A.I.N.HATHARASINGHE

E2340065

Sunday, 13 October 2024

PROJECT REPORT

VEHICLE PARKING MANAGEMENT SYSTEM FOR SLTMOBITEL HEADQUARTERS

CONTENT

1. SECTION 01 – INTRODUCTION

This project report, presented by A.I.N. Hatharasinghe, a student at the University of Moratuwa pursuing a Bachelor of Information Technology, outlines a comprehensive plan to address a key challenge in the field of IT. The proposed Vehicle Parking Management System aims to enhance efficiency and user experience by leveraging innovative technologies such as Visual Studio, C#, and an SQL database. The system will provide real-time parking availability, automate parking spot allocation, and implement access control for authorized vehicles, significantly reducing congestion and wait times. Through rigorous research and development, this project not only addresses operational inefficiencies but also contributes valuable insights and practical applications to the broader academic and professional community.

2. SECTION 02 - SIMILAR SYSTEMS

1. Automated Parking Management System for University Campuses
A university implemented an automated parking system where students and staff register their vehicles online. The system tracks available parking spots in real-time, allowing users to book spots before arrival. It uses RFID for vehicle identification at the gates. Data about parking occupancy and usage patterns is stored in an SQL database, and users can check the availability of parking spaces through a mobile app.

2. Shopping Mall Parking System
This system is designed to manage a multi-level parking facility for a shopping mall. It provides features such as parking spot allocation based on vehicle type (car, motorcycle), registration for monthly parking, and real-time updates on available parking spots. SQL Server is used to store user and vehicle information, while C# is used to build the management software.

3. Corporate Office Parking System
A large corporate office developed an automated parking system where employees register their vehicles. The system uses an SQL database to store vehicle and employee details, while C# handles the core functionality. The system features an RFID scanner at the gate, which reads vehicle tags and allows entry if the vehicle is

registered and an available parking spot is allocated. It includes real-time availability tracking for designated employee parking areas.

4. Airport Parking Management System
This system is implemented in airport parking lots, where thousands of cars need to be managed efficiently. The system uses IoT sensors to detect the availability of parking spaces in real-time. C# is used to develop the management software, which communicates with the SQL Server database to store and retrieve parking space availability and vehicle registration data. The system also integrates payment gateways for automatic parking fee deductions.

3. SECTION 03 - SOLUTION

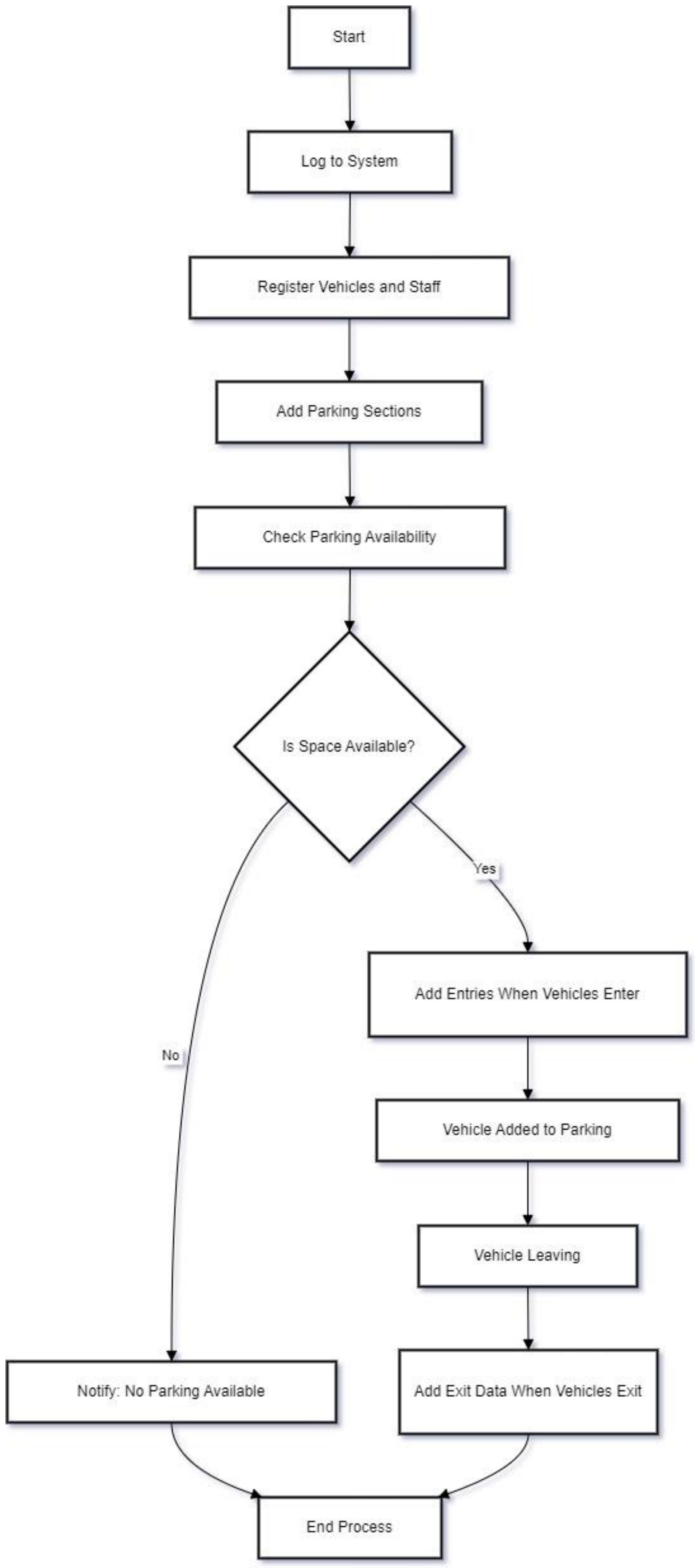
3.1. Functional Requirements

- Real-time Parking Availability Monitoring: Display available parking spots at the entrance.
- Vehicle Registration: Register staff vehicles with unique identifiers.
- Parking Spot Allocation: Automatically assign parking spots to registered vehicles.
- Database Management: Store vehicle data, staff details, and parking availability status.
- Reporting and Analytics: Generate reports on parking space utilization and provide decision-support analytics.

3.2. Non-Functional Requirements

- Scalability: Support multiple users and vehicles simultaneously.
- Security: Ensure secure access to the system, protecting user and vehicle data.
- Performance: Provide real-time updates with minimal latency.
- Reliability: Ensure consistent availability of the system without downtime.
- Usability: Offer a user-friendly interface with intuitive navigation.
- Maintainability: Allow easy updates and bug fixes to the system.
- Efficiency: Optimize the system to minimize resource consumption

3.3. Flowcharts


Start
Log to System
Register Vehicles and Staff
Add Parking Sections
Check Parking Availability
Is Space Available?
Yes
No
Add Entries When Vehicles Enter
Vehicle Added to Parking
Vehicle Leaving
Add Exit Data When Vehicles Exit
Notify: No Parking Available
End Process

3.4. Pseudocode

```
Step 1: Start the process
Step 2: Log in to the system
Step 3: Register vehicles and staff
Step 4: Add parking sections
Step 5: Check parking availability
Step 6: Decision point - Is there space available?
Step 7: If yes, add vehicle entry when vehicle enters
        Step 8: Add vehicle to parking
        Step 9: Process when the vehicle is leaving
        Step 10: Add exit data when the vehicle exits
        Step 11: End the process
Step 7: If no space, notify that parking is unavailable
Step 8: End the process
```