

A PROPOSED OFFERING OF PARKING MANAGEMENT SYSTEM FOR SM MALL OF ASIA

A Design Documentation Presented to the
Faculty of Datamex College of Saint Adeline, Inc.

In Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Information Technology

By:

Mallory Makyla P. Chua

Ben O. Onde

Sean Roscoe D. Panday

Kim Razel N. Torrazo

August 2025

DESIGN DOCUMENTATION

INTRODUCTION

This document is aimed at giving an elaborate design of the Parking Management System (PMS) that is being produced on the SM Mall of Asia. This document will serve as a reference and guide to all the workforce in the project including the development team, testers, system administrators, as well as the stakeholders. It creates a clear and systematic description regarding the construction of the system, the major sections, the way bearing on which the parts will interact with, and the overall deployment and maintenance of the system after it is capable of being used.

The early presentation of these details would prepare all the members of the project so that they are heading one direction and receive all the details required, thereby preventing duplication of work, or scope changes towards the end of the project. It, also, gives the client and decision makers an opportunity to discuss the proposed design prior to development, to ensure that the system satisfies the needs of operation and business requirements. In brief, it is a roadmap to the developers as well as a win or lose tool that helps build the stakeholders package to maintain coherence, effectiveness and accountability within the project lifecycle.

Overview

The PMS is a locally hosted, web-based system designed to simplify and automate mall parking operations. It will allow attendants to register vehicles, automatically calculate parking fees based on time, and provide administrators with real time monitoring of parking slots. Reports on daily or periodic parking activities will also be generated. The system will run on a local computer network within SM Mall of Asia and will not require internet access.

Scope of Design Document

This document covers the system architecture, database design, user interface design, component design, data flows, security measures, performance requirements, error handling, deployment plan, and maintenance strategy for the PMS. It also includes revision history for tracking changes.

SYSTEM ARCHITECTURE

The Parking Management System is selected to operate independently due to the reason that it is reliable, secure, and works 24/7. Cases of internet failure or external systems malfunction under any circumstances that involve shutdown should not cause dissolution of the parking facility and a related independent set-up can ensure that customers can have services all the time. The local storage and processing of all the important data including information about the vehicles, availability of the parking slots faster execution and also lessen over reliance on the external server or using the services of some third parties. This design also enhances the security since exposure to online threats like hacking is not limited. Being a fully committed, independent solution, a solution as a standalone will ensure that the process of parking management will go on flawlessly without necessarily being connected to the network. Broadly, the stand-alone system would be best when it comes to achieving efficiency and stability in the parking processes.

SYSTEM ARCHITECTURE OVERVIEW

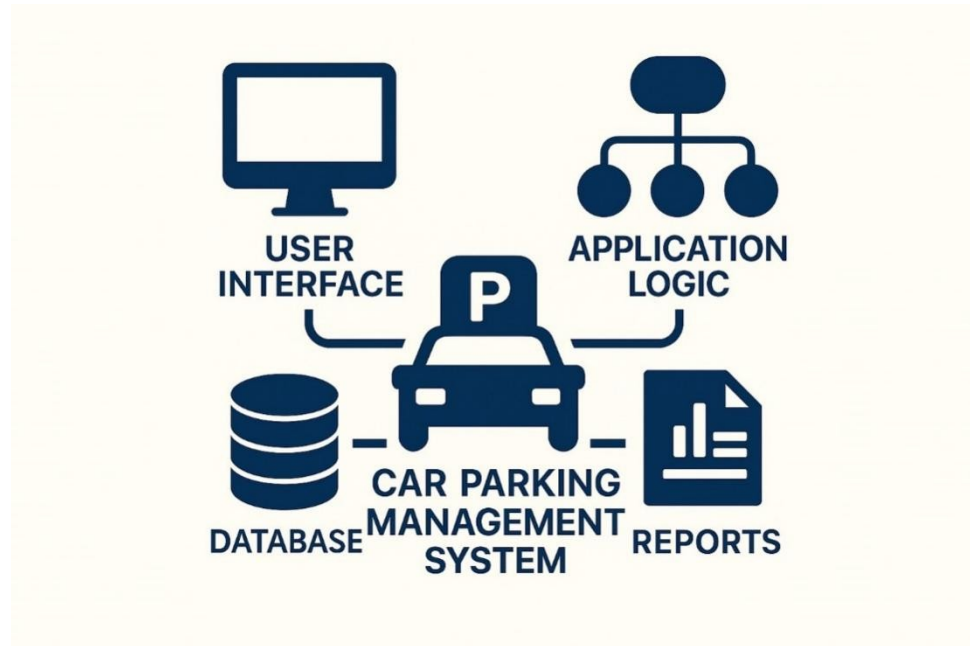


Figure 1. Dataflow diagram for Parking Management System

High Level Component

User Interface (Frontend – Visual Basic)

- Staff (Attendants): Use the Visual Basic application to log vehicle details, assign parking slots.
- Admin (Supervisors/Managers): Access the admin dashboard in the application to monitor overall parking operations.

Application Logic (Backend – Visual Basic Code):

- Processes vehicle registration, calculates parking fees, monitors slot usage, and generates reports.

Enforces role-based access:

- Staff: Limited to daily operations (registration, fee processing).
- Admin: Full access, including reports, system monitoring, and user management.

Database (SQL Server – managed via SSMS)

Stores all records, including:

- User accounts (UserID, Username, Password, Role: Staff/Admin)
- Vehicle records (PlateNumber, EntryTime, ExitTime, Fee)
- Parking slots (SlotID, Status)
- Transactions (TransactionID, VehicleID, Duration, FeeAmount, Date)

Interactions

- Staff enter vehicle details in the Visual Basic UI.
- Application Logic validates the data and sends SQL queries to the SSMS database.
- Database saves and updates records.
- Admins use their dashboard to query the database, generate reports, and monitor usage.

Database Design

RDB Diagram

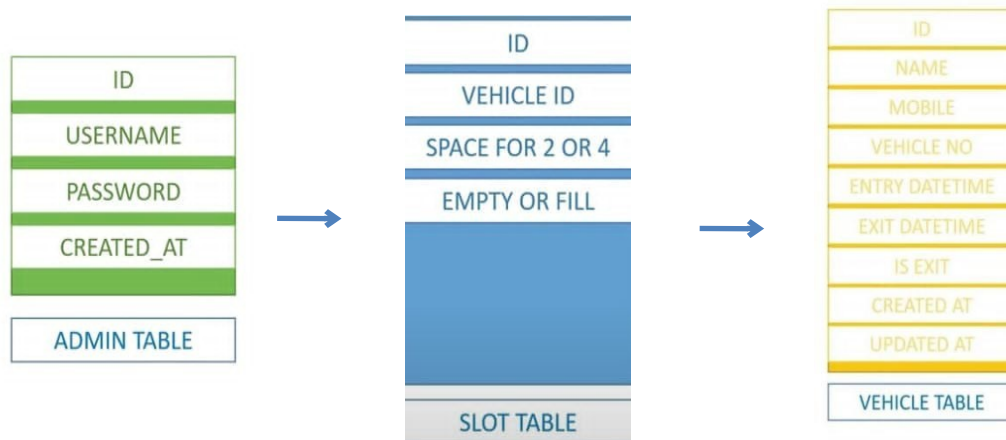


Figure 2. Relation Database for Parking Management System

Relationships

- A Vehicle is assigned to a Parking Slot.
- A Transaction is linked to a Vehicle.
- Users manage these operations.

Database Tables and Fields

- **Vehicle:** VehicleID, PlateNumber, EntryTime, ExitTime, Fee
- **Parking Slot:** SlotID, Status (Available/Occupied)
- **Transaction:** TransactionID, VehicleID, Duration, FeeAmount, Date
- **User:** UserID, Username, Password, Role (Admin/Staff)

Data Normalization

- Database normalized to 3rd Normal Form (3NF).
- Removes redundant data (e.g., parking fee stored only in transactions, not in multiple tables).
- Ensures integrity with foreign keys (VehicleID in Transaction table).

USER INTERFACE DESIGN

Initial Setup Page - The Initial Setup Page is used to create the system's first administrator account, which will manage and configure the parking management system. Once the account is set up, the system becomes ready for use.

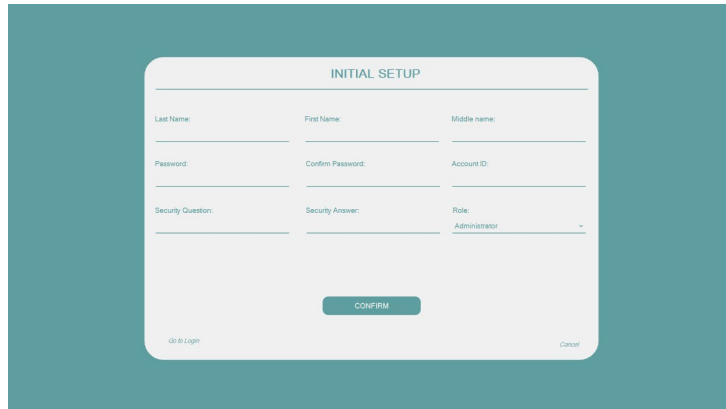
The image shows a 'INITIAL SETUP' form with a teal background. The form is a light gray rounded rectangle with the title 'INITIAL SETUP' at the top. It contains several input fields: 'Last Name', 'First Name', 'Middle name', 'Password', 'Confirm Password', 'Account ID', 'Security Question', 'Security Answer', and 'Role'. The 'Role' field is a dropdown menu with 'Administrator' selected. At the bottom of the form is a teal 'CONFIRM' button. In the bottom left corner of the form is a small link 'Go to Login' and in the bottom right corner is a small link 'Cancel'.

Figure 3. Initial Setup Page

Login Page - The Login Page provides secure access to the Parking Management System. Users enter their registered username and password to access the system based on their role (e.g., Administrator, Staff, or User). This ensures only authorized accounts can manage or use the parking system.

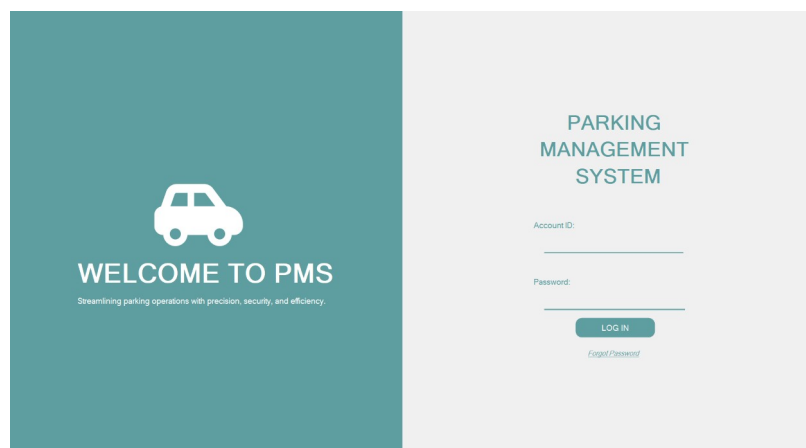
The image shows the 'PARKING MANAGEMENT SYSTEM' login page. It has a split background: teal on the left and light gray on the right. The teal section contains a white car icon, the text 'WELCOME TO PMS', and the tagline 'Streamlining parking operations with precision, security, and efficiency.' The light gray section contains the title 'PARKING MANAGEMENT SYSTEM' and login fields for 'Account ID' and 'Password'. Below these fields is a teal 'LOG IN' button and a small link 'Forgot Password'.

Figure 4. Login Page

Forgot Password - The Forgot Password feature allows users to securely reset their account credentials if they forget them. By verifying their registered email or answering a security question, users can create a new password and regain access to the Parking Management System.

FORGOT PASSWORD

Account ID:

Security Question:

Security Question Answer:

Current Password:

New Password:

Confirm New Password:

[CONFIRM](#)

[Back to Login](#)

Figure 5. Forogt Password Page

Dashboard Page - The Dashboard serves as the central control panel of the Parking Management System. It provides an overview of key information such as available and occupied parking slots, recent transactions, alerts, and system statistics. From here, administrators and staff can quickly access core features like managing slots, monitoring parking activity, and viewing reports.

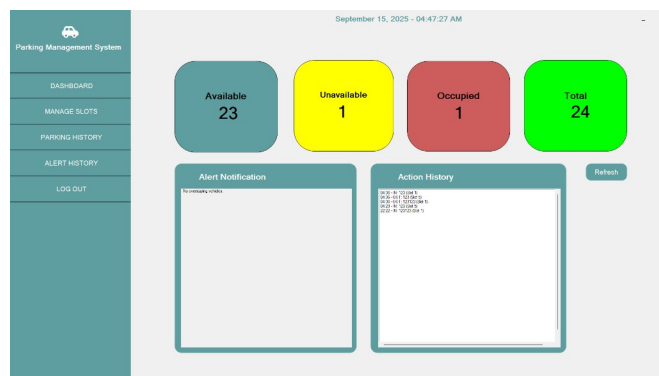


Figure 6. Dashboard Page

Manage Slot Page -It allows administrators and staff to register, update, or remove vehicle records in the system. It stores essential details such as plate number, vehicle type, and owner information, ensuring accurate tracking and management of parked vehicles.



Figure 7. Manage Slot Page

Parking History Page - The Parking History Page provides a record of all past parking transactions. It displays details such as vehicle plate number, entry and exit time, duration of stay, and payment history, allowing administrators and staff to review and track parking activities.

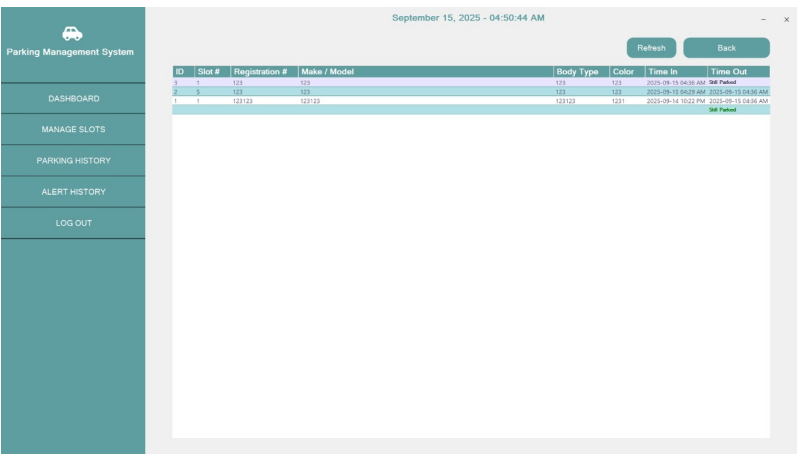


Figure 8. Parking History Page

Alert History Page -The Alert History Page records all system-generated alerts, such as parking violations, unauthorized access, or slot availability issues. It allows administrators and staff to review, track, and manage past alerts for better monitoring and security.



Figure 7. Alert History Page

Design Principles and Usability Considerations

The system will prioritize clean and simple layouts, user-friendly navigation, and consistent input forms. These considerations are crucial since administrative staff must be able to use the system with minimal training. Error-prevention techniques such as mandatory fields and data type validation will also ensure correct data entry.

Component Design

The Parking Management System is divided into several core components. Each component has a specific role but works together to ensure smooth operation.

1. User Interface (UI)

- Provides screens for staff and administrators.
- Includes pages such as Login, Home, Add Vehicle, Manage Vehicle, and History.
- Allows users to enter details and view system outputs.

2. Application Logic

- Acts as the core processor of the system.
- Handles vehicle registration, slot assignment, fee calculation, and ticket generation.

3. Database

- Stores all important records such as vehicles, users, transactions, and parking slots.
- Ensures data integrity and easy retrieval.
- Supports report generation by providing historical data.

4. Authentication and Security

- Validates user login credentials.
- Ensures only authorized users can access system features.
- Protects sensitive data using encryption.

5. Reporting Module

- Provides administrators with real-time and historical reports.
- Tracks slot usage, fees collected, and daily transactions.

6. Error Handling and Logging

- Validates user inputs to avoid mistakes.
- Logs system errors, failed logins, and transaction issues.
- Helps administrators monitor and troubleshoot system performance.

Dependencies

User Interface depends on Application Logic: The User Interface (UI) is the front-end component used by attendants and administrators to interact with the system. However, it does not process business rules or handle data directly. Instead, it depends on the application logic layer to validate inputs, calculate fees, assign slots, and generate reports. This separation ensures that the UI remains simple, while the backend handles complex processes.

Application Logic depends on the MySQL Database: The application logic requires a reliable database to store and retrieve persistent data. Every operation—such as vehicle registration, slot status updates, and transaction logging—relies on SQL queries executed against the MySQL database. The database ensures integrity by enforcing relationships between tables (e.g., Vehicles, Parking Slots, Transactions, and Users). Without the database, the backend cannot maintain accurate parking record.

All Modules Are Interconnected through Backend Services: The PMS follows a modular architecture, but all modules (authentication, vehicle management, fee calculation, slot monitoring, reporting) are tied together by the backend services. For example, the authentication module must interact with the user table in the database, while the fee calculation module must access both the vehicle and transaction tables. These backend services act as the "glue" that keeps the system consistent and ensures smooth communication between UI, logic, and database layers.

Security Design

The security design of the PMS focuses on ensuring that sensitive parking data and user information are protected from unauthorized access.

Requirement

The system must provide secure login functionality, ensuring that only authorized staff and administrators can access its features. Role-based access is implemented so that staff members have limited privileges (e.g., vehicle registration and fee processing), while administrators have extended rights (e.g., generating reports, managing users, and overseeing system logs).

Authentication and Authorization

The system uses a username and password validation mechanism. Each user is required to log in with unique credentials, and the backend validates these against stored records in the database. Role-based privileges determine what actions a user can perform: for instance, attendants can register vehicles, while admins can generate reports. This prevents unauthorized access to sensitive operations.

Data Protection

Passwords stored in the database will be encrypted using hashing algorithms to ensure that even if the database is accessed maliciously, plain-text passwords cannot be retrieved. Since the system is locally hosted within the mall's intranet, external exposure is minimized, reducing risks of online attacks. Access logs will track login attempts, helping detect suspicious activities.

Performance Design

The PMS must perform efficiently to handle heavy mall traffic and peak parking hours.

Requirements

All essential operations—such as registering a vehicle, updating parking slots, and calculating fees—must be completed in under 3 seconds to ensure a smooth flow of vehicles and avoid long queues.

Optimization Strategies

The system will use SSMS on frequently queried fields (e.g., PlateNumber, VehicleID) to speed up searches. Caching will be used for reports and dashboard views, ensuring that commonly accessed data loads quickly without repeated database queries. Database normalization up to 3NF ensures that redundant data is eliminated, which also improves performance.

Performance Testing Plan

The system will undergo stress testing to simulate peak mall traffic, such as hundreds of vehicles entering and exiting within a short time. Load tests will be conducted to measure the system's ability to handle simultaneous operations by multiple attendants and administrators without slowdown.

ERROR HANDLING

Error handling ensures that the system responds gracefully to problems, while logging provides a history of issues for troubleshooting.

Mechanisms

Input validation will be applied to all forms, such as vehicle plate numbers and user login credentials, to prevent errors and malicious entries. If invalid input is detected, the system will show clear, user-friendly error messages. Database connection checks will be performed to ensure data integrity.

Logging

The system will log important events vehicle information and parking time. These logs can only be accessed by administrators to monitor system health and security.

Error Codes and Messages

E01 – Invalid login credentials: Displayed when username or password is incorrect.

E02 – Vehicle not found: Displayed when searching for an unregistered or missing vehicle.

E03 – Database connection error: Displayed if the database is unreachable.

Future error codes may be added as the system grows.

Third Party Integrations

Current Version: The initial PMS release does not integrate with any third-party services. It is designed as a self-contained system operating on a local network to minimize complexity and cost.

Future Enhancements

In future versions, the system could be extended with:

- RFID integration to automatically detect vehicle entries and exits.

- CCTV camera integration with license plate recognition (LPR) for automated logging.
- Mobile application supports customers to check parking slot availability and pre-book slots.

Deployment Plan

The deployment of PMS will follow a structured process to ensure smooth installation and minimal disruption to mall operations.

Deployment Process

1. Install the PMS software on designated mall computers.
2. Set up Visual Basic.NET, and SSMS to run the system locally.
3. Configure the local database and import initial setup data (e.g., users and parking slots).
4. Conduct user training sessions for attendants and administrators.
5. Perform pilot testing before going live.

Hardware Requirements

1. Mall computers with standard specifications (8GB+ RAM, stable processors).
2. Reliable power supply, with UPS recommended to prevent data loss during outages.

Software Requirement

Visual Basic.NET for front end programming
SSMS
(for database management)

Configuration Management

Source code will be managed using Git to track changes, ensure version control, and enable rollback if deployment issues occur.

Maintenance and Support

After deployment, ongoing maintenance ensures the system remains reliable and up to date.

Guideline

A 3-month free maintenance period will be provided post-deployment. During this time, the development team will fix bugs and apply necessary optimizations.

Procedures

Maintenance includes applying patches for bugs, updating libraries, and improving system performance. Each update will first be tested in a staging

Escalation Process

1. Staff or admin reports an issue to the system support team.
2. The development team investigates the bug or performance issue.
3. A patch or update is created, tested, and deployed.
4. If unresolved, the issue is escalated to senior developers or system architects.