CampusFlow

Instructor: Rakesh Matam

Harsh Choudhary
B.Tech, 3rd Year
Roll Number: 2201086
Indian Institute of Information Technology, Guwahati

Date: February 26, 2025

Table of Contents

Problem Description

Late Entry Management:

- Need an application to automate and digitize the student late entry process.

Delivery Management:

- Need a system to streamline the delivery process for student packages and ensure secure verification.

My Solution Approach: Late Entry Management

Key Features:

- **QR Code Scanning:**
  - Security guards scan a student's QR code upon late entry.
  - Details (e.g., name, ID, time of entry) are automatically recorded in the warden's database.
- **Proof Upload Facility:**
  - Students can upload proof (e.g., medical certificate, permission slip) for late entry.
  - Warden can view and verify the proof.
- **Warden Approval/Rejection:**
  - Warden has the functionality to approve or reject late entries based on proof.
  - Automated notifications sent to students about approval status.

My Solution Approach: Delivery Management

Key Features:

- **Integration with popular e-commerce API's:**
  - Automatically fetch order details using APIs.
  - Display order info in the application's student interface.
- **Location Selection and display of order to security:**
  - Students choose delivery location (Boys or Girls Hostel).
  - Display student name and phone number at selected location's security desk.
- **OTP Verification:**
  - Security requests OTP via the application.
  - Students provide OTP when requested.
- **Delivery Confirmation:**
  - Notify students upon successful delivery.

Project Description

- **Objective:**
  - Create a secure and efficient system for managing student gate entries and deliveries.
  - Eliminate manual logging, ensuring transparency, accountability, and convenience.
- **Scope:**
  - **Late Entry Management:**
    - QR code-based entry logging instead of manual registers.
    - Proof submission and approval system for wardens.
    - Automated notifications for students on approval status.
  - **Delivery Management:**
    - Integration with e-commerce APIs for auto-fetching order details.
    - Secure OTP-based verification for package handover.
    - Real-time delivery notifications for students.
- **Target Users:**
  - **Students** – Seamless entry tracking and secure deliveries.
  - **Security Personnel** – QR-based verification and OTP authentication.
  - **Wardens** – Approve/reject late entries based on proof.
  - **Delivery Staff** – Ensure secure and verified package handover.

Functional & Non-Functional Requirements

Functional Requirements:

- QR Code Scanning for Late Entry
- Permission Proof Management
- Order Notification and OTP Verification

Non-Functional Requirements:

- Security and Privacy
- Reliability
- Performance

Design Approach

Development Model:

- **Iterative Model**
  - Enables incremental development with continuous feedback
  - Allows for refinement in each iteration based on testing results
  - Reduces risk by addressing issues early in the development cycle

Architecture:

- **Model-View-Controller Architecture**
  - **Model:** Handles data, business logic, and database interactions
  - **View:** Manages UI components and presentation layer
  - Facilitates independent development of frontend and backend components

Tech Stack

Backend:

- **Node.js** - JavaScript runtime
- **Express.js** - Web framework
- **MongoDB** - NoSQL database
- **Firebase** - Authentication
- **JWT** - Secure authentication

Frontend:

- **React Native** - Cross-platform

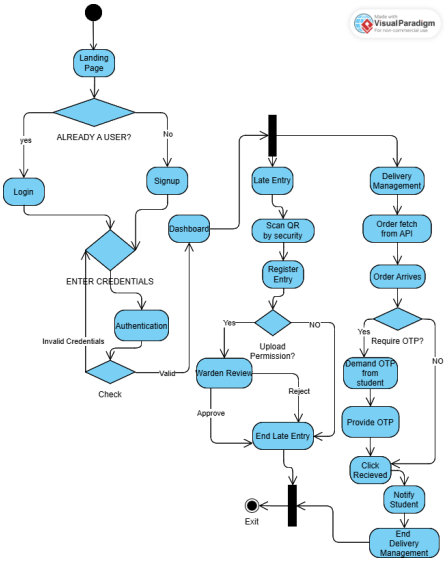
QR Code Technology:

- **QR Generation** - `qrcode`
- **QR Scanning** - `react-native-camera`
- **Data Encoding** - Student details

APIs:

- E-commerce tracking APIs
- Notification services

System Architecture



Progress Till Now

- **Completed Tasks:**
  - Requirements analysis and system architecture design
  - Authentication system using Firebase and JWT
  - Database setup for user and personnel management
  - Backend implementation:
    - Student routes (Login, Signup, Entry Details, Proof Upload)
    - Admin routes (Entry Management, Status Updates)
    - Security routes (Entry Registration)
  - Controller implementation for core functionality

Future Plan of Action

- Implementation and testing of backend components
- Building frontend modules
- Integrating frontend with backend
- Comprehensive testing of the application
- Deployment

Thank You!