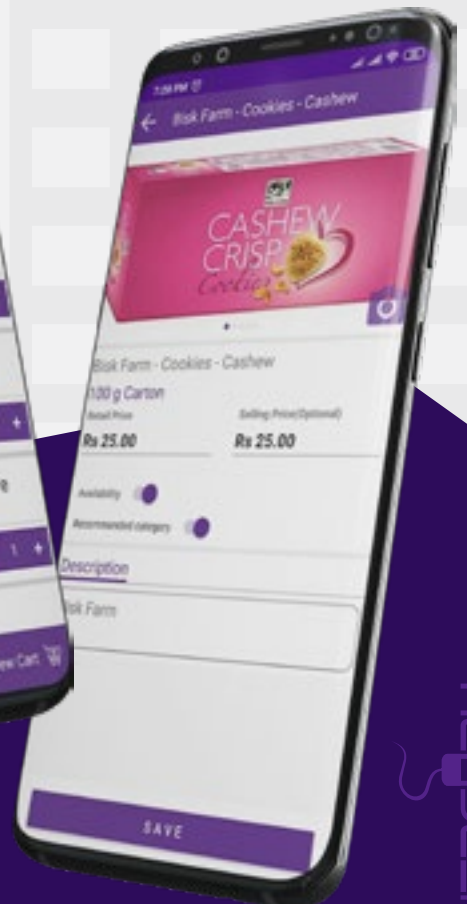
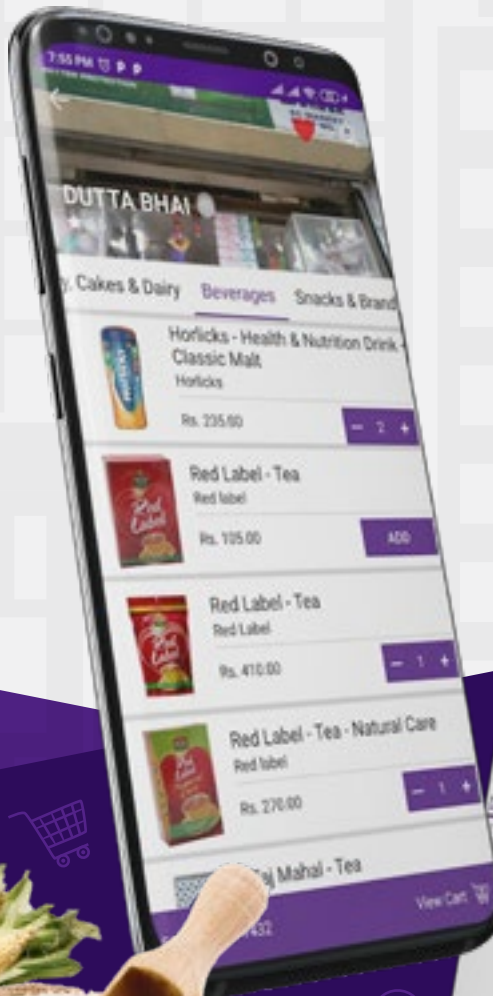




How Webguru Infosystems developed and implemented an Item Order and Delivery App (InstaGin) based system for the start-up Raymit Innovations Pvt. Ltd. The app system offered customers a reliable and secure alternative online marketplace to source their choicest items and get them delivered to their doorstep.

- A Case Study -



An Overview

Mobile apps have become popular tools in the hands of users to receive a host of products or services on the internet. One of the business segments where mobile apps have helped consumers a great deal is eCommerce. Our client, Raymit Innovations Pvt. Ltd., had approached us to develop and implement a mobile app for Item Order and Delivery in the B2B and B2C segments. With the help of this mobile app, the client wanted to reach out to its target customers for the delivery of a range of items such as food, grocery, medicines, garments, sports equipment and accessories, FMCG products, and electronic devices or gadgets, among others. The app allows customers to choose the products of their choice from select stores and notifies the store owners of the purchase. The store owners use the app to connect to the drivers and let them deliver the products to the customers. The system admin manages the entire transactions across the value chain.



The Proposal

Initially, the client had approached us with the requirement to develop the following digital touchpoints for the app users.

- **Item Order Mobile Application (for customers)**
- **Item Delivery Mobile Application (for drivers)**
- **Shop Management Mobile Application (for retailers)**
- **Web-based Panel for System Admin**

As the applications were to be used in India, we suggested building native apps on the Android operating system given its wide market share of more than 95% (Source: gs.statcounter.com.) Our suggestion of building native apps vis-à-vis hybrid apps was based on the observation that the target app users are very choosy about quality parameters like performance, security, features, and usability. They are likely to discard the app suite and settle for a competitor if the former shows any issues related to the above-mentioned quality parameters.

Further, native apps offer the best performance vis-à-vis their hybrid counterparts as they are built for a specific platform and are compiled by the core programming language and APIs of the platform. Also, since the native apps can leverage the device's processing speed given that they don't need any separate interpreter layer (like browser component) to be rendered.

The data protection features of a native app are way ahead than that of a hybrid app. This is due to the fact that hybrid apps use browsers as the interface. There were multiple rounds of discussions with the client to arrive at the features and other requisites for the app suite.

We chose Razorpay as the payment gateway as it has a robust security mechanism – PCI DSS Level 1 compliant, frequent third-party audits, and a dedicated security team to ensure the safety of data. Besides, Razorpay supports a wide array of payment methods such as Debit Card, Credit Card, Net Banking, UPI, EMI, etc.

The Workflow

The Item Order and Delivery app is designed to handle orders from registered customers for purchasing items from various stores (registered with the app) to the customers' locations. The customers can select a store, view the items displayed in that store, and add the item(s) to the cart for purchase. Thereafter, an order is placed by making the online payment.

Once an order is placed, it is viewed by the particular store owner and the system admin of the app. Accordingly, the store owner sends a random delivery notification to the driver (registered with the system) present in the vicinity of the store. Upon accepting the notification request, the respective driver visits the store, collects the item(s), and delivers the same to the customer.

The registered customer can specify a store (not registered with the app system) by choosing its location on the Google map and send a system driven notification to the driver to deliver the items purchased from that store.

The registered store owner can specify a customer (not registered with the app system) by choosing his or her location on the Google map and send a delivery notification to the driver.

In the web admin panel, the system admin will be able to view the orders and payments received and shall reimburse the share of the store and driver by means, which is outside the scope of this case study.



Core Features

- Customers get the list of nearby stores (based on the type of item they are looking for) in the proposed mobile app.
- Clicking on a store will display the items.
- Customers can select items to view details like brand, colour, size, price, dimensions, etc.
- Customers can specify the number of items to be purchased and add the same to the cart.
- Customers can even specify a store (not registered with the app) on Google map and notify the driver to deliver the items purchased from the store.
- Based on the distance (as fetched from Google APIs) between the delivery address specified by the customer and the address of the store, the delivery charge will be calculated and displayed in the app. The delivery charge will be added to the order total to be paid by the customer online.
- Once the order is placed successfully, the concerned store owner, as well as the system admin, will be able to view the same on their mobile apps.
- The store owner will submit a delivery request and the driver (randomly chosen by the system) present nearby to the store will receive a notification about the delivery request.



- The store owner can specify a customer (not registered with the app system) and notify the driver to deliver the items.
- Upon receiving the delivery notification, the concerned driver can accept the same within a specific time period (30 seconds maximum,) or else the request will be forwarded to the next driver nearby.
- This way, the delivery request will be forwarded to a maximum of 3 drivers found within a certain radius from the pickup location. If none of the drivers accepts the request, the request will be dropped and the store owner will be notified. In such a case, the store owner will be required to submit the delivery request again.
- Upon accepting the delivery request notification in the mobile app, the driver can view the location of the store and the customer on Google Maps integrated into his mobile app.
- The driver visits the store, picks up the ordered item(s), and delivers it to the customer.
- The store can update the order status (such as picking up the item and on the way to the customer's location), which the customer(s) can view on his or her mobile app.
- The customer can view the status update of his or her order as to whether the order has been picked up from the store and is on the way for delivery.
- Upon delivery, the driver updates the order status as 'delivered.'
- The customer can provide ratings and reviews to both the store and the driver.
- The system admin can register the stores and drivers from the web-based admin panel.
- The store owners can add the items to the app.
- The system admin can view the orders, amount, and order status. All online payments received from the customers with respect to their orders are deposited into the system owner's account.
- The system admin reimburses the respective share of the order amount to the stores and the respective driver.

The Process

At the outset, we wanted to get into the minds of the end-users and understand their pain points while shopping online. We found the following pain points.

- The UI was often confusing with no clear CTAs
- Checkout and payment processes were difficult to understand



Since the client wanted the app suite to be robust, dynamic, secure, navigable, and responsive, we suggested the frontend be made using the Android native platform for its range of robust security features. So, we went ahead with designing the initial wireframe structure to be vetted by the client using the native APIs of Android SDK in Java programming language.

To manage the three apps, we needed a robust admin panel at the backend in the form of a web portal. This was developed using PHP, a powerful server-side scripting language known for being reliable, fast, and secure. PHP can also be used to create dynamic web pages, which can interact with databases. Further, since PHP is an open-source language, it helped in bringing down the cost of development as well.

PostgreSQL was chosen as the open-source relational database system to build the database at the backend. PostgreSQL was chosen over MySQL as the Item Order and Delivery app is needed to manage a voluminous quantum of data comprising a multitude of items.

In the beginning, the basic design of the three apps and the structure of the database and admin panel were made. Once they were approved by the client, we prepared a wireframe with dummy data to be vetted by the client. The dummy data gave a glimpse into the appearance of the app suite. Thereafter, the task of API (Application Programming Interface) creation for the backend using PHP began to make the app dynamic. The API would serve as the interface to facilitate interactions between multiple software modules, features, or even third-party software.

When the app suite reached a certain threshold level (say 60%,) we began working on the admin panel. Once the three apps and admin panel were completed, the entire app suite was tested comprehensively. This activity was conducted with the involvement of the client. During testing the business logic surrounding the app suite changed/evolved leading to the addition of a few features like the way to change the customer address. Also, multiple admin users were introduced whereby changes made by each admin member were logged.

This was important from the perspective of transparency as every admin member would be able to see the type (and time) of change made in the system. The changes could be adding, editing, or deleting a store, driver, or amount to wallet among others. This feature eliminated the prospect of any arbitrary changes, which could be made by an individual admin member.

Conclusion

The Item Order and Delivery app suite “InstaGin” was developed, tested, and made live in accordance with the objectives of the client. The app suite comprising three apps (for the customer, store, and delivery personnel) has been made seamless, secure, dynamic, and attractive for every stakeholder in the value chain. We thank our client, Raymit Innovations Private Limited for the constant support and guidance.