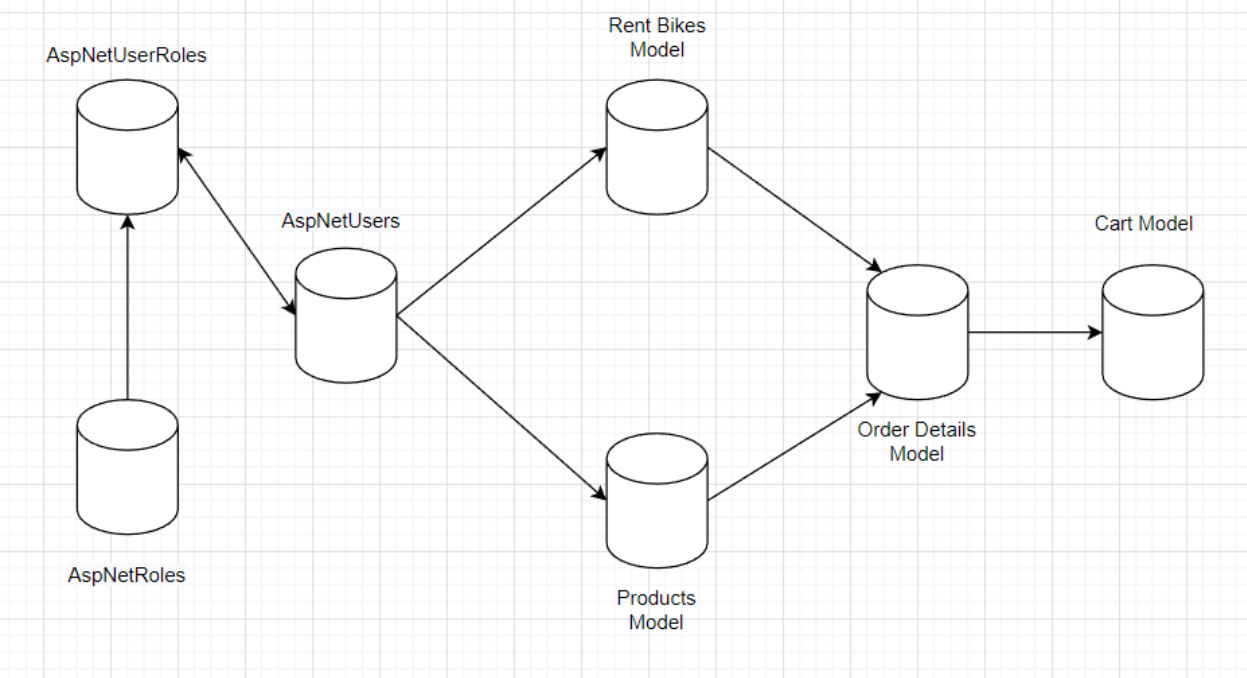

Figure 3:Data Flow Within Tables

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/admins can view the sales page to estimate their earnings. Supervisors can also edit the inventory data of bike and parts/accessories for rent and retail.

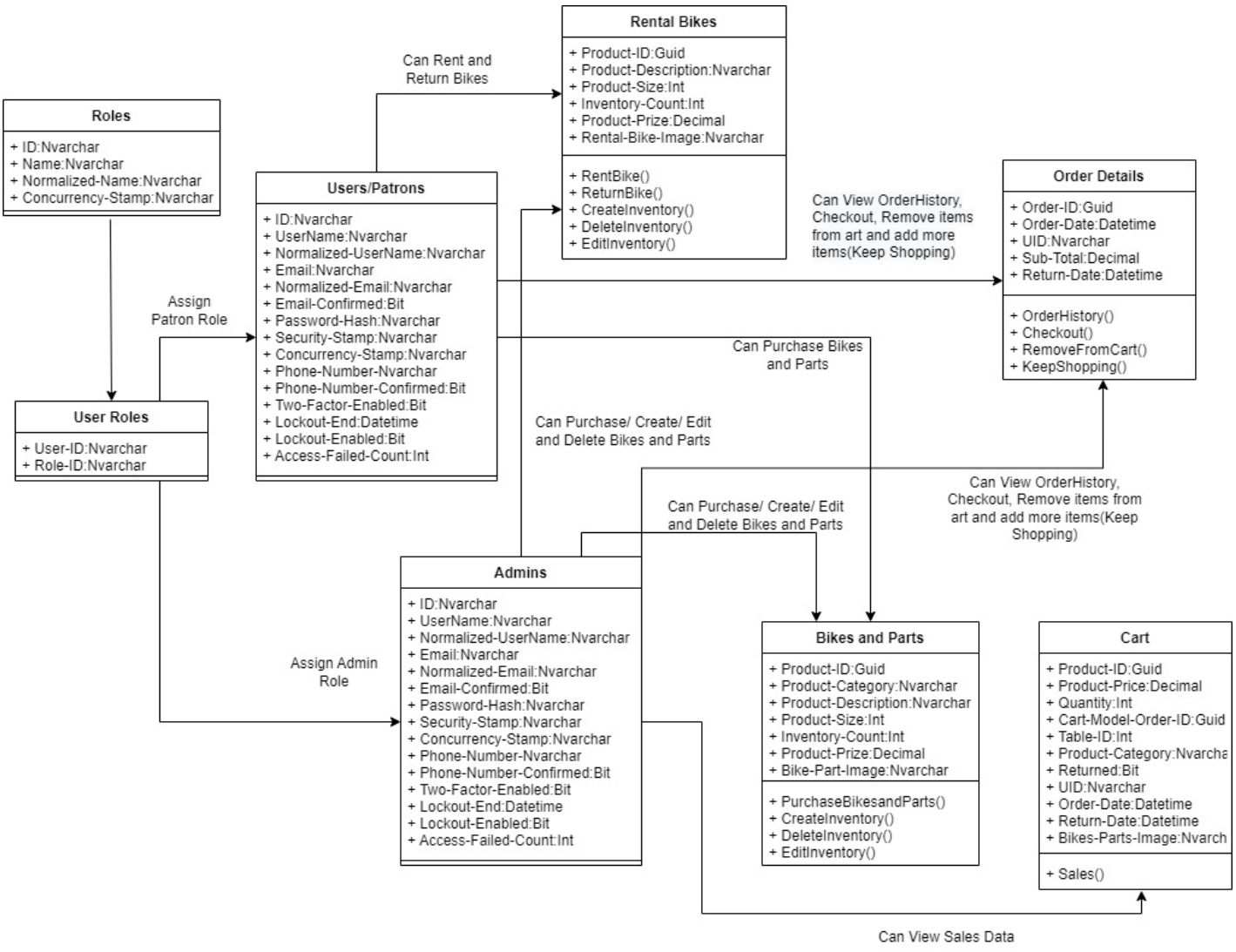

Figure 4: Class Diagram

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. Password can not include username.
- User account will get locked for 24 hours if it enters incorrect password 5 times consecutively.
- Session Management can be implemented by deleting and invalidating the session cookie whenever the user logs out or closes the browser tab.
- 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.
- The URL input would be filtered and sanitized for any kinds of directory traversal attack attempts.
- Anti CSRF and STP tokens would be implemented to prevent CSRF attacks.
- Logging of all user actions would be implemented.
- Email verification of user account would be required upon registering.
- All the HTTP traffic would be redirected to HTTPS traffic.

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/Admins can edit the inventory items.
- Supervisors/Admins can view the sales data.
- A valid user can logout of the system successfully.
- A valid user can change their password.

4.0 References

- https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/
- https://learn.microsoft.com/en-us/dotnet/core/whats-new/dotnet-6
- https://learn.microsoft.com/en-us/aspnet/core/security/anti-request-forgery?view=aspnetcore-6.0
- https://www.yogihosting.com/aspnet-core-identity-username-email-password-policy/