
GitHub Username: mrgupta20

Quech Delivery Provider

Description

This app is to support goods suppliers in managing orders. Orders can be fetched from the backend using rest api. Payment status and order status can be changed from the app.

Supplier can sign up and sign in to the app. Supplier can add detail about them while signing up.

Intended User

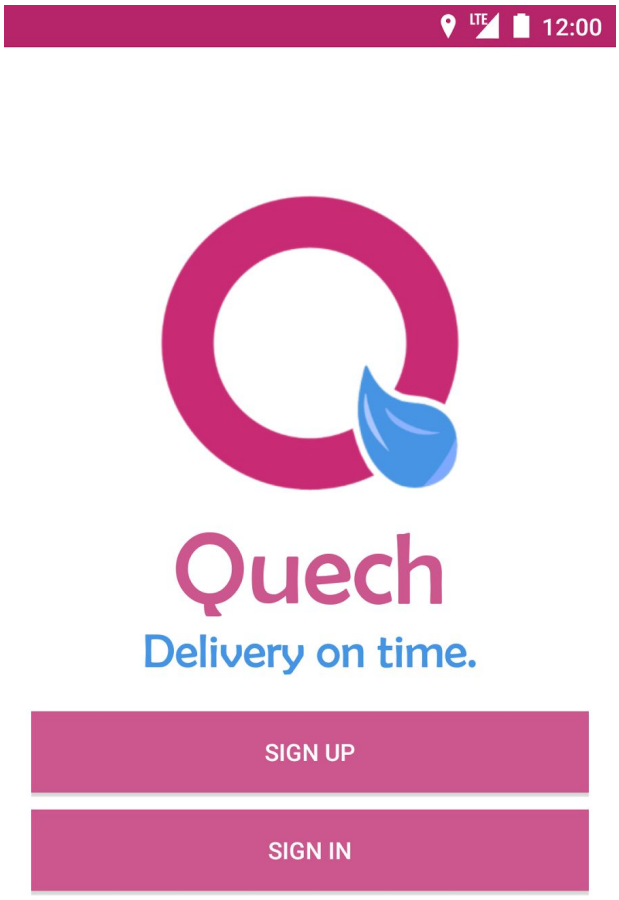
This app is for goods suppliers.

Features

- Sign up - Add personal/store details
- Sign in, Retrieve forgotten password
- Fetch orders from the backend server
- Display list of orders category wise
- Change the status of order

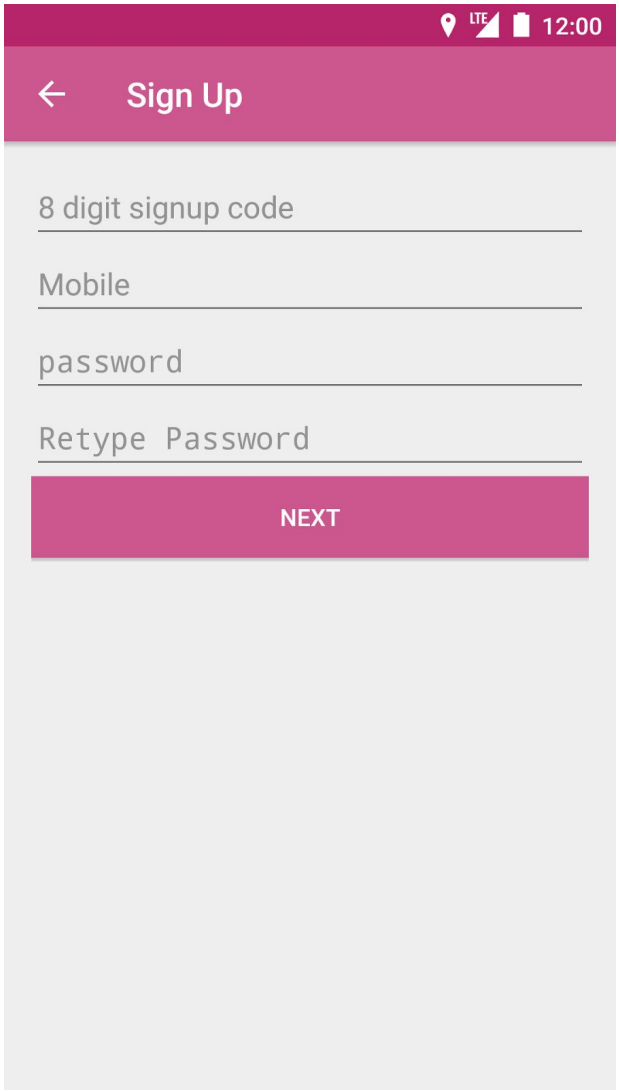
User Interface Mocks

Screen 1



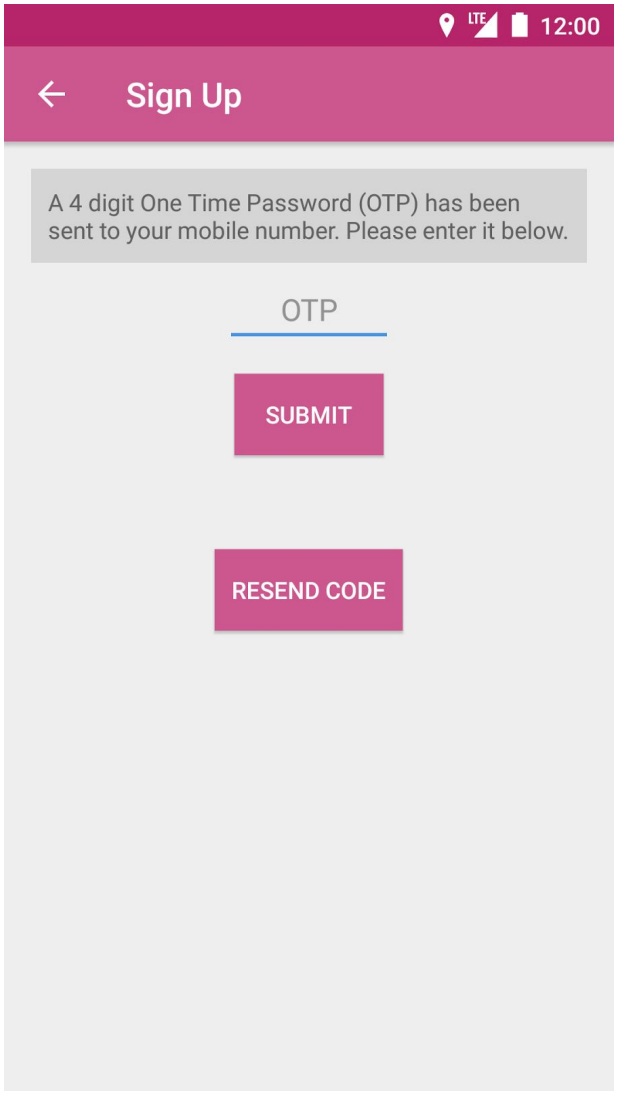
Welcome Screen. From here user can either Sign up or sign in to use the app.

Screen 2



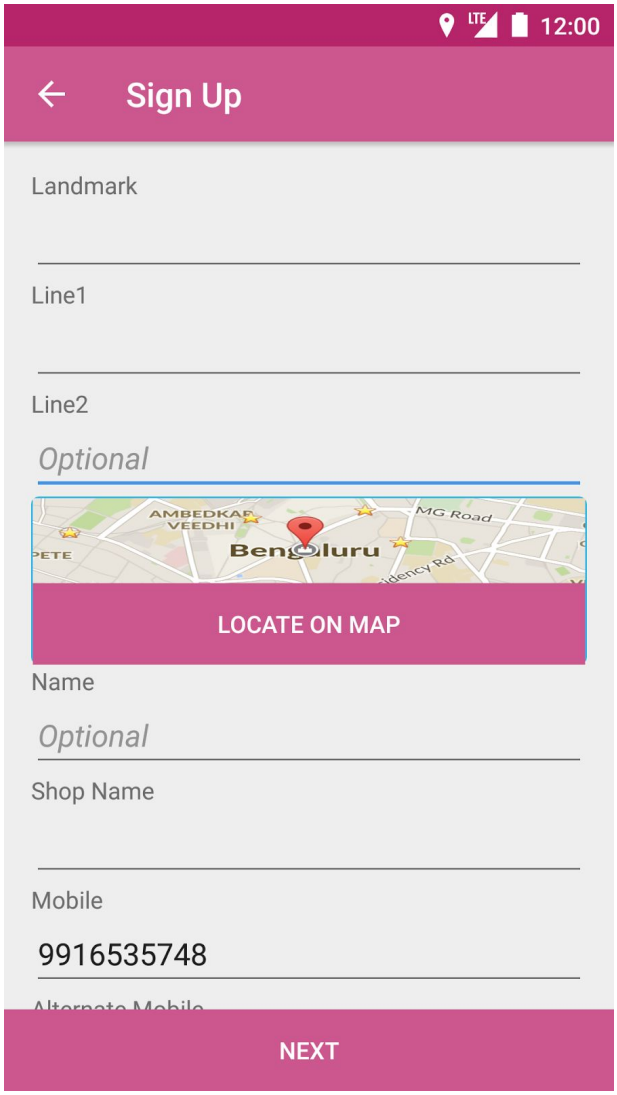
Sign Up Screen - Main Page

Screen 3



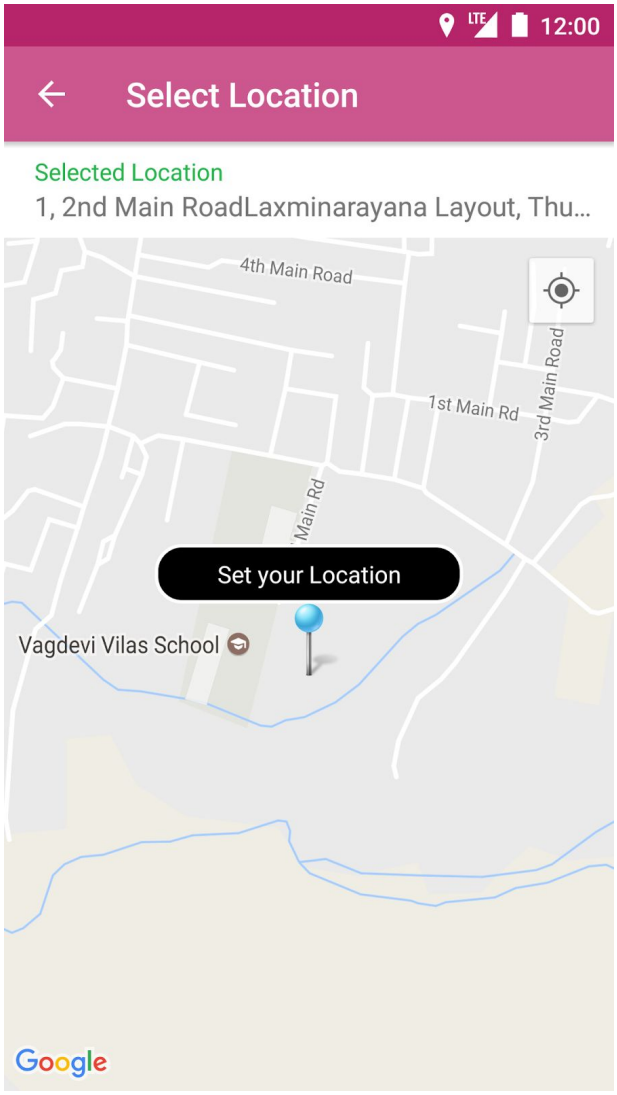
Sign Up Screen - OTP

Screen 4



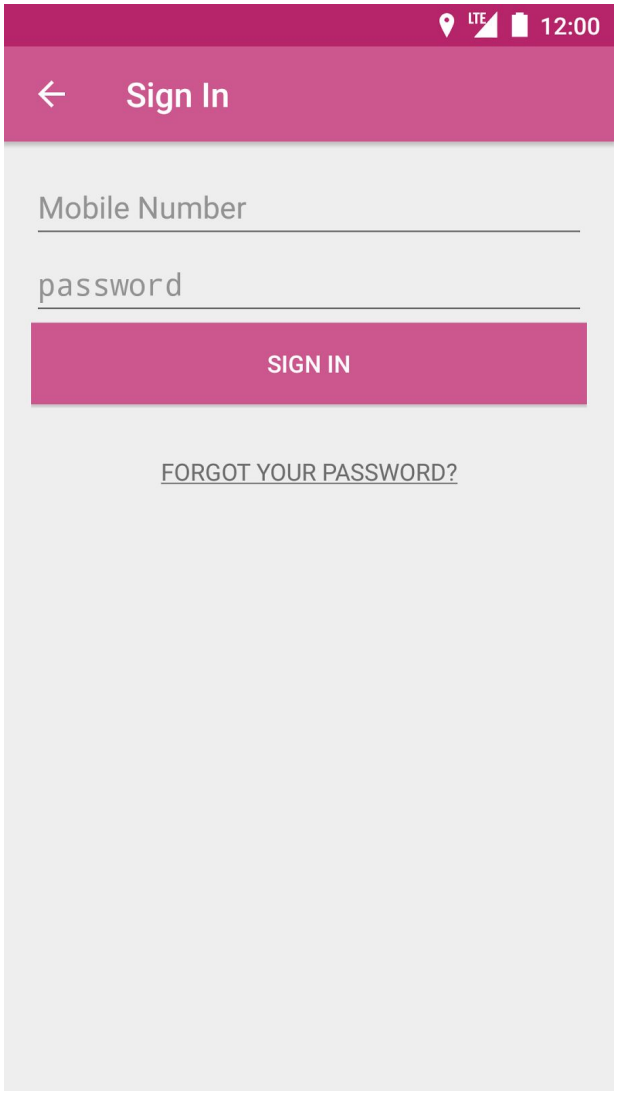
Sign Up Screen - Add personal details

Screen 5



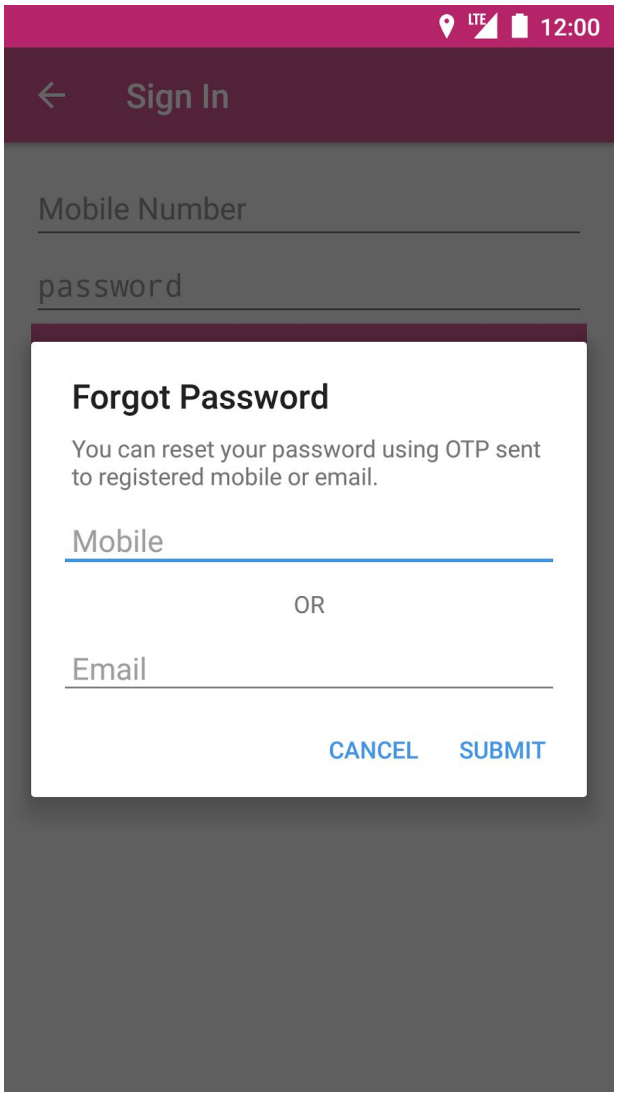
Sign Up Screen - Add GPS location

Screen 6



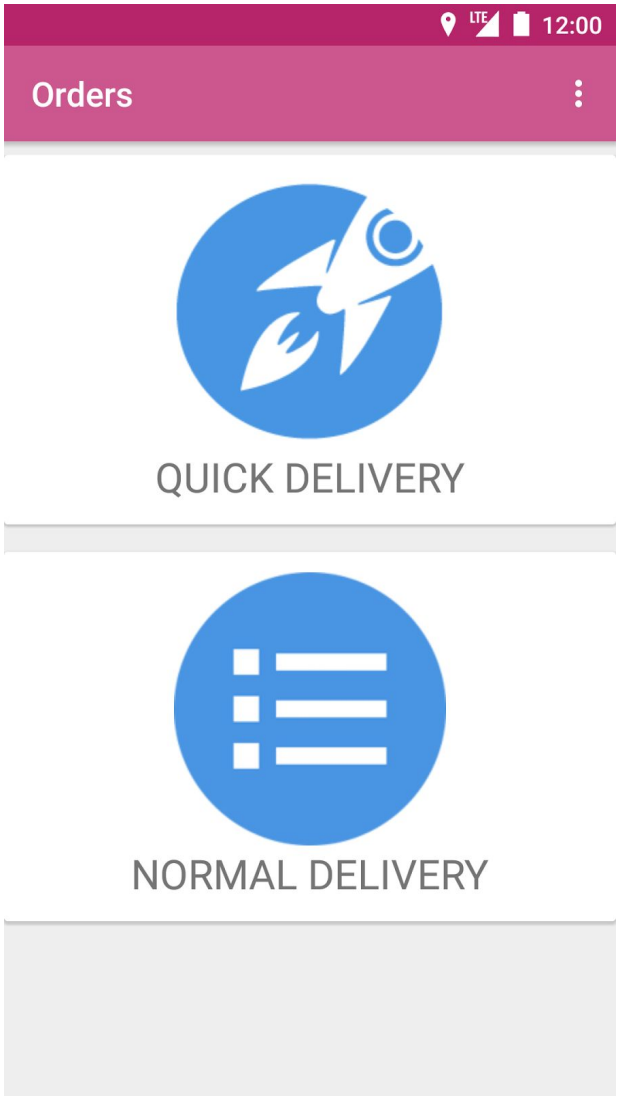
Sign In Screen - Main Page

Screen 7



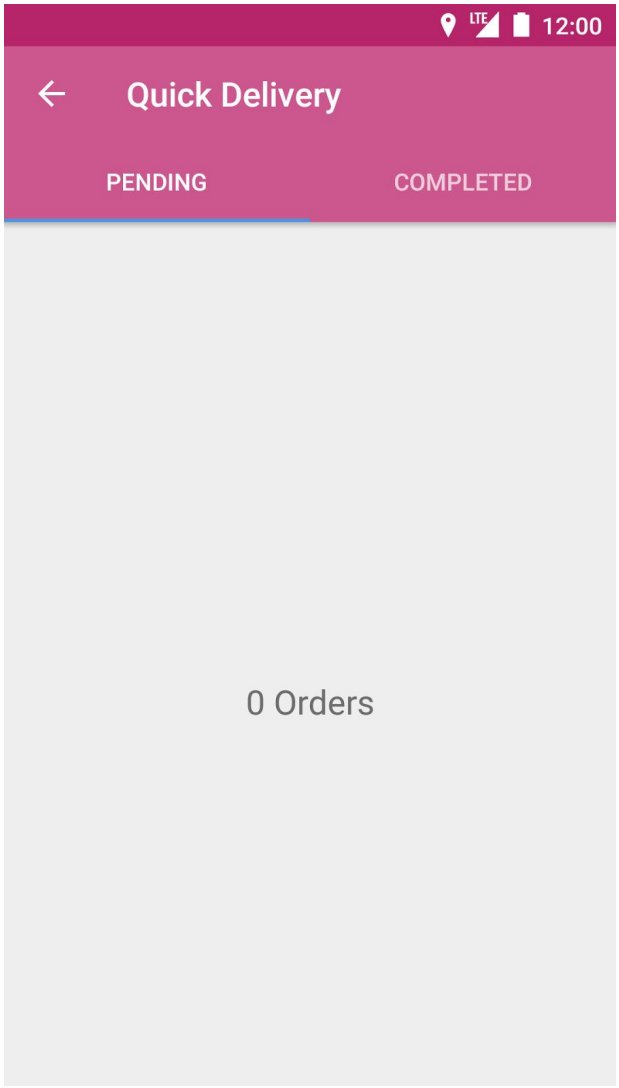
Sign In Screen - Forgot Password

Screen 8



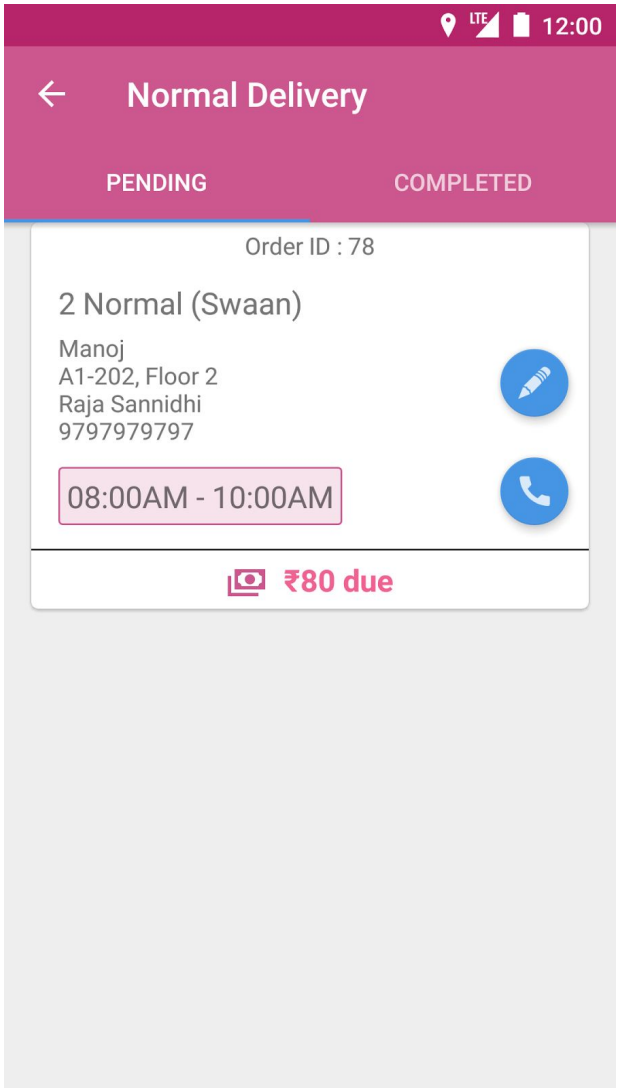
Order Screen - Main Page (Category tiles)

Screen 9



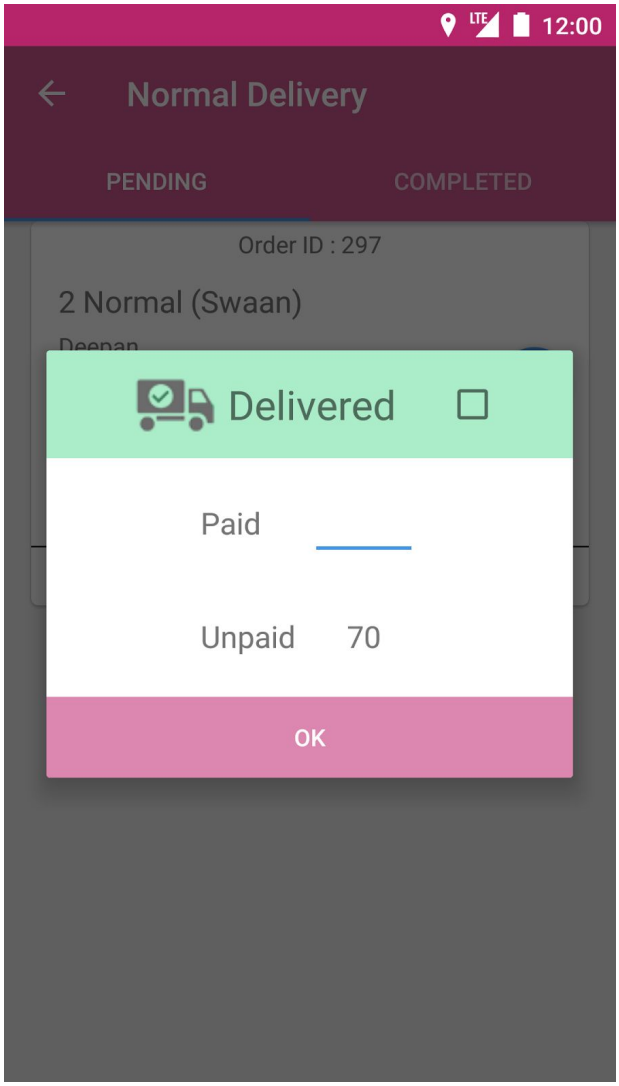
Order Screen - Empty Order List

Screen 10



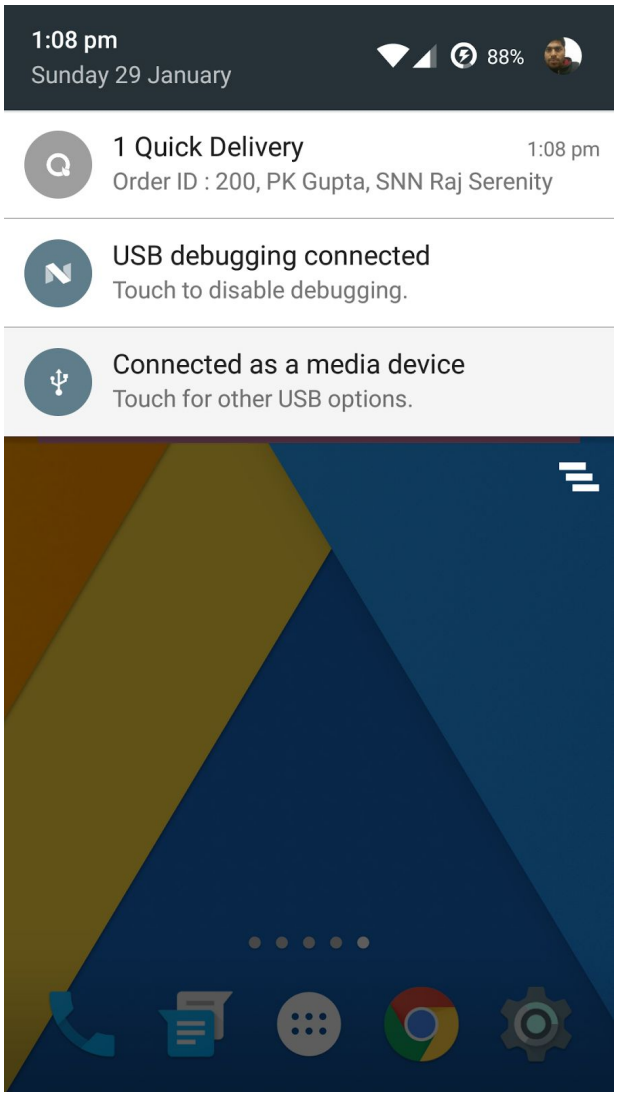
Order Screen - Order List

Screen 11



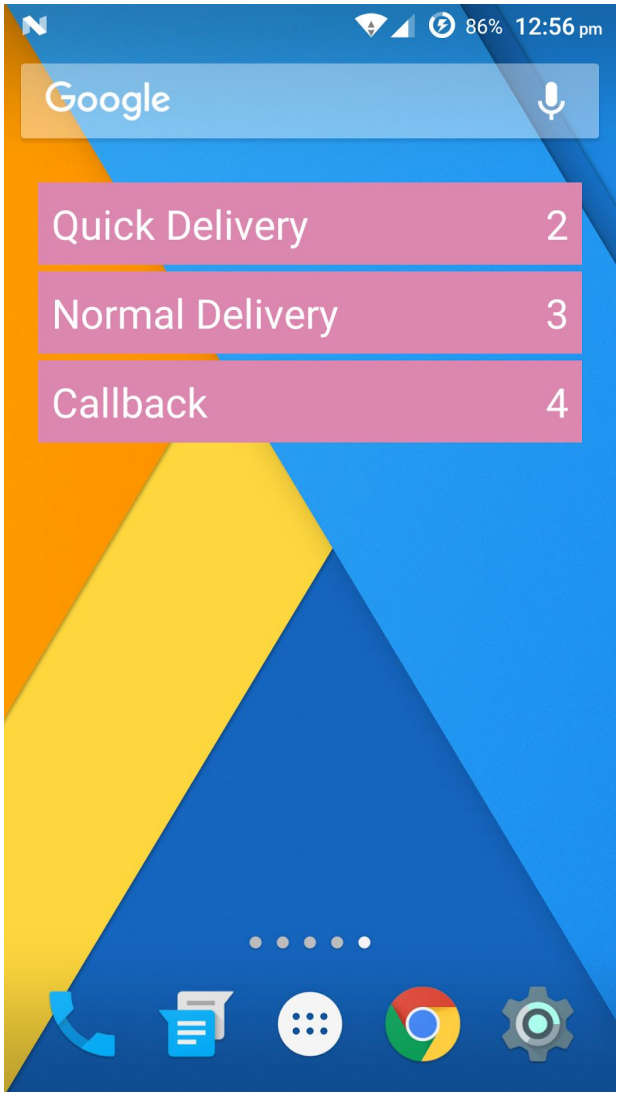
Order Screen - Status Update

Screen 12



Order Notification

Screen 13



App Widget

Key Considerations

How will your app handle data persistence?

1. 1 day of order details will be saved using content provider. It will be used to display the orders when network is not available.

2. When signed in, API Authorization token will be saved in the sharedpreferences to authorize the web api calls.

Describe any corner cases in the UX.

How can i sign in as different user?
To sign in as different user, You have to click on logout button at the order screen home.

Where can i enter details about payment in the order?
While changing status of order, a pop-up opens asking details about paid and unpaid amount. By default all amount is shown as paid. It can be adjusted and saved.

Describe any libraries you'll be using and share your reasoning for including them.

For example, Picasso or Glide to handle the loading and caching of images.

1. Retrofit - It is being used for consuming the REST Api. It provides in house support for parsing JSON and is fast and easy to use.
2. Calligraphy - It is being used to easily implement the custom fonts in the applications.
3. Eventbus - It is being used to pass the events between different android components. It makes easy to implement instead of using interfaces which makes it messy.

Describe how you will implement Google Play Services.

Describe which Google Play Services you will use and how.
Google map will be used to store the supplier location. While signing up, supplier has to select his current location in the map. An Activity containing google map will be opened, which will point to the current location. Though other location can be selected manually.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Choose primary, secondary and other colors.
- Required configurations to be handled by the application

- Configure libraries
- Set up the build variants and product flavors

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Account Home Screen
- Build UI for Sign Up Home Screen
- Build UI for Add Details Screen
- Build UI for Map Activity
- Build UI for Add Items screen
- Build Ui for Sign In Screen
- Build UI for OTP Screen
- Build UI for Forgot Password Screen
- Build UI for Order Home Screen
- Build UI for Empty Order List
- Build UI for Order List Item
- Build UI for Order Status Change Screen

Task 3: Add Maps functionality

In the maps activity, load the current location and show in the center of the screen.

- User can move the pointer to anywhere in the map and select the location
- User can click on a button to move to the current location
- A textbox is displayed to show the selected location address details.

Task 4: Network Handling

Create base classes for handling networking.

- Connect to the main endpoint
- Handle any errors

Task 5: Add Widget

Add widget for the application.

- Design layout for the widget

- Update data in the widget

Task 6: Add Notification

User should get notification for new orders if he is not on the app screen. Tapping on the notification opens the order list screen.

- Set up the notification categories
- Handle the notification click actions

Task 7: Add Content Provider

Create content provider .

- Connect to the main endpoint
- Handle any errors

Task 8: Connect the UI Flow

Make the navigation of app user friendly .

- Sign Up to order status update
- Sign In to order status update
