GitHub Username: sabkayar

Take Order & Distribute

Description

App takes order from a group of users/clients, the items to be ordered can be configured as a one time process, finally it groups order based on item so that the distribution can be made easily. App will be developed in java language with stable (No alpha, beta or RC) versions of all libraries, Gradle and Android Studio.

Once I went to the cafeteria/canteen with my friends group of around 12 people and i was told to take orders from everyone so that only a single order can be placed to the canteen, i wrote names and food items in simple notepad and went to the canteen with that list, now what problems I faced with that simple list are as follows.

1. Calculating total for each food item ordered by my friends, entire process was manual and time taking and can lead to mistakes.
2. Suppose I was smart enough to do point 1, but now it comes to distribution of order, it's a kind of headache as everyone might not remember what has he/she ordered.
3. Estimated price calculation is also required before going to give order as one should have enough money while standing in a queue for giving orders.
4. As I was using my phone for taking/writing orders so I may be the only one who will be active/running among entire friend circle.

Solutions for the above problems:-

Entire manual process mentioned in point 1, 2, 3 is automated by the Take Order & Distribute app and for point 4 the order list and distribution list can be shared by simple text message so that order completion job can be distributed among friends.

Intended User

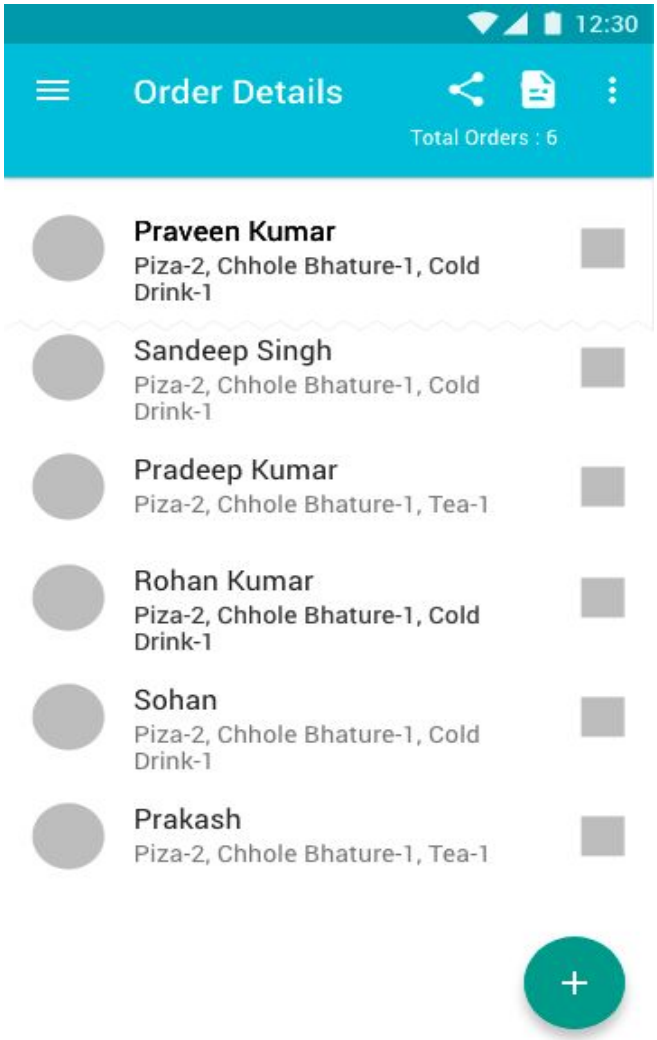
Students, Families or any group of people who want to order something in a group

Features

- Take order
- Share order
- Distribute order
- Configure items(Add,Delete,Edit)

User Interface Mocks

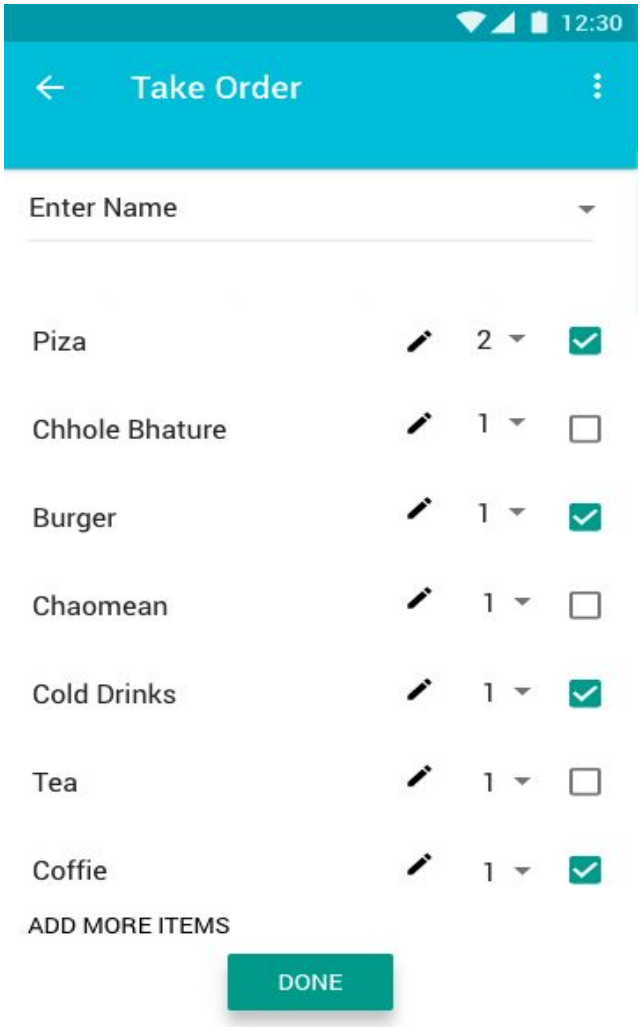
Screen 1 - Order Details



Order Details :- First screen of app having features mentioned below.

1. User can take order by clicking on floating action button.
2. Order can be deleted by left swipe and edited by clicking on list item.
3. Order details can be shared by clicking on share icon placed in toolbar.
4. Order summary can be seen by clicking on the icon next to share icon in toolbar.
5. Total number of orders are labeled at top in actionbar.
6. Option to Add/Edit Items in menu items.
7. Option to Clear order details in menu items.

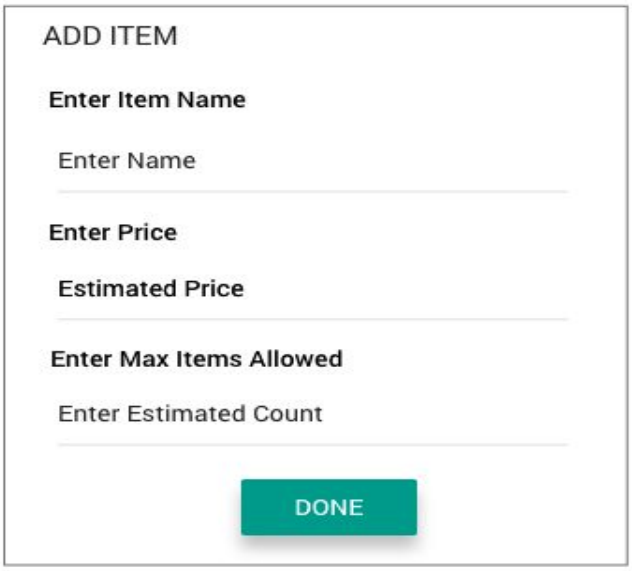
Screen 2 - Take Order



Take Order:- The screen of app having features mentioned below.

1. User can take order by entering the name of the user followed by selecting items.
2. Item name can be edited by clicking on the edit icon.
3. More items can be added by clicking on text label “ADD MORE ITEMS”.
4. The order can be saved either by pressing back navigation button or done button.

Screen 3 - Add/Edit Item



Add Item/Edit Item:- The screen of app having features mentioned below.

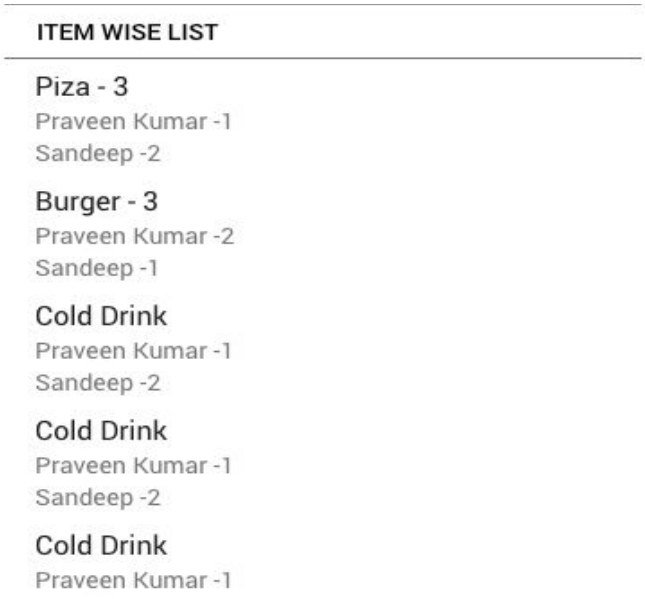
1. In case of Add Item all three fields “item name”, “item price” and “max item allowed” to be added.
2. In case of Edit Item all three fields “item name”, “item price” and “max item allowed” needs to be edited.
3. Pressing done button saves the changes in local database and populate item on Take Order screen.

Screen 4 - Order Summary

Order Summary:- The screen of app having features mentioned below.

1. Item wise list is displayed to the purpose of distribution of the item.
2. Estimated total price and price per item is also displayed.
3. Order summary can be shared to avoid device dependency.

Screen -5- Widget UI Mock



Widget UI:- The widget screen of app having features mentioned below.

1. Item wise list is displayed for the purpose of distribution of the item.

Key Considerations

How will your app handle data persistence?

App will use Room database library with live data for handling data persistence.

Describe any edge or corner cases in the UX.

App will be available in both portrait and landscape mode, data persistence during screen rotation is handled using view model.
Network connectivity will be checked before any network service call and same will be prompted to user.

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

Room library for maintaining data persistence, Timber for showing logs, ButterKnife for handling clicks and other events

Describe how you will implement Google Play Services or other external services.

1. Implement Google Sign in and authentication with firebase.
2. Use of firebase realtime database for fetching/storing data to server.

Describe how you will support accessibility in app.

All the components on screen like text size, button size,color contrast, margin, icons content descriptions will be there to support accessibility.

Describe how the app manages string resources.

All text values will be placed in strings.xml file.

Next Steps: Required Tasks

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

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

- Create Android Studio Project
- Configure libraries by adding dependencies in project

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for Order Details Activity(Main Activity)
    a. Build list item for each order detail

    b. Setup action bar layout with title,menu items, and other labels.
- Build UI for Take Order Activity
    a. Build list item for take order activity.
    b. Setup action bar layout for handling back navigation.
- Build UI for Add/Edit item dialog box
- Build UI for Order Summary Activity
    a. Build list item for item wise list

Task 3: Setup local database

Describe the next task. List the subtasks. For example:

- Create Tables
    a. ITEMS (ID, ITEM_NAME(varchar), ITEM_PRICE(int - default 0), MAX_ALLOWED (int - default 0).
    b. NAMES (ID, NAME(varchar))
    c. ORDER_DETAILS (ID(Foreign Key references to ITEMS), NAME(varchar), ITEM_NAME(varchar), ITEM_PRICE(int), ITEM_COUNT(int))
- Write queries to assessing/writing data to the database
- Setup live data on a table which notify any changes to database

Task 4: Setup RecyclerView

Describe the next task. List the subtasks. For example:

- Create RecyclerView Adapter and View holder
- Add features for deletion of data on left or right swipe of list item.
- Create model class for list item to be used by recycle view.
    a. Items (itemName, itemPrice, maxAllowed)
    b. OrderDetails (name, itemName, itemPrice, itemCount)

Task 5: Handling Errors

Describe the next task. List the subtasks. For example:

- Handle null value errors occured due to empty editText.
    a. Name field on Take order screen
    b. Item Name, Item Price(optional - default 10), Expected count(optional, default - 4)

Task 5: Widget Implementation

Describe the next task. List the subtasks. For example:

- It will display a list view with a list item having two textViews. First one having item name with count and second one having names of users for the purpose of distribution.
- App will use intent service for updating listview in widget.

Add as many tasks as you need to complete your app.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"