

# MISSOURI STATE UNIVERSITY DEPARTMENT OF COMPUTER SCIENCE



# Hostel Management

Akhil Goud Burra Pranav Jeksani Sundar Teja Akula

Supervisor Dr. Belkhouche

February 21, 2024

# **Project Description**

# **Project Summary**

While we pursuing a degree in India, the rising number of educational institutions has led to a corresponding increase in private hostel facilities for student accommodation. However, we have personally encountered several challenges: The primary issue was relying on offline options for booking facilities, which created multiple difficulties:

- 1. Offline booking options for private hostel rooms near the university involve waiting for room assignments for a very long time.
- 2. The absence of an online system for booking private study rooms within the hostel will always give us trouble in emergency situations.
- 3. Reporting lost items in the hostel was restricted to offline channels, making us wait until the next working day to register a complaint.
- 4. Lack of online access to view whether the books are available in the library for my specific stream will always be an irritation to physically check.

To address these challenges, we want to develop a simple and clean web application that resolves the issues. This web app will offer an online booking system for hostel rooms, study rooms, a complaint registration process, and an online system to check if the books are available for the specific stream. This ensures a more convenient and accessible hostel management system.

#### Software Features

# **End User**

#### Feature 1

Login – For the login, you must log in first before you can access the system.

#### Feature 2

Sign Up – For the sign-up, you will fill in the forms, such as your username, password, and email address.

# Hostel Room Booking Module

#### Feature 1

End-users can book a hostel room by sending a "Request" to the admin user.

#### Feature 2

End-users can book a hostel room monthly or daily.

#### Software Features

### **End User**

# Hostel Room Booking Module

### Feature 3

End-users can view whether the request is accepted or rejected by the administrator.

# Study Room Booking Module

# Feature 1

End users can book the hostel study room monthly or daily.

# Posting Missing Items Module

# Feature 1

End user can post the missed items by identifying name of the item, item description, contact information, uploading lost item image.

# Viewing Paid Transactions Module

# Feature 1

End user can view all the transactions that they made for booking hostel room and study room.

### Payment Module

#### Feature 1

After end user sends a request for hostel room booking, the administrator can see the request and upon the availability of rooms, admin will accept or reject the request.

If admin accepts the request, end user will get an option to pay. If admin rejects the request, end user will not get the option to pay.

#### **Software Features**

### End User

# Fetching Library Books Module

# Feature 1

End users will have the capability to fetch all the books for their specific stream and their availability status.

#### Feature 2

End users will be authenticated by authentication layer.

#### Feature 3

End users are protected by authorization layer. So, end users are restricted to perform only fetch operations.

#### Feature 4

End users are restricted to fetch 10 times per minute.

# Admin User

#### Feature 1

Login – For the admin login, you must login first before you can access the system.

# Feature 2

Sign Up—For the admin sign up, in-built Django framework provides an option for creating superusers or admin users.

# Hostel Room Management Module

#### Feature 1

admin user can add rooms by specifying Room No and No of beds.

# Feature 2

admin user will also have to an option to view all his added rooms.

# Software Features

### Admin User

# Viewing New Requests Module

# Feature 1

Admin user can accept or reject the request made by the end user by checking the availability of the room.

# Viewing Paid Transactions Module

### Feature 1

admin user can view all the transactions that are made by the end users.

# Managing Library Books Module

#### Feature 1

admin user will have capability to create, Read, update, and Delete the books, streams, and book availability status.

#### Feature 2

admin users are protected by authentication layer.

#### Feature 3

admin users are protected by authorization layer. So, they can perform create, update and delete operations.

# Feature 4

Admin users are restricted to fetch 20 times per minute.

# Software Development

# Product Backlog

### To Do List

# **End Users**

- Item 1 Login System: Allowing end users to login, with their credentials.
- Item 2 Registration System: Allowing end users to register, with their username, password, and email address.
- Item 3 Home: Create a home screen, to display any important information.
- Item 4 Logout: Create a end user logout functionality.

### **Admin Users**

- Item 5 Login System: Allowing admin users to login, with their credentials.
- Item 6 Registration System: Allowing admin users to register, with their username, password, and email address. After registration, manager will have capabilities to add the user to admin group.
- Item 7 Home: Create a home screen, to display any important information.
- Item 8 Logout: Create a admin user logout functionality.

# Hostel Room Booking Module

# **End Users**

- Item 9: Create a method to send the request to admin user for booking Hostel Room.
- Item 10: Create a method which allows end users to book the hostel room monthly or daily.
- Item 11: Allow the end users to see weather the sent request is accepted or rejected by the admin user.

# Software Development

# Product Backlog

### To Do List

# Hostel Room Management Module

#### **Admin Users**

- Item 12: Create a method where, admin users can add the rooms by specifying RoomNo and No of Beds.
- Item 13: Create a method where, admin users can view all the rooms that they added.

# Study Room Booking Module

#### **End Users**

• Item 14: Create a method where, end users should have an option to book the hostel study room monthly or daily basis.

# Viewing New Requests Module

#### Admin Users

• Item 15: Create a method where, admin users should have a option to accept or reject the request that has been sent by the end user.

# Payment Module

#### **End Users**

• Item 16: Create a method where, End users should have an option to pay the payment only if, admin user accepts the request sent by the enduser and end user should not get an option to pay only if, admin user rejects the request.

# Software Development

# **Product Backlog**

# To Do List

# Viewing Paid Transactions Module

### **End Users**

• Item 17: Create a method where, End user should able to view all the transactions that they made for booking hostel room and study room.

# **Admin Users**

• Item 18: Create a method where, admin user should able to view all the transactions that are made by the end users.

# Posting Missing Items Module

# **End Users**

• Item 19: Create a method where, End users should able to post the missed items by identifying name of the item, item description, contact information, uploading lost item image.

# Software Development

# **Product Backlog**

### To Do List

# Fetching Library Books Module

#### **End Users**

- Item 20: Create a method where, End users should have the capability to fetch all the books for their specific stream and their availability status.
- Item 21: End users should able to authenticate by the token based authentication system.
- Item 22: End users should be restricted to perform only fetch operations.
- Item 23: End users should be restricted to fetch 10 times per minute.

# Managing Library Books Module

### **Admin Users**

- Item 24: admin user should have the capability to create, Read, update, and Delete the books, streams, and book availability status.
- Item 25: admin users should be protected by authentication layer.
- Item 26: admin users should be restricted to perform create, update and delete operations.
- Item 27: Admin users should be restricted to perform CRUD operations 20 times per minute.

# Software Development

# Product Backlog

# Supplementary Tasks

#### Library Management

# Database Design

- Item 28: Identify the Entities and their Attributes.
- Item 29: Identify the relationships for the entities.
- Item 30: Derive the Functional Dependencies.
- Item 31: Identify the problems that exists in the unnormalized tables.
- Item 32: Normalize the created Tables to Third Normal Form.

# Setting Up Interface Setup

- Item 33: Create a setup for End User Authentication.
- Item 34: Create a setup for Admin User Authentication.
- Item 35: Create a setup for Registering the users.
- Item 36: Create a setup for adding the users to manager group and for deleting the user from manager group.

# Testing

- Item 37: Create the setup for testing Interface URL Endpoints.
- Item 38: Create a setup for writing the test cases for individual components.

### Documentation

• Item 39: Setup the Open API swagger to document the URL Endpoints Methods.

Item 40 Dark Theme: Implement dark light theme.

# Software Development

# **Sprint Backlog**

- Task 1: Implement Item1 End user Login System.
- Task 2: Implement Item2 End user Registration System.
- Task 3: Implement Item4 End user Logout System.
- Task 4: Implement Item5 Admin user Login System.
- Task 5: Implement Item6 Admin user Registration System.
- Task 6: Implement Item8 Admin user Logout System.
- Task 7: Implement Items 28,29.
- Task 8: Implement Items 30, 31, 32.
- Task 9: Implement Items 33, 34.
- Task 10: Implement Items 35, 36.
- Task 11: Implement Items 37, 38, 39, 40.

# Software Development

# **Sprint Backlog**

- Task 12: Implement Stream Model by looking at Item 32.
- Task 13: Implement Item 37 for Stream Model.
- Task 14: Implement Item 38 for Stream Model.
- Task 15: Implement Items 20 and 24 for Stream Model.
- Task 16: Implement Items 21 and 25 for Stream Model.
- Task 17: Implement Items 22 and 26 for Stream Model.
- Task 18: Implement Items 23 and 27 for Stream Model.
- Task 19: Implement Item 39 for Stream Model.
- Task 20: Implement Items 9, 10, 11.
- Task 21: Implement Items 7, 12, 13, 15 and 18.

# Software Development

# **Sprint Backlog**

- Task 22: Implement Book Model by looking at item 32.
- Task 23: Implement Items 20, 24 for Book Model.
- Task 24: Implement Item 21 for Book Model.
- Task 25: Implement Item 25 for Book Model.
- Task 26: Implement Item 22 for Book Model.
- Task 27: Implement Item 26 for Book Model.
- Task 28: Implement Item 23 for Book Model.
- Task 29: Implement Item 27 for Book Model.
- Task 30: Implement Items 37, 38 for Book Model.
- Task 31: Implement Item 39 for Book Model.
- Task 32: Implement Items 14, 16.
- Task 33: Implement Items 17, 18.

# Software Development

# **Sprint Backlog**

- Task 34: Implement Availability Model by looking at item 32.
- Task 35: Implement Items 20, 24 for Availability Model.
- Task 36: Implement Item 21 for Availability Model.
- Task 37: Implement Item 25 for Availability Model.
- Task 38: Implement Item 22 for Availability Model.
- Task 39: Implement Item 26 for Availability Model.
- Task 40: Implement Item 23 for Availability Model.
- Task 41: Implement Item 27 for Availability Model.
- Task 42: Implement Items 37, 38 for Availability Model.
- Task 43: Implement Item 39 for Availability Model.
- Task 44: Implement Item 19.