Enterprise Application Development

Cachy Courier

Group Members:

- RANGISETTY NAVYA SAI - S20170010122 - navyasai.r17@iiits.in
- TUNUGUNTLA OOHA - S20170010167 - ooha.t17@iiits.in
- SAI SREE NITHYA - S20170010050 - saisreenithya.g17@iiits.in
- PUPPALA HRITHIK - S20170010115 - hrithik.p17@iiits.in

Description:

Our Project is An online Application which helps customers to transport their couriers to other parts of the City.

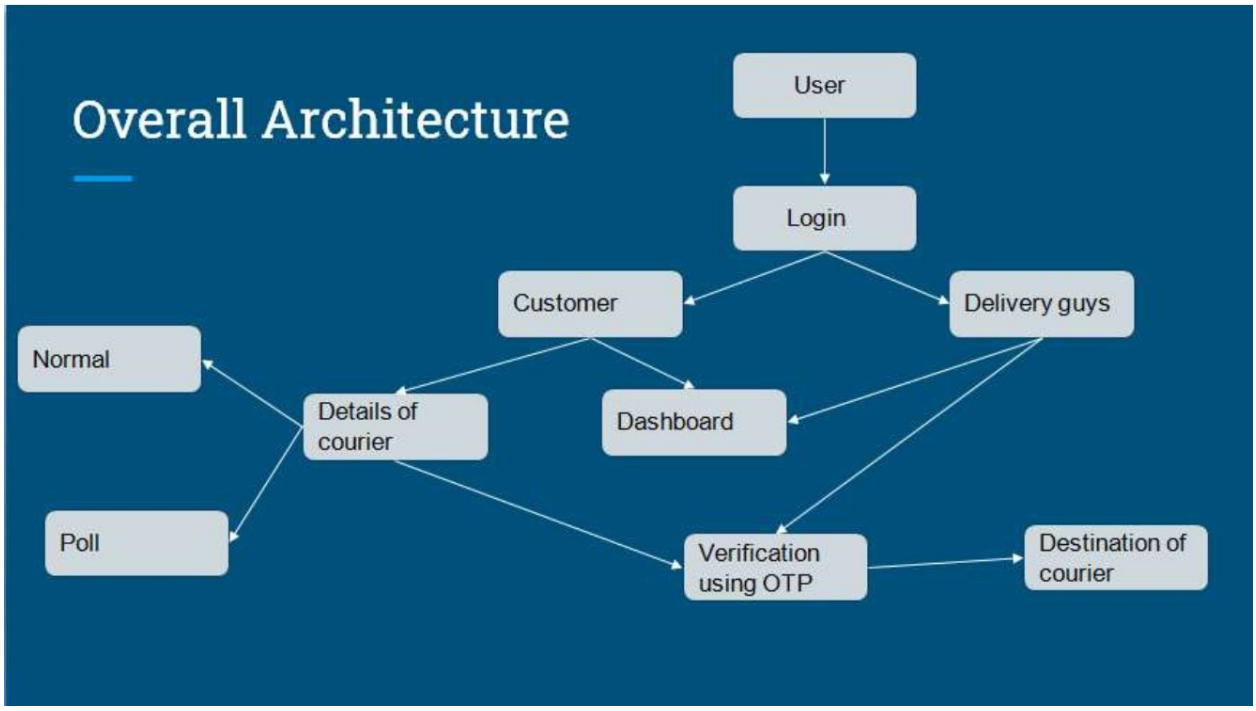
Introduction:

Mobile Courier Service System is a special courier system where all courier transactions are done via a mobile phone. The system also pre-informs the user about how much it will cost them to send the package .The mobile courier service system also provides a map for the courier agents which will enable them to easily navigate to their pickup or delivery point.

Motivation :

- This Simple and hassle-free Application helps people to save their time and energy.
- They can get their forgotten items from home to the workplace by our app in no time.
- This application can take-up immediate orders so that customers can save it for the future if any discrepancies are raised.

Architecture :



Tech Stack :

- Backend: NodeJS
- DataBase: MongoDB
- User Interface: Flutter App, Flutter Web

Modules

Module - 1:

User Portal and Placing orders

- User Authentication (Login / Sign Up).
- Delivery Executive Authentication (Login / Sign Up).
- Users can book Delivery Requests now or Later . For Completing the booking , User Needs to add the Details of the Package .

Status : Completed (User Can Login and Fill the Details for Booking the Deliveries for Parcels)

Module - 2:

Assigning Delivery Executive

- Users Will be assigned a Delivery Executive.
- Based on the Location of user using *Google Maps API ,* A Delivery executive is Assigned

Status : Completed (When a User Books for Delivery , A Nearest Delivery Executive is Assigned)

Module - 3:

Notifications

- When A Delivery Executive is Assigned to Particular Parcel , Delivery Executive gets an Email regarding the Details of The Parcel and Destination Address Which are Retrieved from Database .
- Similarly Users also get an Email regarding Delivery Executive Assignment .
- It is Implemented Using Node Mailer in node.js

Status: Completed (Notifications Are Implemented when the User and Delivery executive are Connected)

Module - 4:

Payments

- Managing Payments made By the Customer to the Application.
- Two Modes Are Implemented i.e.Cash On Delivery and Payment Through Various Online Applications.

Status : Completed (User Can Make Their Online Payments via net banking , paytm , UPI and Debit/Credit Cards)

Module - 5:

Live Tracking :

- When a Delivery Executive Starts to deliver a Courier then , *get_location* service of delivery executive gets started and user can get Live Location of Delivery Executive on Google Maps using API .
- When the Delivery executive delivers the courier , *get_location* service gets Stopped.

Status : Completed (The Live Location of The Executive can Be Seen by the User) .

Module - 6:

Optimised Routing (Pickups and Deliveries)

- Delivery Executive Should get the Best route towards picking the Delivery.
- Designing the Best path Algorithm in case of Normal Deliveries . Pool Algorithms

Status: Completed (Optimistic Path Algorithms are Made So that couriers are Sent in Less time and Less Effort with less cost .)

Module - 7:

Frontend :

The entire Frontend (Flutter Application) is Made User Friendly and developed with Decent UI .

Status : Completed (Flutter Frontend is Made for each and every subsection of all the Pages)

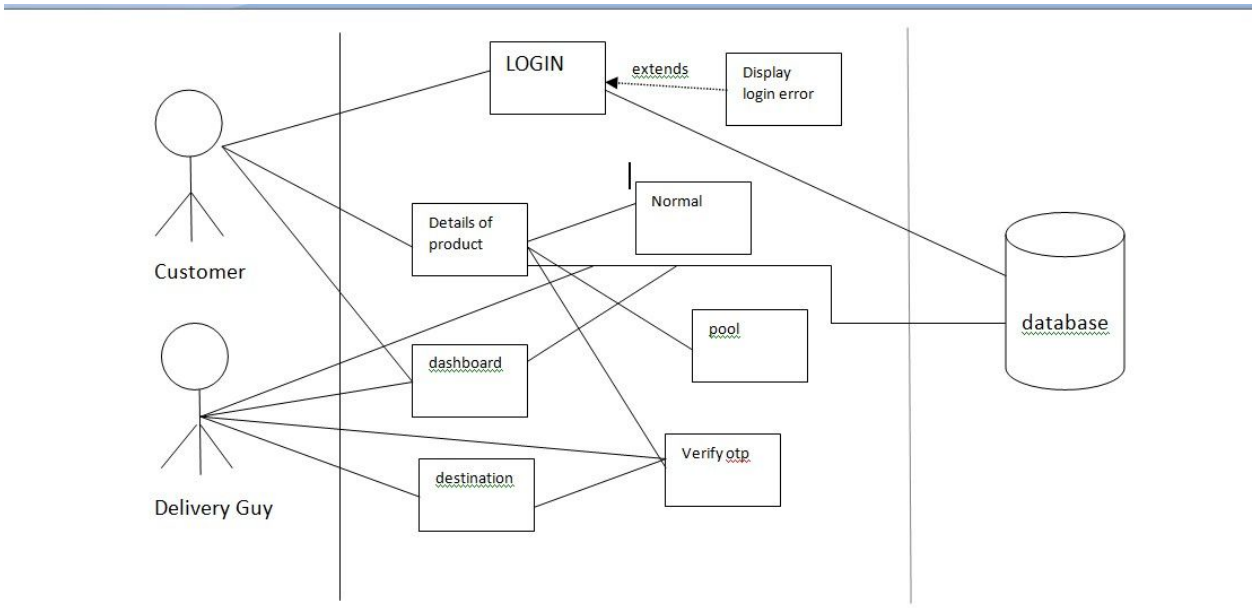
Module - 8:

User Statistics :

- An Overall Statistics of app like how many orders are created, ongoing, delivered.For delivery guys we will show number of orders delivered by him(Monthly wise)

Status : Completed (User Can See Their Monthly Usage of the Application with Neat Visualizations .)

Use Case Diagram :

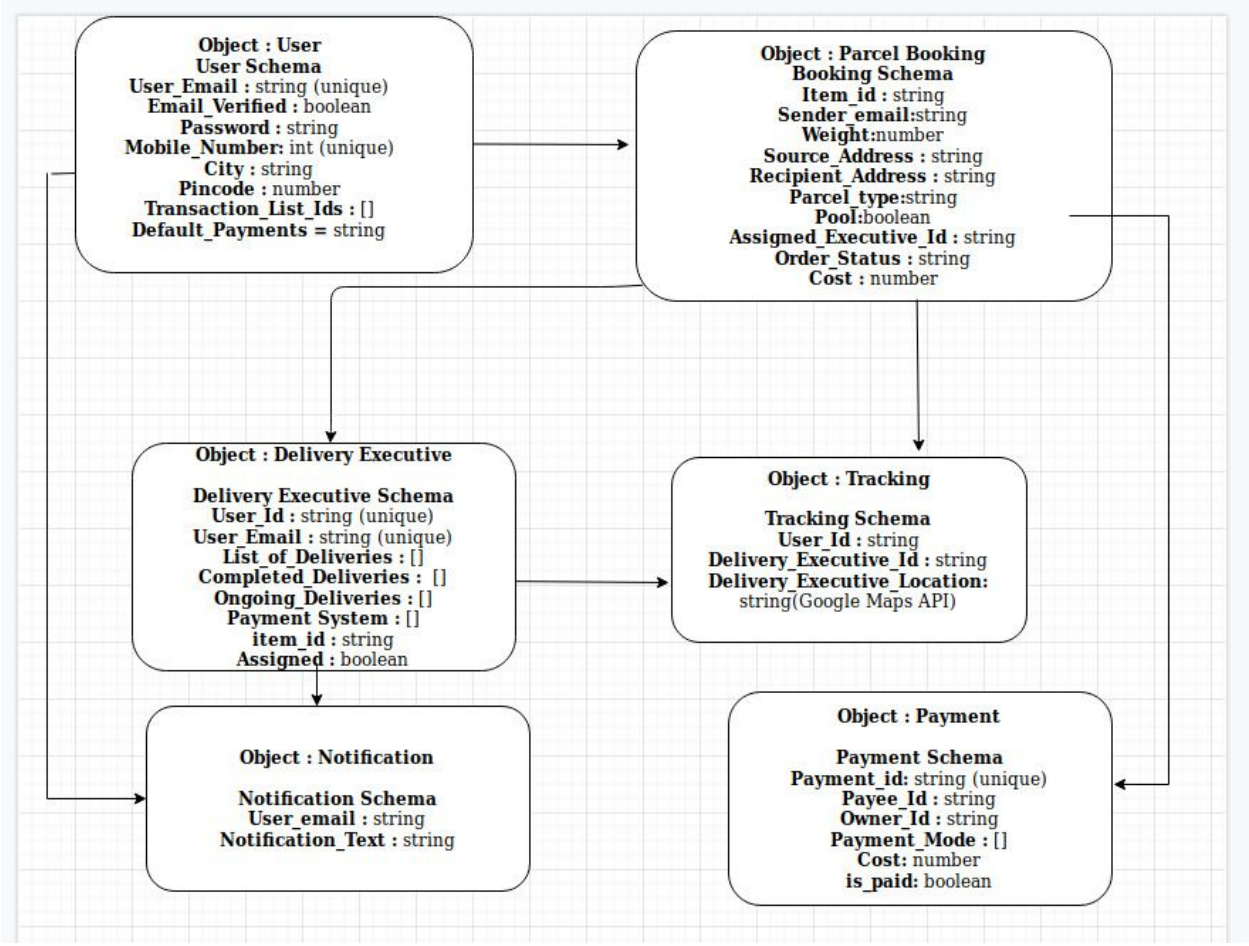


Database :

Mongo DB with NODE JS Backend

NO SQL Database

Database Schema Diagram :



Please Click Here for Clarified Diagram

Details of The Attributes :

User Schema :

- **user_id :** *String, Unique* → A Unique String is Created For Every user.
- **email :** *String, Unique* → Stores the email of the user.
- **email_verified :** *Boolean* → Tells if Email is verified or not
- **password :** *String*
- **Mobile_number :** *Number , Unique* → Acquire Mobile Number of User.
- **City :** *String* → Acquire the City of the User.
- **Pincode :** *Number* → Get the Pincode of the User's City.
- **Transaction_list_id :** *List* → This Stores All The Transactions of that particular user.
- **Default_payments :** *String* → Acquire the Default Payment of the user.

Delivery Executive Schema :

- **email :** *String, Unique* → Stores the email of the Delivery Executive.
- **Deliveries_list :** *List* → This Stores All The Deliveries of that particular Executive.

- **Completed_deliveries :** *List* → This Stores All The Deliveries that are Completed.
- **Ongoing_deliveries :** *List* → This Stores All The Deliveries that are ongoing and yet to complete.
- **Assigned :** *Boolean* → It Determines the Status of the Delivery executive.

Notification Schema:

- **email :** *String*→ Stores the email of the user.
- **notification_text :** *String* → Stores the text of the Notification.

Parcel Booking Schema :

- **Item_id :** *String* → Stores the ID of the Item
- **sender_email :** *String*→ Stores the email of the parcel sender.
- **weight :** *Number* → Stores the Weight of the Parcel
- **recipient_address :** *String* → It Stores the Address of the Receiver
- **Parcel_type :** *String* → It Stores the Type of Parcel that a user wants to send.
- **Pool :** *Boolean* → Tags the Orders as Pool deliveries or Fast Deliveries
- **Assigned_executive_email :** *String* → Stores the Mail of Delivery Executive .

- **Order_status :** *String* → Tells the Status of Order.
- **Cost :** *Number* → It stores the cost of parcel based on Weight.

Tracking Schema :

- **user_email :** *String*→ Stores the email of the user.
- **delivery_ex_email :** *String, Unique* → Stores the email of the Delivery Executive.
- **delivery_ex_location :** *String* → GOOGLE MAPS API

Payment Schema :

- **payment_id :** *String , Unique* →Unique ID for every payment
- **Payee_mail :** *String* → Gives the Email ID of the Payee
- **Is_paid :** *Boolean* → It Tells About the Status of Payment

Github Link for The Code : Github