How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: LavanyaGanganna

Red Chillies

Description

Red Chillies is the food ordering app for Red Chillies(Malabar Cuisine) restaurant. It is located in Milpitas.This app helps people to order kerala food(Indian) in advance and pick on time without any waiting.

Intended User

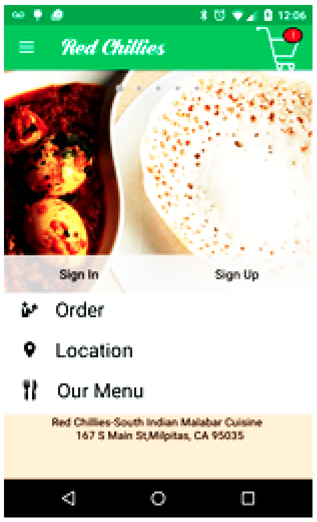
Red Chillies app is perfect for all the people who love Spicy Kerala cuisine ,in and around Milpitas,Santa Clara county.

Features

- Displays full restaurant Menu with photos.
- Uses Google Maps to show direction to Red Chillies,Milpitas
- Integrates paypal to pay money online
- Integrates Google+ login
- Registered User and order information is stored in the backend using Google App Engine Datastore(Objectify)
- Mail is sent from backend to users regarding order summary using Java Mail API.
- The restaurant full menu is stored in local database using sqlite database.
- The restaurant manager can login as admin and view the full orders and delete the order once picked up.

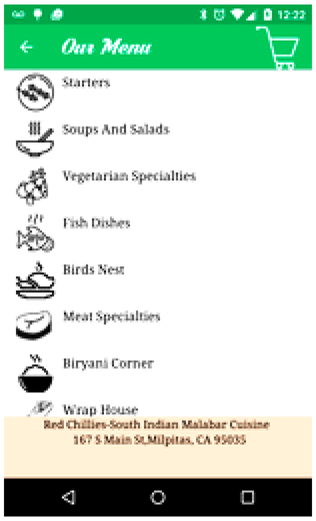
User Interface Mocks

Screen 1


Red Chillies
Sign In
Sign Up
Order
Location
Our Menu
Red Chillies-South Indian Malabar Cuisine
167 S Main St,Milpitas, CA 95035

This is the first screen where user can select the food to order, or check the menu or check the location of Red Chillies.
User has to signup with valid email address.

Screen 2



12:22

Our Menu

Starters

Soups And Salads

Vegetarian Specialties

Fish Dishes

Birds Nest

Meat Specialties

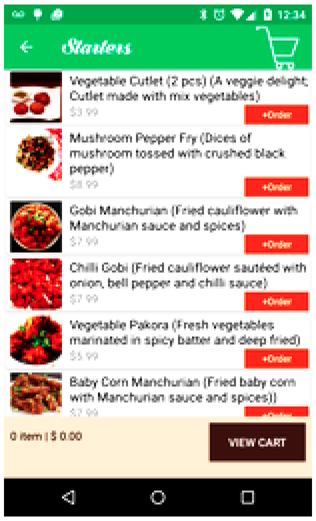
Biryani Corner

Wrap House

Red Chillies-South Indian Malabar Cuisine
167 S Main St,Milpitas, CA 95035

This screen shows all the Food categories in the restaurant menu.

Screen 3



12:34

Starters

Vegetable Cutlet (2 pcs) (A veggie delight; Cutlet made with mix vegetables)
$3.99
+Order

Mushroom Pepper Fry (Dices of mushroom tossed with crushed black pepper)
$8.99
+Order

Gobi Manchurian (Fried cauliflower with Manchurian sauce and spices)
$7.99
+Order

Chilli Gobi (Fried cauliflower sautéed with onion, bell pepper and chilli sauce)
$7.99
+Order

Vegetable Pakora (Fresh vegetables marinated in spicy batter and deep fried)
$5.99
+Order

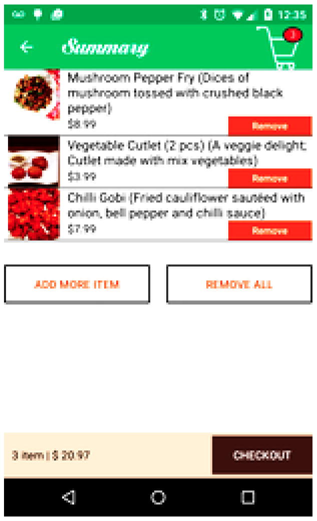
Baby Corn Manchurian (Fried baby corn with Manchurian sauce and spices))
$7.99

0 item | $ 0.00

VIEW CART

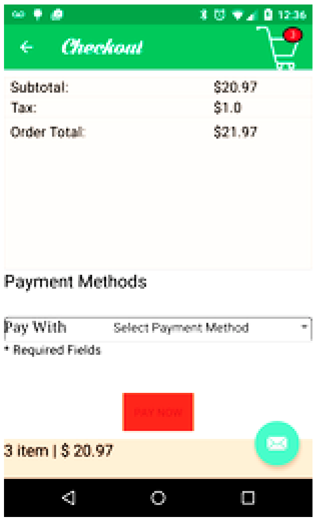
In this screen all the food items in the individual categories are shown.
Clicking on order button adds the individual food items to the shopping cart icon, shown in the toolbar.clicking on View Cart button shows the summary of items ordered.

Screen 4


12:35
Summary
Mushroom Pepper Fry (Dices of mushroom tossed with crushed black pepper)
$8.99
Vegetable Cutlet (2 pcs) (A veggie delight; Cutlet made with mix vegetables)
$3.99
Chilli Gobi (Fried cauliflower sautéed with onion, bell pepper and chilli sauce)
$7.99
3 item | $ 20.97

In this screen , all the items ordered by the user are shown.Clicking on Remove button removes individual item from shopping cart.User can remove all the items or add more items from the order summary screen.

Screen 5


12:36
Checkout

Subtotal:	$20.97
Tax:	$1.0
Order Total:	$21.97

Payment Methods
Pay With Select Payment Method
* Required Fields
3 item | $ 20.97

In this screen,the total amount with tax is shown to the user,the user can select payment method either paypal or gift card.User can proceed with payment by clicking pay now button.

Key Considerations

How will your app handle data persistence?

All the restaurant menu data in this app is stored in a local repository using SQLite. The initial data is bundled as an asset and copied to local database.
A content provider will be used to connect the local repository from the main application.
The users information and order summary is saved in the backend using objectify data access.
Here are the main entities which will be stored.

- User’s Login information
- User’s Order summary
- Restaurant menu

Describe any corner cases in the UX.

The shopping cart menu icon shows the number of items ordered as badge.

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

1. **Picasso:-** For loading images into imageview.
2. **Google Maps:-** For showing directions to Red Chillies restaurant.
3. **Paypal:-** For paying money using credit card or paypal login.
4. **ButterKnife:-**This is view injection library used for generating boiler plate code of field and method binding to views.
5. **Google Play Services:-**In this application I use identity platform Google+ provided by Google to authenticate the user.Also an ad banner will be shown using Admob library provided by Google Play Services.
6. **SqliteAssetHelper:-** To copy the menu from asset folder to local device database.
7. **Objectify:-**To store Users data in the backend server.

Describe how you will implement Google Play Services.

In this application I use identity platform(Google+)provided by Google to authenticate the user.
An ad banner will be shown using Admob library provided by Google Play Services.
Map directions will be shown using maps provided by Google Play Services.
Open the build.gradle and add the

```
compile 'com.google.android.gms:play-services:9.8.0' dependency
```

```
Copy the google.services.json file to app folder.
```

Next Steps: Required Tasks

Task 1: Project Setup

- Create a new project in Android studio using following information:
- ApplicationName:Capstoneproject1
- Company Domain:android.example.capstoneproject1.com
- Packagename:com.example.android.capstoneproject1
- Platform:Phone and Tablet
- Minimum SDK:API 16-Android-4.1
- Modify build.gradle to include all dependencies libraries.
- Design the flow of Project

Task 2: Implementing UI for first Activity

- Design a splash screen
- Build UI for Navigation Drawer
- Add ViewPager using PagerAdapter
- Add listview to display the choices for user

Task 3: Showing the Red Chillies restaurant location

- First get the Google API key from Google Developers console
- Add the access permissions and the API key in the Androidmanifest file
- Using Supportmapfragment,Google Map,Locationmanager show the map with red chillies location

Task 4: Implement Restaurant menu

- Build UI for restaurant menu categories list
- Build UI for each section of menu items
- Build UI for Summary of Items ordered by customer

Task 5: Implement SignUp and SignIn activity

- Build UI with text Input layout
- Add Textwatcher to each field inputs to validate.

Task 6: Implement The Database

- First add the entire restaurant menu entries using DB browser for Sqlite.
- Save this menu.db file in the assets folder.
- Create the DBhelper class and provider class for local database.
- Copy the menu.db database to local database using SqliteassetHelper

Task 7: Implement The backend module using Google App Engine

- Create new Google Cloud Module(backend) and add it to the app.
- Create a Java class(Users) in the backend to hold the user's information.
- Right click on the Users class and generate the custom Google cloud Endpoint.
- Create your own API call in the UserEndpoint class and use it in the app to communicate with backend.
- Create an AsyncTask to connect to the backend from the app
- Deploy the backend module to Google App Engine

Task 8: Integrate PayPal API to pay through paypal account or Credit Card

- Add the dependency for paypal sdk in gradle file
- Create a valid Paypal account in paypal.com
- Log in to the developers.paypal.com and go to the sandbox accounts section
- Create one more sandbox test account with dummy money of 10000$
- We can check the transactions in the sandbox.paypal.com(Login with the test account id and password).
- Click on the create App button and provide the application name to generate ClientId.Use this client Id in the program to configure Paypal.

Task 9: Implement Google+ login

- To use Google services API we need to add gms play services dependency in gradle.
- Get the configuration file from the google developers console
- Copy google-services.json file to the app module
- Using Google Api Client add the google account for sign in

Task 10: Implement Admob

- Sign into the apps.admob.com
- Create Ad unit id for banner ads
- Add gms play services dependency in gradle
- Add the gms meta data and permission in androidmanifest file.
- In order to display the ads add the gms.ads.Adview in the xml file.

Task 11: Sending mails from backend

- Once the customer successfully completes payment for his order, mail is sent with the order summary to customer's mail address using Javamail API in the backend.
- In the google cloud platform register the sender's email address.
- The javamail Api is included in the App engine sdk.
- Create a message using a JavaMail `Session` object and send it.

Task 12: Implementing the Redchillies Appwidget

- First create layout with imageview and textview to display the widget
- Next create a appwidgetprovider xml file with all metadata like widget minheight,width,resize height,preview image.
- To continue create a appwidgetprovider java class to build the widget view and launch app on touch.
- The appwidgetprovider uses remoteviews object and pendingIntents to launch the app.
- Add the entry in the Androidmanifest file about the widget.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"