**A PROPOSED OFFERING OF BUS TICKETING AND BOOKING SYSTEM**

**FOR VICTORY LINER INC.**

A Project Proposal 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:

Arellano, Romeo C.

Estigoy, Ron Kyrie

Flores, Joserey G.

Gadi, Susana

August 2025

**DOCUMENT DESIGN**

**BUS TICKETING AND BOOKING SYSTEM**

**INTRODUCTION**

**Purpose of the Document**

This document is an overall design specification for the Bus Ticketing System It acts as a guide for developers, testers, and stakeholders by outlining the system's architecture, database, components, and design principles.

**Overview of the System**

The Bus ticketing system is intended to mechanize the process of ticketing, reserving, and handling bus tickets. It caters to both customers and administrators through functionalities like seat reservation, fare calculation, payment, scheduling, and report generation.

**Scope of the Design Document**

The document addresses the technical architecture, user interface design, data handling, security factors, and deployment plan of the Bus ticketing system . The document excludes training manuals and detailed project management activities.

**System Architecture**

The system shall be a web and mobile-based client-server system, deployed in a cloud environment for scale and reach.

**High-Level Components**

Client Application (PC ): Passengers' and staff's user interfaces.

Application Server: Business processing, ticket handling, and data management.

Database Server: All ticketing, passenger, and schedule information is stored.

Payment Gateway Integration: Secure payment handling thru face to face.

**Deployment Architecture**

Onboard -server deployment.

Clients access over the on board Network

**Communication Protocols SerVER PROTOCOLS (LAN-BASED)**

TCP/IP (within LAN)Used when ticket counters in the same depot are connected to a central server storing booking records

**Database Design**

ERD

**Database has the following major entities:**

User (Passenger/Admin)

Bus

Route

Trip

Ticket

Payment

**Database Tables And Relationships**

**BUS**

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| bus\_id (PK) | INT | Unique ID for bus |
| bus\_number | VARCHAR | Vehicle registration number |
| bus\_type | VARCHAR | AC, Non-AC, Sleeper, etc. |
| capacity | INT | Total number of seats |

*Table 1 BUS Ticketing and Booking System*

**Route**

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| route\_id (PK) | INT | Unique route ID |
| source | VARCHAR | Starting point |
| destination | VARCHAR | End point |
| distance\_km | DECIMAL | Distance |

*Table 2. BUS Ticketing and Booking System*

**Trip**

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| trip\_id (PK) | INT | Unique trip ID |
| bus\_id (FK) | INT Bus | Which bus is used |
| route\_id (FK) | INT Route | Which route |
| departure\_time | DATETIME | Departure |
| arrival\_time | DATETIME | Arrival |
| trip\_date | DATE | Date of journey |

**Seat**

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| seat\_id (PK) | INT | Unique seat ID |
| bus\_id (FK) | INT → Bus | Bus to which seat belongs |
| seat\_number | VARCHAR | Seat label (e.g., A1, B2) |

*Table 3. BUS Ticketing and Booking System*

**Booking**

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| seat\_id (FK) | INT → Seat | Which seat booked |
| passenger\_id (FK) | INT → Passenger | Passenger details |
| booking\_date | DATETIME | When booking was made |
| status | VARCHAR | Booked, Cancelled |

*Table 4. BUS Ticketing and Booking System*

**Passenger**

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| passenger\_id (PK) | INT | Unique passenger ID |
| name | VARCHAR | Passenger name |
| gender | VARCHAR | Male/Female/Other |
| age | INT | Age |
| phone | VARCHAR | Contact number |

**User/Admin**

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| user\_id (PK) | INT | Unique agent ID |
| name | VARCHAR | Agent name |
| role | VARCHAR | Admin, Clerk |
| username | VARCHAR | Login username |
| password | VARCHAR | Encrypted password |

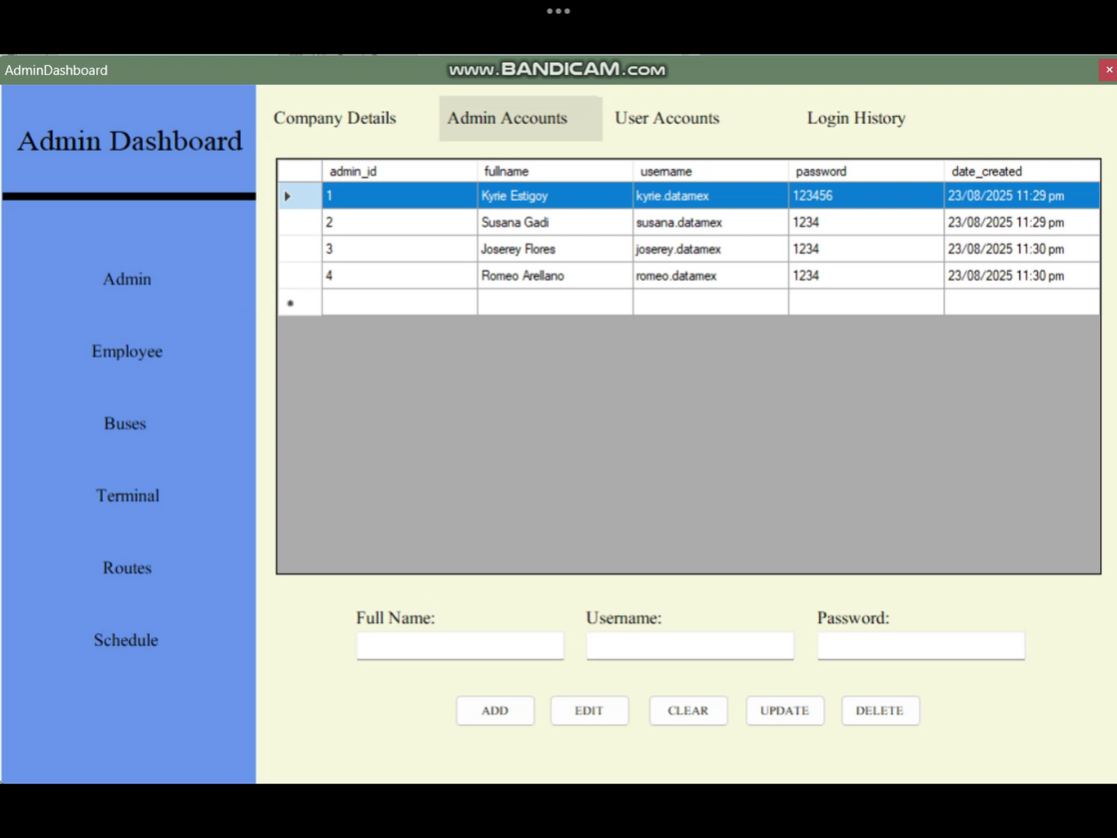
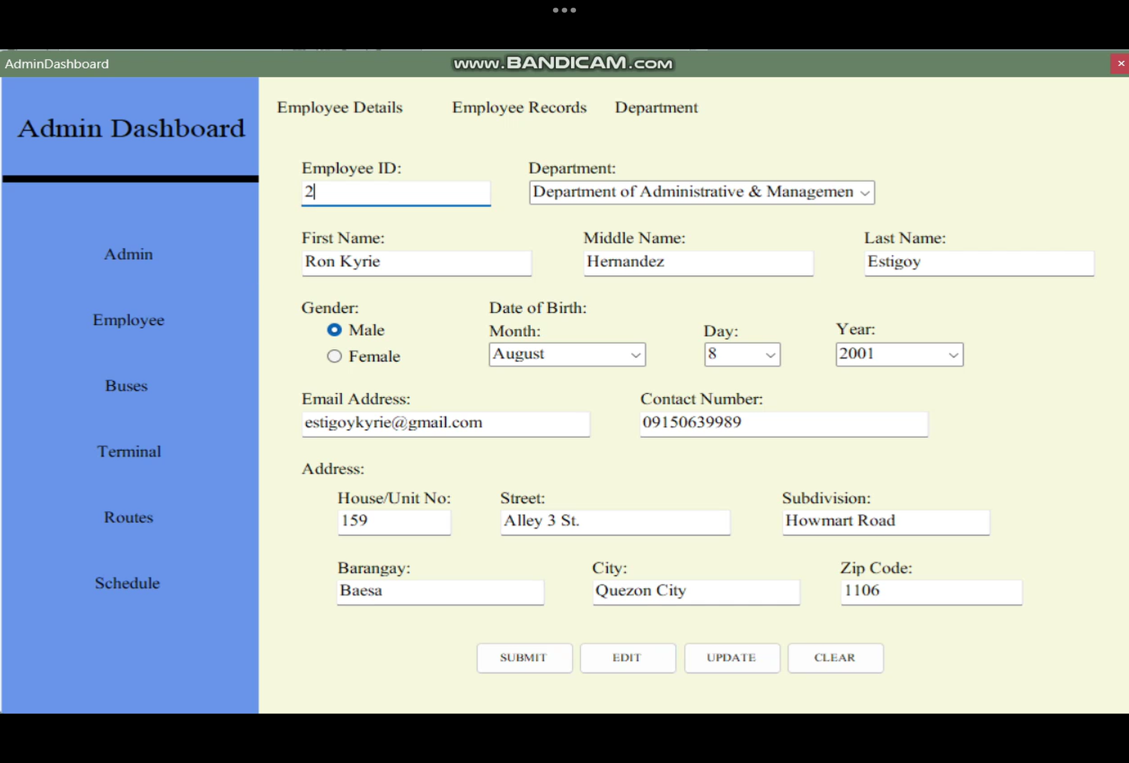
*Table 5. BUS Ticketing and Booking System*

**Data Normalization**

3rd Normal Form (3NF) used to prevent redundancy.

**User Interface Design**

**Admin Interface**

****

****

**Admin UI:** Operations Department

Maintenance & Technical Department

Administrative & Management Department

Customer Service & Support

Employee (Details: ID Number, Name, Age, Address, Contact Number, Position, Hired Date)

**Ui Principles**

Simple navigation (Home, Select, Pay, Confirm).

Responsive design (desktop).

Accessibility compliance (clear fonts, colors, and labels).

**Component Design**

Key Modules

Authentication Module (local login/signup, role-based access

Booking Module (seat reservation, cancellation).

Payment Module (offline settlement)

**Admin Module**

bus, route, schedule Operations Department

Maintenance & Technical Department

Administrative & Management Department

Customer Service & Support

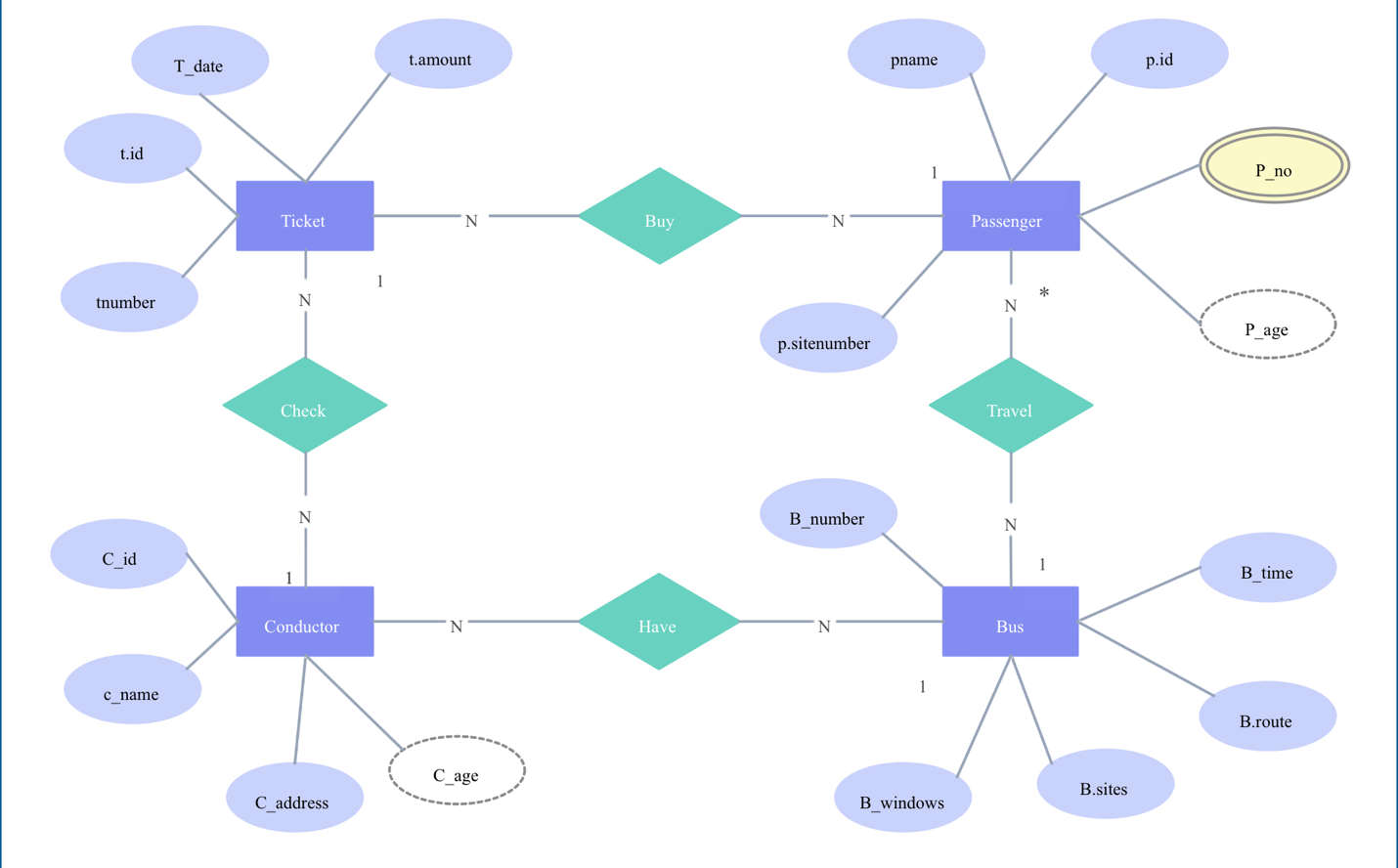
Employee

Notification Module (printed Tickets or QR Code).

**Interface Specifications**

**Dependencies**

Payment gateway (Cash or Card) print of a ticket

**Data Flow Diagrams**

**Security Design**

Authentication: Role-based login with encryption.

Authorization: Passenger/Admin segregation.

Data Encryption: Sensitive data (passwords, payments) encrypted using SSL/TLS.

Backup & Recovery: Local Back Up

**Performance Design**

Requirements: The system must support at least concurrent users (tellers, dispatchers, drivers, passengers at kiosks) operating across LAN-connected terminals

Optimization: Frequently searched fields (PNR, Passenger Name, Route, Seat Number) indexed to improve query performance

Testing: Simulate hundreds of concurrent reservations to ensure system stability

Error Handling Critical failures trigger safe rollback (e.g., if payment fails, reservation is released automatically)

Error Codes: Defined for generic failures (e.g., "ERR101: Seat Reservation ").

**Performance Design**

Performance Requirements and Objectives

The Offline Bus Ticketing System should be fast and responsive so that ticketing agents can serve passengers without delay. The system must allow a ticket to be booked and printed within minutes and seconds and should support transactions per minute during busy hours. It should also work smoothly on standard computers used at bus stations without needing special hardware.

Strategies for Optimizing System Performance

To achieve good performance, the system will use a local database sql so that bookings can be processed instantly without depending on the internet. Data will be saved quickly on the local machine. The system will also use efficient queries and indexing in the database to reduce response times. Since it runs offline, caching will be used to store bus schedules and seat availability, making the booking process faster. The software will be kept lightweight so that it runs well even on older PCs.

Performance Testing Plan

To ensure the system meets performance goals, testing will be carried out under different conditions. The system will be tested for response time, making sure a booking is completed within the required a minutes and seconds. It will also be tested for load handling, checking if it can process multiple bookings at the same time without slowing down. Stress testing will be done by simulating peak hours to see how the system performs when many agents are using it. Finally, long-term testing will be done to check if the system remains stable and efficient after handling transactions.7

**Error Handling Mechanisms And Strategies**

**Graceful Degradation:**

If a module fails ( payment gateway unavailable), the system provides fallback options

Safe Rollback:

If an error occurs during booking or payment, the system automatically cancels the incomplete transaction and releases reserved seats.

Validation & Input Handling:

Prevents invalid entries ( double booking of seats, invalid passenger details).

**Logging Requirements And Specifications**

**Local Logging (offline mode):**

Transactions: Bookings, cancellations, refunds.

Errors: Database failures, sync errors, booking.

**Retention Policy:**

Audit Trail: Every booking and payment transaction must have a corresponding log entry for accountability.

**Error Codes And Messages**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Type** | **Message** | **Description** |
| ERR 1 | Booking Error | Please choose another seat | Duplicate selection prevented |
| ERR 2 | Database Error | Contact admin | Database lost or corrupted |
| ERR 3 | Input Validation | Invalid data entered | Invalaid passenger info or missing fields |
| ERR 4 | System Crash | System must restart | Application crash requiring restart. |
| ERR 5 | Data Corruption | Data integrity issue restored required | Corrupted files or back up needed |

**Third Party Integration**

**QR/Barcode Generator (Local or Library Integration)**

Purpose: Generates secure scannable codes for tickets.

Integration Point: Ticketing module.

Data Exchange:

Input: Passenger booking details (PNR, Seat, Trip ID).

Output: QR/Barcode image embedded in ticket (PNG/PDF).

Offline Handling: Fully local, no internet required

**Step 1: Preparation**

Set up the required hardware (server, workstations, backup devices, network)

**Step 2: Installation**

Deploy the bus ticketing application on local terminals (teller/cashier, dispatcher, driver, and admin workstations)

**Step 3: Configuration**

Configure role-based access control, routes, buses, schedules, and fare tables

**Step 5: Rollout**

Gradual rollout by terminal or branch to minimize disruption.

Provide staff training for tellers, dispatchers, and drivers

**Step 6: Monitoring & Maintenance**

Enable centralized logging and monitoring.

Schedule periodic updates, backups, and system audits.

**Hardware And Software Requirements**

A. Server (for multi-terminal deployments)

Client Terminals (Cashier, Dispatcher, Driver, Admin)

Processor: Dual-core (Intel i3 or Pc)

Additional Hardware

Barcode/QR scanners for ticket validation

Receipt/ticket printer

**Guidelines For System Maintenance And Support**

**Regular Monitoring:**

Check system logs daily for errors, sync conflicts, or failed transactions.

Review server health (CPU, memory, storage, backups).

**Preventive Maintenance:**

Perform database optimization (index rebuilding, cleanup) monthly.

Verify integrity of backups and test restore procedures regularly.

Inspect hardware (printers, scanners, UPS) to ensure proper functionality.

**User Support:**

Provide helpdesk/ticketing system for operators (tellers, dispatchers, admins)

**Procedures For Handling Software Updates, Patches, And Bug Fixes**

**Update Management:**

Updates deployed in a staging environment first, tested before rollout.

Version-controlled releases with documented changelog

**Bug Fix Workflow:**

User reports issue logged in support system.

Developer replicates and fixes bug in development branch.

Fix tested in staging environment.

Approved fix deployed to production (all terminals)

**Escalation Process for Resolving Issues**

Frontline Support (Operators/Local IT Staff : Handles basic issues: password reset, printer problems, simple booking errors

**System Administrator / Internal IT Team**

Manages server/database issues, offline sync failures, and backup recovery

**Escalation Timeline:**

Critical Issues (system crash, data corruption)

**Revision History**

|  |  |
| --- | --- |
| **Dates** | **Description of revision** |
| 08-18-25 | Make an figure agile |
| 08-18-25 | Title of the System |
| 08-18-25 | The system make an offline |
| 08-18-25 | Make an times new roman font |
| 08-18-25 | Make an justify every sentence |
| 08-18-25 | Name in to an alphabetical |
| 08-18-25 | Make good title for system |

**Appendix**:

odd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377–387. https://doi.org/10.1145/362384.362685

Abouelmehdi, K., Beni-Hssane, A., Khaloufi, H., & Saadi, M. (2018). Big healthcare data: Preserving security and privacy. Journal of Big Data, 5(1), 1–18. https://doi.org/10.1186/s40537-017-0110-7

Carroll, J. M. (2000). Making use: Scenario-based design of human-computer interactions. MIT Press.

Singh, Y., & Sharma, A. (2017). Analysis of performance testing and its tools. International Journal of Computer Applications, 164(3), 1–5. https://doi.org/10.5120/ijca2017913662

Industry / Technical References

MySQL. (2023). Optimization and indexing. Oracle. https://dev.mysql.com/doc/refman/8.0/en/optimization.html

PostgreSQL Global Development Group. (2023). Backup and restore. PostgreSQL Documentation. https://www.postgresql.org/docs/current/backup.html