

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**”

Description

Intended User

Features

User Interface Mocks

Screen 1

Screen 2

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: Your Next Task

Task 4: Your Next Task

Task 5: Your Next Task

GitHub Username: Bheema-hb

Application Name : ParkMe

Description

ParkMe : Whenever I go out with my car the most problematic situation I face is to look for parking. Especially city like Bangalore,Mumbai are so crowded that we won't get space to park our vehicle. And another challenge is to find out the parking location. So with this app I am trying to solve that problem by showing the parking location near by them. Here going to consider the

key aspects of number of parking slots available, parking charges, Showing users parked duration, etc

Intended User

This app is for any vehicle owners

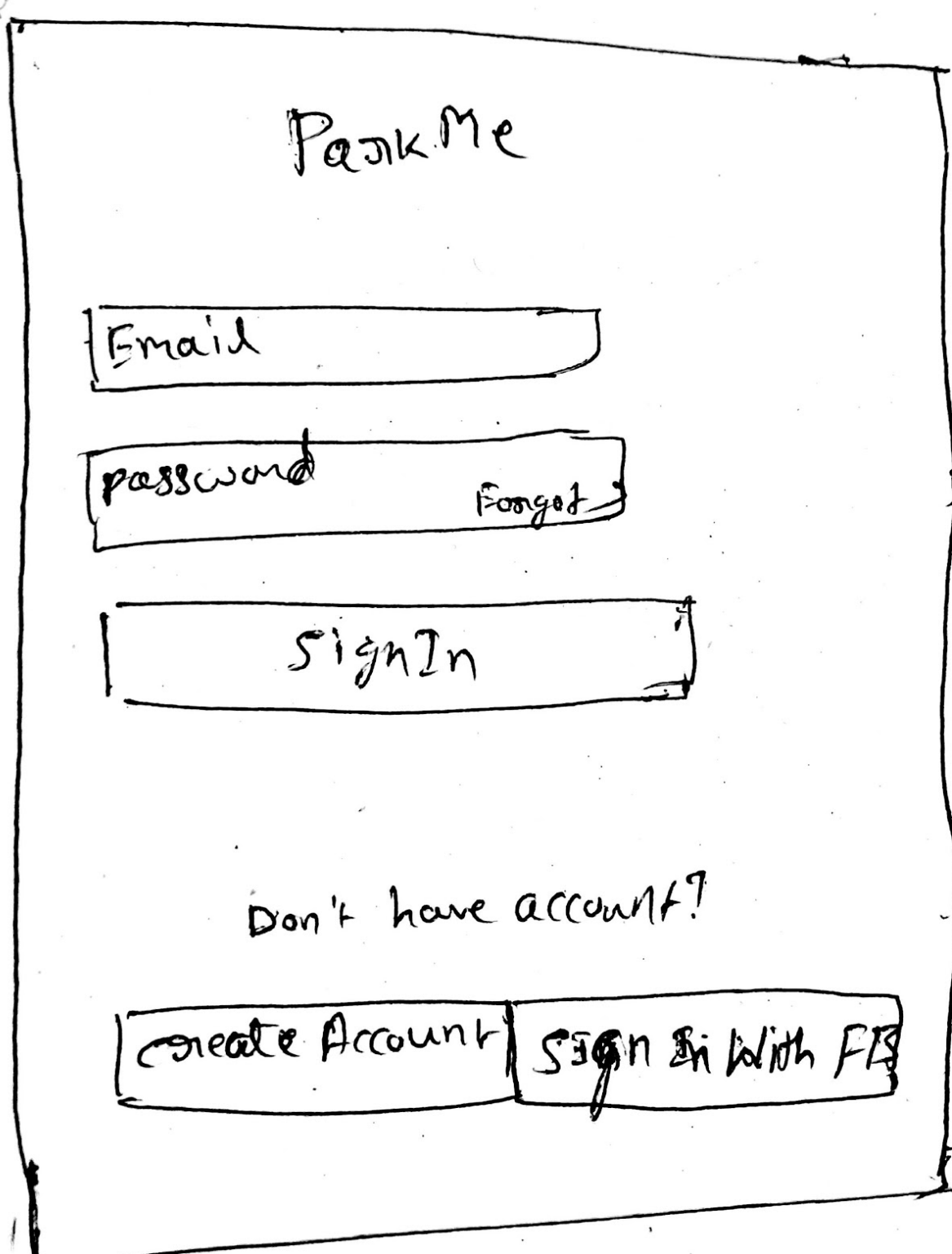
Features

- Facebook login
- Default system login
- Map with markers to show the parking locations
- Timer clock to notify user the parked duration
- Showing total payable amount

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Login Screen



A hand-drawn sketch of a login screen for an application called "ParkMe". The screen is enclosed in a rectangular border. At the top, the text "ParkMe" is written in a cursive, handwritten style. Below this, there are three input fields. The first is labeled "Email". The second is labeled "password" and has a small "Forgot" link to its right. The third is a large button labeled "signIn". Below the input fields, the text "Don't have account?" is written. At the bottom, there are two buttons: "create Account" and "signIn With FB".

ParkMe

Email

password [Forgot](#)

signIn

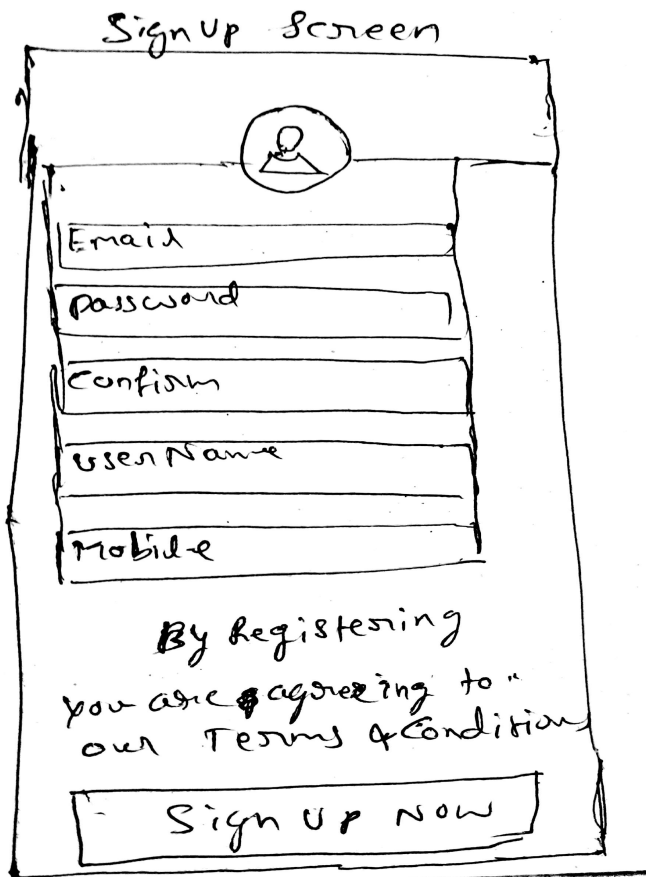
Don't have account?

create Account [signIn With FB](#)

This is a login screen where user can login via their ParkMe account.

SignUp Screen

SignUp Screen



Email

Password

Confirm

User Name

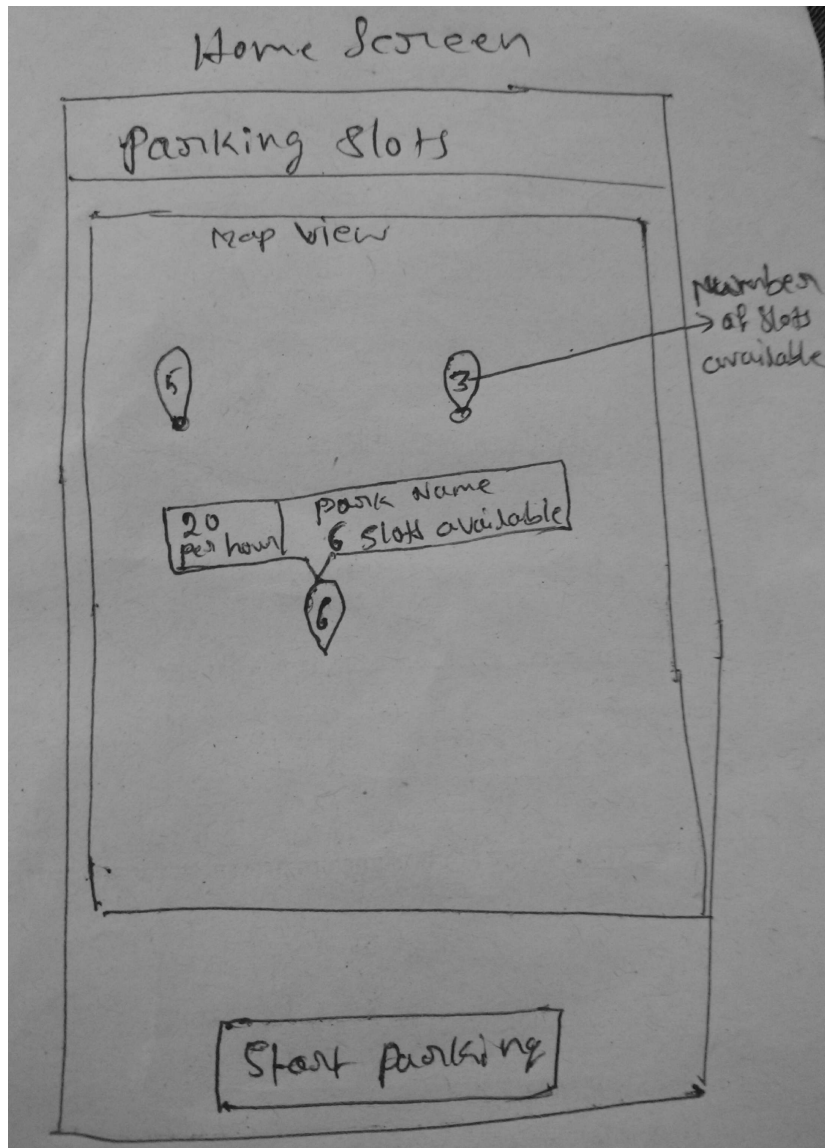
Mobile

By registering
you are agreeing to
our Terms & Conditions

Sign Up Now

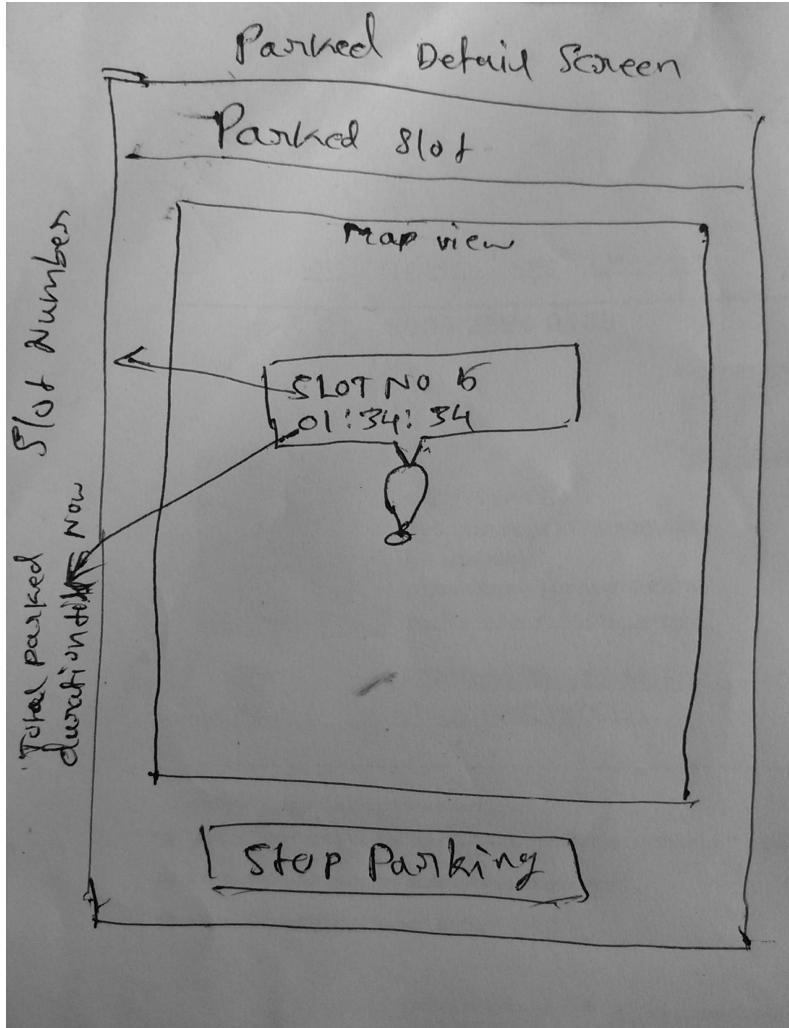
Here has to enter few details like email, password, username and mobile number. If he has signed up via facebook then most of these will captured automatically

Home Screen



This is a screen where i will show available parking near to the user. Here map will have marker at each parking location with number of slots. When user click on marker, I will show price and available slots. To start the parking user has to select/click the marker and has to click on start parking button

Vehicle Parked screen



This is a screen when user starts parking. It shows the parking slot and the time ticker. That is the total duration of parking.

Payment Summary

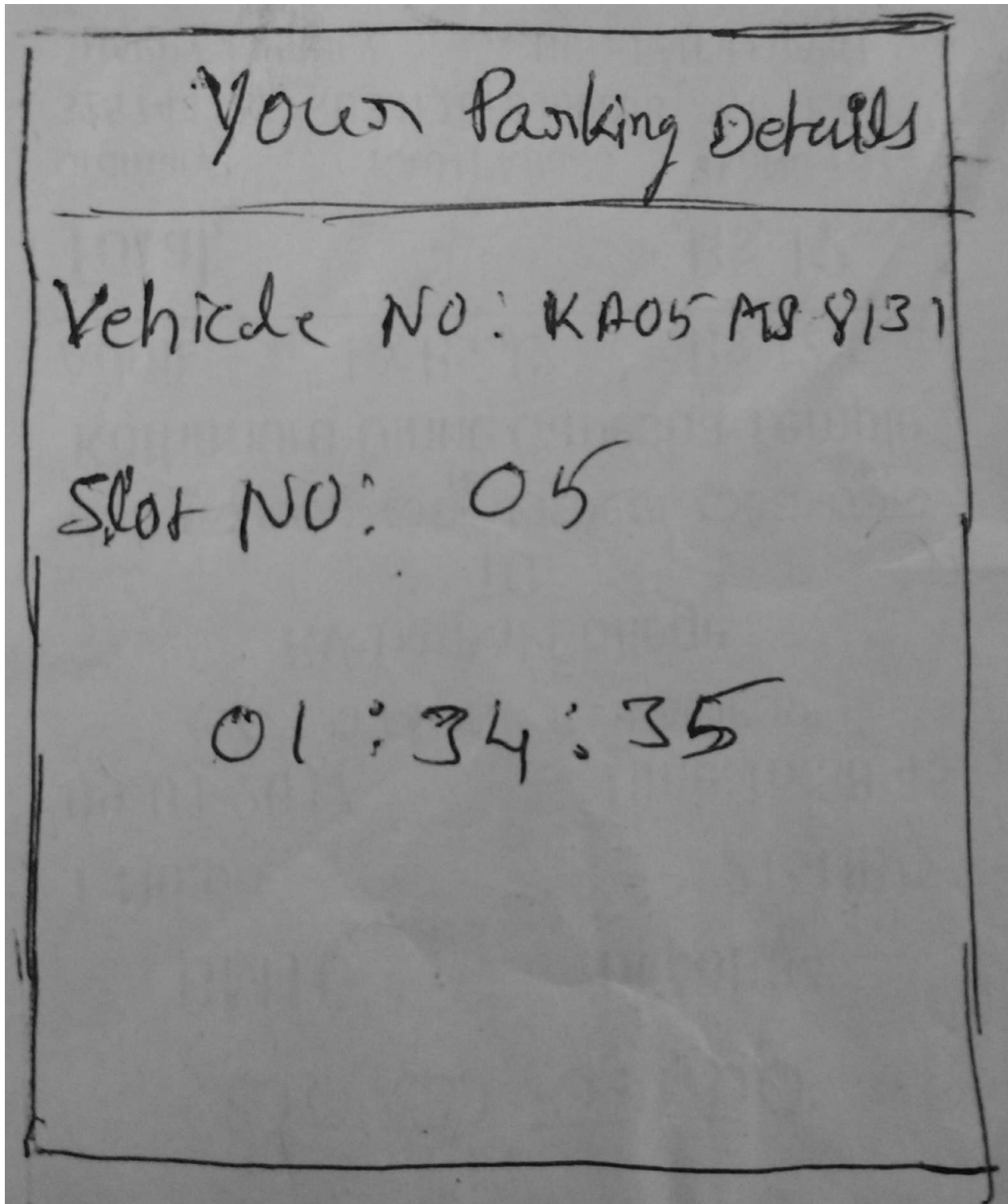
X Payment Summary

| | |
|----------------|----------------|
| Location : | Location Name |
| Date : | Parked time |
| Arrival : | Assigned time |
| Exit : | Exit time |
| Vehicle : | Vehicle Number |
| Alloted Slot : | Slot Number |
| Price | Price money |
| Other fee | |
| Total | Total Amount |

Pay Now

This is the payment summary screen.

App Widget



Key Considerations

How will your app handle data persistence?

When user selects the parking slot and start the parking then I am going to store that parking data like parked start time, parking slot, parking name, etc. using sqlite db.

I will Content provider to store and access data. So whenever user returns to app again it will resume to parked Screen by accessing these data. I will use loader to attach these data to view.

Describe any corner cases in the UX.

I am not going to develop backend for this application. So the app will work with the dummy apis. For this I am using mockey to host my api responses.

So the parking slots, payment summary apis returns dummy responses.

SignUp and Login I have given just to show app complete flow.

Login will work even if you enter invalid credentials.

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

Using Facebook library for facebook sign up.

Volley for networking

Gson for parsing the responses

I will use the IntentService for all web API

Describe how you will implement Google Play Services.

Using play-services-maps and play-services-location to show the map with marker in the app and to fetch the user current location respectively

Next Steps: Required Tasks

Task 1: Project Setup

The project setup involves following subtasks

- Project skeleton creation
- Volley setup
- Facebook Integration

Task 2: Implement UI for Each Activity and Fragment

- Splash screen
- Login screen implementation
- Signup screen implementation
- Home screen Activity
- Navigation Menu

Task 3: Integrating apis and other UI elements

- Integration dummy login, signup api
- Home Map fragment screen implementation
- Implementing marker on the map with marker window
- Integrate google location api to fetch user current location

Task 4: Handling start and stop parking

- Handle the start parking action and its details
- Modifying the home fragment to show parked details
- Handle stop parking action

Task 5: Payment summary

- Payment Summary screen implementation

Task 6: App widget

- App widget provides implementation
- It provides information about the currently parked vehicle and its total parked duration

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**"