System Design Document

for

Car Rental System

Prepared by Group

Name: Car Rental System

Atluri Charitha Sri	**B220213CS**	**atluri_b220213cs@nitc.ac.in**
Bandapalli Venkata Durga Sahithi	**B220753CS**	**bandapalli_b220753cs@nitc.ac.in**
Bharath Nayak Bhukya	**B220224CS**	**bharath_b220224cs@nitc.ac.in**
Bhukya Sai Lokesh	**B220227CS**	**bhukya_b220227cs@nitc.ac.in**

CONTENTS

1 PURPOSE

The primary purpose of this System Design Document (SDD) is to provide a comprehensive description of the Car Rental System (CRS). This document explains the purpose,architecture, data models, and functionality of the CRS, serving as a crucial reference for the development team, system administrators, and all stakeholders involved in the project.
Additionally, it lays the foundation for effective project management, quality assurance, and system testing.

1.1 Document Objectives

1. **System Architecture:** This section offers a detailed explanation of the architectural components of the Car Rental System (CRS), covering hardware, software, and network elements essential for its operation. It describes the structure of the system, including server infrastructure, database management, and user interface components, highlighting how these elements interact to facilitate smooth and reliable car rental operations. The architecture is designed to support real-time vehicle availability checks, customer data handling, secure payment processing, and seamless booking and return processes, ensuring efficient and user-friendly service.

2. **Data Model:** This section describes the database schema, entity-relationship diagrams, and data structures used in the Car Rental System (CRS). It outlines the database's organization, detailing the relationships among various entities such as customers, vehicles, rental transactions, and payment records. The data model is designed to support efficient data retrieval and storage, ensuring accurate tracking of vehicle availability, customer details, rental agreements, and transaction history.

1.2 Target Audience

1. **Development Team**: This includes developers, database administrators, and IT professionals responsible for the design, development, and maintenance of the CRS. The document provides them with key insights into the system's architecture, data model, and functionality to support effective development and deployment.

2. **Car Rental Administrators**: Car rental managers, executives, and decision-makers within the organization. This document helps them understand the system's technical aspects and how it supports operational needs, including vehicle inventory management, rental transactions, and customer service.

3. **Quality Assurance and Testing Teams:** Teams responsible for testing the CRS to ensure it functions correctly, meets performance expectations, and adheres to security standards. The document serves as a foundation for developing test cases, ensuring data accuracy, and evaluating system performance.

4. **Car Rental Staff**: Employees interacting directly with the CRS, such as rental agents, customer service representatives, and support staff. This document offers them a better

understanding of the system's functionalities, helping them efficiently handle tasks related, and customer manager.

1.3 Definitions, Acronyms and Abbreviations

- CRS: Car Rental System
- SDD: System Design Document
- DBMS: Database Management System
- ERD: Entity-Relationship Diagram
- NITC: National Institute of Technology, Calicut
- 1NF: First Normal Form
- 2NF: Second Normal Form
- 3NF: Third Normal Form

1.4 References and Acknowledgments

- Fundamentals of Database Systems by Ramez Elmasri
- https://dbdiagram.io/
- https://erdplus.com/
- https://mermaid.live/

2 Assumptions and Constraints

2.1 Assumptions

The following are assumptions made while developing the project:

Data Migration: Assuming that the CRS has access to existing data related to vehicles, customer records, rental histories, and other relevant information, which will be migrated into the new database system without significant data quality issues.

Hardware and Software Availability: Assuming that the necessary hardware, servers, database management software, and development tools required for the CRS project are available and meet the project's technical requirements.

Data Cleansing and Transformation: Assuming that any data cleansing or transformation required during the migration process is within the project scope and manageable, ensuring that the data in the new system is reasonably accurate.

Data Security Measures: Assuming that appropriate data security measures, including access controls and encryption, are in place to protect the CRS database from unauthorized access and potential security breaches, especially for sensitive customer and information.

Scalability: Assuming that the CRS is designed to be scalable, capable of accommodating potential future growth in terms of vehicle inventory, customer accounts, rental transactions, and additional features to meet the evolving needs of the car rental business.

User Acceptance: Assuming that the CRS will be accepted and effectively utilized by car rental staff and management, with adequate training and a user-friendly interface to support their day-to-day operations.
.

Maintenance and Updates: Assuming that the project team will provide regular maintenance, updates, and enhancements to the CRS to ensure it continues to function optimally, remains secure, and meets changing requirements in the car rental industry.

2.2 Constraints

The design contains the following constraints:

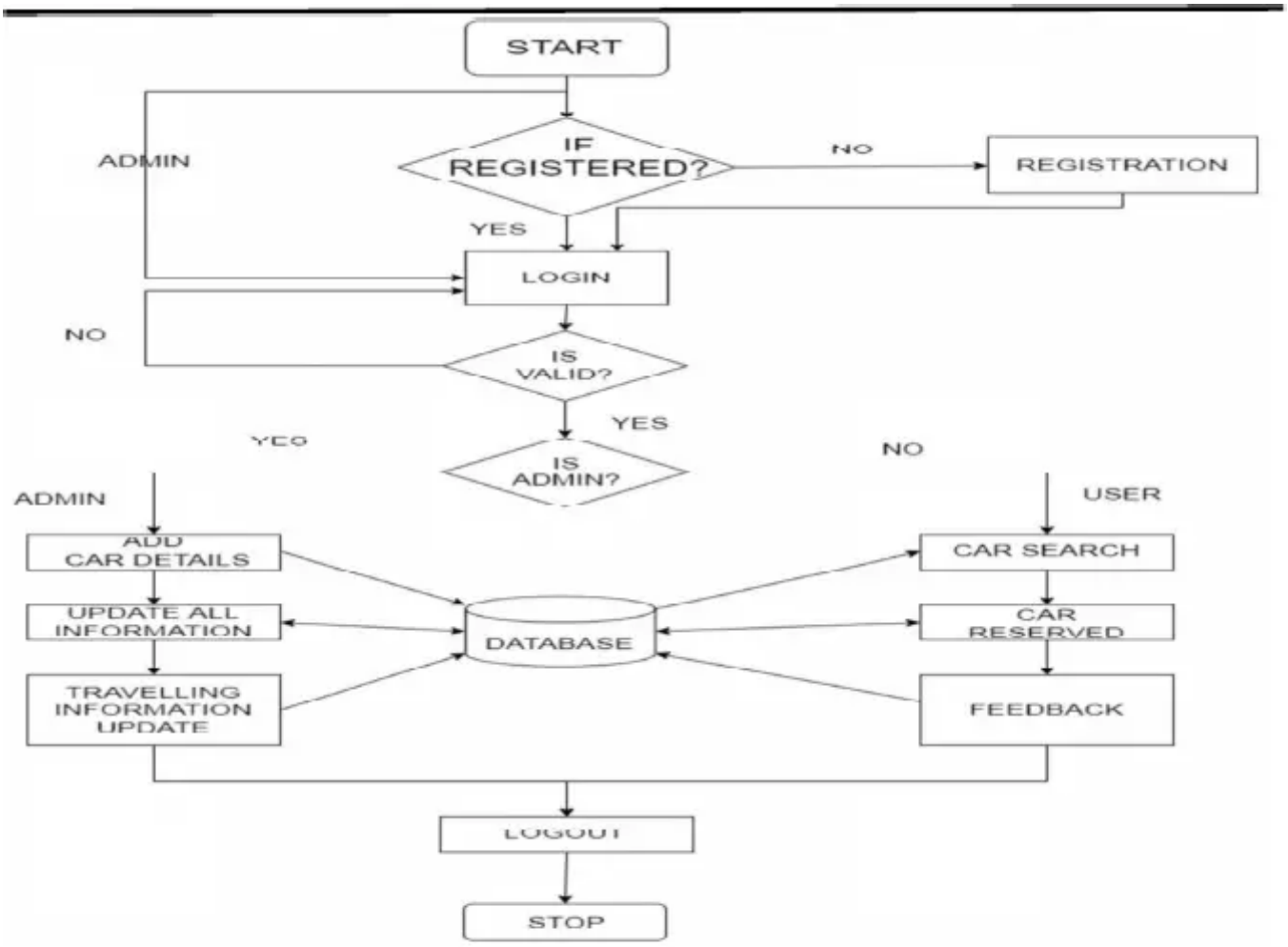
Development and Maintenance Constraints: The CRS software is exclusively designed, delivered, and maintained by the project team to ensure it meets the unique requirements of the car rental business.

Admin Access Constraints: Admin access to the system is protected with unique, secure credentials to maintain the security and integrity of administrative functions, such as vehicle inventory management, customer data access, and financial transactions.

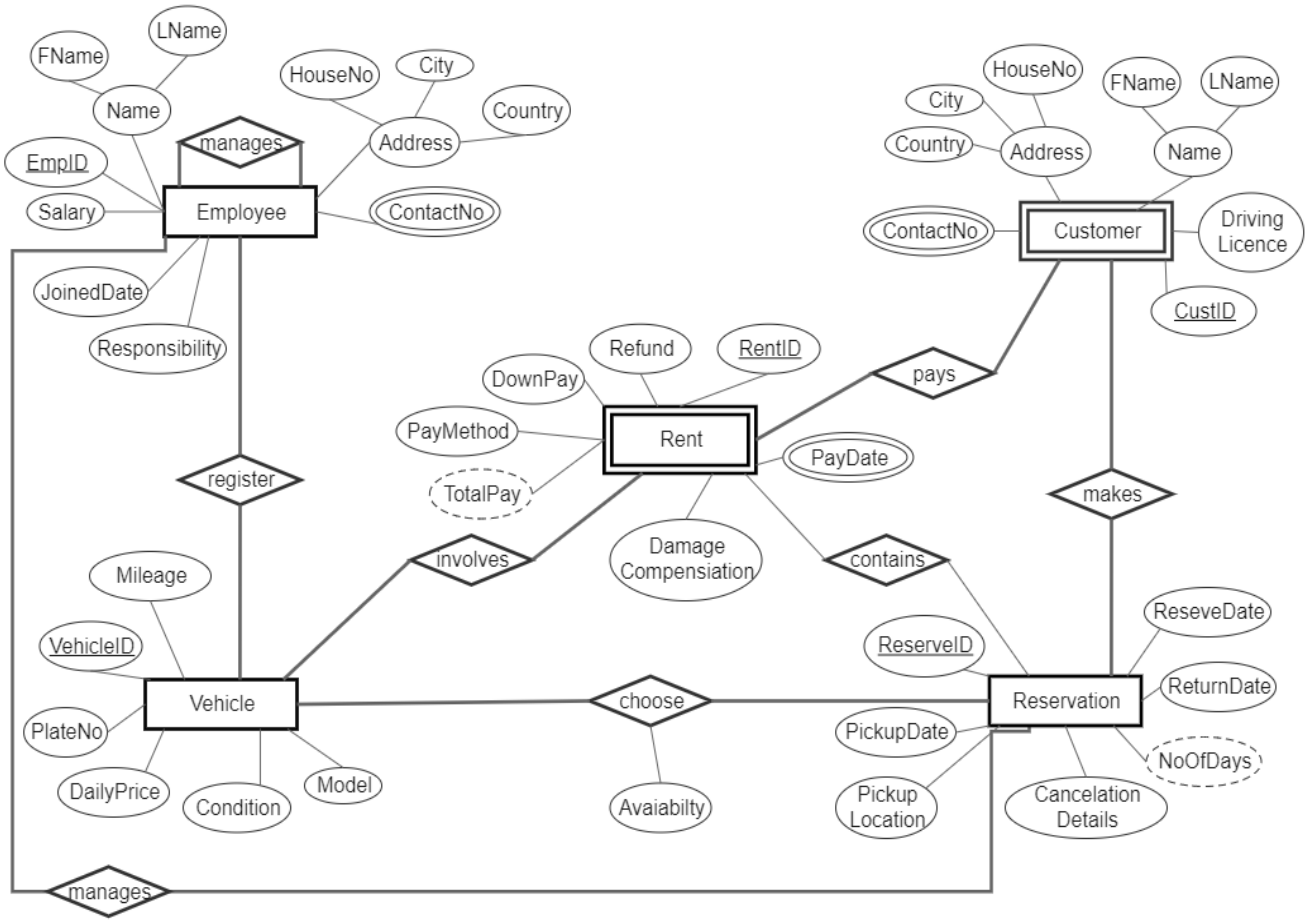
User Role-Based Dashboard Constraints: Each employee or user has unique credentials and is provided with a personalized dashboard that aligns with their specific role within the system, such as admin, rental agent, or customer service representative, ensuring they have access only to the features necessary for their responsibilities.

3 Database Functions

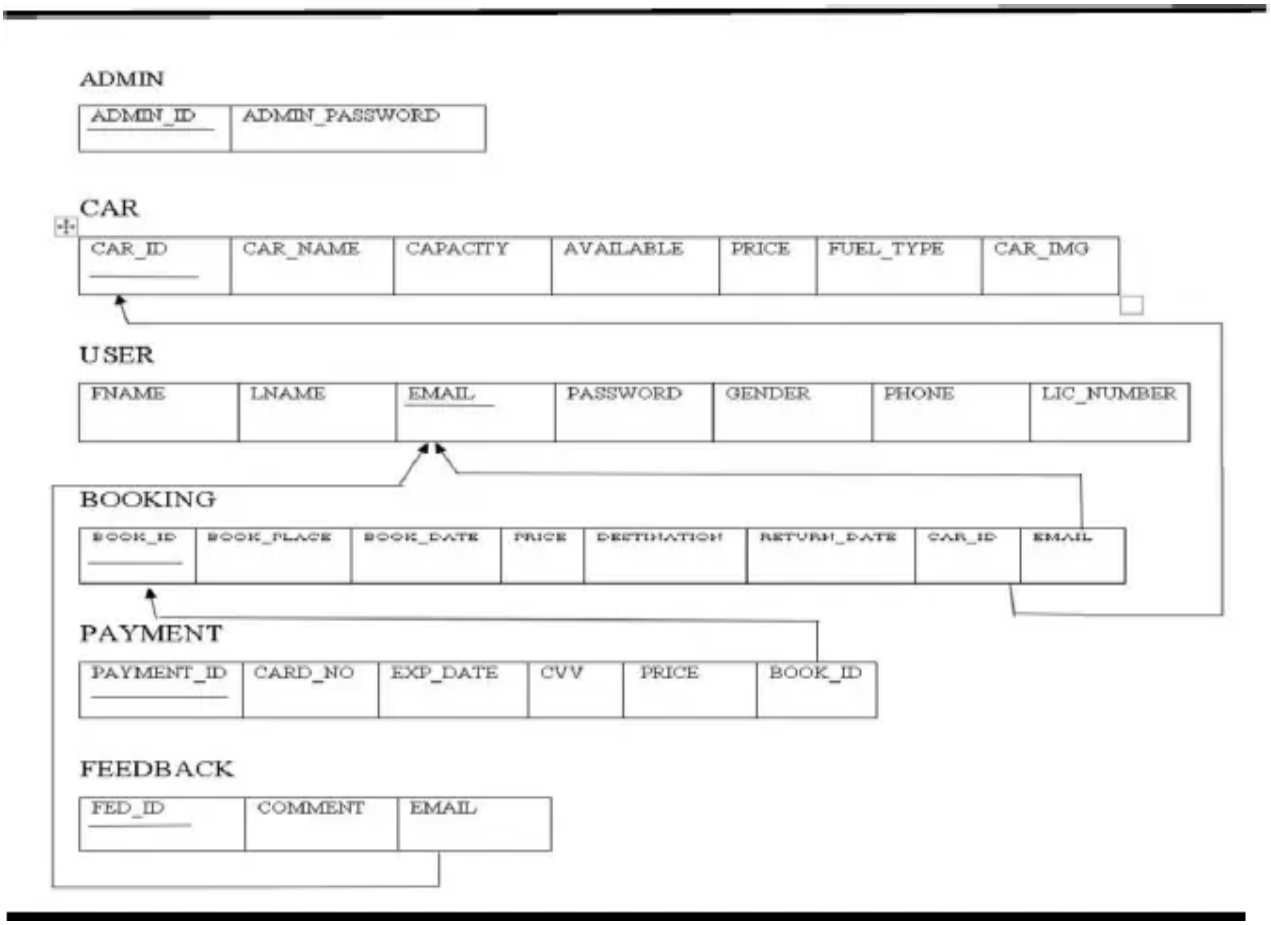
3.1 Control Flow



3.2 Entity-Relation Model



3.3 Relational Schema



3.1 Normalization

- 1NF - The tables are in 1NF, as there are no multivalued or composite attributes. Each table cell contains atomic values, and each record is unique. Hence the database is 1NF normalized.

- 2NF - The tables are already in 1NF as proved above. There are no partial dependencies, that is, there are no non-prime keys solely dependent on only one part of a primary key in any of the tables. Hence the database is 2NF normalized.

- 3NF - The tables are already in 2NF as proved above. There are no transitive functional dependencies in the schema. There are no non-prime keys that are dependent on another non-prime key in any specific table. Hence the database is 3NF normalized.

3.1 Schema Description

1.ADMIN: it stores the details of admin**.**

- **ADMIN_ID**: User name of the admin.
- **ADMIN_PASSWORD**: password of the admin

Field name	Data type	Constraints	Description
Admin_ID	Varchar(45)	Not Null	Id of role
Admin_Password	Varchar(45)	Not null	Password of Admin

2. CAR: It gives the details about the car

- **CAR_ID:** Id given to car done by auto increment.
- **CAR NAME**: Name of the car.
- **CAPACITY:** Seat capacity.
- **AVAILABLE:** Availability of car
- **PRICE:** Price of the car.
- **FUEL_TYPE**: Car fuel type.
- **CAR_IMG:** Image of car

Field name	Data type	Constraints	Description
Car_ID	Integer	Not null	Id of car
Car_Name	varchar(45)	Not null	Name of Car
Capacity	Integer	Not null	Capacity of Car
Available	varchar(45)	Not null	Availability
Price	int	Not null	Rental Price
Fuel_Type	varchar(45)	Not null	Type of Fuel
Car_Image	varchar(45)	Not null	Visual Image of Car

3. **USER:** It stores the details of user**.**
    - **FNAME:** first name of user.
    - **LNAME:** last name of user.
    - **PASSWORD:** password of user.
    - **GENDER:** gender of user.
    - **PHONE:** phone number of user.
    - **LIC_NUMBER:** license no of user.

Field name	Data type	Constraints	Description
FNAME	varchar(45)	Not null	First Name
LNAME	varchar(45)	Not null	Last Name
EMAIL	varchar(45)	Not null	Email
PASSWORD	varchar(45)	Not null	Password
GENDER	varchar(10)	Not null	Gender
PHONE_NUMBER	integer	Not null	Phone Number
LICENCE NUMBER	varchar(45)	Not null	Liscence Number

4. **BOOKING:** it gives the booking details of user.
    - **BOOK_ID:** Booking id done by auto increment.
    - **BOOK_PLACE:** place of booking.
    - **BOOK_DATE:** date of booking.
    - **PRICE:** price of the car.
    - **DESTINATION:** destination.
    - **RETURN_DATE**: return date.
    - **CAR_ID:** id given to car and foreign key associated with booking.
    - **EMAIL:** email of user and foreign key associated with booking.

Field name	Data type	Constraints	Description
BOOK_ID	Int(11)	Primary Key	Book ID
BOOK_PLACE:	varchar(45)	Not null	Return place
BOOK_DATE	date	Not null	Return date
PRICE	Int(11)	Not null	Cost
DESTINATION	varchar(45)	Not null	Destination
CAR_ID	Int(11)	Foreign key	Car ID
EMAIL	varchar(45)	Not null	Mail ID
RETURN_DATE	Date	Not null	Return date

5. **PAYMENT:** It provides payment option for users.
    - **PAYMENT_ID:** id given to payment and done by auto increment.
    - **CARD_NO:** card number.
    - **EXP_DATE:** expiry date of card.
    - **CVV:** CVV of card.
    - **PRICE:** price of car.
    - **BOOK_ID:** id given to booking and foreign key associated with booking.

Field name	Data type	Constraints	Description
PAYMENT_ID	Int(11)	Primary Key	Payment ID
CARD_NO	varchar(45)	Not null	Card No
EXP_DATE	varchar(45)	Not null	Exp date
CVV	Int(11)	Not null	CVV
PRICE	Int(11)	Not null	Cost
BOOK_ID	Int(11)	Foreign Key	Book ID

6. **FEEDBACK**: it provides user to give their feedback.
    - **FEED_ID**: id given to the feedback done by auto increment.
    - **COMMENT**: Message about their experience.
    - **EMAIL: email of user and foreign key of user associated with feedback.**

Field name	Data type	Constraints	Description
FEED_ID	Int(11)	Primary Key	Feedback
COMMENT	Text	Not null	Comment
EMAIL	varchar(45)	Not null	Email