BUS MANAGEMENT SYSTEM

Sujal Gauchan, Miraj Gansi, Yug Pokharel, Helan George Adhikari.

B.S.C (Hons) Computer Softwarica Collage of

IT and Ecommerce, Coventry

UniversitySt4008CEM: Computing Activity

Led Learning project 1

Ayush Kaji Dangol

Feb 1, 2024

Table Content

Bus management System……………………....1

Introduction……………………………………4

Aim……………………………………………5

Objectives……………………………………..6

Problem Statement………………………….....7

Features………………………………………10

Functional Requirements…………………….11

Non-functional requirement………………….12

Scope…………………………………………13

Development Methodology..............................15

Methodology…………………………………16

7

Tools and Technologies……………………...17

Conceptual Diagram…………………………19

System Architecture………………………….20

Project Plan…………………………………..22

Prototypes……………………………………23

Developed System…………………………...25

BUS MANAGEMENT SYSTEM

Introduction

In the dynamic and fast-paced world of transportation, efficient management of bus service is crucial to ensure the smooth function of public or private transporting system. The bus service, providing an organized and user-friendly platform for managing various aspects of the bus fleet.

AIM

Our application aim is to enhance the efficiency,

accuracy, and overall effectiveness of bus

operations, benefiting both the service provider

and the passengers.

OBJECTIVES

* Efficient Bus Operations
* Passenger Information Management
* Resource Optimization
* Data Accuracy and Integrity
* Improved customer Experience
* Administrative Oversight
* Bus and Route Management
* Scheduling and Timetable
* Passenger Reservation System

PROBLEM STATEMENT

The Bus Management System project aim to address significant challenges in the current state of bus transportation operation.

Presently, the reliance on manual processes for managing bus schedules, routes, and passenger information results in inefficiencies and inaccuracies. The lack of automation leads to difficulties in real-time updates and tracking, impacting the reliability and punctuality of bus services.

Additionally, the absence of centralized system contributes to resource underutilization, affecting the optimization of buses, fuel, and maintenance. Passengers face inconvenience due to limited access to timely information, online booking options, and communication channels. The overall lack of comprehensive oversight tools for administrators further complicates decision-making and strategic planning. Consequently, the bus management system project endeavors to mitigate these challenges by introducing automation, centralized data management, and enhanced user interface to streamline operations, improve resource allocation, and enhance the overall efficiency and user experience od bus transportation services.

Features

* Route management
* Ticketing and Fare management
* Passenger Information System
* Safety and Security
* Mobile Applications
* Environmental Sustainability
* Integration with other System
* Reporting and Analysis
* Real-time Communication

Functional Requirements

Functional Requirements means what the system should do, it include the features and functions:

* User Authentication and Authorization:
* Route Management:
* Bus Scheduling:
* Ticket Booking and Reservation:
* Passenger Management:
* Bus Tracking and Management:
* Diver and Conductor Management:
* Payment and Fare Management:
* Integration with External System:

# Non-functional requirements

Non-functional Requirements means how the system performs a certain function area follows:

* Usability: Easy to use and eye-catching UI.
* Maintainability: Easy to maintain the system.
* Changeability: Easy to modification and delete of the data.
* Capacity: It holds a lot of the data at a time approximately 300.
* Scalability: Not too many issues in upgrading the system.
* Portability: Only valid in the Operating system.
* Recoverability: system should handle the failure data without any loss of data.

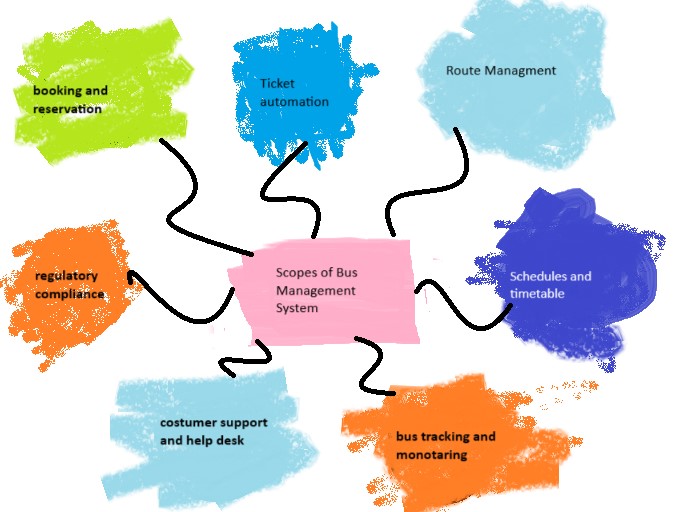
Scope

The scope of a Bus Management System encompasses the various features, functionalities, and capabilities that the system is designed to provide. The scope defines the boundaries and objective of the system. Here is an overview of the scope of Bus Management System:

* Route Management
* Scheduling and Timetables
* Booking and Reservation
* Bus Tracking and Monitoring
* Regulatory Compliance
* Customer Support and Help Desk

Figure 1

Scopes of System



Development Methodology

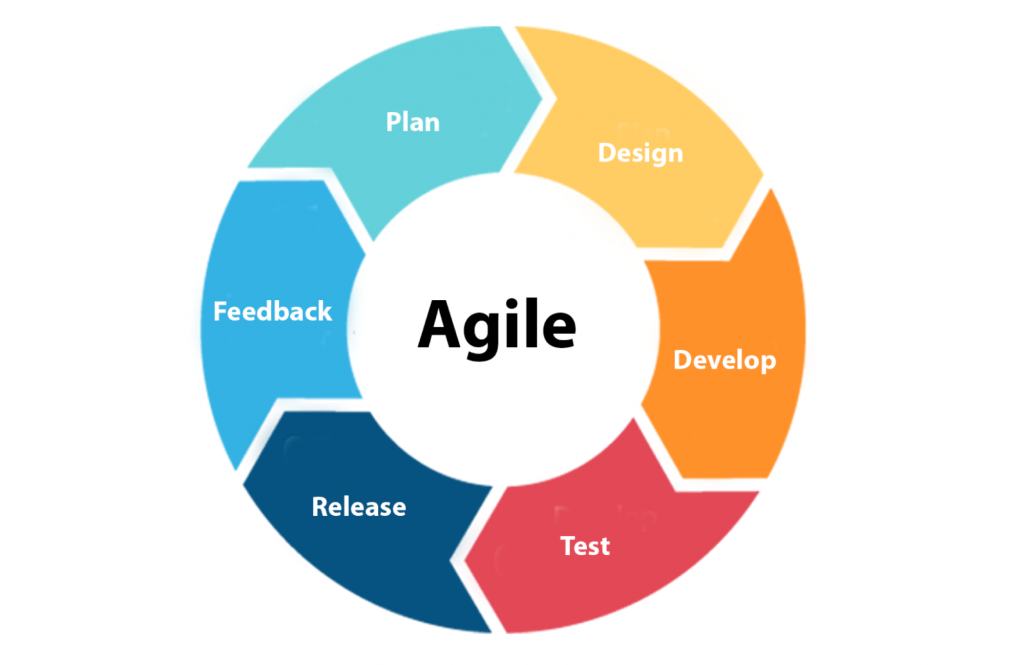
Development of software requires a Software Development life cycle (SDLC) which gives the full information about the development of software step by step. It is crucial software effective and efficient.

**Methodology**

IN projects, we employ the rapid development process, which produces flexible software quickly. This is the most recent methodology that has gained popularity recently. Initially, the team members create a strategy for finishing the project. The system's front endis designed during the subsequent cycle generally employing a figma, and Visual Paradigm interact with users. Developed the strategy and

regularly assessed how it performed in real-world setting. Following the discovery of systemic fault. We fix the system's mistakes.

Figure:2

*Agile Methodology*

Tools and Technologies

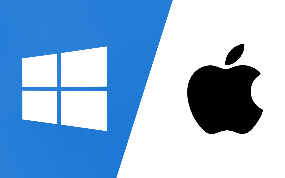
There are several tools and technologies used to build this software. They are given below.

* **Windows and MacBook** as working platforms.
* **Adobe pdf Reader** for feasibility study and requirement analysis.
* **Figma** for the design phase.
* **Python** with tkinter library and **Visual Studio Code** for coding phase.
* **MySQL** used for database in Development phase.
* **Visual paradigm** for conceptional diagram.
* **Git and GitHub** uses for version control.
* **Discord, Messenger** **and** **Instagram** used for feedback and s=discussions.
* **Google** used for search.
* **Word** used for creating a project plan.

Figure 3

*Tools and Technology used*

**Platform Design Project Plan**



**Feasibility study**

**Requirement analysis version control Coding**





**Commination Conceptual Diagram Research**





**Database coding**

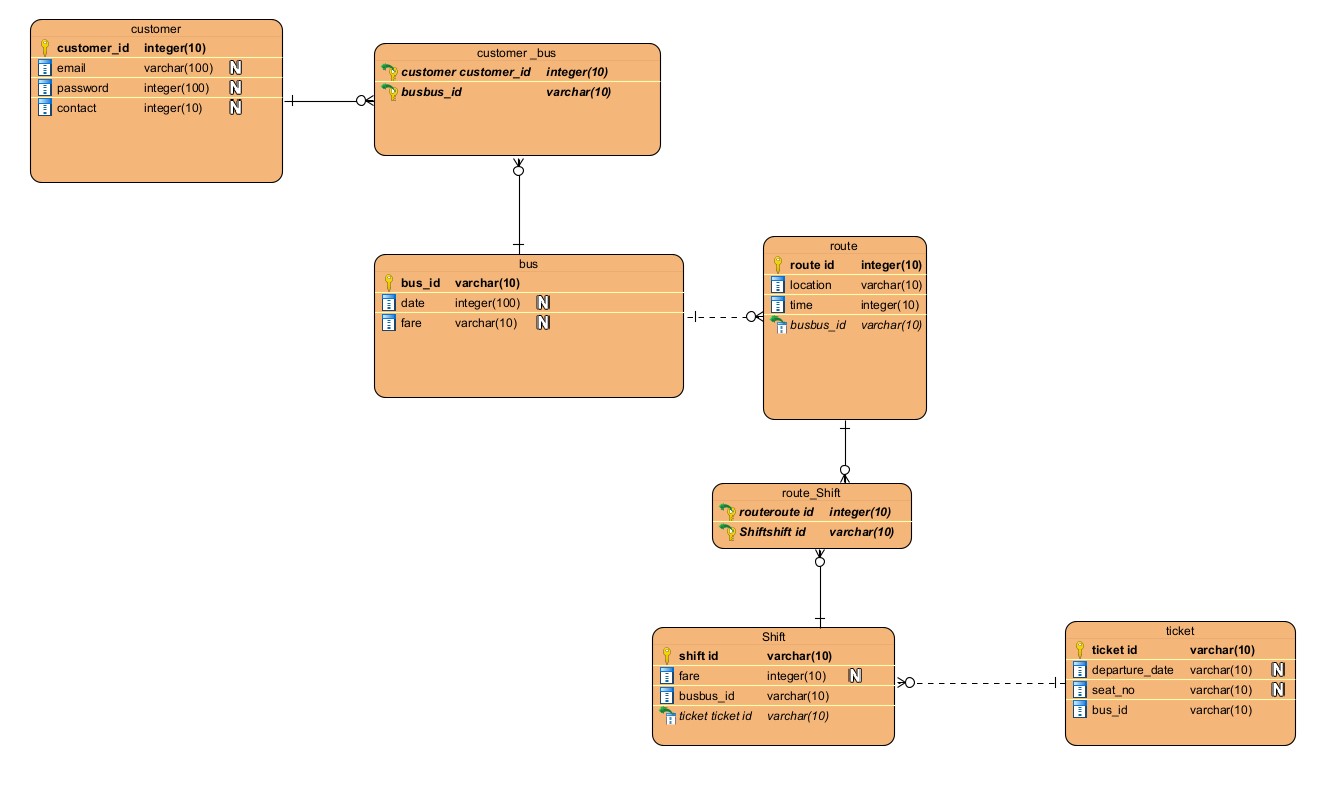


Conceptual Diagram

AN Entity-relationship diagram shows the relationship between entities in the system.

Figure:4

*ER Diagram*

**

System Architecture

The system architecture of bus management System comprises interconnected components designed to streamline and optimize bus operations. At its core, the architecture includes a user interface (UI) with distinct interfaces for passengers, divers, and administrators.

The passenger interface enables travelers to

access bus schedules, make reservations, and track busses, while the Drivers interface assists bus operators in managing routes and receiving updates. Administrators interact with a comprehensive Administrator interface to oversee the system's overall functionality, encompassing bus routes, schedules, and user accounts. The system relies on robust database, housing information such as bus details, passengers' records. A booking and Reservation System facilitates ticket transactions securely, incorporating a Payment Gateway for online purchases. Feedback and Reporting mechanisms gather user input and provides administrators with valuable insight. Authentication and security layers ensure secure access, and a notification system keeps users informed about reservations and bus updates. Additionally, the architecture involve integration with payment system.

This comprehensive architecture aims to create a seamless and efficient Bus Management System, emphasizing scalability, reliability, and security in its design and implementation.

Project Plan

A project plan is necessary to track our

efforts, progress, and work remaining and

give all information about the project. Due

to this chart, information the pace of the

project and how many days requires to fulfil

complete the work.

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Start Date | End Date | Duration |
| Feasibility Study |  |  |  |
| Requirement Analysis And Specification |  |  |  |
| Design/prototype |  |  |  |
| Coding And Unit Testing |  |  |  |
| Integration And System Testing |  |  |  |

Prototypes

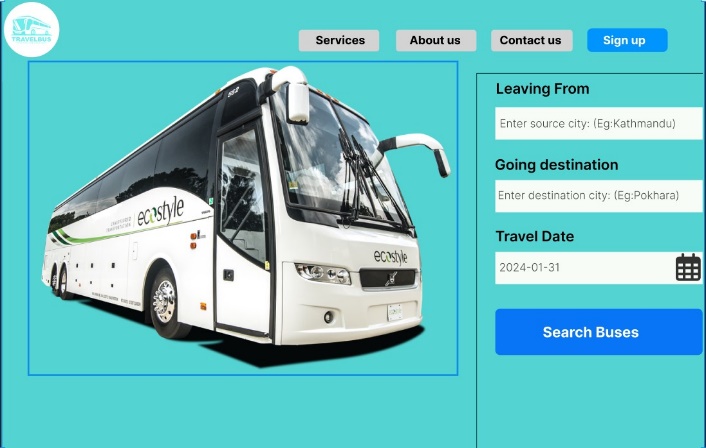
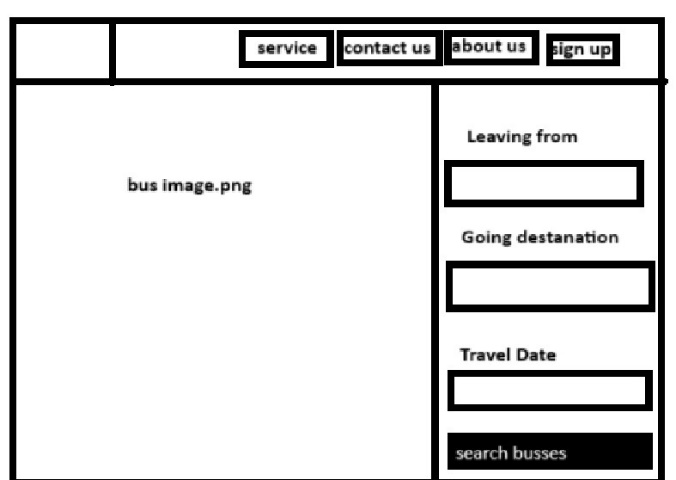
Prototypes are the preliminary version of the product. Throughout the early phase if growth and development, prototypes are primitive sketches or sketches from an architecture, good, any computer program that are used to show case and evaluate the features, functionality and design.

These models, which can be interactive, functional, or give stakeholders, developers, and consumers a concert grasp development, prototyping is a useful technique for modifying requirements, obtaining feedback, and spotting possible problems. It helps teams interact and make well-informed decisions by helping them understand design ideas, graphical interfaces, and user interactions.

Depending on the requirements and complexity of the project, prototypes can take a variety of shapes, ranging from low-fidelity drawings to high-fidelity interactive simulations.

Figure:5

*High fidelity and low fidelity design*

**

Development System

The Bus Management System display a user fronted for search buses and was created employing Python's Tkinter, who framework and MySQL.

It shows ticket booking that provide information about available sites. The graphical interface components are built by Tinker, while database handling is made easier with MySQL.

TO address needs, such as error handling, validations, and a larger features set for an effective parking management platform, additional adjustments and adaptation are advised.

Login UI

The Login User Interface of the Bus Ticket Management System offers a visually appealing and intuitive way for users to authenticate themselves

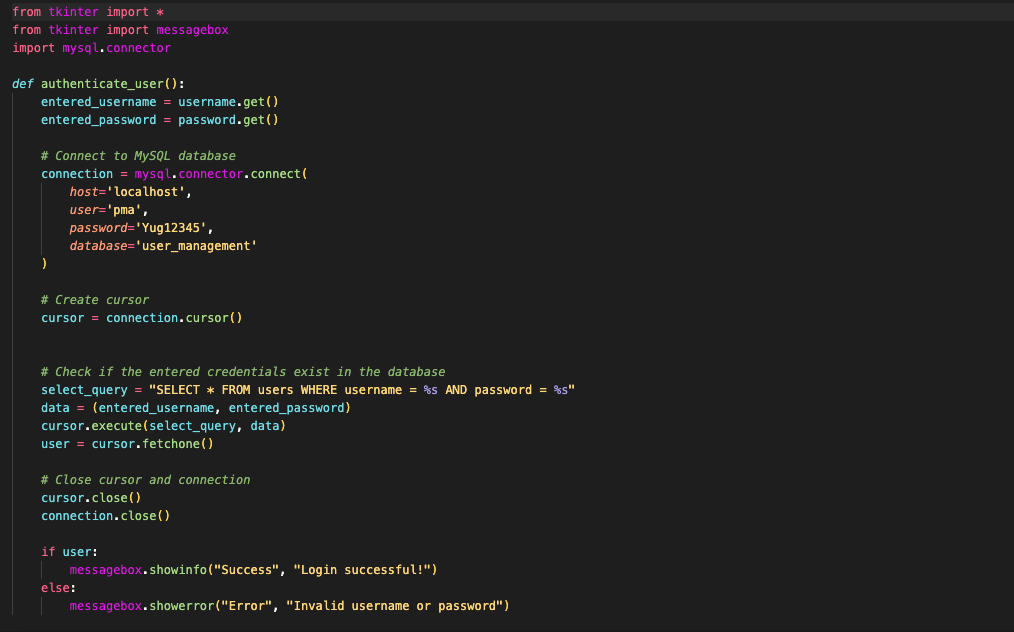
It was created in Python with Tkinter and

has an eye-catching design with password and email/phone input fields. Strong security and data management are ensured by the MySQL database supporting this login system. Only logged-in users are permitted access to the system since the user's credentials are verified against the database. User convenience and security are improved with the addition of features like password masking and a "Forgot Password" option.

Additionally, the system promotes new

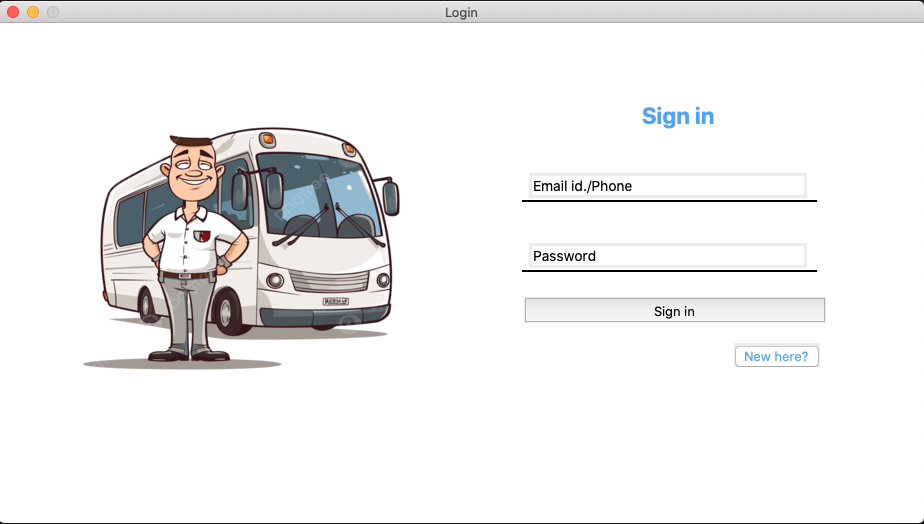
registrations by way of a "Sign Up" button. This all-encompassing strategy helps to make the Bus Ticket Management System effective, safe, and user-focused.

Login Page and Code









Costumer Details UI

A bus management system customer details UI service as the user interface for passengers to input their information when booking bus ticket.

This interface typically includes fields for essential customer details, such as full name, contact number, email address, address, and date of birth. Additionally, the UI incorporates section for booking specifics, like trip date, departure and arrival location, seat preferences, and bus class or type. To facilitate secure transaction, the UI includes a payment information section, allowing customer to input details like payment method, cardholder name, card number, expire date, and cvv/cvc.

The user-friendly design ensures a seamless and efficient booking process for customers, enhancing their overall experiences with the bus management system.

*System UI and code*

CODE AND SS

Update/Delete UI

In the lasts iteration of the bus management system UI, significant updates and refinements have been introduced to enhance user experience and system efficiency.

The interface has undergone a modernized redesign, featuring a clean and contemporary aesthetic, responsive layout, and improved navigation for seamless interaction across devices. The booking details section now empowering users to visually select preferred seats. Redundant fields in the costumer details from have been eliminated to streamline the input process, and outdated features and technologies have been removed to maintain system relevance and security. The payment section has been refined, eliminating obsolete methods and simplifying steps for more straightforward transaction.

This updates collectively aim create a more user-centric, intuitive, and efficient bus management system UI. User testing and feedback have been integral to these enhancements, ensuring alignment with user expectations and preferences.

*System UI (II) and code*

*Code and ss*

**System Testing**

Black box testing is a type of software testing that focuses on evaluating an application's functionality without taking into account the implementation details or underlying code structure.

Using this approach, testers view the software as a "black box," interacting with the input and observing the result to see how the system responds.Making sure the software meets all requirements and functions correctly from the user's point of view is the main goal.Testers create test cases based on inputs, expected outcomes, and system requirements without having any prior knowledge of how the product works inside.

This method helps identify faults, highlight issues that impact the user experience, and confirm that the application matches user expectations and anticipated functionality.

Test 1

Test 2

Test 3

Test 4

Test 5

Test 6

Version control

Youtube :

GitHub :

**GitHub**

github photo;

**Conclusion**

The team made software that solves the problems of parking by using the latest tools and technologies gaining knowledge from the classroom in the topic of mathematics, Programming and Algorithm, and Software Design. Try to use the full concepts and ideas about the programming and effective tools.

The team project solves the old base pen-paper-based parking management system which is mainly focused on a person's name and contact details. But in this software, we introduced a lot of tools for making software. The system has passed both functional and non-functional requirements. All the errors are correct from the beginning of the project. All testing has successfully passed this system. Data are secure for the customer that is not violent by others. The GUI is also a main part of the system that is so much attractive, effective, and impactful to the customers.

The team make software for the first time and gained a lot of experience from it, making software from scratch is challenging but a chance to learn new things also. All the efforts will lead to a sweet result in the end, good coordination is to solve the projects easily. A tutor is supportive and friendly to students throughout the process. Expanding knowledge about making software and handling challenges smartly is important in the upcoming days for one good IT Cain.

**Reference**