GitHub Username: apoorva-asgekar

Kidventures

Description

Ever struggled to decide where to take the kids for a couple of hours or even to spend the entire day? Ever wondered if the fancy restaurant your best friend has suggested for a dinner with family has food options for your little one? This app answers all those questions and provides you with more resources to ensure your family outings are adventurous for all the right reasons.

This app lets parents -

- Search for kid-friendly activities locally - personalize the search as per the age of their kids, category like museums, parks etc.
- Look for kid-friendly food options.
- Check out kids menus, bathroom/changing table options, stroller renting options, kids activities
- Review an exciting place they recently visited with their kids to help out other parents looking for similar options.
- Give a kidventurous rating to this place.

Intended User

There are lots of apps which provide reviews for places nearby but parents often get lost in the millions of reviews trying to see if this is a good place for them to visit with their kids. This app provides the parents with a platform to explore and share child specific reviews with other fellow parents containing things important to parents such as diaper changing stations or kids menus or even special kids programs run by parks.

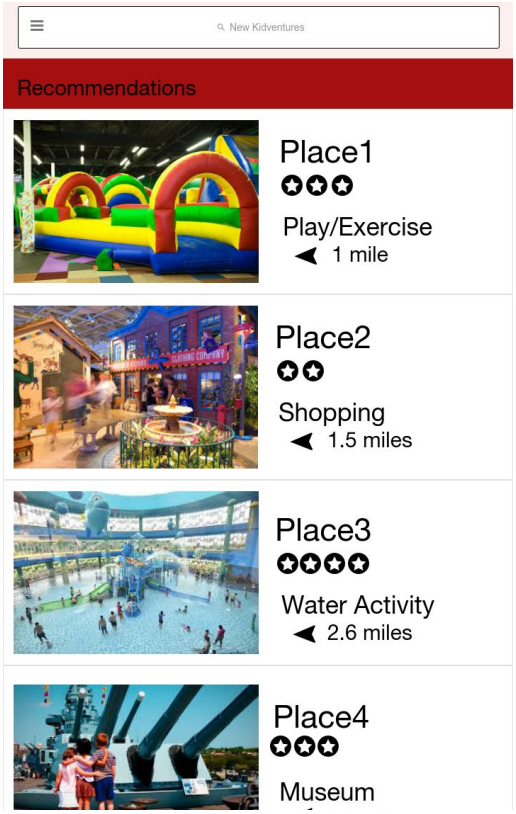
Features

This app enables the parents to :

- Create a profile with preferences such as kid's ages.
- See kid-friendly recommendations nearby.
- Search for nearby places which meet your search criteria - for example museum, water park etc.
- Provide reviews with kid friendly specifics like if the place has a kids' menu, if it has a changing station.
- Option to upload photos of places, menus, maps etc.
- Rate a place as kid-friendly on a scale of 1 to 5 stars.
- Maintain a list of favourite places.
- Maintain personal notes about each place which they might not want to share with the world.

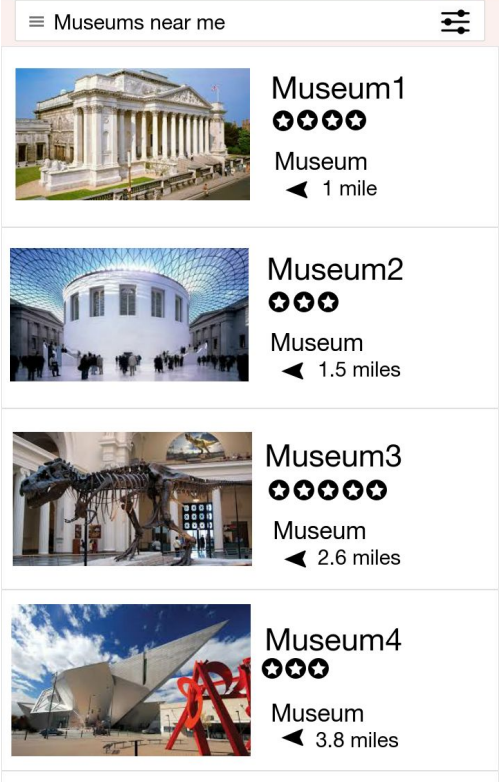
User Interface Mocks

Screen 1 - Homepage



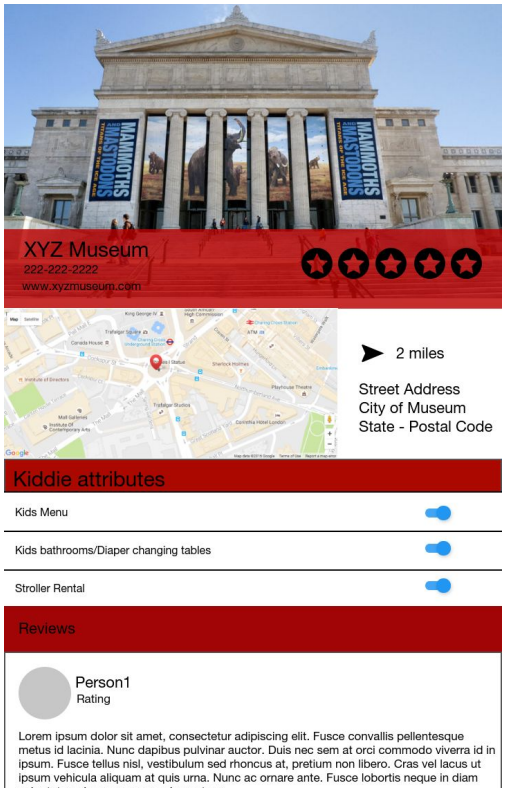
The homepage will display a list of recommended places nearby and a search bar. The user can then enter search keywords to get specific search results.

Screen 2 - Search Results



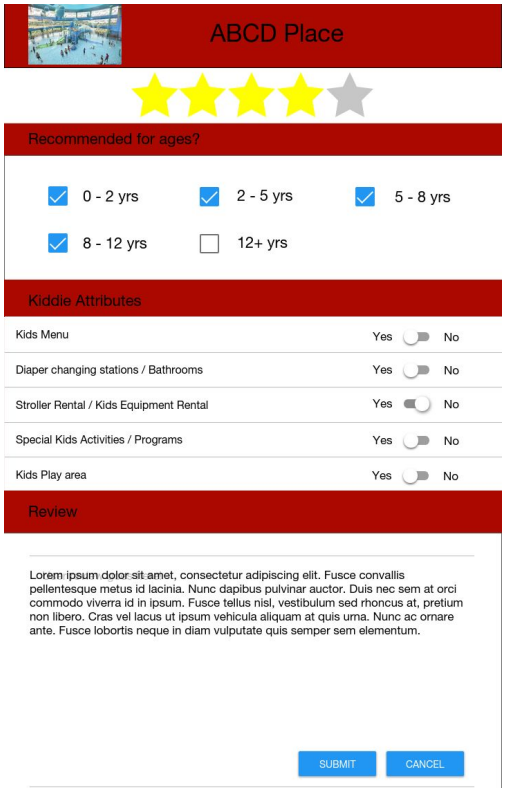
The search results will display a result for specific query by the user and will give an option to the user to filter the results based on certain “Kiddie attributes” like - has kid’s menu, has changing stations etc.

Screen 3 - Place Detail



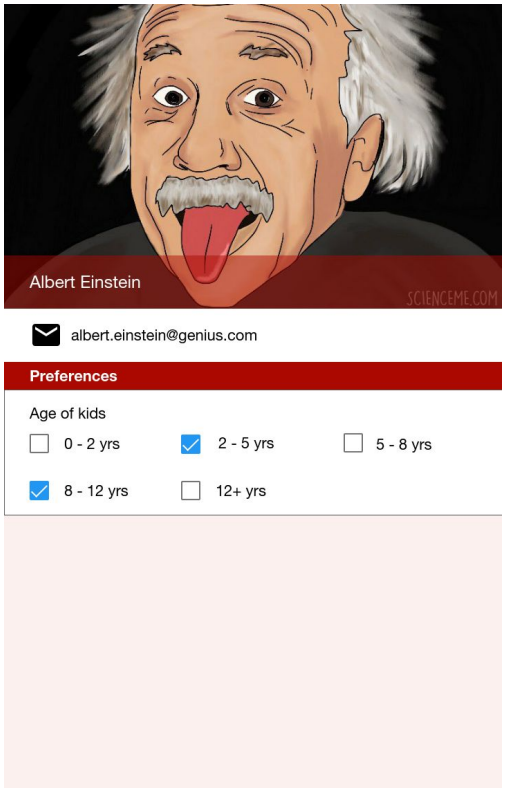
The place detail page will open when a user clicks on a place from either the recommended list or the search results. It will display all specifics about a place including the rating, reviews and any special child related attributes. It will also display the location on a map and the distance from the place.

Screen 4 - Review Page



The review page will open when a user decides to write a review for a particular place. It will have certain child specific questions, a rating and an option to write an in-depth review for a place.

Screen 4 - User Profile



The user profile will maintain a profile picture, an email address used for signing up and some general user preferences which can be used to filter the results for a user or to provide recommendations.

Key Considerations

How will your app handle data persistence?

The app uses Firebase Realtime Database to store the app information such as user profiles, reviews, ratings etc.
It will use Google Cloud Storage to store the photos uploaded by users.
To store personal notes that each user might want to add for a place - which they do not wish to share with the world - the app will use Sqllite database and store the data locally on the user's device.

Describe any edge or corner cases in the UX.

- It the user hits a back button before submitting a review - they will be re-directed back to the Place detail's page for which they had started the review.

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

- Will use Picasso for loading the images from Google Cloud Storage.
- Will use Butterknife for binding views.

Will add to this list if any other library was deemed necessary during project implementation.

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

- Google Location API - to detect the user's current location in order to provide recommendations nearby.
- Google Places API - to extract basic information about the places - like address, phone, website etc. These will be displayed to the user in the place details page along with the child specific recommendations.

Next Steps: Required Tasks

Task 1: Project Setup

The project setup consists of the following activities:

1. Creating a project in the latest version of Android Studio
2. Decide "Target SDK" and "Min SDK" version depending on the features being implemented.
3. Add required libraries to the app gradle file and do a gradle sync.

Task 2: Implement UI (XMLs) for Each Activity and Fragment

Sub-tasks include:

- Build UI for Homepage
- Build UI for Search Results (include tablet view)
- Build UI for Detail Page (include tablet view)
- Build UI for User Profile
- Build UI for Review Page

Task 3: Implement SQLlite database

This includes the following subtasks:

- Create a Contract
- Create a DBHelper
- Create a ContentProvider

Task 4: Implement Firebase Realtime database

Includes the following subtasks:

- Create a firebase project
- Create the database rules
- Create a firebase connector
- Implement Firebase Authentication

Task 5: Implement code for Each Activity and Fragment

Sub-tasks include:

- Implement code to display homepage details
- Implement code to display search page details
- Implement code to display results page details
- Implement code to display user profile
- Implement code to display user entry page review.
- Wire up all the activities/fragments with the databases.

Task 6: Implement Google Services

This includes the following subtasks:

- Implement connection to Location Services.
- Implement connection to Places API.

Task 7: Get data from Google services

This includes the following subtasks:

- Implement methods to get data required from Places API.
- Implement business logic to fetch search results based on query + preferences.

- Implement methods to get data required from Location API.
- Use location data to filter results.

Task 8: Incorporate Material Design elements

This includes the following subtasks:

- Implement Enter/Exit transitions
- Configure - element shadows/elevations.

Task 9: Implement Widget

This includes the following subtasks:

- Implement Widget UI (XML)
- Implement widget business logic.

Task 10: Implement UI and Unit test

This includes the following subtasks:

- Draw up test scenarios
- Implement necessary tests.
