



Shail Desai

CIT 38800 - 20013

Java Project Report

April 28th, 2024

JAVA PROJECT REPORT

❖ Introduction:

The Campus Transportation Application stands as a hallmark of technological innovation, strategically engineered to streamline the transit experience within the academic realm. Anchored by the formidable Oracle Database, this application epitomizes the pinnacle of data integrity and transactional efficiency. It offers a conduit through which students can effortlessly align their schedules with pre-determined transportation routes, a testament to our institution's commitment to operational excellence. This report aims to expound upon the application's functionalities, elucidate its database interactions, and highlight the seamless user experience it provides, all while underscoring the meticulous attention to detail that characterizes its integration into campus life.

❖ What Does This Application Solve?

The Campus Transportation Application addresses several key challenges faced by educational institutions in managing and coordinating transportation for students:

- **Efficiency in Scheduling:** It resolves the complexity of aligning transportation schedules with individual student timetables, ensuring that students have access to transportation that fits their unique academic schedules.
- **Ease of Access:** The application simplifies the process of accessing transportation services by allowing students to log in with their Student ID, thereby removing the need for complex booking procedures.
- **Data Management:** By utilizing the robust Oracle Database, the application ensures secure and reliable management of student information, transportation schedules, and payment transactions.
- **Resource Optimization:** The preselection of routes by administrators, based on data-driven insights, enhances route planning and vehicle utilization, reducing wasted trips and optimizing transport resources.
- **User Experience:** With a focus on user-centered design, the application provides a straightforward and intuitive interface that reduces the learning curve and improves overall satisfaction with transportation services.

- **Payment Processing:** It streamlines the financial transactions related to transportation services, providing a secure and straightforward process for students to pay for their commutes.
- **Verification and Validation:** The application ensures that only eligible students can access transportation services by verifying their details against the preselected routes and schedules.
- **Transparency:** By providing digital receipts with detailed payment information and a barcode for service access, the application fosters transparency and trust between the service provider and the students.

❖ Oracle Database Tables:

1. CAMPUS_USERS:

- Holds the account details for individuals who can access the transportation portal. Fields include a unique user identifier, username, and password. The primary key is UserID, which ensures each record is unique.

2. CAMPUS_STUDENTS:

- Contains student-specific information, linking to the CAMPUS_USERS table via UserID. It includes the student's name, department, major, and scheduled pickup and drop-off times. The primary key is StudentID, and it has a foreign key constraint linking to UserID from CAMPUS_USERS.

3. PAYMENT_DETAILS:

- Records the payment transactions made by students for transportation services. It tracks payment details such as amount, date, method, and associated card information. New fields like FirstName, LastName, ExpiryDate, CardNumber, and CVV have been added to complete the payment profile.

4. CAMPUS_ROUTES:

- Lists the transportation routes available for students. Each route has a unique identifier and a descriptive name. The primary key is RouteID, uniquely identifying each route.

5. USER_RECEIPT:

- Captures details about the receipts generated post-payment. It contains a ReceiptID, the payment identifier, and the date and description of the transaction. A foreign key constraint ensures PaymentID references the corresponding record in PAYMENT_DETAILS.

6. ROUTES:

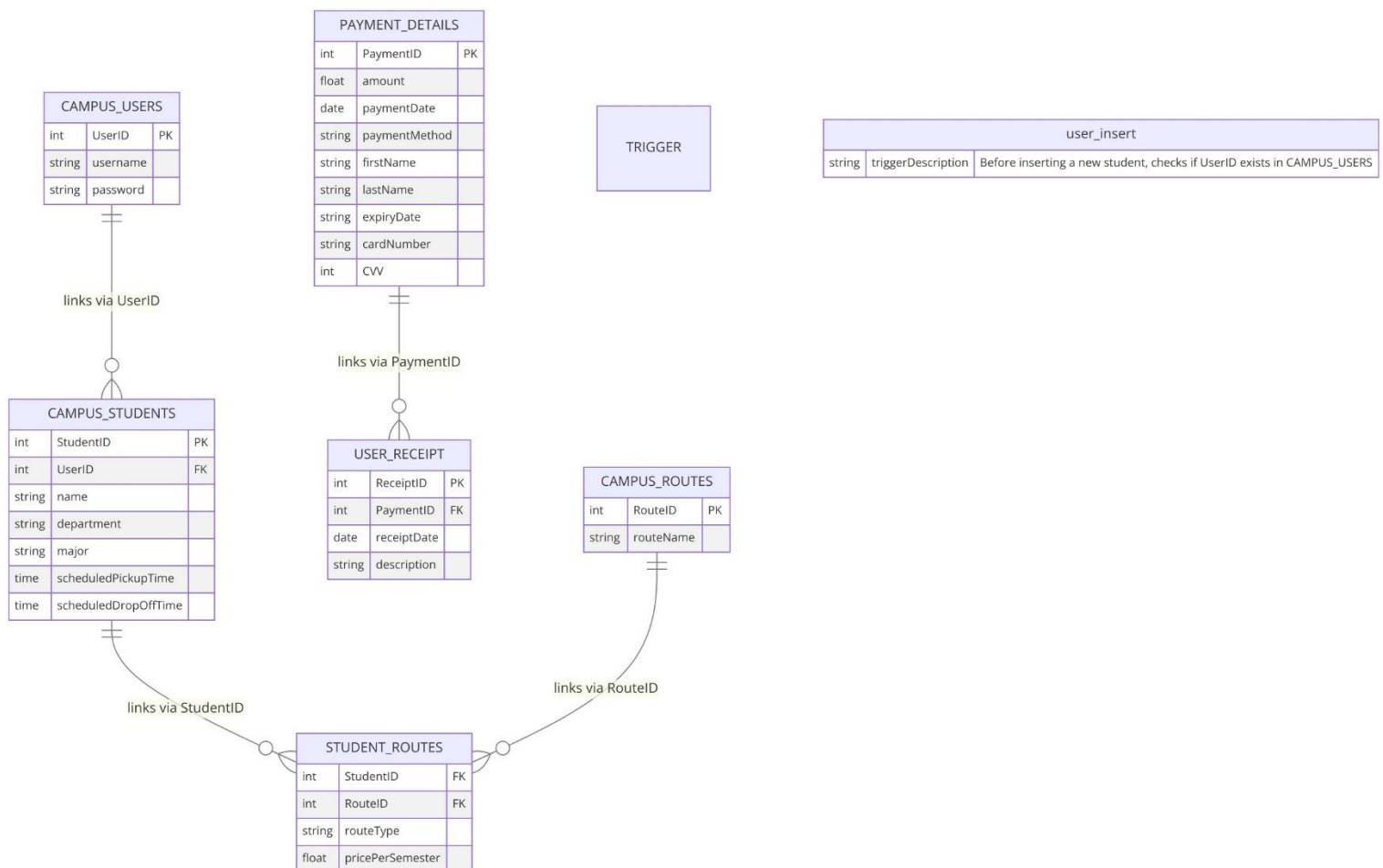
- Similar to CAMPUS_ROUTES, it seems to detail specific routes and their prices per month. The primary key is RouteID, and it includes RouteName and PricePerMonth.

7. STUDENT_ROUTES:

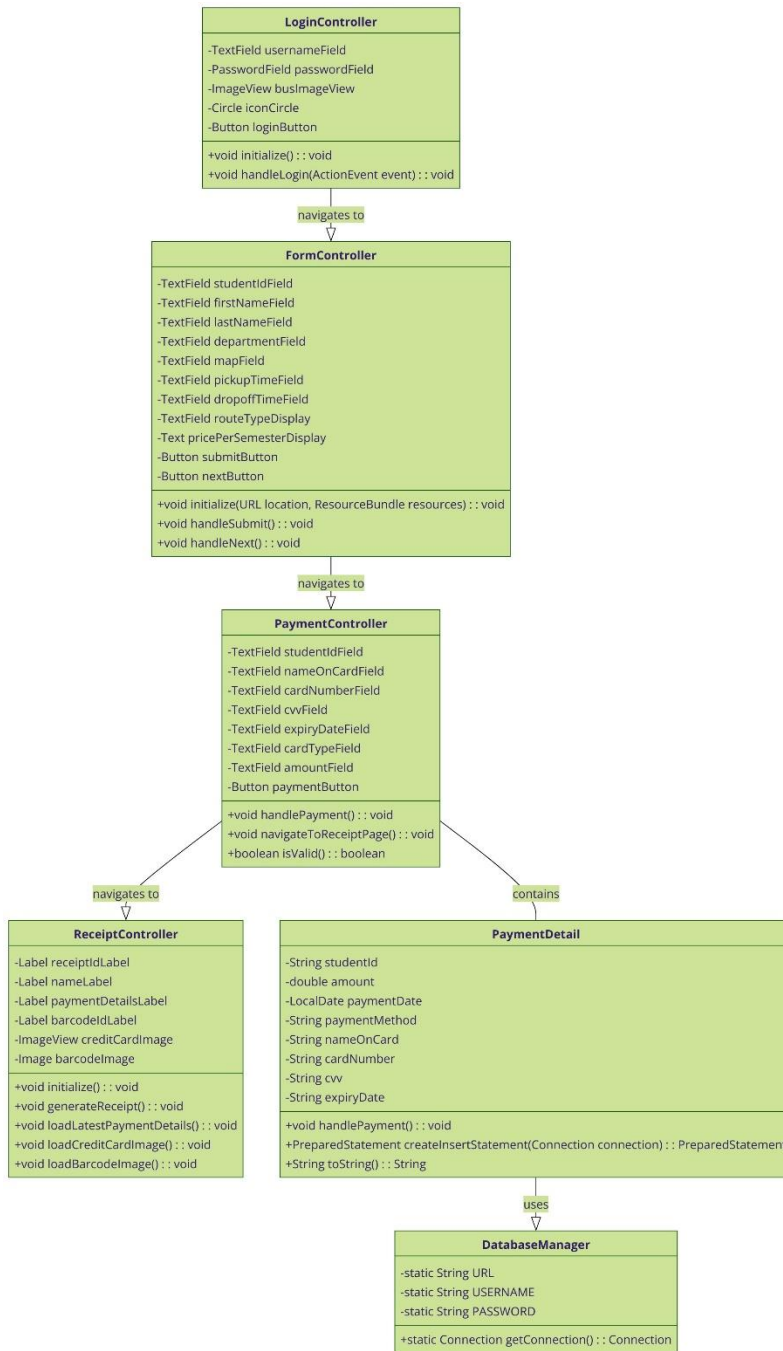
- Maps students to their selected routes, including the type of route and price per semester. This table seems to facilitate tracking which student is assigned to which route and at what cost.

8. TRIGGER user_insert:

- Not a table, but an important database object that ensures referential integrity. Before inserting a new student into CAMPUS_STUDENTS, it checks if the UserID exists in CAMPUS_USERS; if not, it assigns a random existing UserID.



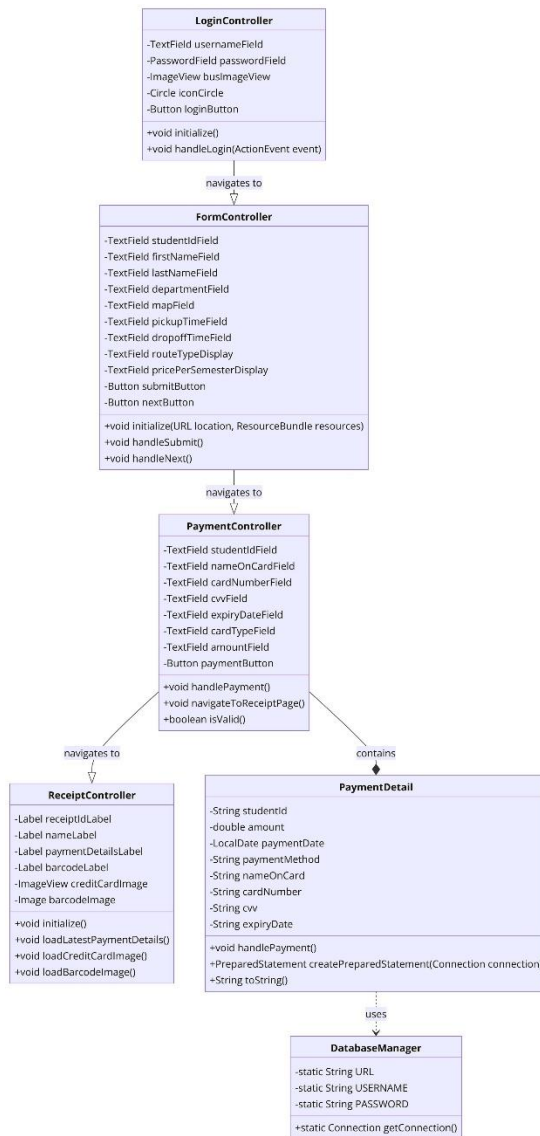
❖ System Design Diagram:



- **LoginController:** The user starts at the login screen, where they must enter their username and password. The LoginController class has fields for username and password inputs, an image view, a circle icon, and a login button. It has methods to initialize the screen and handle the login action.
- **FormController:** Once successfully logged in, the user is taken to a form, controlled by the FormController. This form collects details about the student, including student ID, name, department, major, and transportation scheduling information like pickup and drop-off times. It also includes route type and price per semester. This controller has methods to initialize the form, submit the data, and navigate to the next screen.
- **PaymentController:** After submitting the form, the user is directed to make a payment. The PaymentController manages the payment process, taking inputs for student ID, name on card, card number, CVV, expiry date, card type, and amount. It has methods to handle payment submission, navigate to the receipt page, and validate the information entered.
- **ReceiptController:** Following a successful payment, the ReceiptController takes over to present a receipt to the user. It contains labels for receipt ID, name, payment details, and barcode image. It has methods to initialize the view, load the payment details, and load the barcode image.
- **PaymentDetail:** It's a class that contains the details of the payment, such as student ID, payment amount, date, method, and card information. It has methods for handling the payment and creating a prepared statement to interact with the database.
- **DatabaseManager:** This class is used to manage the database connections and statements. It has static fields for the URL, username, and password required to connect to the database, and a method to get the database connection.

The "navigates to" and "contains" labels indicate the direction of navigation and class relationships within the application. The user flow is Login → Fill Form → Make Payment → Show Receipt, with data management handled by PaymentDetail and DatabaseManager classes.

❖ Class Diagram:



❖ Conclusion

In summation, the Campus Transportation Application has been meticulously engineered to transform the landscape of campus mobility. Through the strategic integration of a robust Oracle Database, the application successfully automates and refines the process of managing transportation logistics. The relational tables—from `CAMPUS_USERS` to `STUDENT_ROUTES`—work in concert to create a cohesive and responsive data architecture. This ensures a harmonious balance between operational efficiency and user experience. As

evidenced by the data schema and functionality, the application stands as a testament to the power of technological advancement in fostering a connected and efficient academic community. It is the synergy of these elements that culminates in the delivery of a superior transportation service, setting a new benchmark for educational institutions worldwide.

