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: KosratDAhmad

My Prayers

Description

My Prayers app indicates you Muslims Prayer timing, the Mosques near you, the Qibla direction wherever you are, and some other features. My app solve a problem which is prayer times are calculated from some algorithms based on longitude and latitude of your place but some countries or cities didn't depend on this algorithms for the prayer times, they have a fixed time table, for example I'm from Kurdistan of Iraq we have a fixed time table for prayers that are differ from the one that based on longitude and latitude and the apps on the Play Store almost depends on longitude and latitude so we can't use them on my country. Two years ago I create an app for my peoples named My Prayers which is only for Kurdistan of Iraq with poor design and performance but used fixed prayer times and now I realized that there are some other cities or countries have fixed time table for prayers like (Aachen, Birmingham, Dubai, Emirates, Iraq, London, Munchen, Paris, Syria, and Kurdistan of Iran). I want to solve this problems by creating an app depends on both location (longitude and latitude) and fixed prayer times to cover over all world.

Intended User

Muslim Religion

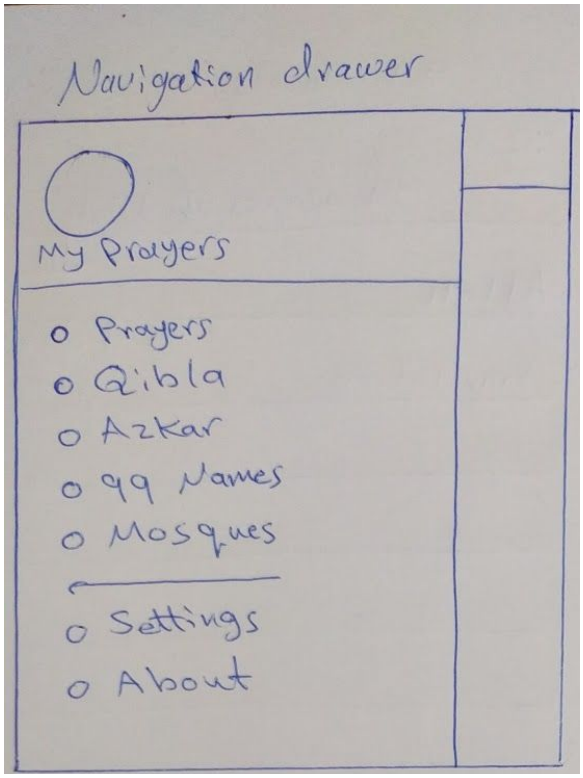
Features

- Display prayer times based on location and fixed.
- A Compass to show the Qibla direction.
- 99 names of ALLAH with sound.
- Common Muslim Azkars.
- Display near Mosques.
- Hijri Date.
- A widget that shows today's prayers times.
- Notification or Alarm for each prayer with the ability to adjust their times.
- Ability to select notification tone (Athan) from the SD card.
- Switching the phone to silent automatically during prayer times, with settings for each prayer.
- Ability to adjust prayer times manually.
- A Watch face to show prayer times in the watches.
- App fully translated to: English, Kurdish, Arabic, and Persian.

User Interface Mocks

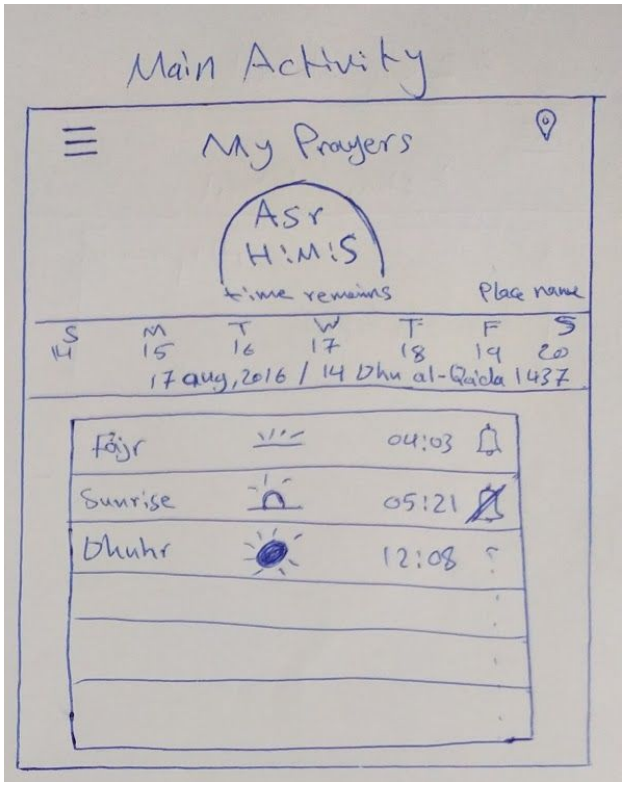
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



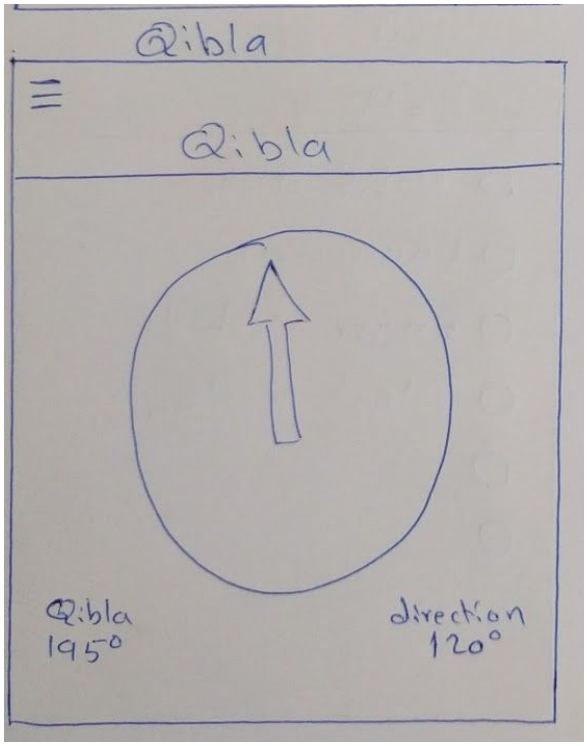
This is Navigation drawer of the app which contains all sections of the app and settings

Screen 2



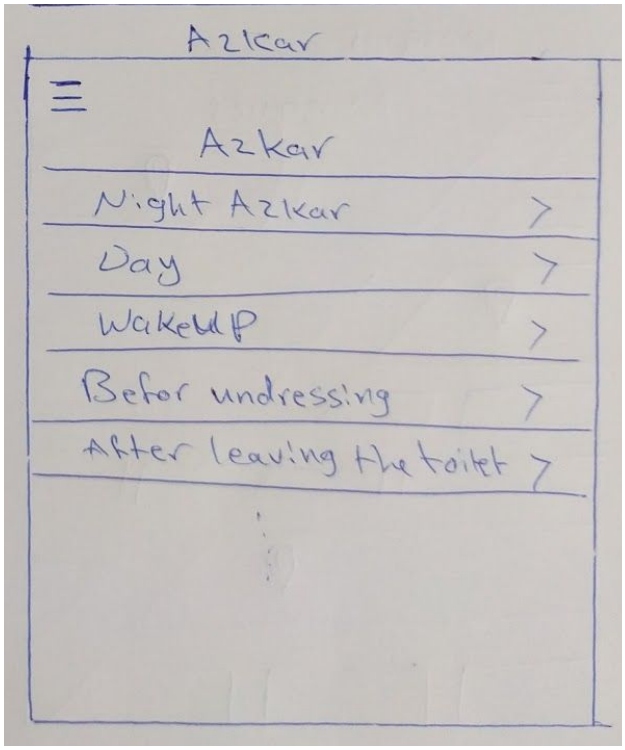
This is the main activity which display the prayer times of your place at the top we have a progress to show you how many time remains for the next prayer after that we have a calendar to shows Julian and Hijri Date and you can change the date of the prayer times. Finally we have a list of prayer times with ability to enable notify or not.

Screen 3



This is shows you the direction of the Qibla

Screen 4



This is the list of common muslim azkars when you tap on it, it will show you the azkar with translation.

Key Considerations

How will your app handle data persistence?

I will create a database with the following tables:

- countries: columns (country_code, country_name, country_languages).
- cities: columns (city_name, country_code, latitude, longitude).
- static_prayer: columns (method, method_name, city, date, fajr, sunrise, dhuhr, asr, maghrib, isha).

Then I will fill the tables with data and build a Content Provider to communicate with this database. The shared preferences will be used for saving simple values for later use, such as whether push notifications are turned on or off.

Describe any corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

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

- Android Support Library (appcompat-v7, design, recyclerview-v7, and cardview-v7) for creating an app with material design and support older version of android.
- Butter Knife for Field and method binding for Android views.
- Schematic to generate content provider.
- Material-dialogs to support material dialog style for older android version.

Describe how you will implement Google Play Services.

- Google Play Service Admob: I will add ads to my app to make money.
- Google Play Service Analytics: for analysing my user and improve my app.
- Google Play Service Location: for getting location of the user for display prayer times
- Google Play Service Map: for shows near by Mosques

Next Steps: Required Tasks

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

Task 1: Project Setup

- Open app with Android Studio
- Ensure JDK is set to 1.8
- Set configuration to "app"
- Click 'run' to compile and install on a device

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for Navigation Drawer
- Build UI for MainActivity

- Build UI for Qibla Activity
- Build UI for Azkar Activity
    - Build UI for Azkar Detail Activity
- Build UI for 99 Names Activity
- Build UI for Mosques Activity
- Build UI for Settings

Task 3: Implement Google Play Services

Describe the next task. For example, “Implement Google Play Services,” or “Handle Error Cases,” or “Create Build Variant.”

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

- Implement Google Play Services Admob in main activity
- Implement Google Play Services Location
- Implement Google Play Services Map in Mosques activity
- Implement Google Play Services Analytics

Task 4: Your Next Task

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

- Create layout
- Something else

Task 5: Your Next Task

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

- Create layout
- Something else

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

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**"