

## Lunch Finder Application Development Plan

---

[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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

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

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

**GitHub Username:** [mudasar](#)

# Happy Lunch

## Description

This application will be used to order lunch in office environment. Specially where there is industrial / office area and only a couple of restaurants are there round the corner. The restaurants have few number of food options they offer in Lunch menu and it is always packed on the lunch time as everyone is hitting the restaurant.

This application will help the customers to view and order the food for lunch without standing in the long queue and looking at the daily specials from all the participating restaurants.

Customers can view the daily specials and can browse the regular menu and choose what fancy them. It'll also show the price and the waiting time before they can pickup the food.

## Intended User

Intended users of the application are office workers in places where there are large offices around and only a couple of restaurants exists nearby.

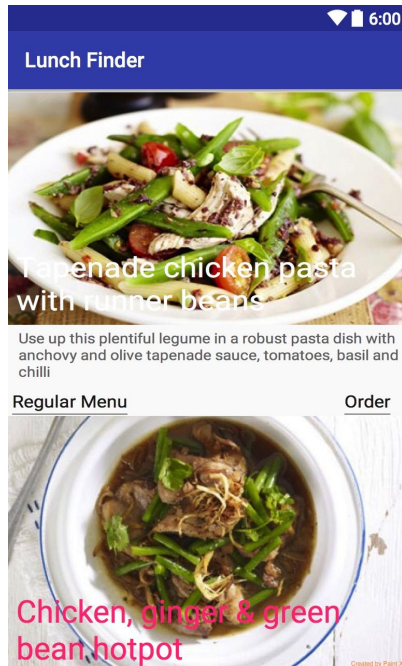
## Features

The major features provided by this application are listed below:

- View Dish of the day from all participating Restaurants
- View Dish of the Day's details
- View Regular restaurant menus
- Save personal details
- Order food
- Notify when to collect the food

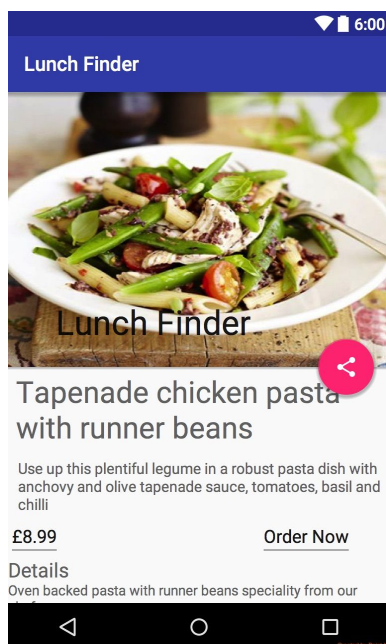
## User Interface Mocks

### Screen 1



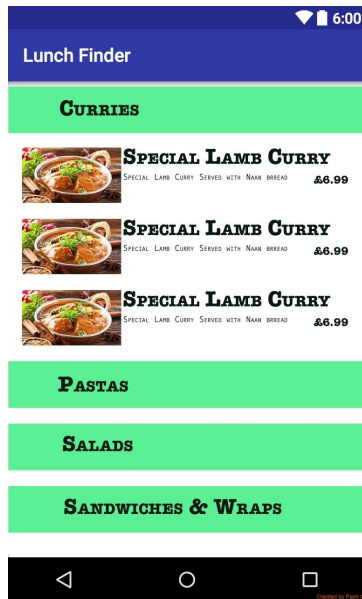
Dish of the day from each of the category will be listed in cardviews.

### Screen 2



Details of the dish of day with instructions, price and waiting time along with sharing options and order functionality.

## Screen 3



This screen will help to choose something to order from the regular menu.

## Screen 4

A screenshot of a mobile application interface titled "Lunch Finder". The top status bar shows a Wi-Fi icon, a battery icon, and the time 6:00. Below the title bar, there is a text prompt: "Please enter your basic information to be used for your orders". Below this prompt are four input fields: "Your Name", "Email", and "Phone", each followed by a horizontal line for text entry. Below the "Phone" field is a button labeled "LOCATION". At the bottom of the form is a button labeled "SAVE". At the bottom of the screen is a black navigation bar with three white icons: a back arrow, a circle, and a square.

This screen will help the customer to save his/her basic details which will be sent to restaurant when placing the order.

## Key Considerations

### How will your app handle data persistence?

Sqlite will be used to store data on the device. Firebase will be used to load the data from online sources. A background service will be handling the notifications.

### Describe any corner cases in the UX.

Any time if a user is not on the home activity pressing back button will take the user to one step up in the navigation. For example If a user is browsing through dish of the day details pressing back button will take the user to home activity.

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

Glide to handle and load images  
Firebase client to connect with Firebase

## Next Steps: Required Tasks

### Task 1: Project Setup

- Create Project in Android Studio
- Create Git repository and readme file
- Configure external libraries

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

- Build UI for MainActivity
- Build UI for Customer Details / Signup
- Build UI for Dish of the Day Details
- Build UI to order the food
- Add UI layouts for Tablet

### **Task 3: Your Next Task**

- Adding Tests for Business logic
- Implement Data Storage
- Save User input data to database
- Link Data and UI
- Create Notification

### **Task 4: Your Next Task**

- Implementing Business logic for loading data from online sources
- Implement Error Handling

### **Task 5: Your Next Task**

- App Testing
- Bug fixing
- Application Signing and Distribution