Capstone Stage 1

Contents

Author: Asha Ramanjanaiah

GitHub Username: Asha Ramanjanaiah

Hindu Temples

Description

Hindu Temples is an app which provides online presence for the Temples. Temples should register with this app to post temple information such as temple open/close timings, events, location etc. This app helps devotees to be aware of events, festivals and pooja's, so that they can visit temples on the occasions.

Temple administrative department should reach out to Hindu Temple app admin to enroll the temple details for the first time. Later, temple administrator will be provided with login credentials to maintain temple timings and events info.

App is written solely in the Java Programming Language.

Intended User

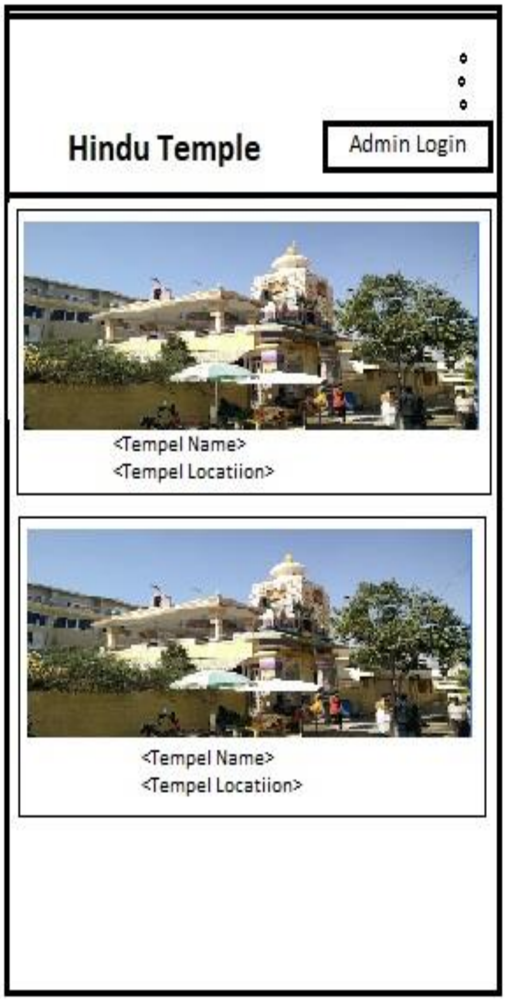
This is an app for Families belong to Hindu religion.

Features

List of main features of my app.
1. Saves temple name, location and images.
2. Saves temple opening time, closing time and day.
3. Saves events that will be happening at temple with date and time.
4. Displays Temple information in the first screen.
5. Displays Temple timings, events and location in the second screen.

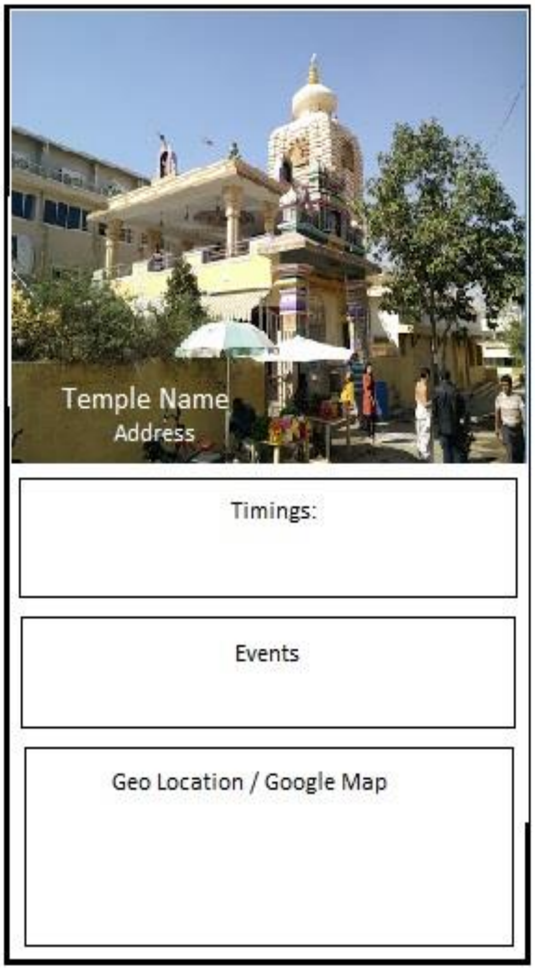
User Interface Mocks

Screen 1



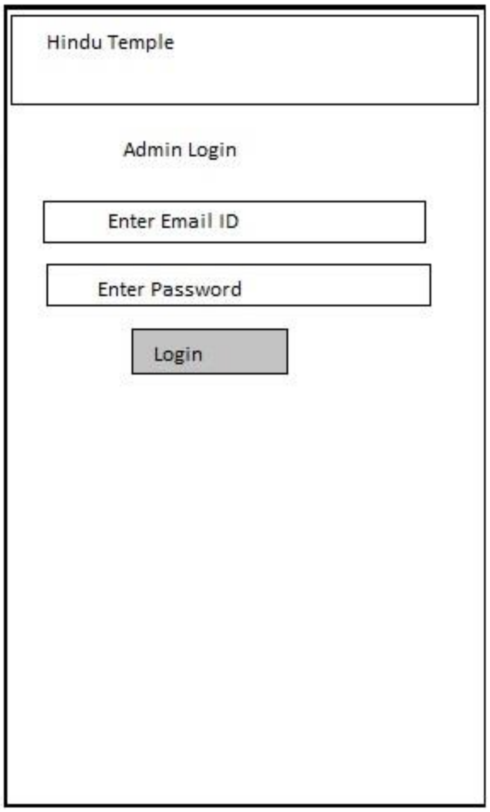
This is the first screen of the app which displays temple name, image and location.

Screen 2



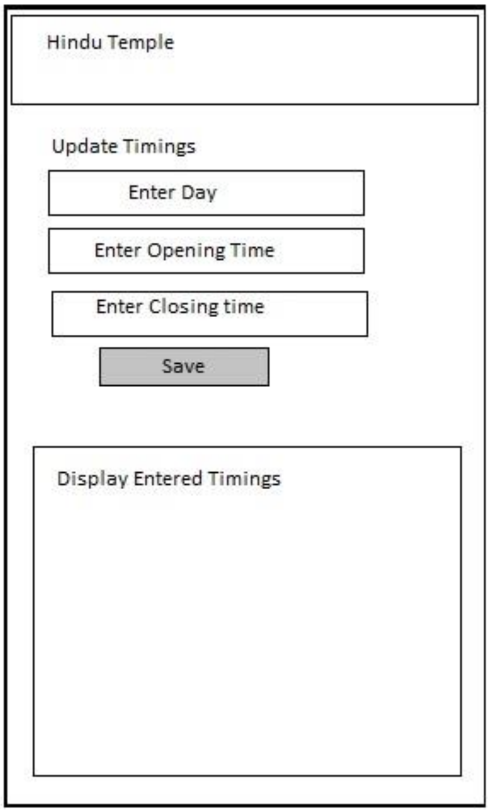
This is the second screen of the app. User will be navigated to this screen once clicked on card view in first screen. This screen displays temple timings, events and maps.

Screen 3



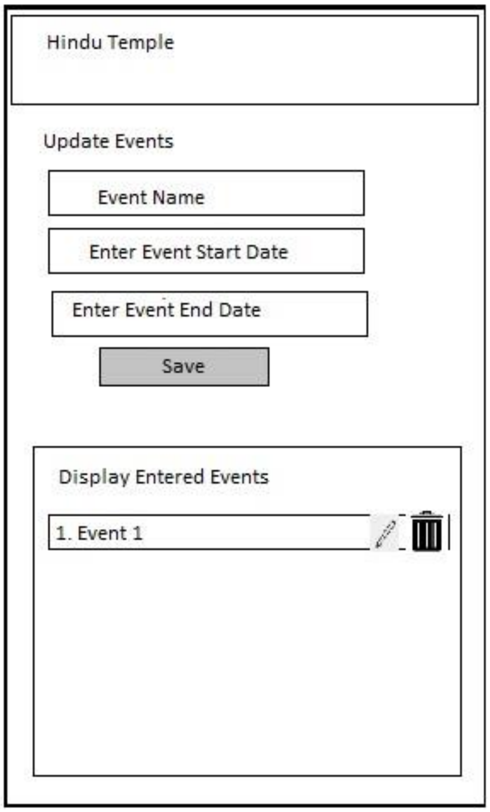
This is the admin login screen.

Screen 4



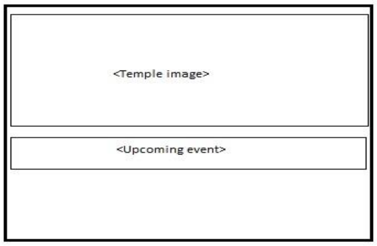
Once admin is logged in with the provided login credentials, this page will be displayed. Admin can add/update the timings here.

Screen 5



This page is to update event Information.

Widget UI



Key Considerations

How will your app handle data persistence?

I will be using Firebase Realtime Database to handle data persistence.
I'm planning to create 3 nodes/tables:

- Temples – includes temple id, name and location.
- Timings – includes day, opening time, closing time.
- Events – includes event name, day, time

I will be linking user to the data in these nodes.

```
"temples": {{
        "userUID": {
                templeId: "43242";
                templeName: "Hanuman"
                templeLocation: "Bangalore"
        }
}
```

Describe any edge or corner cases in the UX.

I don't see any edge cases as of now.

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

I will be using Picasso to handle the loading and caching of images.

Describe how you will implement Google Play Services or other external services.

To make the Google Play services APIs available to your app:

1. Open the build.gradle file inside your application module directory. Add a new build rule under dependencies for the latest version of play-services.

2. Ensure that your top-level build.gradle contains a reference to the google() repo or to maven.

3. Save the changes, and click Sync Project with Gradle Files in the toolbar.

Next Steps: Required Tasks

Task 1: Project Setup

Setting up Firebase.

1. Installing Firebase SDK.
2. Adding project in firebase console.
3. Adding Firebase Realtime Database to the project.
4. Adding Firebase Authentication to the project.

Task 2: Creating admin login Activities

1. Creating Login activity
2. Creating Update Temple Info activity to add/update temple information.
3. Creating another activity to add/ update temple timings and events.

Task 3: Creating UI for Activities and Fragments

1. Creating an Activity and a fragment to display temple information in cardview.
2. Build UI for detail page which displays information about temple timings, events and map.

Task 4: Including Admob and maps

1. Adding dependencies.
2. Including ads and map in UI.

Task 5: Creating widget

- Building UI for widget.
- Writing an IntentService for the communication between widget and app.

Task 6: Creating build variant

- Setting up signing configuration, and including keystore and passwords in repository.
- Creating release build variant.

Versions of Gradle, libraries and Android studio

- Android studio version – 3.3.2
- Gradle version – 4.10.1
- Google play services version – 4.2.0
- Firebase core version – 16.0.8
- Firebase database – 16.1.0
- Firebase Authentication – 16.2.0
- Picasso – 2.5.2

Supporting Accessibility

- Following material design best practices, to allow all the users, including users with disabilities, to navigate and interact with the app more easily.
- Including content labeling, touch target size, color contrast, and view attributes.

Storing resources

- Strings – strings.xml
- Colors – colors.xml
- Themes – styles.xml

Google Play/Firebase services

- Admob
- Maps