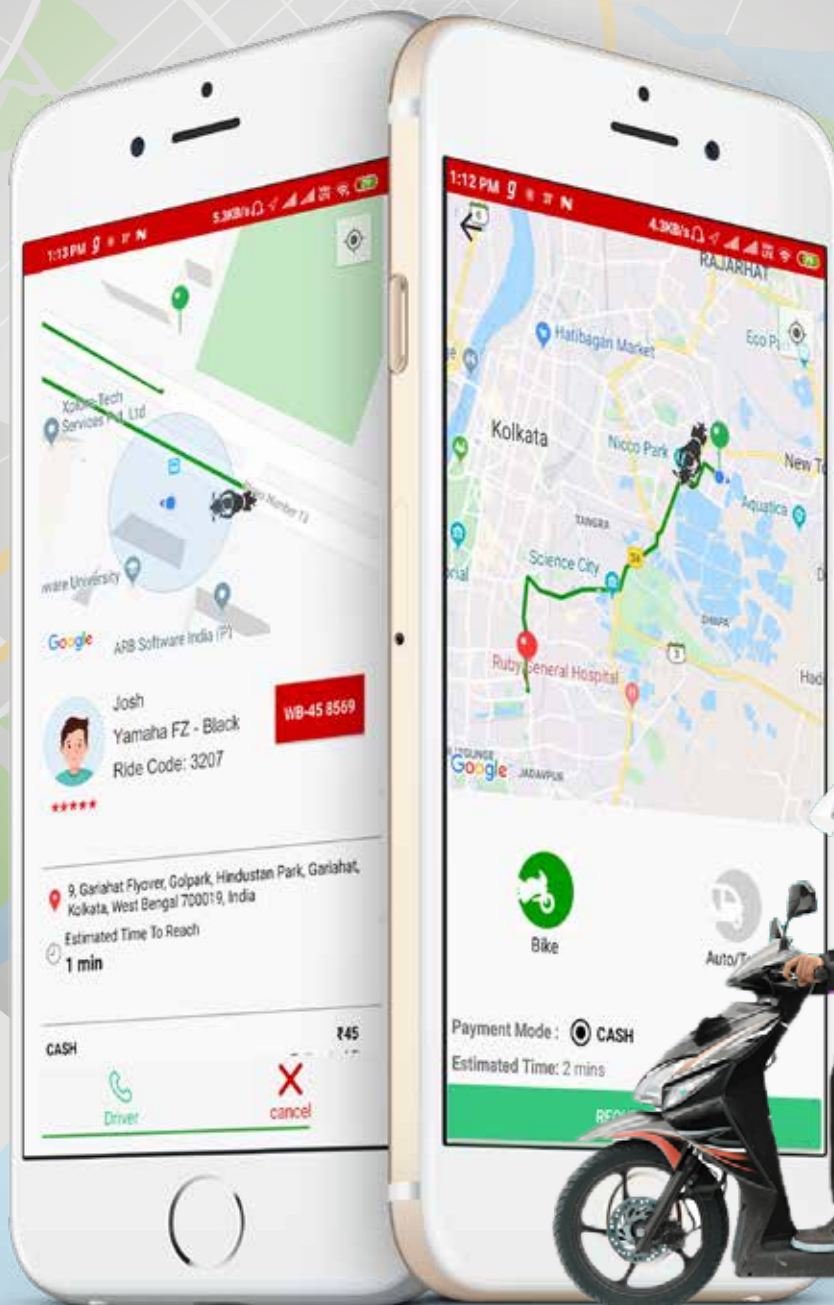




Case Study For Bike Ride Booking Mobile App

How Webguru Infosystems developed an interactive, architecturally sound, easy-to-use, and cost-effective bike-to ride booking app system for passengers/ commuters in the districts of Purulia, Bankura, and Hooghly in West Bengal.



An Overview

Transportation is a major challenge/requirement for people residing in the districts of Bankura, Purulia, and Hooghly in West Bengal, India. Sensing the requirement, EASY2WAY Bike Ride Private Limited (the client) thought of introducing a mobile app-based platform for passengers in the above-mentioned districts to avail a bike or toto service and escape the ordeal of waiting endlessly in the queues for buses. Besides, the app platform gives the owners of bikes or totos an earning opportunity. As an experienced digital enterprise with the technical expertise to develop both native and hybrid apps for various industry segments, Webguru Infosystems got into the act and delivered the ride-booking app system to the satisfaction of the client.



The Proposal

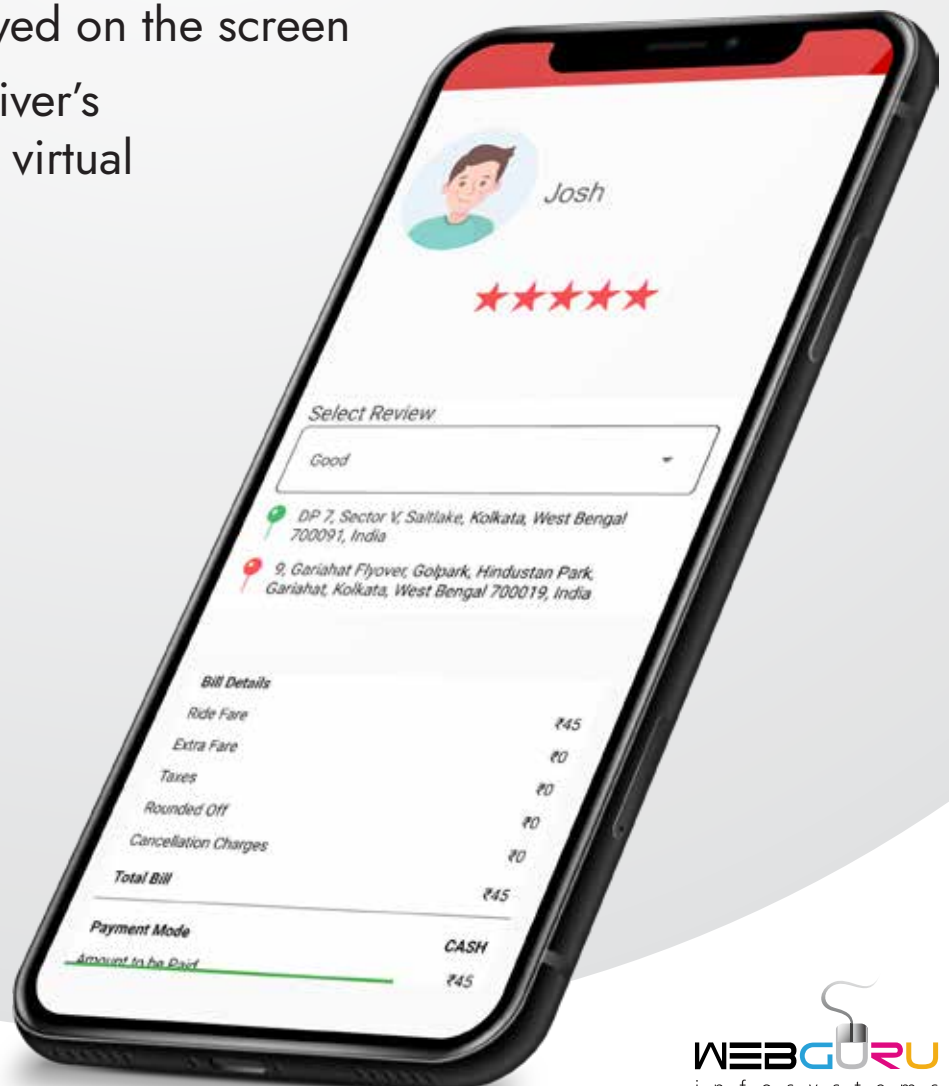
The proposed bike/toto ride booking app is to allow the passengers to book a bike or toto (a 3-wheeler battery operated vehicle) for taking a ride from a pickup location to a drop-off location.

The bike/toto ride booking system to contain the following components:

- Android mobile app for passengers aka Bob Ride
- Android mobile app for drivers aka Bob Rider
- Web admin panel for the system owner

The key features of the ride-booking system

- One-click login
- GPS tracking
- Estimated fares displayed on the screen
- A quick view of the driver's balance amount in the virtual wallet, anytime
- Ratings and reviews



The Workflow

At the outset, the driver needs to register for the app-based system by submitting a few documents such as driver's license, insurance, RC book, etc., followed by the system admin approving the registration after verifying the documents from the backend. Thereafter, the workflow of the ride-booking system is described as under:

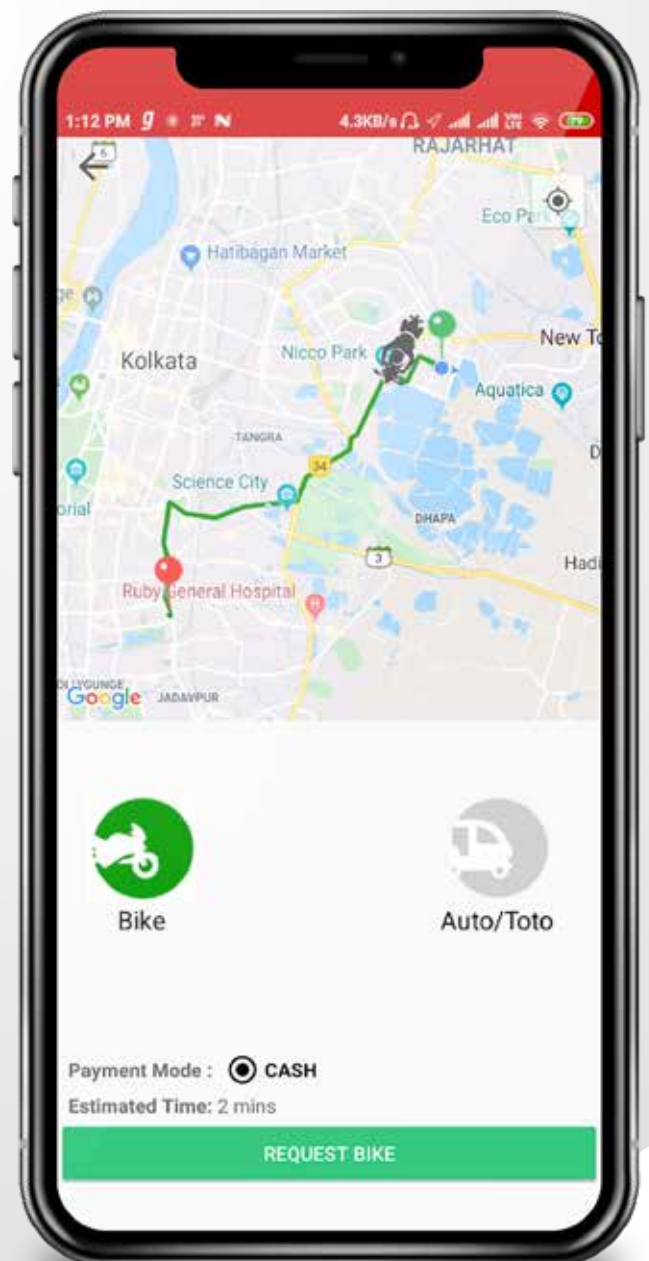
- Once a passenger enters the destination, his or her pick-up location is taken using GPS or entered manually by the passenger from Google's Location List. Thereafter, the passenger has to enter the type of vehicle he or she wants to travel in, say a bike or toto. And if it is a toto, then he or she has to mention the number of seats to be booked.
- Upon receiving the booking notification, the concerned driver needs to accept that booking request within a specific time period (max 30 seconds), or else the request gets forwarded to the next driver in the vicinity. In this manner, a booking request can be forwarded to a maximum of 3 drivers (if found within a certain radius from the pickup location), that is, if the first or the 2nd driver doesn't accept the booking request.

Note 1 : To get the booking request notification, the driver needs to maintain a minimum balance amount paid to the system admin beforehand.

Note 2 : If none of the drivers accepts the booking request, the same gets dropped and the passenger is notified. In such a case, the passenger needs to submit the booking request again.



- After accepting the booking request from the passenger, the driver specifies the fare amount from a drop-down menu in the Bob Rider app (driver's app) and the fare amount gets displayed on the Bob Ride app (passenger's app).
- Once the passenger accepts the fare, he or she receives an OTP, which needs to be given to the driver at the time of taking the ride.
- Once a booking request gets confirmed, the passenger is notified with the bike or toto details (make, model, color, and registration number), fare and the tentative time to arrive at the pickup point. Also, the contact information of the passenger and the driver gets exchanged via push notification, and becomes visible in the respective apps.
- The driver's geo-location (as fetched by the driver's app from the location service of the Android mobile phone) is shared so that the passenger is able to track the driver on the Google Map in his or her mobile app.
- The user can cancel the ride within 2 minutes of booking or before the driver arrives at the pickup location, whichever is earlier, else a cancellation fee gets levied, which the user needs to pay along with the fare for the next ride.
- Once the trip is completed, the driver marks 'End Trip' from his / her mobile app, and accordingly the status of the trip gets updated and displayed on the driver app, passenger app, and the system admin. The passenger can make the payment in cash to the driver.



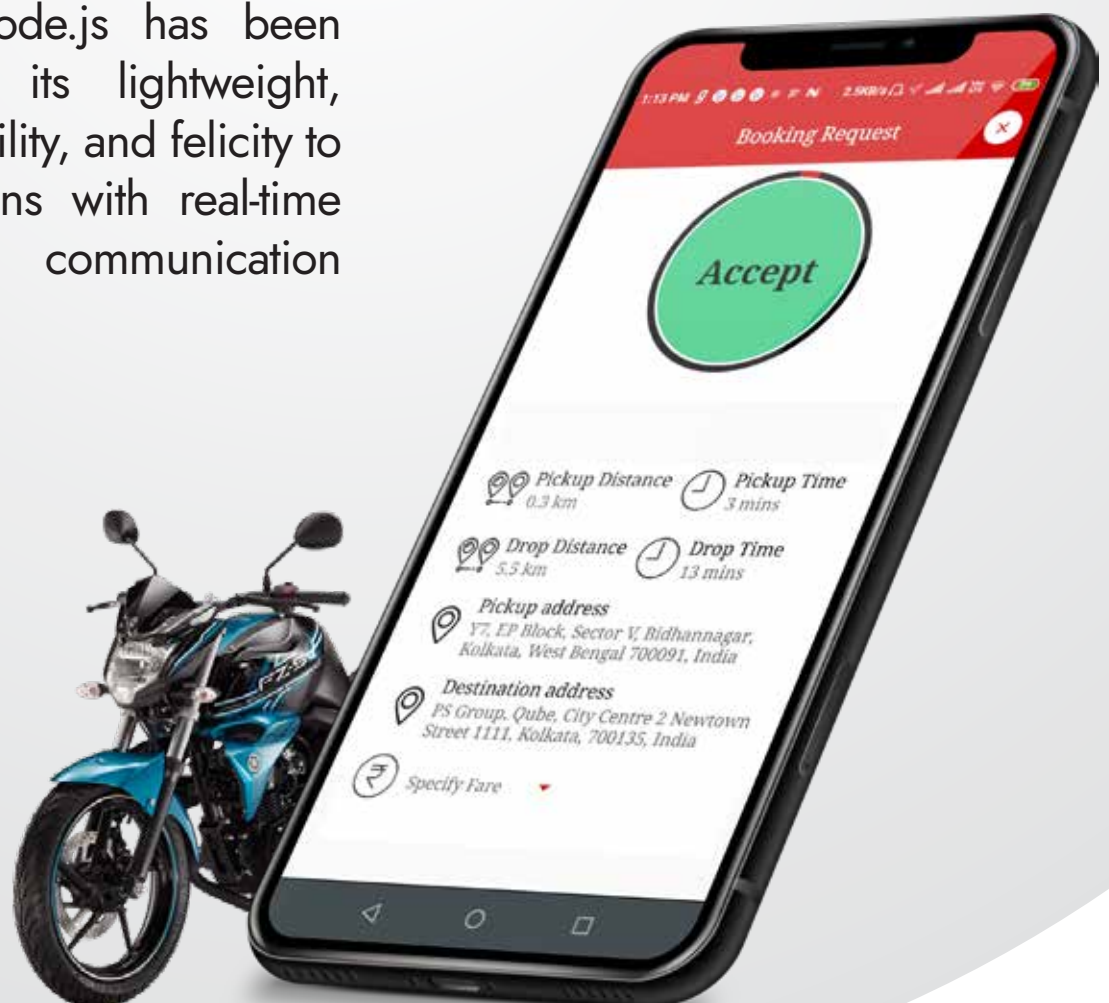
The Development Process

The ride-booking system has been built in two phases as described below:

Phase I

We built the two mobile apps (Bob Ride and Bob Rider) on the Java-based Android native platform as people in the three rural districts of West Bengal (Bankura, Purulia, and Hooghly) are most likely to have Android-based smartphones. The technology stack to build the backend web panel has been Node.js and MongoDB.

MongoDB has been chosen to build the app's database as it is better suited for projects with a high rate of I/O operations. The open-source Node.js has been used due to its lightweight, flexibility, scalability, and felicity to build applications with real-time navigation and communication features.



Further, as Node.js fills the gap between frontend and backend applications by using the non-blocking I/O approach, it significantly increases the efficiency of the development process. According to the non-blocking I/O approach, more than one passenger requests can be initiated in parallel without having to respond to the request of the first passenger. The admin backend web panel manages the backend settings, passenger requests, drivers, view the activities of drivers and passengers and generate reports.

The frontend app for the driver (Bob Rider) to have a virtual wallet wherein he or she has to keep a predetermined amount (to be fixed by the system owner). And for every ride completed, the amount stored in the virtual wallet to get deducted by a specified amount as platform charge – to be fixed/managed by the system owner. To enable the passengers and drivers to track each other's location, we used Google Maps Directions API for route configuration, Google Location Services API to track the respective locations and Pubnub to share live locations across the system modules.



The various features of the app (both driver and passenger) are as follows:

Registration: The app system allows registration (signup and login) for the passenger and driver by using social media or a mobile number/password. As an additional requirement, the profile info of the driver is required to be verified and authenticated by the system admin.

Profile editing: Both passengers and drivers can edit their profiles containing photos and info such as name, phone number, email, etc. Also, the driver's app should have information about the vehicle and the driver's license. Both the driver and passenger can view each other's profiles to inculcate a sense of security.

Displaying the cost of a ride: The passenger should know the estimated fare beforehand as specified by the driver.

Ratings and Reviews: Both passengers and drivers can rate each other and give reviews on the experience once the trip is over. The reviews help the admin to understand the strengths and challenges associated with the service and make suitable improvements.

Booking and payments history: The drivers and passengers can view details about the previous trips; date, cost, time, name of driver/passenger, pickup and drop locations, ratings, etc.



Phase II

The client came with a requirement to send a voice message alert to the driver if he or she is found to be offline when a passenger makes a request. However, according to the latest Android versions (from Oreo onwards), tracking of a user's location (or any other activity) from the backend when the user is offline has been done away with due to security reasons. So, we suggested to send a message to all the drivers (irrespective of their app being offline or online) within a certain distance and based on their last known location when a passenger request is generated.

Similarly, a message to be sent to the passenger making a booking request informing him or her about the unavailability of any driver in the vicinity at that moment. The passenger to be alerted through a message notification subsequently if a driver near the pickup location comes online within a specified period.

Conclusion

The ride-booking app system has been made operational after a thorough testing exercise and in accordance with the business specifications as laid down by the client. We hope the app, in some way, will help to solve the transportation woes of people living in the districts of Bankura, Purulia, and Hoogly, and generate income opportunities for the drivers of bikes and totos. We thank the client for the constant support and guidance provided to us during the planning, development, and testing phases.



Want To Develop A
Custom Application
For Your Business?

Talk to Our Experts



An ISO 9001:2015 Certified Company

www.webguru-india.com