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: **https://github.com/ArnalShoorukov**

Prayer Time App

Description

Prayer Time App is the ultimate tool for getting accurate prayer times and Qibla direction, no matter where you are. Its modern and simple to use interface offers you immense power.

Intended User

Prayer Time App is a must have companion for all Muslims over the world. 100% free and without internet.

Features

Prayer Time App comes to all Muslims with big new features that is needed all the day to all Muslims around the world like :

Prayer Times:

★ Prayer times for 200.000+ cities in 200+ Countries: Fajr, Sunrise, Dhuhr, Asr, Maghrib, Isha
★ Automatic location detection using Network or GPS, or finding the location manually by searching (Without Internet)
★ A Compass to show the Qibla direction.
★Choose Islamic prayer calculation method applicable to you.
★ Choice your Madhab / Juristic Method (Hanafi, Shafi, Maliki or Hanbali)

Prayer Time Settings:

★ Changing default settings of prayer calculation method and Juristic Method
★ 12/24-hour format

Remembrance and Supplication

Prayer Time App is an application consisting of Beneficial Islamic Duas daily important Supplication (duas) and Remembrance (azkar) categories. It will help Muslims of all ages including Muslim kids and adults to know, learn, memorize, and recite different supplications for daily life and other occasions.

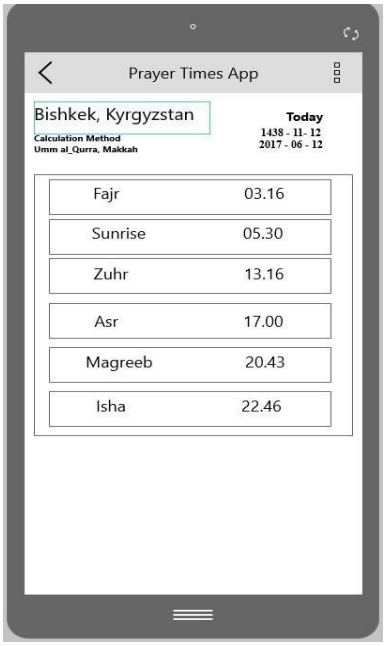
★ Dua Categories meant for achievement in different instances are available: Morning/Evening, Restroom, Prayer ....

★ Supports phones and tablets

User Interface Mocks

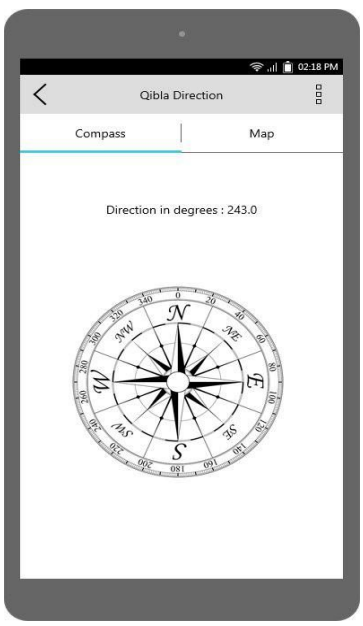
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

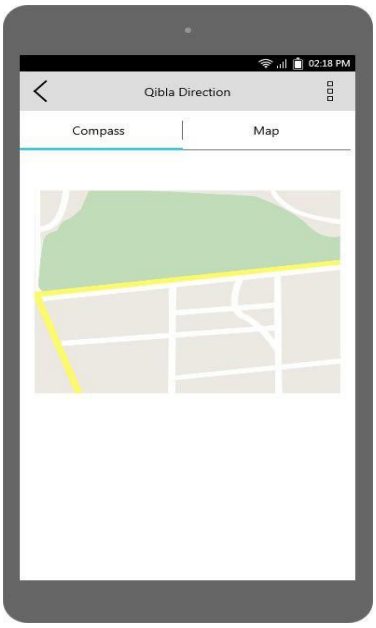
Screen 1



This is my first screen, where users can find most important information for the current day. Like a current day prayer time, date, location and calculation method for prayer time.

Screen 2



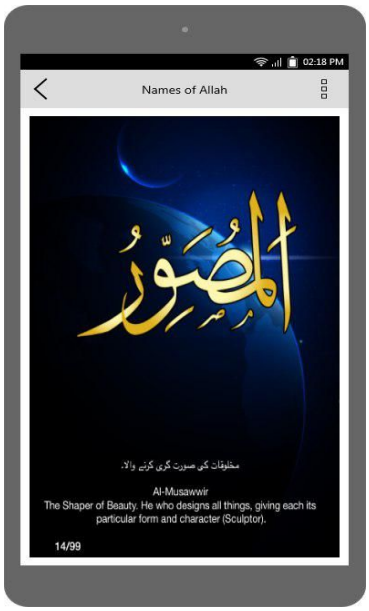


In this screens users can find direction to Qibla, either by using compass or map.

Screen 3

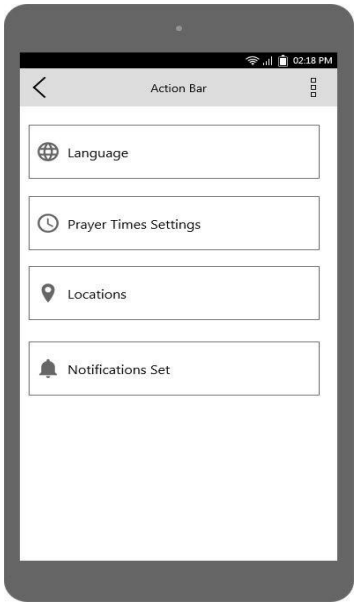
In this screen users can find daily duas which are important for every Muslim.

Screen 4



In this screen users can find all 99 Names of Allah which mentioned in Quran and Sunnah of Prophet.

Screen 5

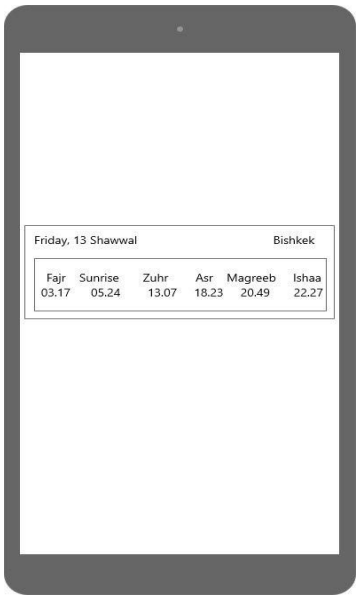


This screen for settings, users may change desired Islamic Calculation Method, time format and Juristic Methods for calculation of daily prayer.

Screen 6

This screen gives to users all information about us, how to contacts for further advices, complains or any wishes. Also to follow us in social media pages.

Screen 7



Home screen widget for showing daily prayer times.

Key Considerations

How will your app handle data persistence?

I will create new Content Provider for my application and it will handle persistence. Setting and other data done by using shared preferences.

Describe any corner cases in the UX.

My app consist from couple of activities, while passing from main activity to another screen it may let wait user some small delay, map activity may let user also wait until it find current location, also user may wait for few time time while activities where goes from listview to detail view.

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

I am going to use for my project prayer times calculation library, which is Islamic Calculation Library (ICLib). This library contains calculations/algorithms needed specifically by muslims and people in muslim countries, such as prayer times, qibla direction, and Hijri conversion.

Also I will use PrayTimes class where provided formulas for calculation, location, Qibla degree etc..

Describe how you will implement Google Play Services.

Google Analytics: For tracking user engagement on app and do improvement based on those results.

Google AdMob: Monetize app and earn money showing relevant ads in app.

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

First I will create basic setup of project and upload it to github.
List of sub task:

- Create basic project
- Configure all the dependencies
- Create app skeleton with navigation drawer
- Upload to GitHub and start working on project

Task 2: Home Screen

- Create main view
- Create main view with sample data
- Implement ICLib library
- Implement PrayTimes class
- Set location
- Show prayer times

Task 3: Qibla Screen

- Find Qibla using compass
- Find Qibla using map

Task 4: Dua Screen

- Create screen with dua
- Create content provider

- Favorite dua implementation
- App uses Loader to retrieve data and show dua's which favorited by user

Task 5: Names Screen

- Create screen with names
- Play audio by pressing the Name

Task 6: Settings Screen

- Create screen of settings
- app uses an AsyncTask to collect da according to Location and calculation method settings

Task 7: About Screen

- Create screen with our information
- Links for our social media
- Share function

Task 8: Home screen widget

- Create home screen widget
- Daily prayer info show to home screen widget

Task 7: Implement Google Play Services

- Google AdMob integration
- Google Analytics integration

Task 8: Improve Design

- Handle landscape orientation
- Handle for tablet users
- RTL support

Task 9: Testing

- Testing on various devices
- Bug fixing

Task 10: Deployment

- Generate signed apk

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**"