Dutchess County Bus Transportation System

PROJECT PHASE 08 [THE FOUR-CE]

Sindhuja Ravikanth
Shanmukha Chowdary Nalla
Gaurav Bapurao Sherla
Katipally Chanakya Vardhan Reddy

TEAM MEMBERS INTRODUCTION

- ➢Sindhuja Ravikanth
- ➢Shanmukha Chowdary Nalla
- ➢Gaurav Bapurao Sherla
- ➢Katipally Chanakya Vardhan Reddy

OUTLINE

- System Description
- Entity-relationship model
  - External Models
  - Conceptual Model
- Enhanced Entity Relationship Diagram
- SQL database development
- Loading data and performance enhancements
- Graphical User Interface

SYSTEM DESCRIPTION

- **User Entity**: The User entity represents the passengers utilizing the DCBTS. Contains UserID, first name, last name, date of birth (DOB), email, phone number, username, and password.
- **Admin Entity**: The Admin entity manages system changes and oversees the functioning of the DCBTS. Contains AdminID, username, password, email, phone number, and EmployeeID.
- **Bus Entity**: The Bus entity describes the properties of each bus in the DCBTS fleet. Contains BusID, BusNumber, Capacity, LicensePlateNumber, and BusType.
- **Ticket Type Entity**: The Ticket Type entity defines the various ticket categories available to users. Contains TicketTypeID, TypeName, Price, ValidityPeriod, and description of applicable age groups.
- **Payment Entity**: The Payment entity records all transactions related to ticket purchases. Contains PaymentID, Amount, PaymentDate, UserID (foreign key), and TicketTypeID (foreign key).
- **Reservation Entity**: The Reservation entity manages seating reservations on buses. Contains ReservationID, PaymentID (foreign key), BusID (foreign key), ReservationDate and NumberOfReservations.
- **Bus Route Entity**: The Bus Route entity defines the routes within the DCBTS, identified by RouteID. Contains StartLocation, EndLocation, and the BusID (foreign key) associated with each route.

SYSTEM DESCRIPTION

- **Route Stop Sequence Entity**: The Route Stop Sequence entity details the stops along a route. Contains StopSequenceID and is associated with a RouteID (foreign key), StopID, NumberofStops, and ArrivalTime.
- **Employee Entity**: The Employee entity represents individuals employed in the DCBTS. Contains EmployeeID, DepartmentID (foreign key), FirstName, LastName, and Position.
- **Department Entity:** The Department entity outlines the organizational structure of the DCBTS. Contains DepartmentID, DepartmentName, and Location.
- **Schedule Entity**: The Schedule entity manages bus schedules, uniquely identified by ScheduleID. Contains BusID (foreign key), RouteID (foreign key), DayFlag, DepartureTime, and ArrivalTime.
- **Bus Stop Entity**: The Bus Stop entity describes all bus stops within the DCBTS. Contains StopID, StopName, and Location.
- **Notification Entity**: The Notification entity communicates important messages to users or administrators. Contains NotificationID, HolidayID (foreign key) or ReservationID (foreign key), Message, DateAndTime, UserID (foreign key), and AdminID (foreign key).
- **Holidays Entity**: The Holidays entity identifies public holidays during which bus services may be affected. Contains HolidayID, HolidayDate, and Description, NotificationID (foreign key).

ENTITY RELATIONSHIP MODEL

- **Entities and Attributes**: User, Admin, Bus, Ticket Type, Payment, Reservation, Bus Route, Route Stop Sequence, Employee, Department, Schedule, Bus Stop, Notification, Holidays
- **Multivalued Attributes**: Ticket Type: Price, ValidityPeriod, Route Stop Sequence: NumberofStops, Reservation: NumberOfReservation
- **Composite Attributes**: User and Employee: FirstName and LastName
- **Derived Attributes**: Payment: Amount (derived from Ticket Type Price)
- **Weak Entity Notification**: Depends on Admin or User
- **Strong Entities**: Bus, Bus Route, Ticket Type, User, Bus Stop, Schedule, Holidays, Route Stop Sequence, Payment, Reservation, Employee, Department
- **Participations**:

- **Total Participation**

  - Each payment associated with a user.

  - Each department is associated with an employee.

  - Each route stop sequence is associated with a bus stop.

  - Each bus is associated with many routes and schedules, and vice versa.

- **Partial Participation**

  - Users may or may not have associated records in other tables.

  - Not all administrators need associated employee records.

  - Not all holidays may have associated notification records.

  - Bus may or may not be associated with reservations.

  - Not all notifications need to be associated with User and Admin.

  - Not all reservations may be associated with notification records.

EXTERNAL MODELS

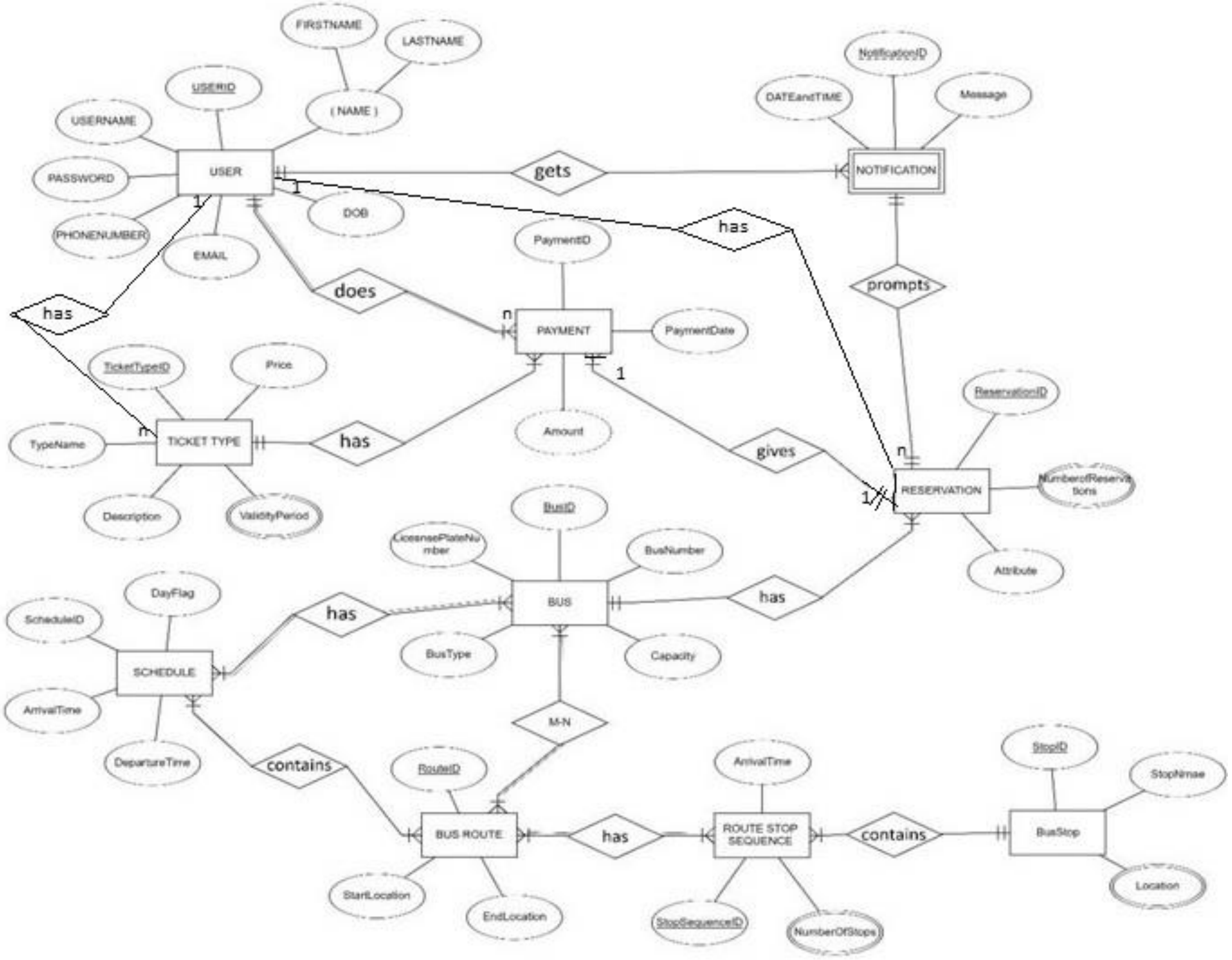


Fig 1: External ER USER POV

EXTERNAL MODELS

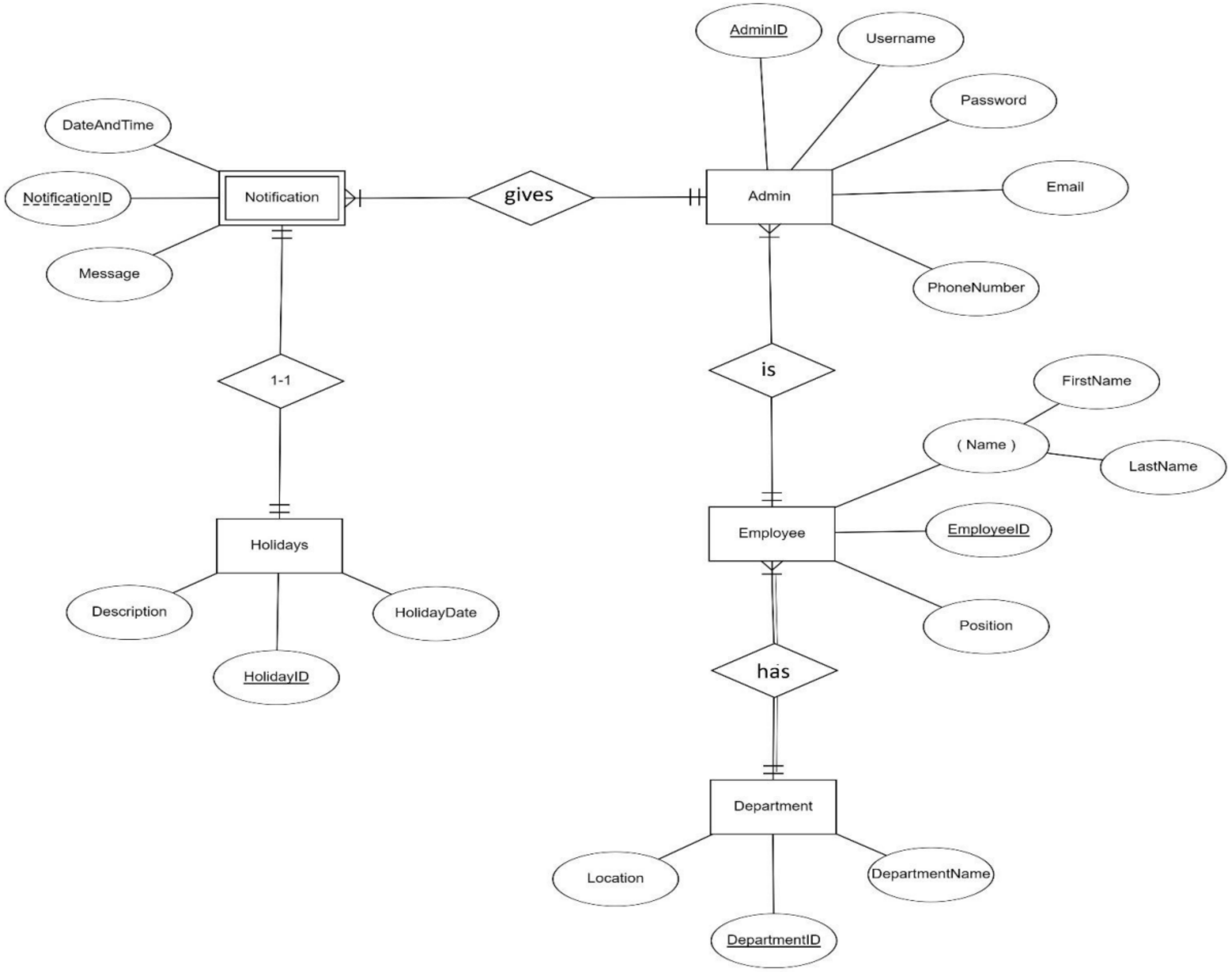


Fig 2: External ER ADMIN POV

CONCEPTUAL MODEL

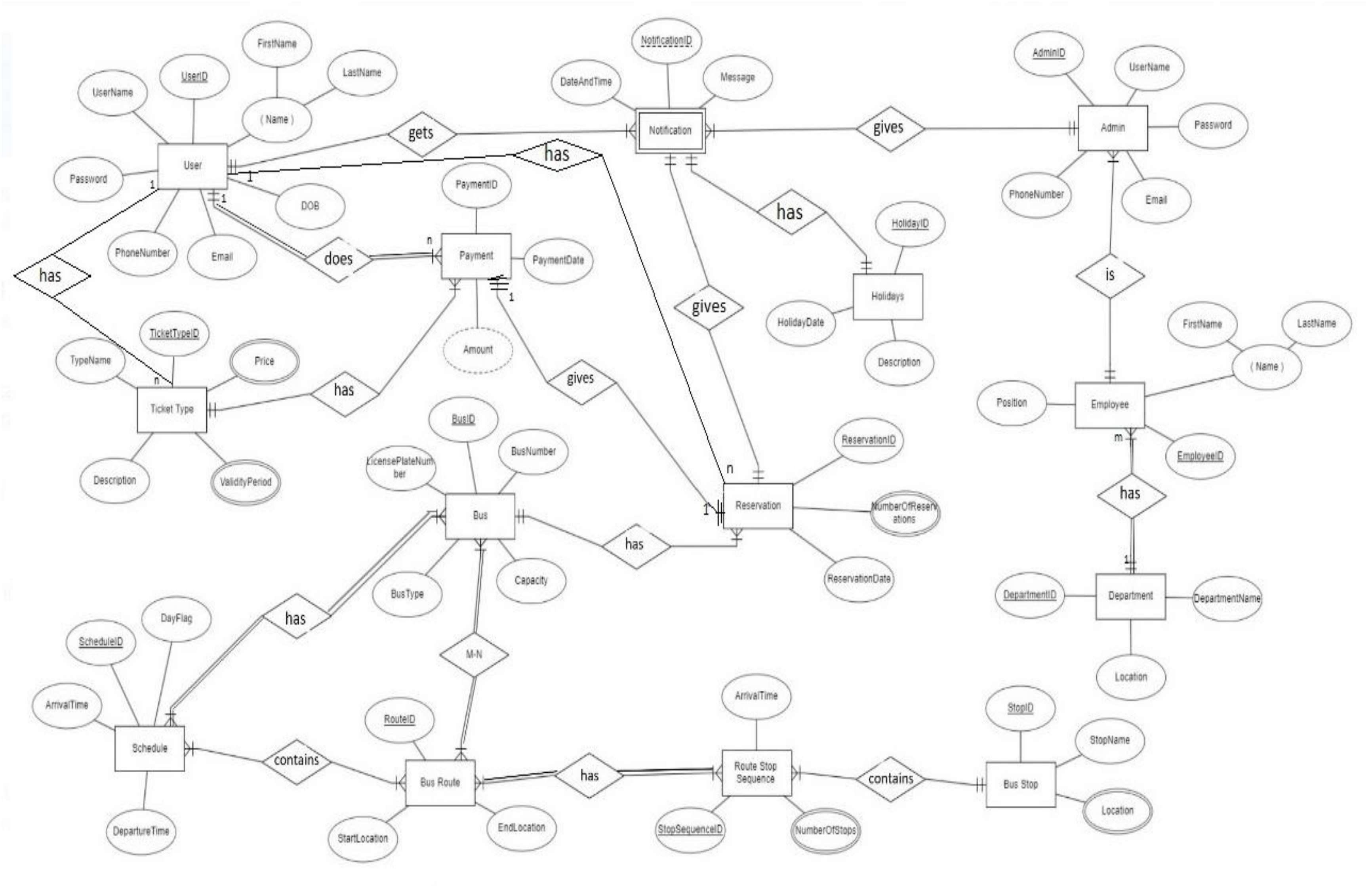
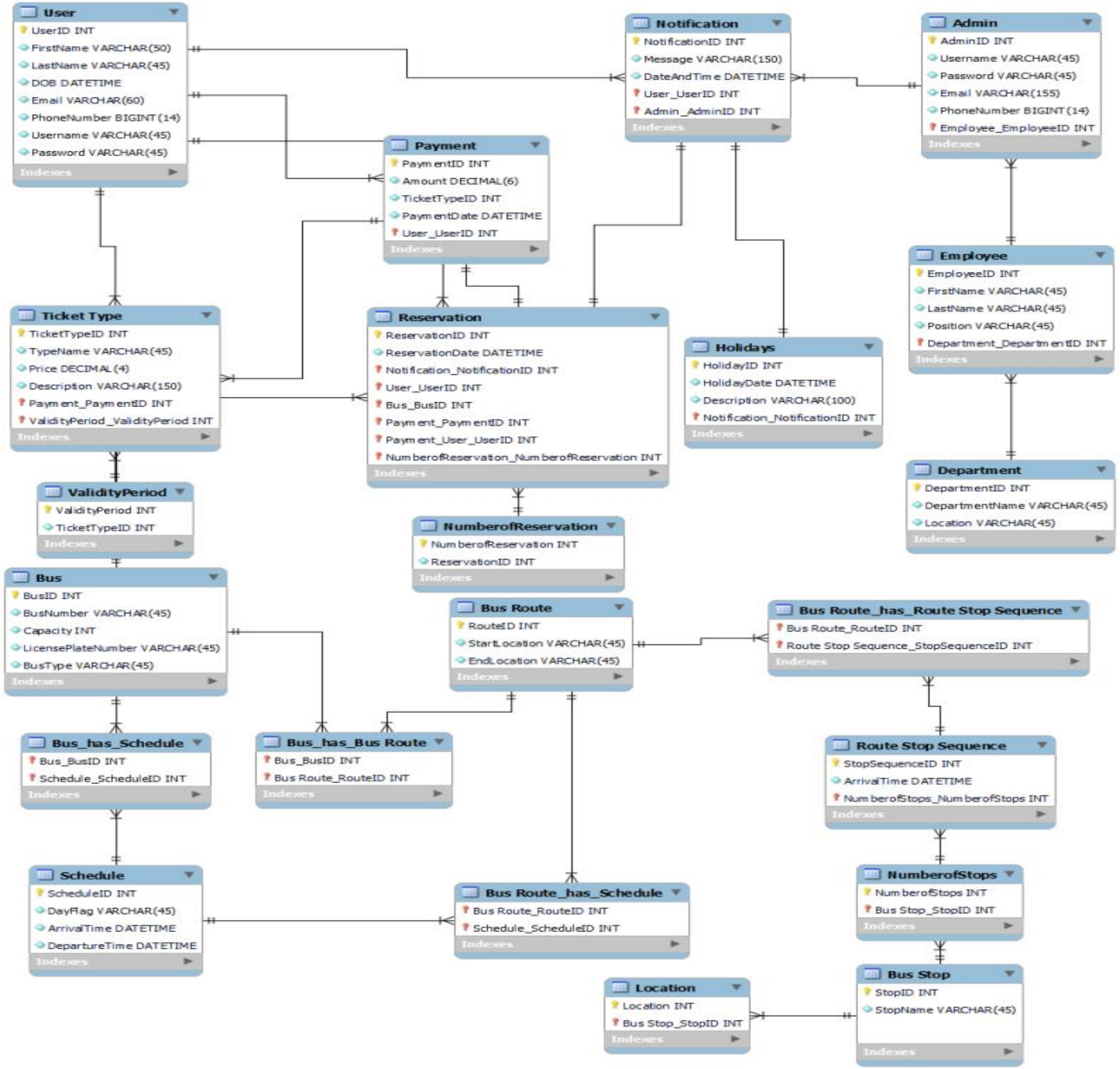


Fig 3: ER Conceptual Diagram

ENHANCED ENTITY RELATIONSHIP DIAGRAM



User
- UserID INT
- FirstName VARCHAR(50)
- LastName VARCHAR(45)
- DOB DATETIME
- Email VARCHAR(60)
- PhoneNumber BIGINT(14)
- Username VARCHAR(45)
- Password VARCHAR(45)

Notification
- NotificationID INT
- Message VARCHAR(150)
- DateAndTime DATETIME
- User_UserID INT
- Admin_AdminID INT

Admin
- AdminID INT
- Username VARCHAR(45)
- Password VARCHAR(45)
- Email VARCHAR(155)
- PhoneNumber BIGINT(14)
- Employee_EmployeeID INT

Payment
- PaymentID INT
- Amount DECIMAL(6)
- TicketTypeID INT
- PaymentDate DATETIME
- User_UserID INT

Employee
- EmployeeID INT
- FirstName VARCHAR(45)
- LastName VARCHAR(45)
- Position VARCHAR(45)
- Department_DepartmentID INT

Ticket Type
- TicketTypeID INT
- TypeName VARCHAR(45)
- Price DECIMAL(4)
- Description VARCHAR(150)
- Payment_PaymentID INT
- ValidityPeriod_ValidityPeriod INT

Reservation
- ReservationID INT
- ReservationDate DATETIME
- Notification_NotificationID INT
- User_UserID INT
- Bus_BusID INT
- Payment_PaymentID INT
- Payment_User_UserID INT
- NumberofReservation_NumberofReservation INT

Holidays
- HolidayID INT
- HolidayDate DATETIME
- Description VARCHAR(100)
- Notification_NotificationID INT

Department
- DepartmentID INT
- DepartmentName VARCHAR(45)
- Location VARCHAR(45)

ValidityPeriod
- ValidityPeriod INT
- TicketTypeID INT

NumberofReservation
- NumberofReservation INT
- ReservationID INT

Bus
- BusID INT
- BusNumber VARCHAR(45)
- Capacity INT
- LicensePlateNumber VARCHAR(45)
- BusType VARCHAR(45)

Bus Route
- RouteID INT
- StartLocation VARCHAR(45)
- EndLocation VARCHAR(45)

Bus Route_has_Route Stop Sequence
- Bus Route_RouteID INT
- Route Stop Sequence_StopSequenceID INT

Bus_has_Schedule
- Bus_BusID INT
- Schedule_ScheduleID INT

Bus_has_Bus Route
- Bus_BusID INT
- Bus Route_RouteID INT

Route Stop Sequence
- StopSequenceID INT
- ArrivalTime DATETIME
- NumberofStops_NumberofStops INT

Schedule
- ScheduleID INT
- DayFlag VARCHAR(45)
- ArrivalTime DATETIME
- DepartureTime DATETIME

Bus Route_has_Schedule
- Bus Route_RouteID INT
- Schedule_ScheduleID INT

NumberofStops
- NumberofStops INT
- Bus Stop_StopID INT

Location
- Location INT
- Bus Stop_StopID INT

Bus Stop
- StopID INT
- StopName VARCHAR(45)

Fig 4: Enhanced Entity Relationship Diagram

SQL DATABASE DEVELOPMENT

- **Database Creation**: Created a database ‘Dutchess_county_bus_transportation_DBMS_project’.
- **Table Creation**: Defined tables for entities such as User, Bus, Bus Route, Ticket Type, Department, Employee, Admin, Bus Stop, Notification, Holidays, Route Stop Sequence, Payment, Reservation, Bus_has_Bus Route, Bus Route_has_Route Stop Sequence, Schedule, Bus_has_Schedule, Bus Route_has_Schedule, Payment_has_Reservation.
- **Relationships**: Established relationships between tables using foreign key constraints. Example: User makes many Payments, User receives many Notifications.
- **Normalization**: Ensured data integrity by defining primary keys and enforcing foreign key relationships. Used normalization techniques to organize data efficiently.
- **Data Types**: Choose appropriate data types for each attribute to optimize storage and retrieval. Example: Used VARCHAR for names, and DATETIME for date and time information.
- **Indexes**: Created unique and regular indexes to improve query performance. Example: Created unique indexes for primary keys and other attributes where uniqueness is required.
- **Constraints**: Implemented constraints such as PRIMARY KEY, UNIQUE, and FOREIGN KEY to maintain data accuracy. Example: Defined a primary key for UserID in the User table.
- **Data Integrity**: Ensured data integrity by specifying ON DELETE and UPDATE actions in foreign key constraints. Example: Set actions to NO ACTION in certain relationships to prevent unwanted changes.
- **Engine Selection**: Specified the storage engine (InnoDB) for each table to provide features like transactions and referential integrity.

LOADING DATA AND PERFORMANCE ENHANCEMENTS

➢ **Loading Data**:

1. **Manual Insertion**: The process involves inserting parent table records before child table records, ensuring foreign key references exist. Foreign key checks are temporarily disabled to allow inserting records with invalid foreign key references and then re-enabled after insertion.

2. **CSV Import**: Data is imported from CSV files into respective tables using the LOAD DATA LOCAL INFILE command. The correct file path, field separator, enclosure, and line terminator are specified. IGNORE 1 ROWS is used to skip the header row, and column mappings are ensured.

➢ **Performance Enhancements**:

1. **Insertion Optimization**:
   - Bulk INSERT Statements: Utilizing bulk INSERT statements instead of separate ones for improved performance.
   - Disabling Indexes and Foreign Key Checks: Temporarily disabling foreign key checks and indexes during insertion and re-enabling them afterward.
   - Multiple VALUES Lists: Using multiple VALUES lists in a single INSERT statement for efficiency.
   - CSV Data Import: Loading data from CSV files using LOAD DATA instead of individual INSERT statements, minimizing context switches.

2. **Normalization**:
   - First Normal Form (1NF): Ensuring each table cell contains a single value, and records are unique.
   - Second Normal Form (2NF): Eliminating partial dependencies by making sure non-prime attributes are fully functionally dependent on the primary key.
   - Third Normal Form (3NF): Removing transitive dependencies by creating new tables (e.g., Bus_has_Schedule) to address them.

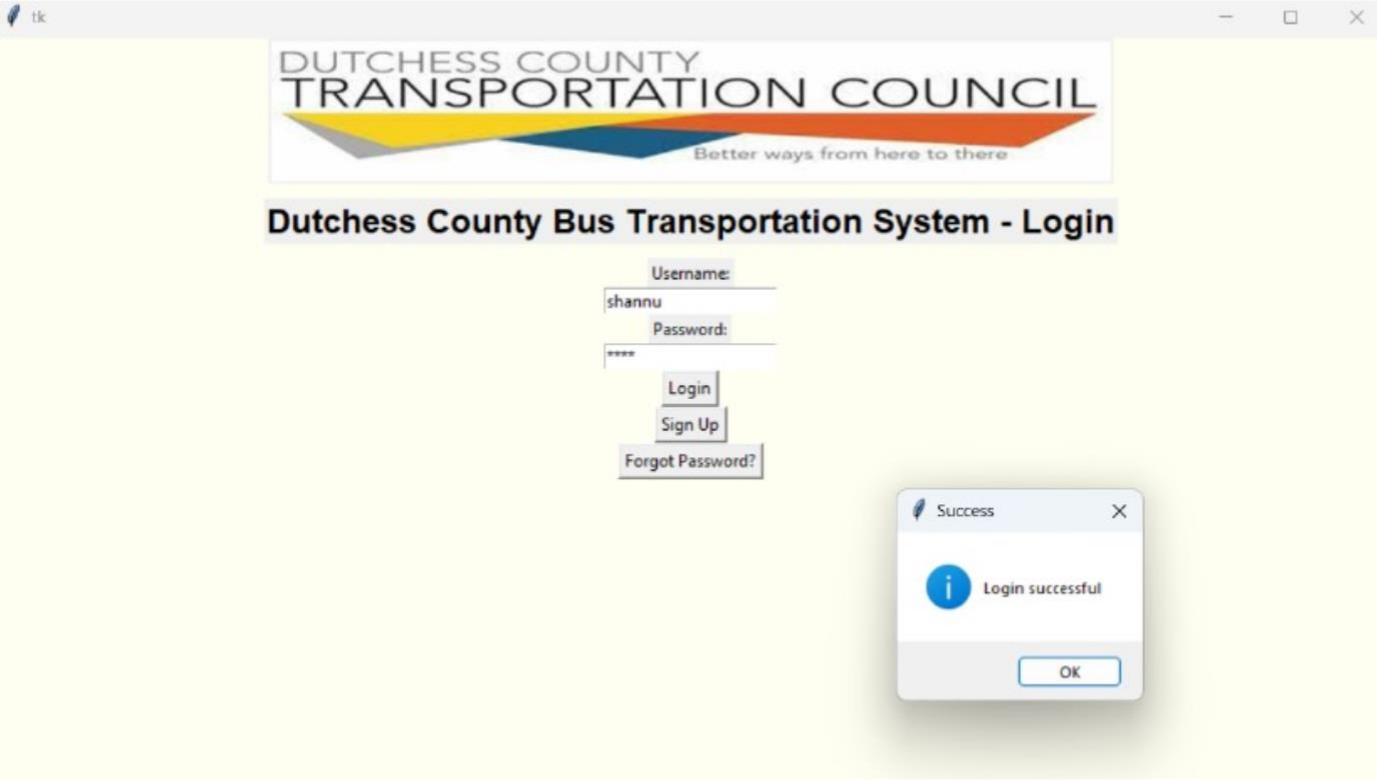
GRAPHICAL USER INTERFACE

Graphical User Interface has the following elements:

- **Database Connectivity**:
  - The application establishes a connection to a MySQL database for user authentication and data storage.
  - Queries are executed using the MySQL connector library.
- **Image Display**:
  - The login window displays an image fetched from a URL using the requests library.
  - The image is processed using the Pillow (PIL) library and displayed in the Tkinter window.
- **Code Structure**:
  - The code is organized into functions and utilizes Tkinter for GUI elements.
  - Error handling is implemented for database-related errors.The

GRAPHICAL USER INTERFACE

- **Login Page:**
  - The GUI includes a login page with entry fields for username and password.
  - Users can attempt to log in, and their credentials are checked against a MySQL database.
  - Buttons for "Sign Up" and "Forgot Password?" provide additional functionalities.



Dutchess County Bus Transportation System - Login

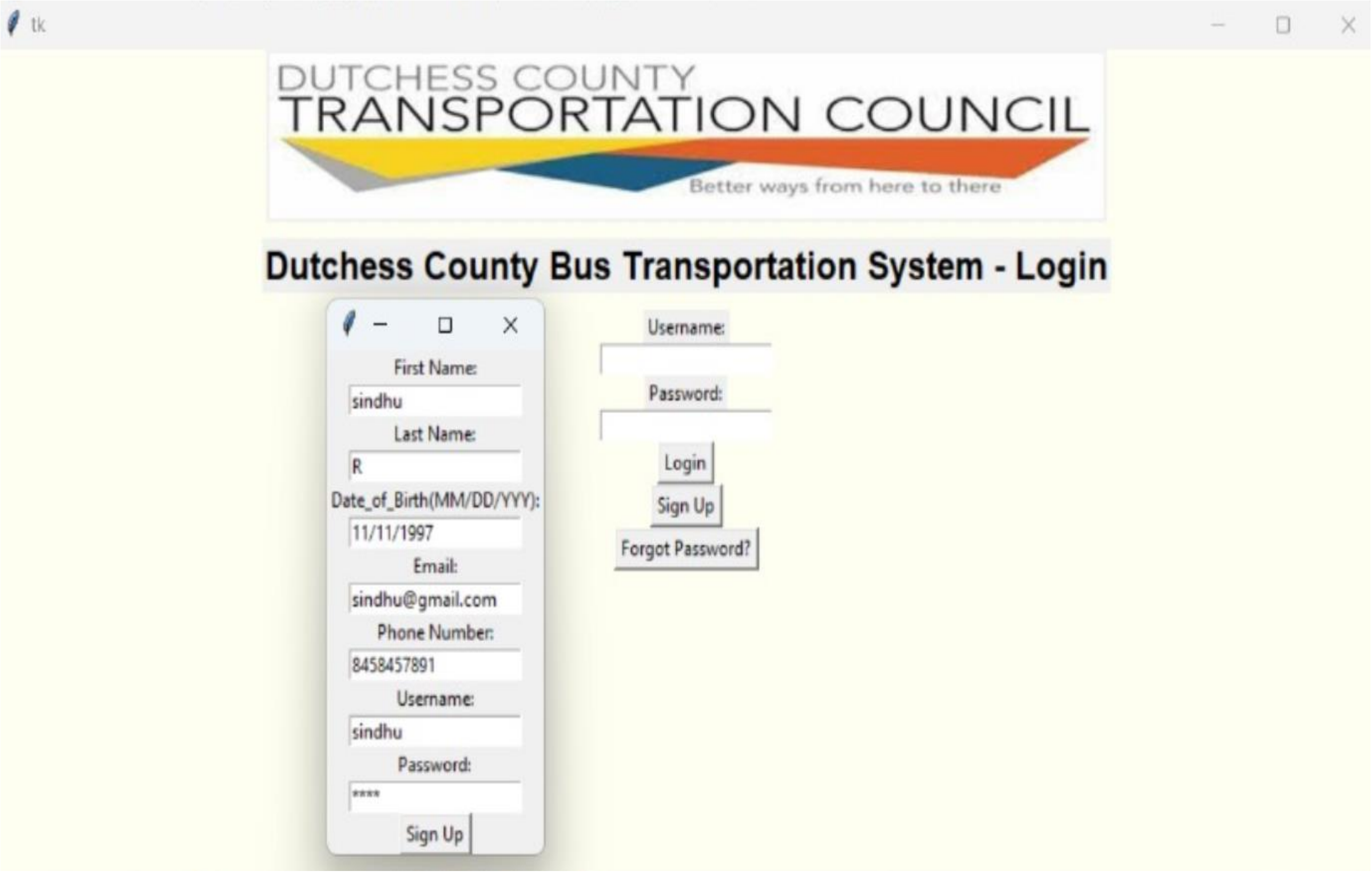
Username:
shannu
Password:
****

Login successful

Fig 5: Login Page

GRAPHICAL USER INTERFACE

- **Sign-Up Page:**
  - A separate window allows users to register by providing their first name, last name, email, username, and password.
  - User information is inserted into the MySQL database upon submission.



Dutchess County Bus Transportation System - Login

First Name: sindhu

Last Name: R

Date_of_Birth(MM/DD/YYY): 11/11/1997

Email: sindhu@gmail.com

Phone Number: 8458457891

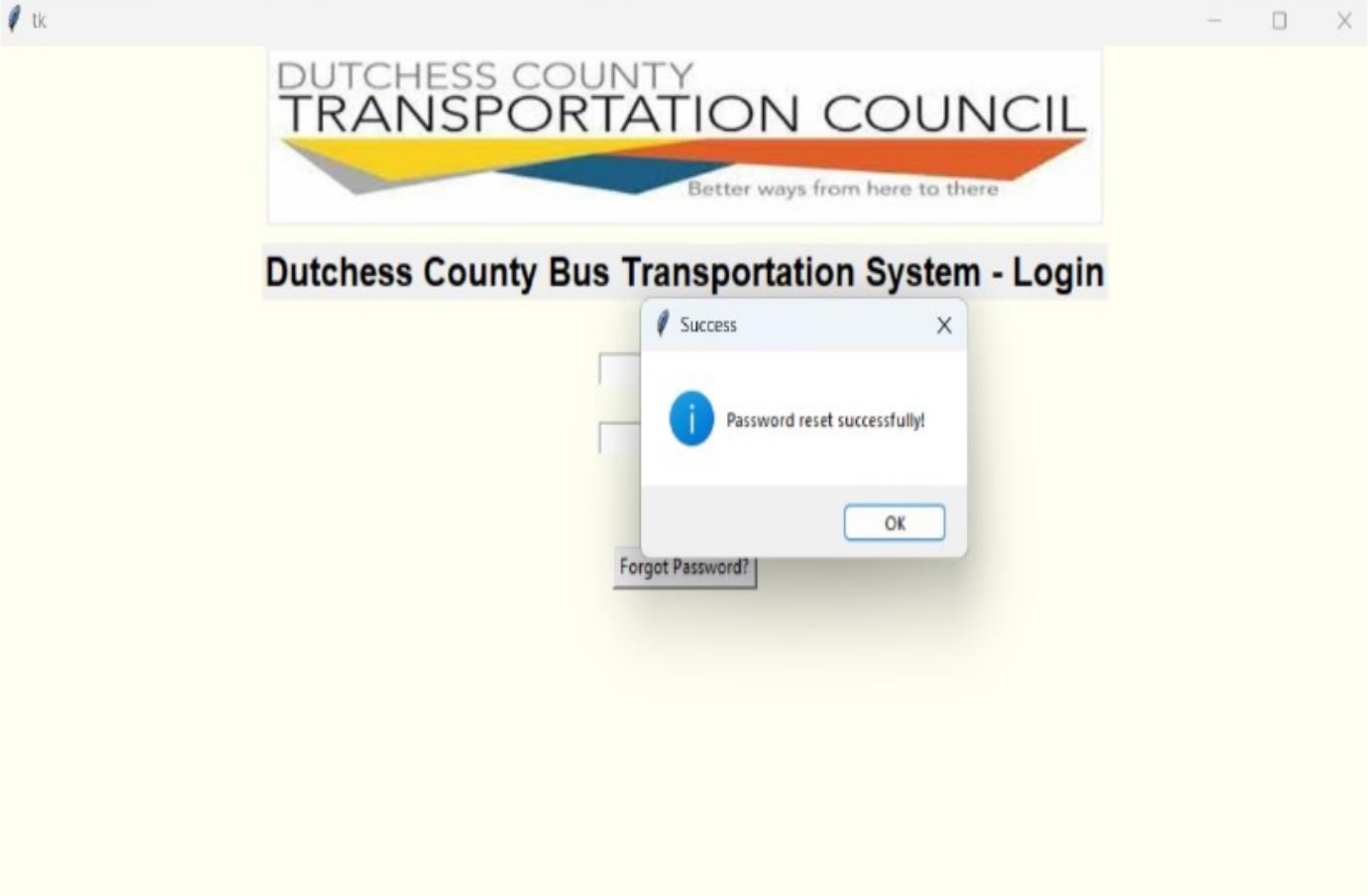
Username: sindhu

Password: ****

Fig 6: Sign up Page

GRAPHICAL USER INTERFACE

- **Forgot Password Page:**
  - Users can reset their password by entering their username and new password.
  - The application updates the password in the database.



Dutchess County Bus Transportation System - Login

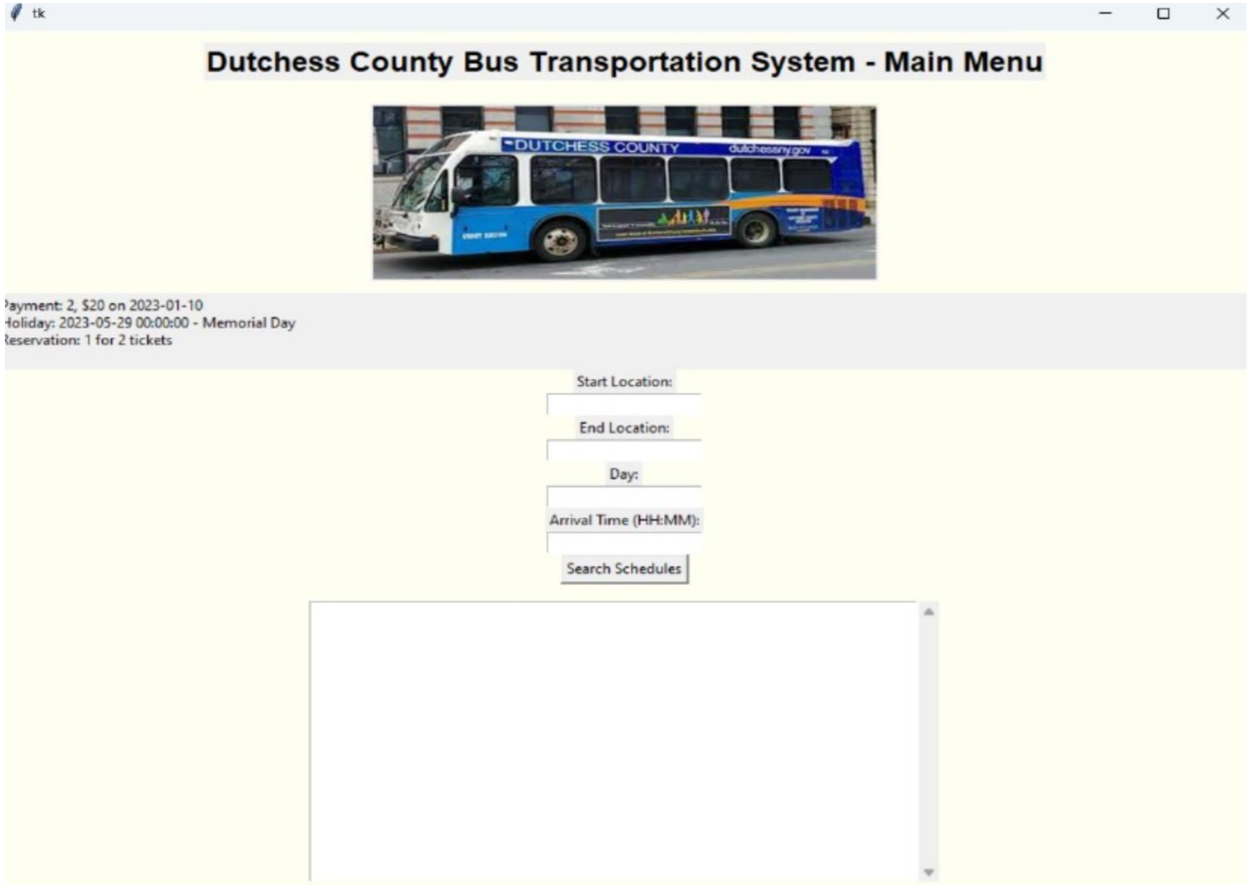
Success

Password reset successfully!

Fig 7: Forgot Password Page

GRAPHICAL USER INTERFACE

- **Main Menu Page:**
  - After successful login, users are directed to the main menu.
  - Entry fields and buttons allow users to input transportation details and perform actions like searching schedules or making payments.



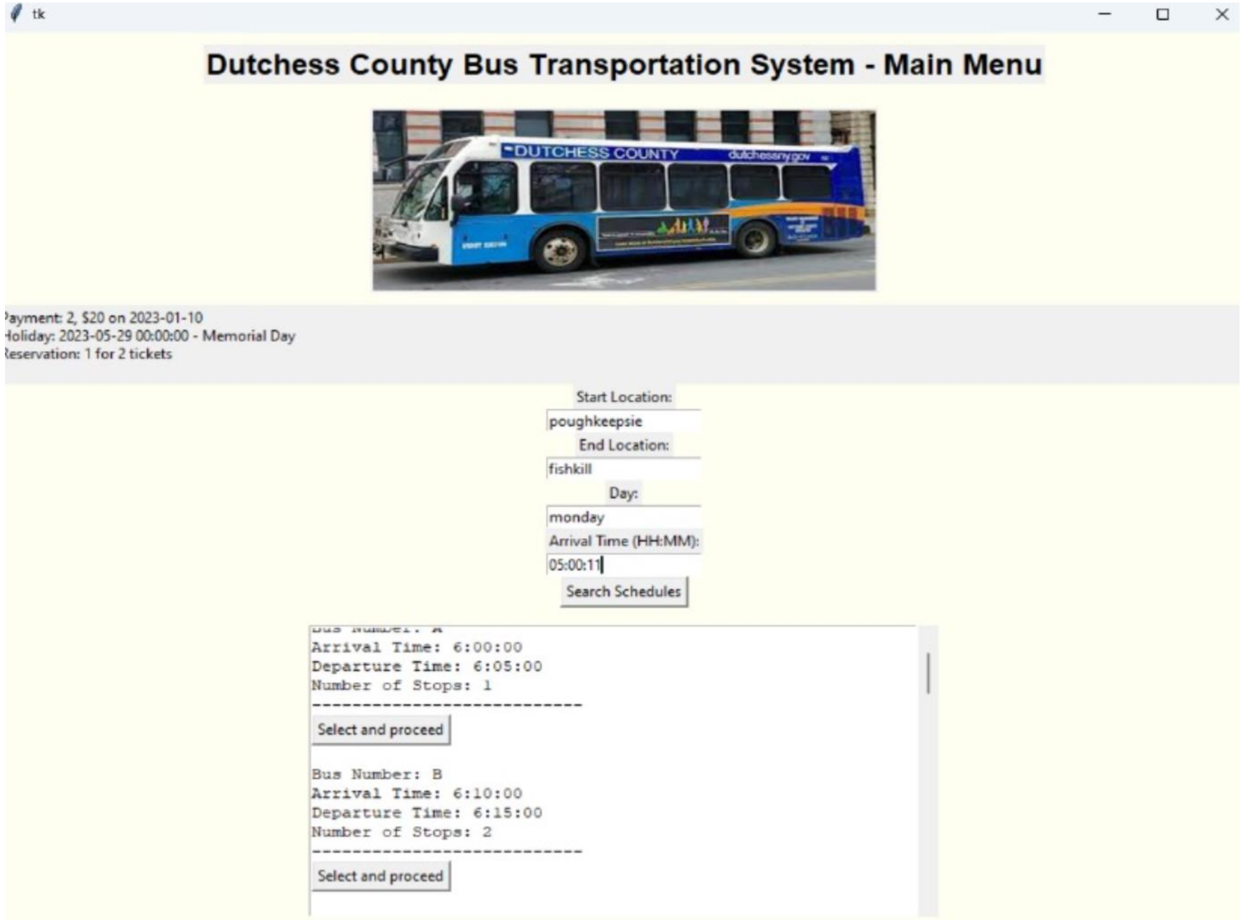
Dutchess County Bus Transportation System - Main Menu

```
ayment: 2, $20 on 2023-01-10
Holiday: 2023-05-29 00:00:00 - Memorial Day
Reservation: 1 for 2 tickets
```

Fig 8: Main Menu Page

GRAPHICAL USER INTERFACE

- **Search Page (Action Page):**
  - The search functionality involves retrieving bus schedule information based on user input (start, end, day, arrival time).
  - The results are displayed in a text widget, and users can select a schedule to proceed to payment**.**



Dutchess County Bus Transportation System - Main Menu

Payment: 2, $20 on 2023-01-10
Holiday: 2023-05-29 00:00:00 - Memorial Day
Reservation: 1 for 2 tickets

Start Location: poughkeepsie
End Location: fishkill
Day: monday
Arrival Time (HH:MM): 05:00:11

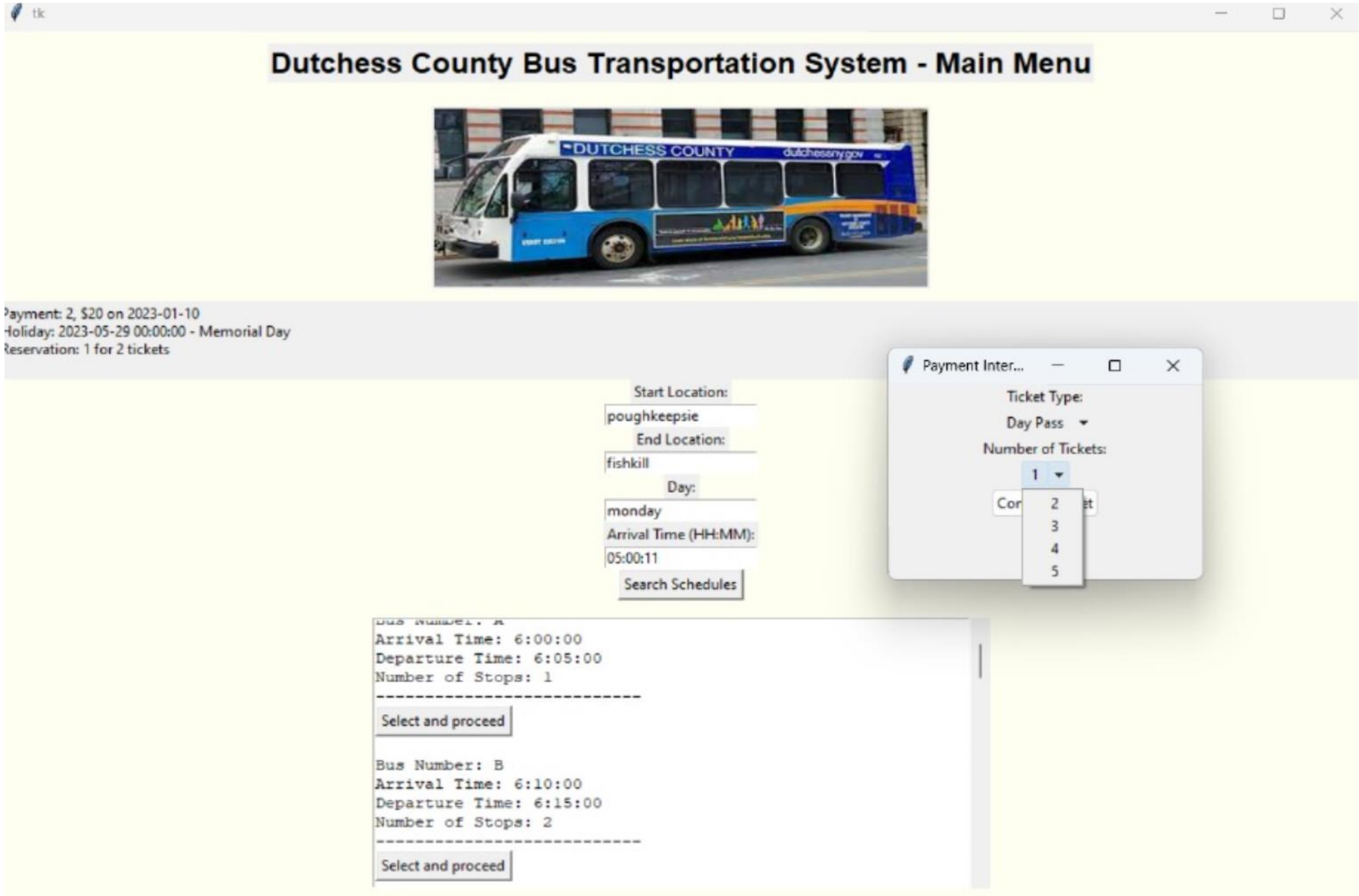
```
Bus Number: A
Arrival Time: 6:00:00
Departure Time: 6:05:00
Number of Stops: 1
---------------------------

Bus Number: B
Arrival Time: 6:10:00
Departure Time: 6:15:00
Number of Stops: 2
---------------------------
```

Fig 9: Search Page

GRAPHICAL USER INTERFACE

- **Payment Page:**
  - There is a placeholder for the payment functionality, indicating the intent to implement payment features.
  - The specific payment implementation is not provided in the code.



tk

Dutchess County Bus Transportation System - Main Menu

ayment: 2, $20 on 2023-01-10
Holiday: 2023-05-29 00:00:00 - Memorial Day
Reservation: 1 for 2 tickets

Start Location: poughkeepsie

End Location: fishkill

Day: monday

Arrival Time (HH:MM): 05:00:11

```
Bus Number: A
Arrival Time: 6:00:00
Departure Time: 6:05:00
Number of Stops: 1
----------------------------
```

```
Bus Number: B
Arrival Time: 6:10:00
Departure Time: 6:15:00
Number of Stops: 2
----------------------------
```

Payment Inter...

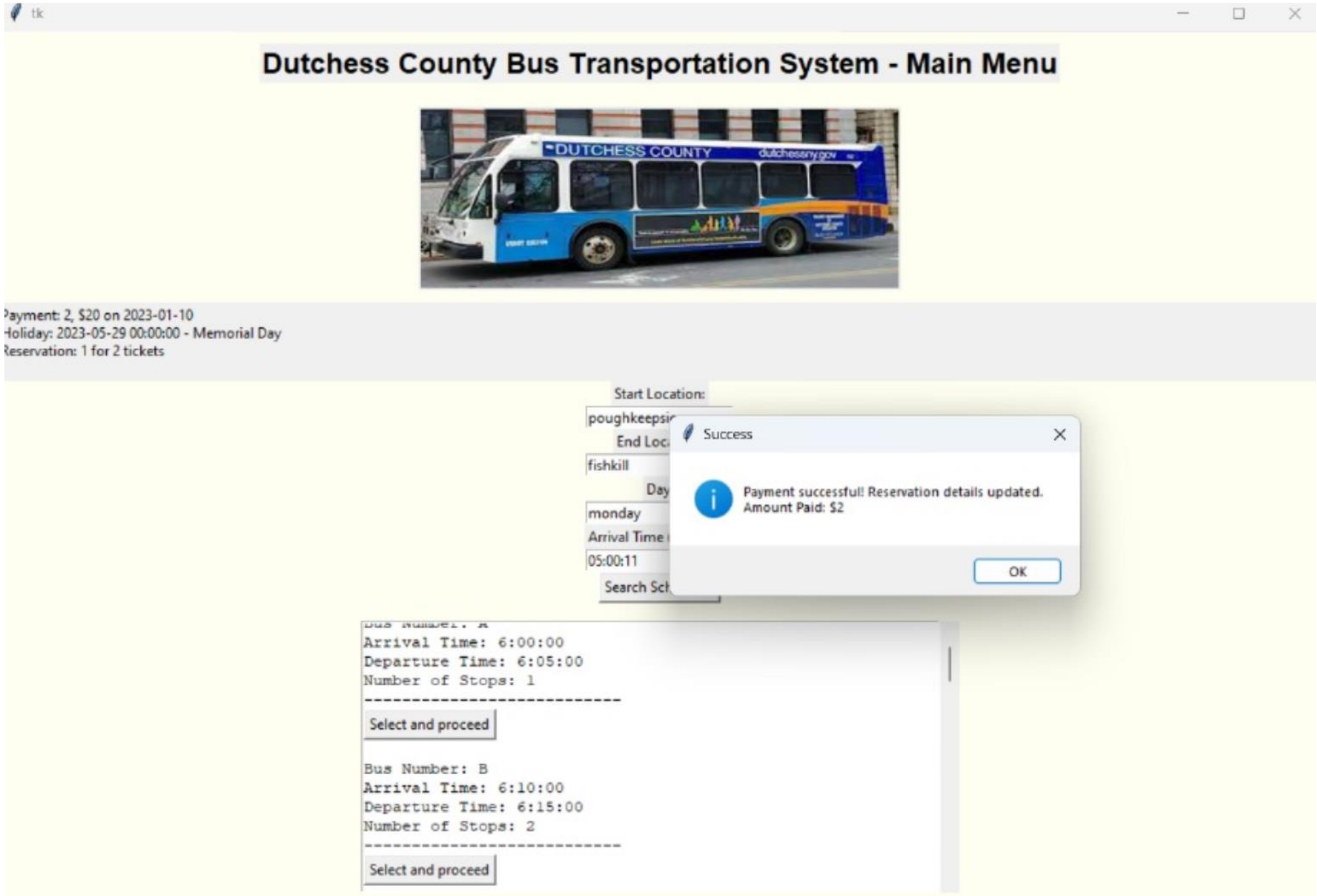
Ticket Type: Day Pass

Number of Tickets: 1

Fig 10: Payment Page

GRAPHICAL USER INTERFACE

- **Payment Successful Page:**
  - The Below figure shows how the payment is successful once the user makes the payment.



Dutchess County Bus Transportation System - Main Menu

```
ayment: 2, $20 on 2023-01-10
oliday: 2023-05-29 00:00:00 - Memorial Day
eservation: 1 for 2 tickets
```

Start Location: poughkeepsi

End Loc: fishkill

Day: monday

Arrival Time: 05:00:11

Success

Payment successful! Reservation details updated.
Amount Paid: $2

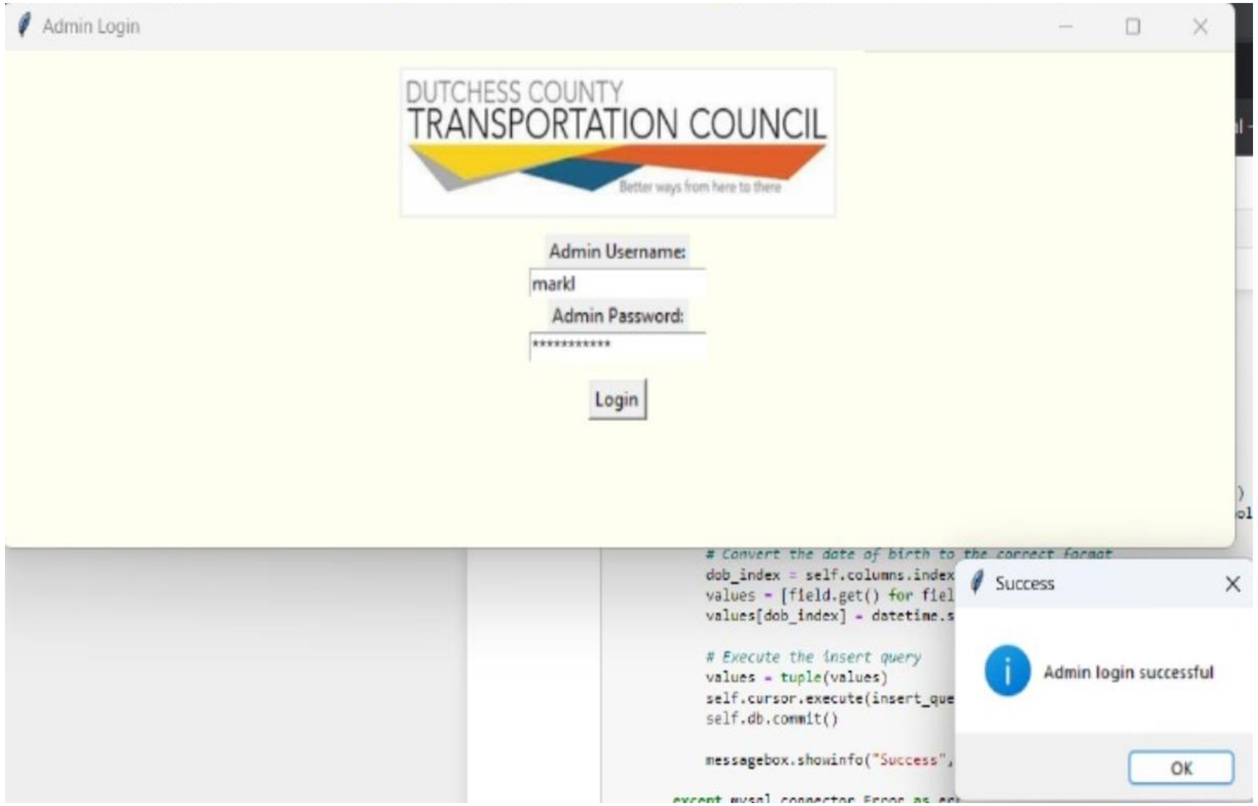
```
Bus Number: A
Arrival Time: 6:00:00
Departure Time: 6:05:00
Number of Stops: 1
---------------------------
```

```
Bus Number: B
Arrival Time: 6:10:00
Departure Time: 6:15:00
Number of Stops: 2
---------------------------
```

Fig 11: Payment Success Confirmation

GRAPHICAL USER INTERFACE

- **Admin Login Page:**
  - The GUI includes an admin login page with entry fields for admin username and admin password.
  - Admins can attempt to log in, and their credentials are checked against a MySQL database.



Admin Username: markl

Admin Password: ***********

```
        # Convert the date of birth to the correct format
        dob_index = self.columns.index
        values = [field.get() for fiel
        values[dob_index] = datetime.s

        # Execute the insert query
        values = tuple(values)
        self.cursor.execute(insert_que
        self.db.commit()

        messagebox.showinfo("Success",

    except mysql.connector.Error as er
```

Admin login successful

Fig 12: Admin Login Page

GRAPHICAL USER INTERFACE

- **Admin Main Menu Page:**
  - The application includes admin pages for managing users, bus routes, and ticket types.
  - Admins can update user information, manage routes, and handle ticket-related details.

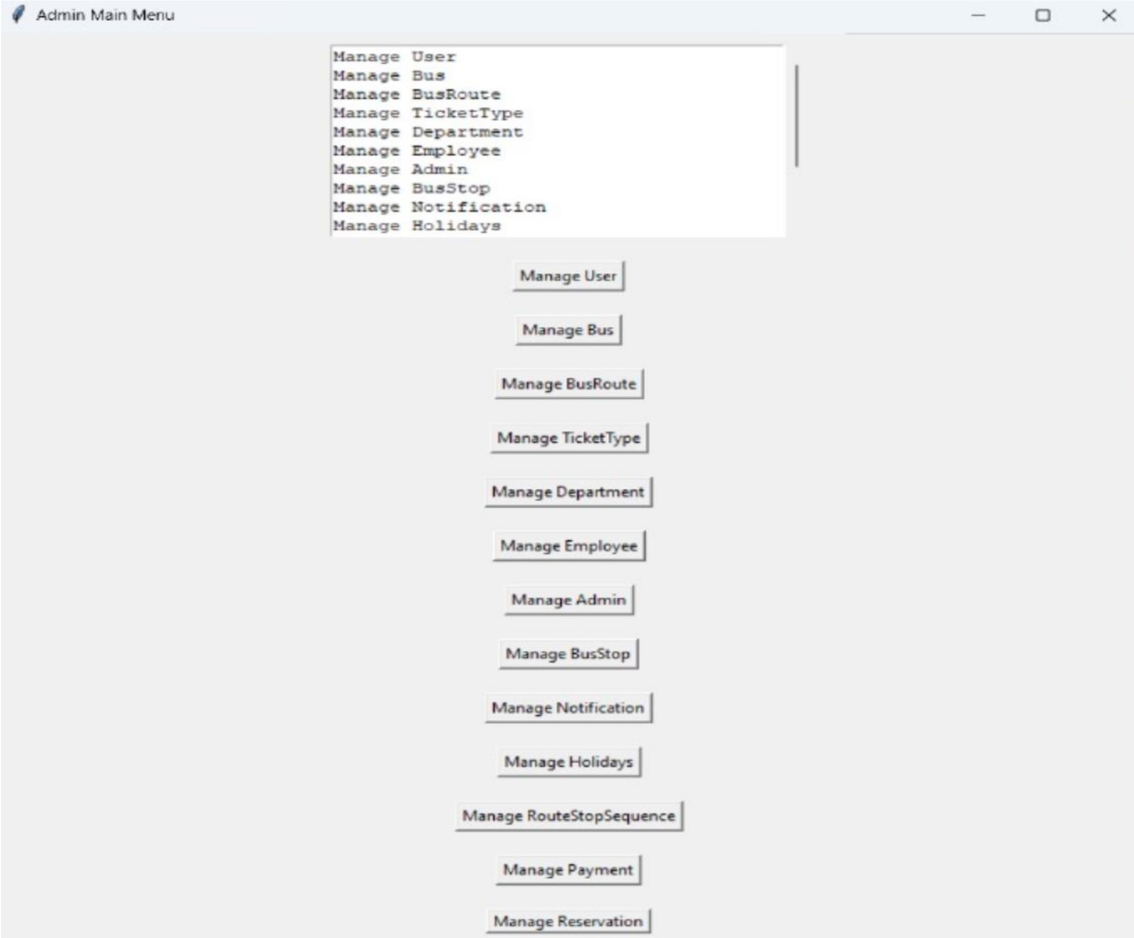
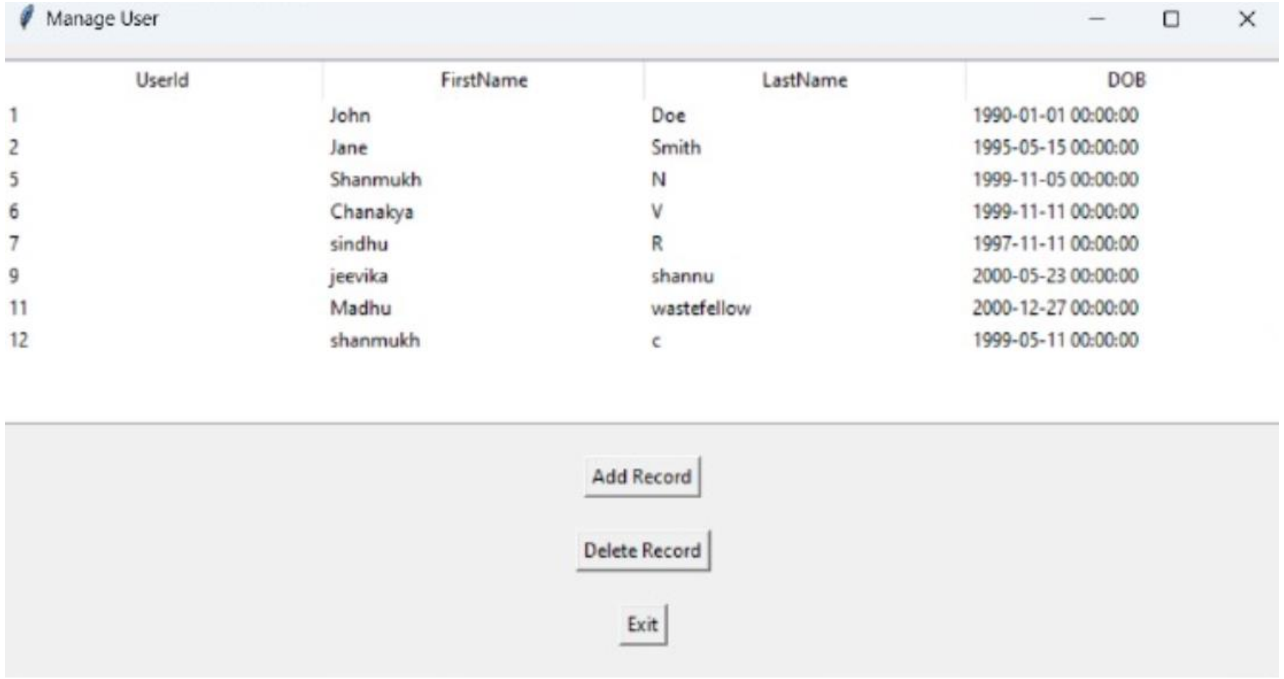


Fig 13: Admin Main Menu Page

GRAPHICAL USER INTERFACE

- **Manage User Page:**
  - The application includes managing various entities. Eg: User
  - Admins can add or delete user information.
  - Adding and deleting is common for all entities.

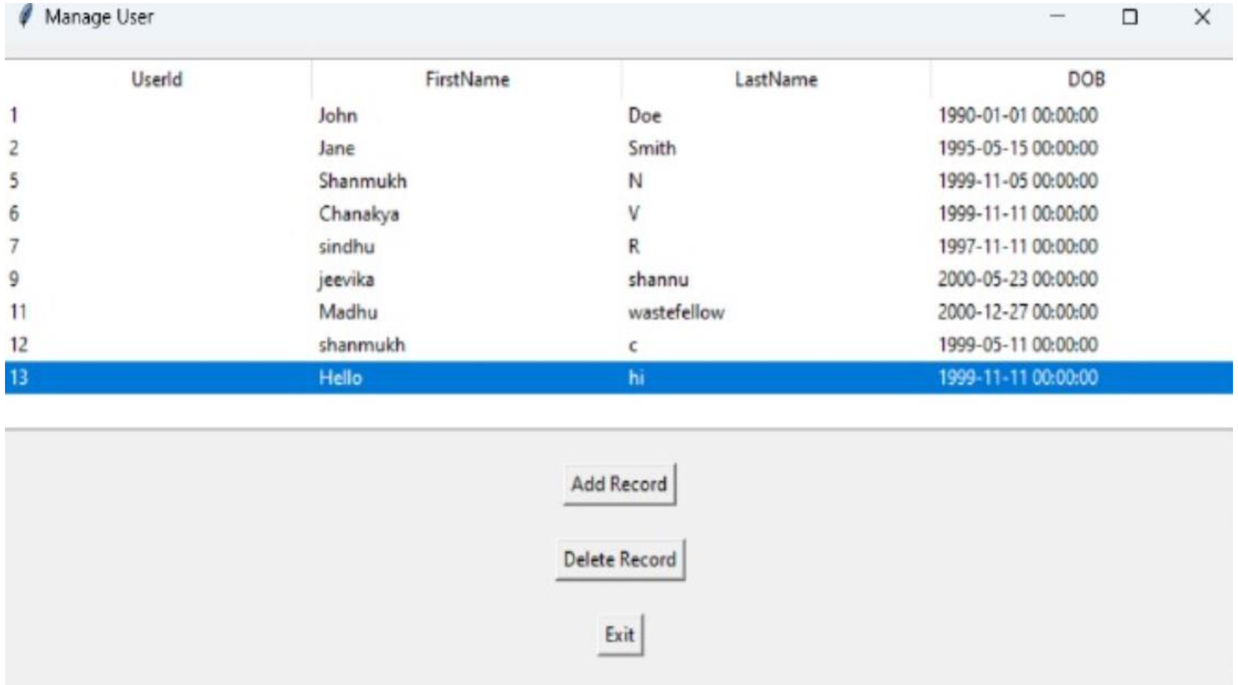


UserId	FirstName	LastName	DOB
1	John	Doe	1990-01-01 00:00:00
2	Jane	Smith	1995-05-15 00:00:00
5	Shanmukh	N	1999-11-05 00:00:00
6	Chanakya	V	1999-11-11 00:00:00
7	sindhu	R	1997-11-11 00:00:00
9	jeevika	shannu	2000-05-23 00:00:00
11	Madhu	wastefellow	2000-12-27 00:00:00
12	shanmukh	c	1999-05-11 00:00:00

Fig 14: Manage User Page

GRAPHICAL USER INTERFACE

- **Delete record:**
  - Upon selecting a particular User and clicking on Delete Record, the respective record will be deleted.

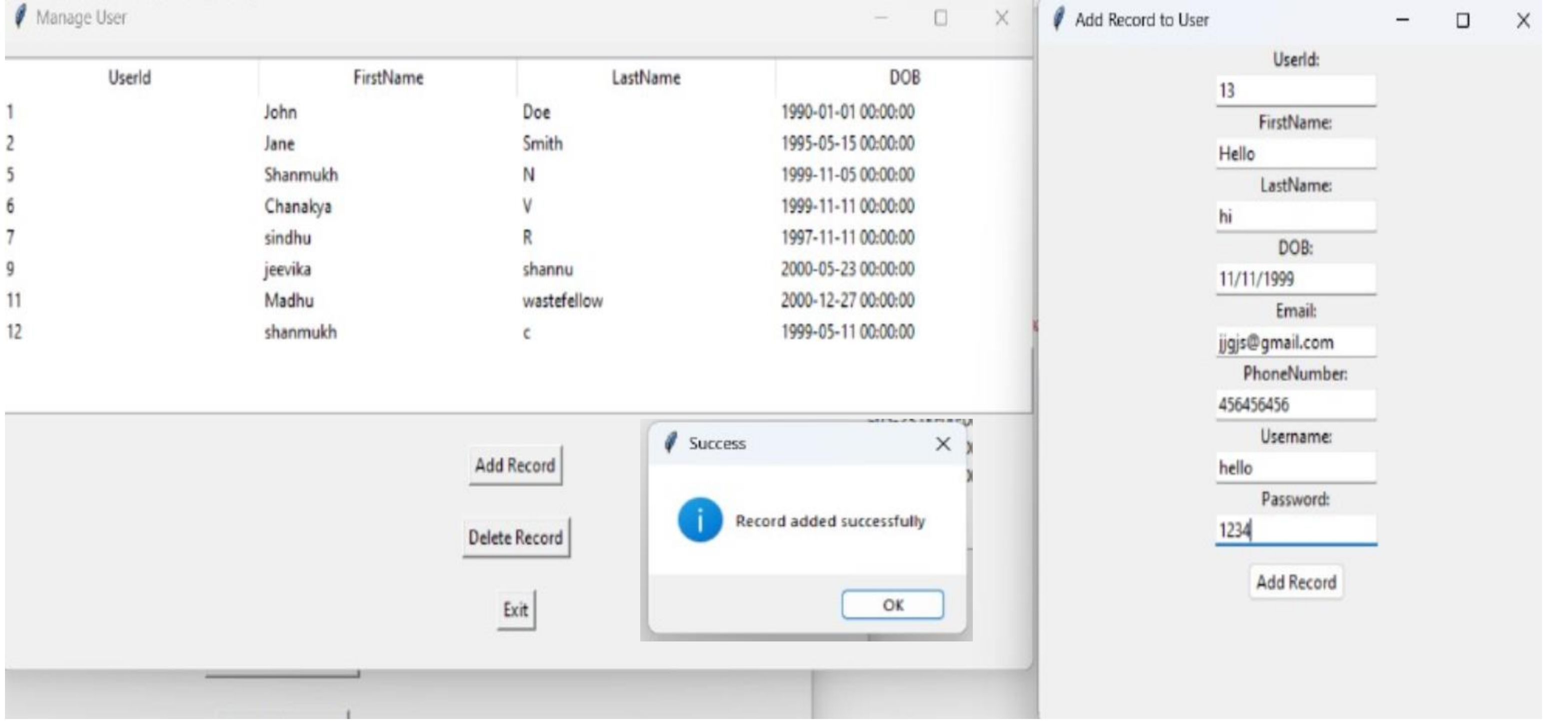


UserId	FirstName	LastName	DOB
1	John	Doe	1990-01-01 00:00:00
2	Jane	Smith	1995-05-15 00:00:00
5	Shanmukh	N	1999-11-05 00:00:00
6	Chanakya	V	1999-11-11 00:00:00
7	sindhu	R	1997-11-11 00:00:00
9	jeevika	shannu	2000-05-23 00:00:00
11	Madhu	wastefellow	2000-12-27 00:00:00
12	shanmukh	c	1999-05-11 00:00:00
13	Hello	hi	1999-11-11 00:00:00

Fig 15: Delete Record

GRAPHICAL USER INTERFACE

- **Add record:**
  - Upon clicking on Add Record, a new window will open up that will take the input from the user based on whose values the new User will be created.



Manage User

UserId	FirstName	LastName	DOB
1	John	Doe	1990-01-01 00:00:00
2	Jane	Smith	1995-05-15 00:00:00
5	Shanmukh	N	1999-11-05 00:00:00
6	Chanakya	V	1999-11-11 00:00:00
7	sindhu	R	1997-11-11 00:00:00
9	jeevika	shannu	2000-05-23 00:00:00
11	Madhu	wastefellow	2000-12-27 00:00:00
12	shanmukh	c	1999-05-11 00:00:00

Success

Record added successfully

Add Record to User

UserId:	13
FirstName:	Hello
LastName:	hi
DOB:	11/11/1999
Email:	jjgjs@gmail.com
PhoneNumber:	456456456
Username:	hello
Password:	1234

Fig 16: Add Record

REFERENCES

- Dutchess County Public Transportation Application:
  - https://www.dutchessny.gov/Departments/Public-Transit/Public-Transit.htm
- NYC Transit: MTA Subway and Bus:
  - https://transitapp.com/region/new-york
- Moovit:
  - https://moovitapp.com/nycnj-121/poi/en

THANK YOU