

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Create Local Data Store](#)

[Task 4: Implement Business Logic](#)

[Task 5: Bring up UI](#)

[Task 6: Handle Errors](#)

**GitHub Username:** [Michael Huang](#)

# TinyPOS

## Description

TinyPOS is a point of sale client for restaurants. Its goal is to provide a simple and reliable solution to the restaurant users. With this app, you can easily pull out online orders. It eliminates all traditional complexities, providing you the easy interface and the simple hardware installation. It's a flexible solution for saving your money and time.

## Intended User

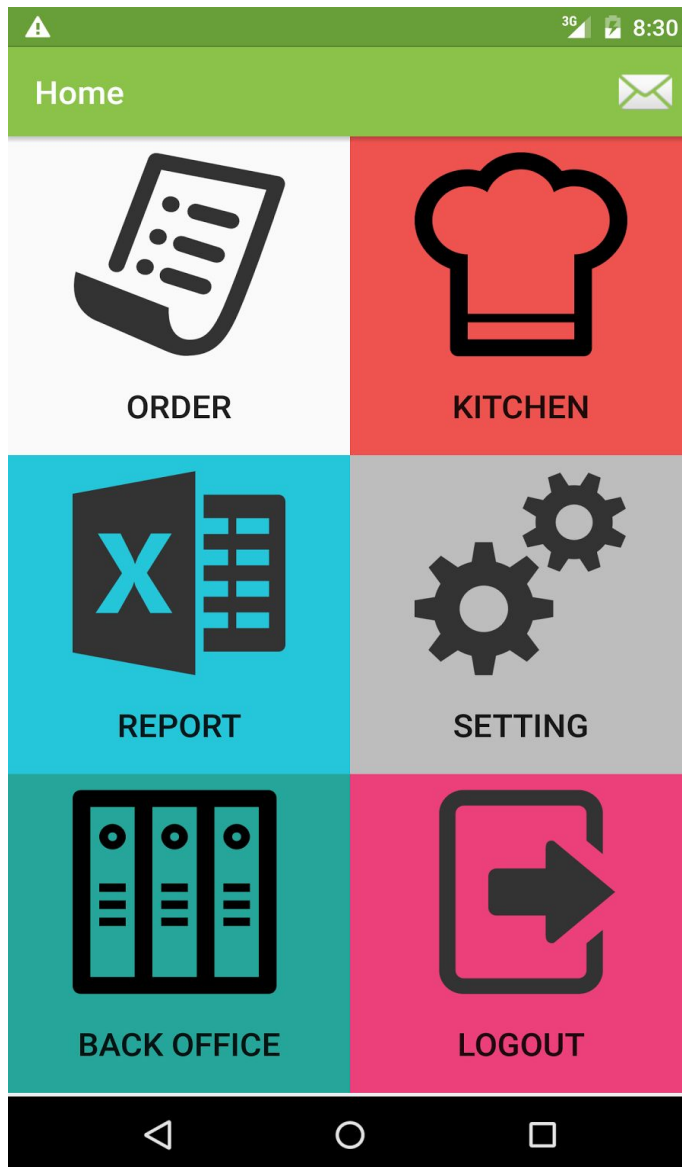
This app is designed for Restaurant, Deli Store, Food Court, etc.

## Features

- Online orders.
- Fast address lookup.
- Delivery time estimation.
- Simplified user interface.

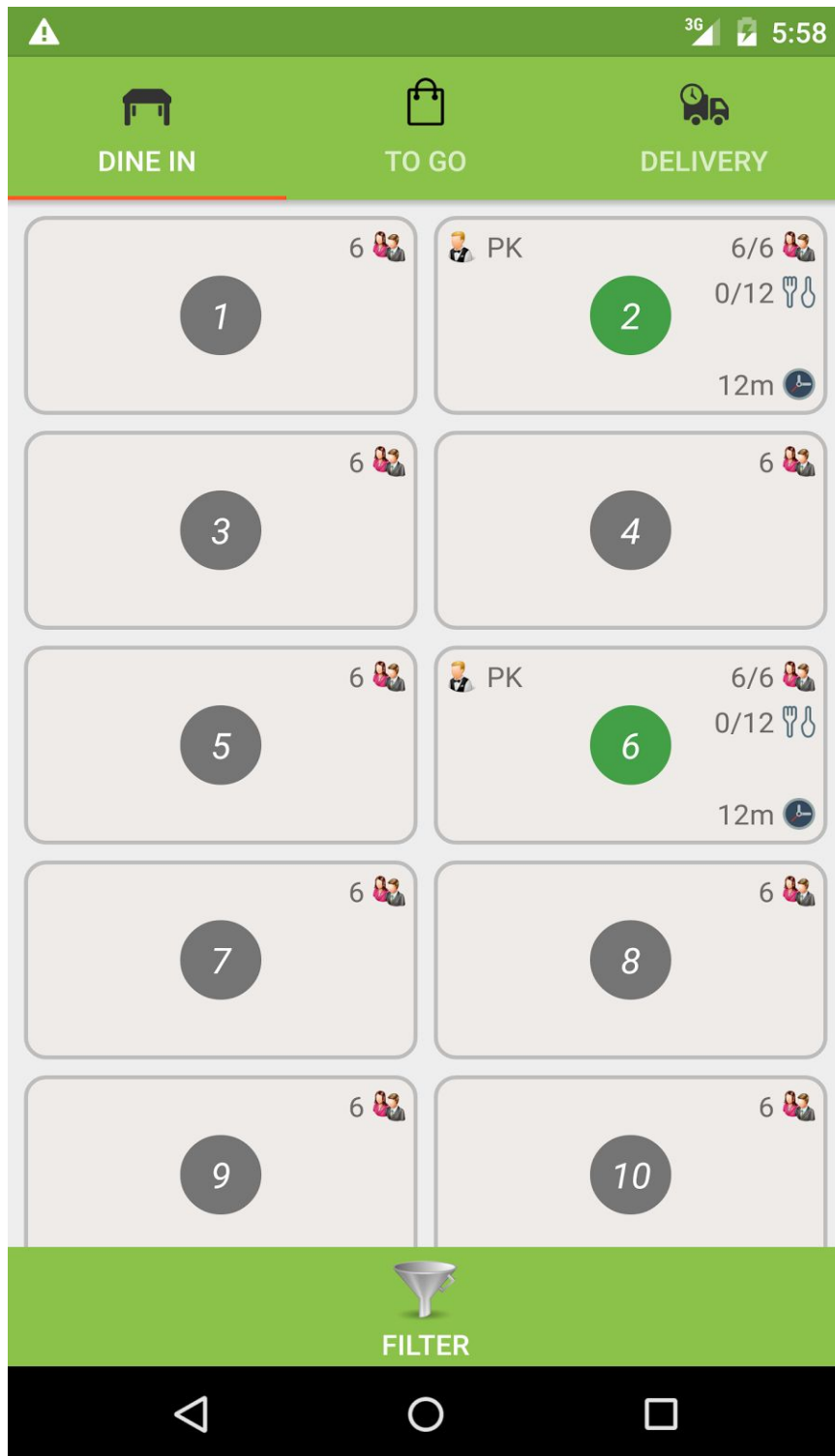
## User Interface Mocks

### Screen 1



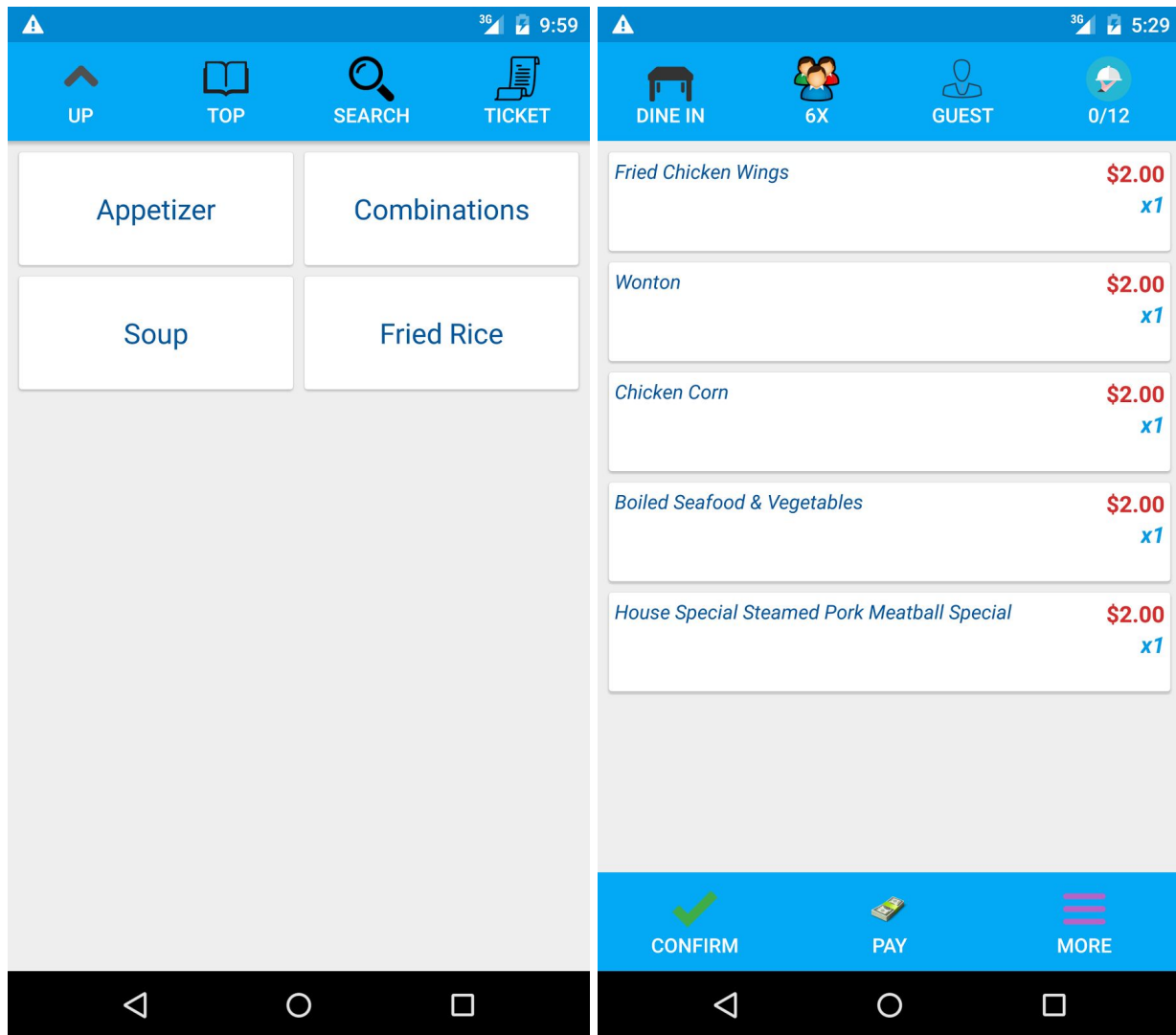
Home Menu

## Screen 2



Order home, including Dine In, To Go, and Delivery

## Screen 3



Order ticket, single activity, swipeable between food menu and ticket detail.

## Screen 4

The screenshot shows a mobile application interface for a kitchen menu. At the top, there is a red header bar with a white back arrow, the word "Kitchen" in white, and a white three-dot menu icon. Above the header, a status bar shows a warning icon, 3G signal, battery level, and the time 1:55. Below the header, there are three white cards, each representing a menu item. Each card has a grey header with an ID, a green "1x" quantity, the item name, and a checkbox. The items are: "100-9 Wonton", "100-10 Chicken Corn", and "100-11 Boiled Seafood & Vegetables". At the bottom of the screen, there is a red bar with a green checkmark icon and the word "CONFIRM" in white. Below this is a black bar with three white navigation icons: a back arrow, a circle, and a square.

ID	Item Name	Quantity	Checkbox
100-9	Wonton	1x	<input type="checkbox"/>
100-10	Chicken Corn	1x	<input type="checkbox"/>
100-11	Boiled Seafood & Vegetables	1x	<input type="checkbox"/>

CONFIRM

Kitchen - Progressing food items.

## Key Considerations

### How will your app handle data persistence?

The data is stored on the server, but the app also has its own copy of food menu and progressing orders. The data saved on the local database can be accessed through Content Provider.

### Describe any corner cases in the UX.

- If the device disconnects from the server, it's still able to take orders. Once the device reconnects to the server, the built-in service will synchronize any changes.
- if an order is currently being modified by multiple devices, only one device can make the change to that order successfully.

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

GSON - Extract data from Server.

Volley - Communicate with the server or any HTTP access.

Butter Knife - Bind views and reduce code.

Picasso - Load and Cache Images.

### Describe how you will implement Google Play Services.

Maps, Places - Provide hints when creating a new address and get routes for delivery orders.

## Next Steps: Required Tasks

\* Since this project should focus on Android app, I am not going to include the structure of server implementation here, but I will attach the source.

### Task 1: Project Setup

- Update SDK.
- Configure libraries.
- Import icon package.
- Create signing task.

## Task 2: Implement UI for Each Activity and Fragment

- Build UI for LoginActivity
- Build UI for HomeActivity
- Build UI for OrderMainActivity
  - DineInFragment
  - ToGoFragment
  - DeliveryFragment
- Build UI for OrderActivity
  - OrderMenuFragment
  - OrderTicketFragment
- Build UI for KitchenActivity
- Build UI for ReportActivity
- Build UI for SettingActivity
- Build UI for CustomerActivity

## Task 3: Create Local Data Store

- Design table layout.
- Implement Content Provider.
- Create a service to synchronize data.

## Task 4: Implement Business Logic

- Split the logic into small parts and implement them..
- Implement helper libraries.
- Implement server API call.
- Implement google service.

## Task 5: Bring up UI

- Implement event listeners.
- Configure activity stack.
- Bind views to data store.

## Task 6: Handle Errors

- Add exception handlers as needed.

- Add unit tests.
- Apply stress testing.