GitHub Username: deepakvadgama

Radhe Krishna Bhakti

Description

This app is for all people who want to progress spiritually by doing *bhakti* (devotion) to Radhe Krishna.

All saints have given a single message of loving God. But attaching the mind to divine is difficult. We are always busy and stressed in this material world performing our duties and the knowledge of this spiritual path is dispersed and sometimes confusing. This app will provide you with clear understanding of path of love. It will help you increase your attachment to God by telling you about God's engrossing tales known as *leelas*. It will inspire you with quotes of saints and will help you practice devotion by participating in chantings.

This app provides:

1. Inspirational Quotes from Saints like Tulsidas, Swami Vivekanand, Kabir etc.
2. Photos of Radhe Krishna and other Hindu deities like Shiva, Ram, Hanuman etc.
3. Engrossing Stories/Leelas of God's incarnations like Krishna, Ram etc
4. Videos of Kirtans
5. Videos of Lectures given by Saints and their disciples
6. Ability to search for information of specific saints or deities
7. Ability to save work/art of your favorite saint or deity.

Note: This app does not endorse or condemn any religion or belief. It simply provides people inspiration to help towards goal of loving God (in this case, God's form of Radha and Krishna). We believe in message of One-God-many-forms and would love to keep adding more content to this app to help everyone.

Intended User

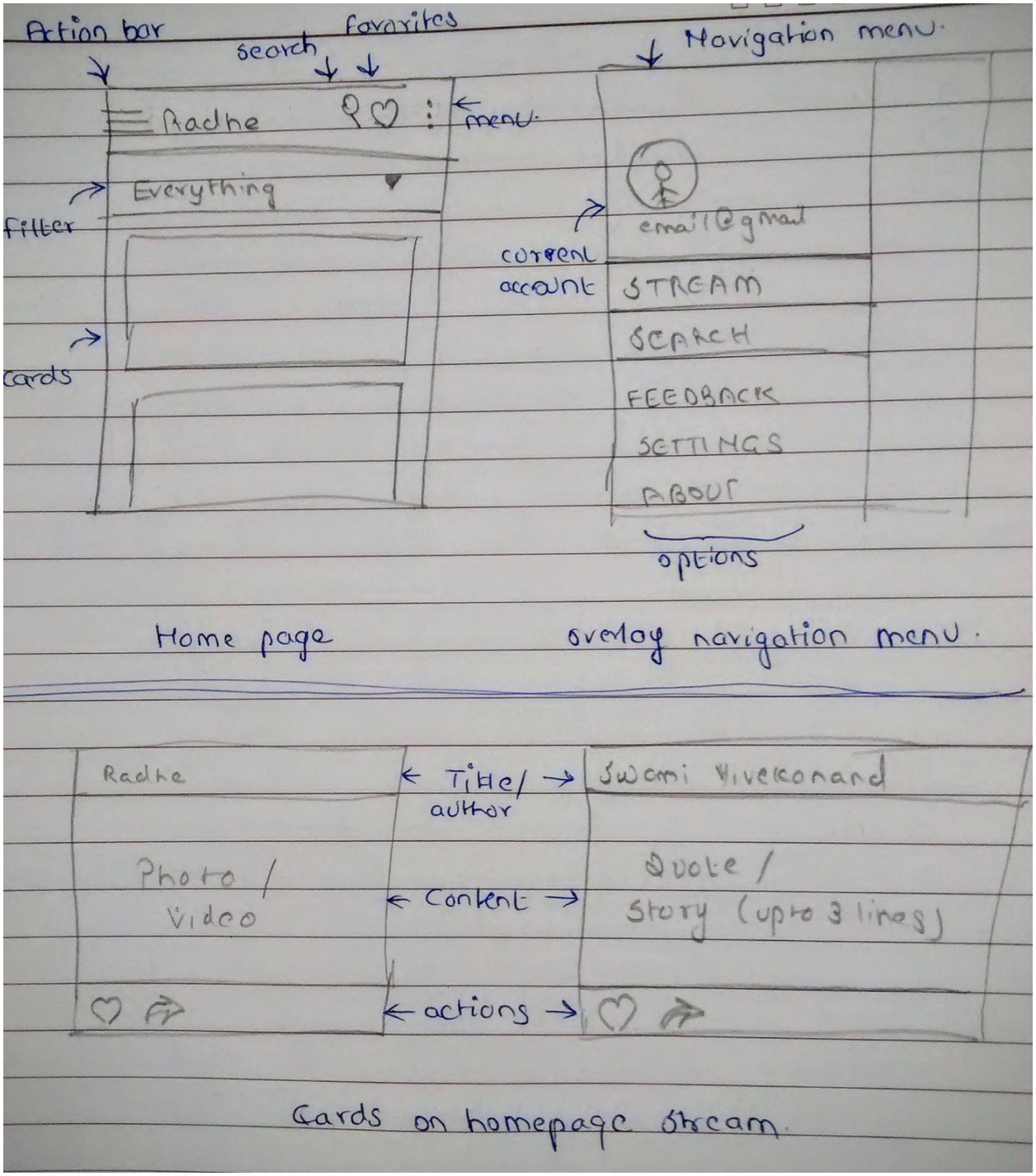
Everyone who is inclined towards spiritual path and want to know more about Radha and Krishna, and various saints.

Features

- Stream of photos, quotes, videos (YouTube only), stories.
- Ability to search for specific content based on author, title and type.
- Ability to save favorite content.
- Ability to use Google account to synchronize favorites across devices.
- Ability to Share content.
- Widget functionality that shows Quotes (updated daily).
- Notifications.

User Interface Mocks

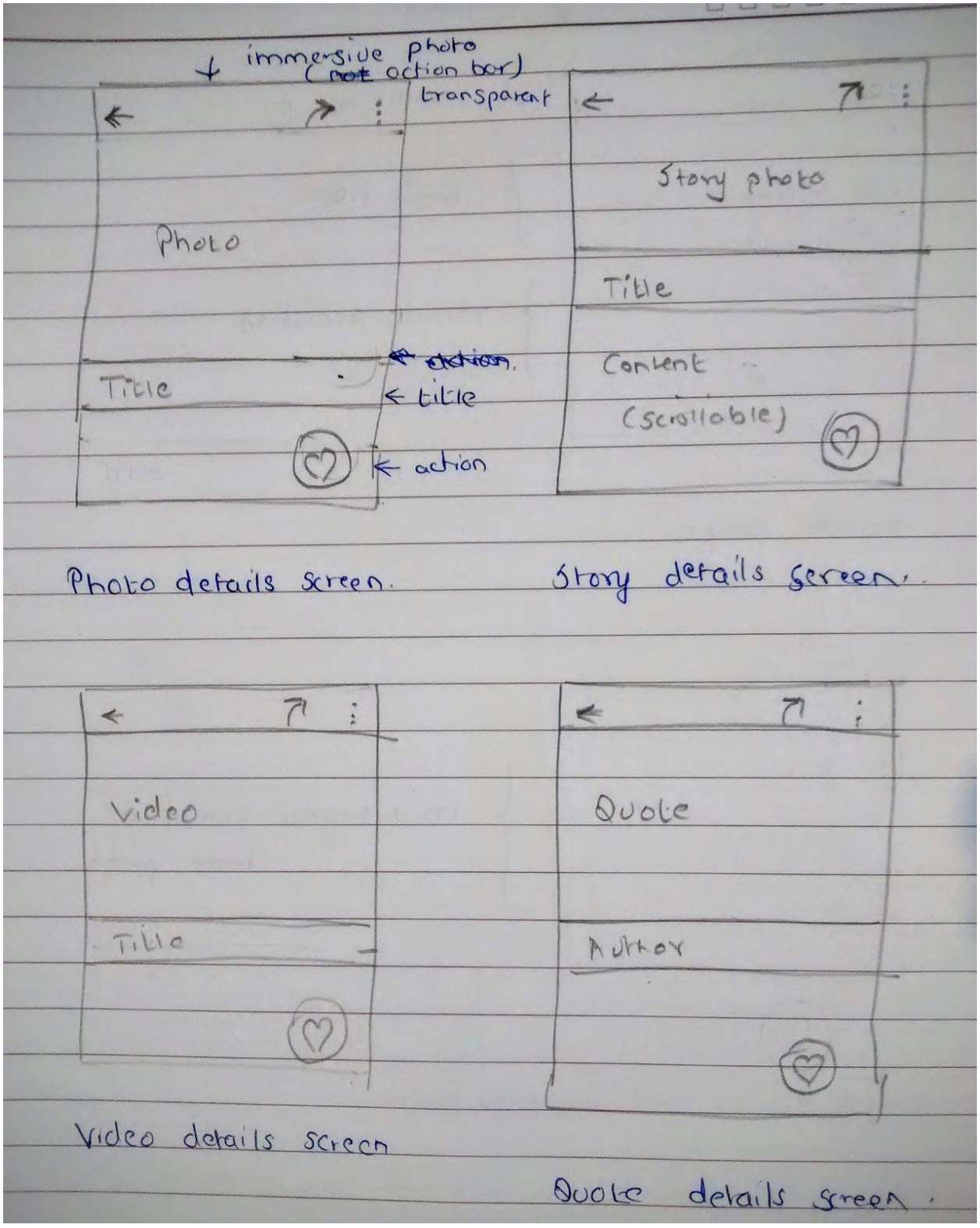
Screen 1: Home page



Home page will contain

- Actionbar with title of the app and favorites, search & menu buttons
- Stream with list of cards showcasing type of either photo/video (kirtan/lectures) or text (story/quote).
- Overlay navigation menu, with options of stream, search, feedback and about.

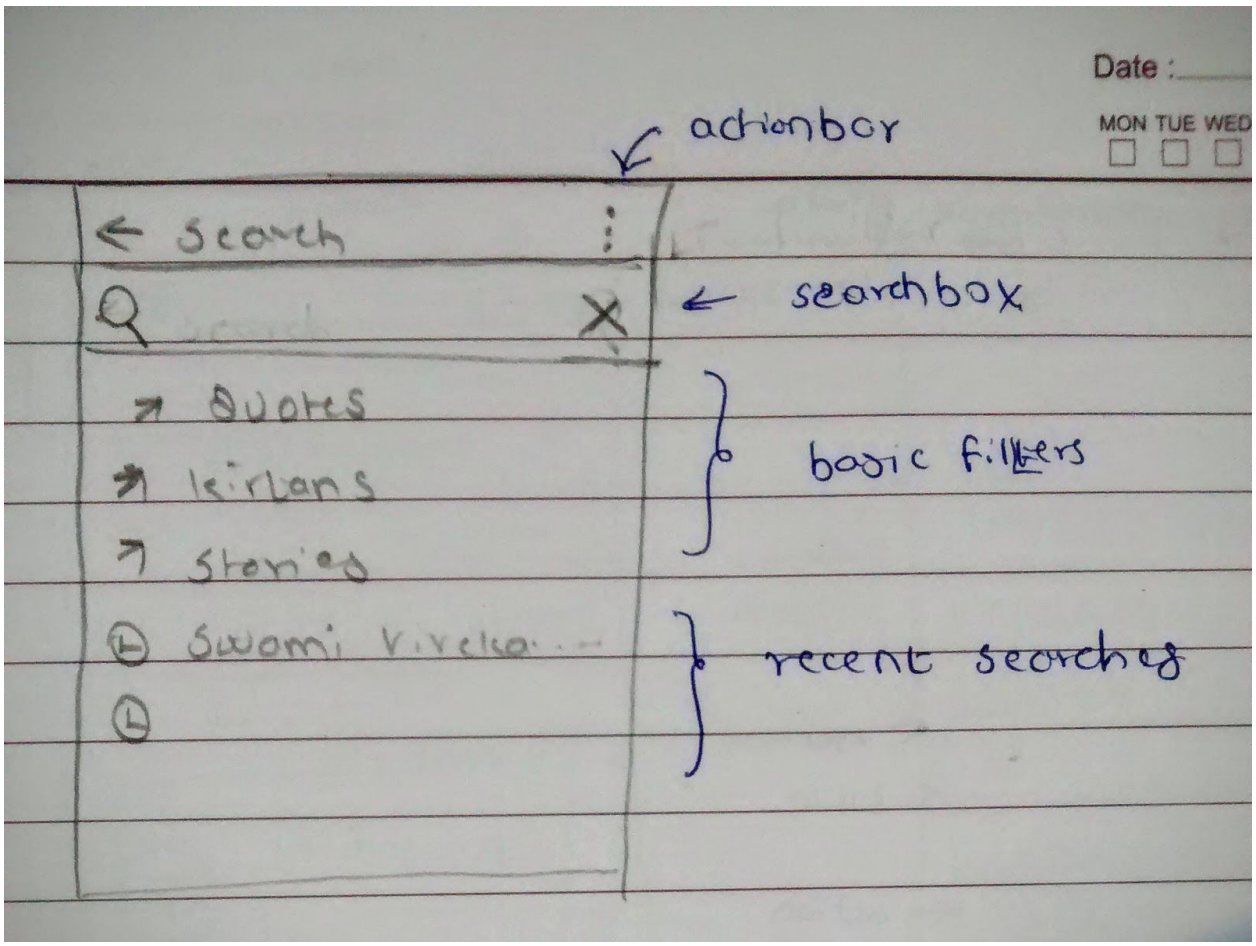
Screen 2: Details Screen



Details screen with specific layout for each type (photo, video, quote and story).

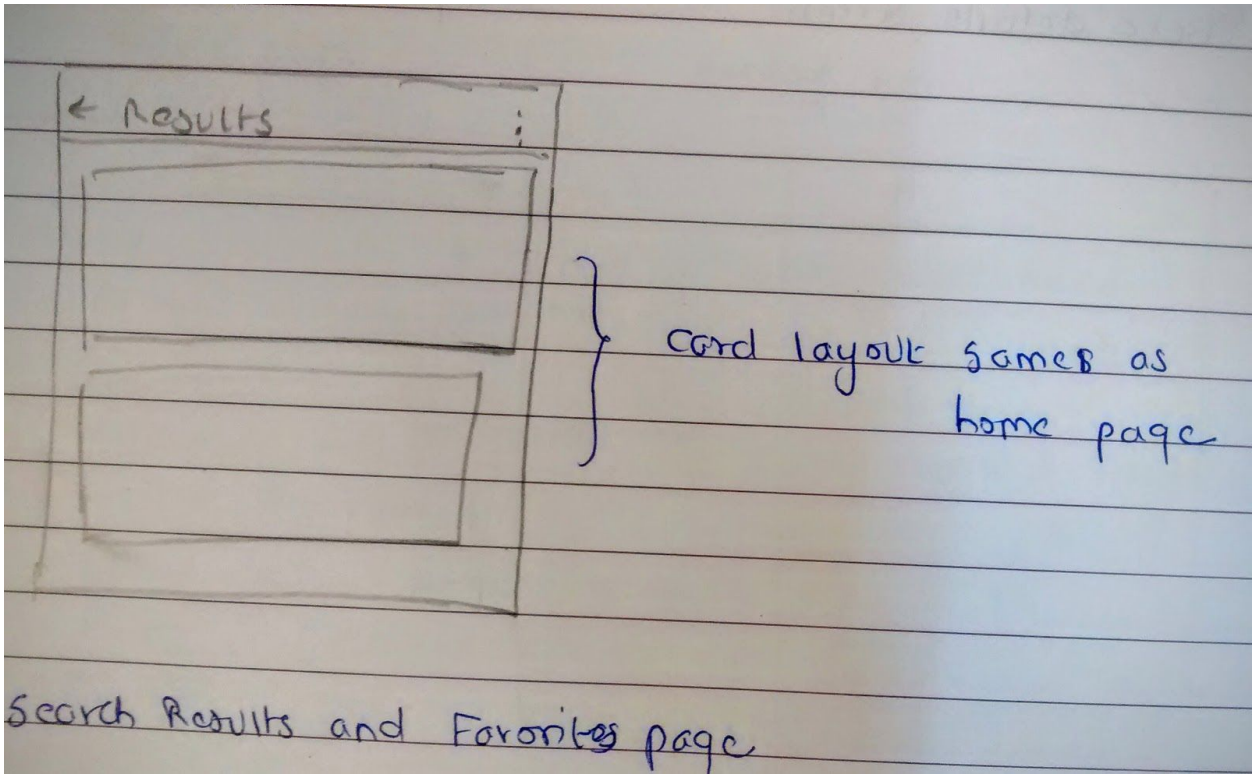
- Photos screen will be immersive with transparent title bar.
- Title bar will have share option and menu button
- Each screen will have a title and subtitle.
- Each screen will have FAB (for adding to favorites).

Screen 3: Search Screen



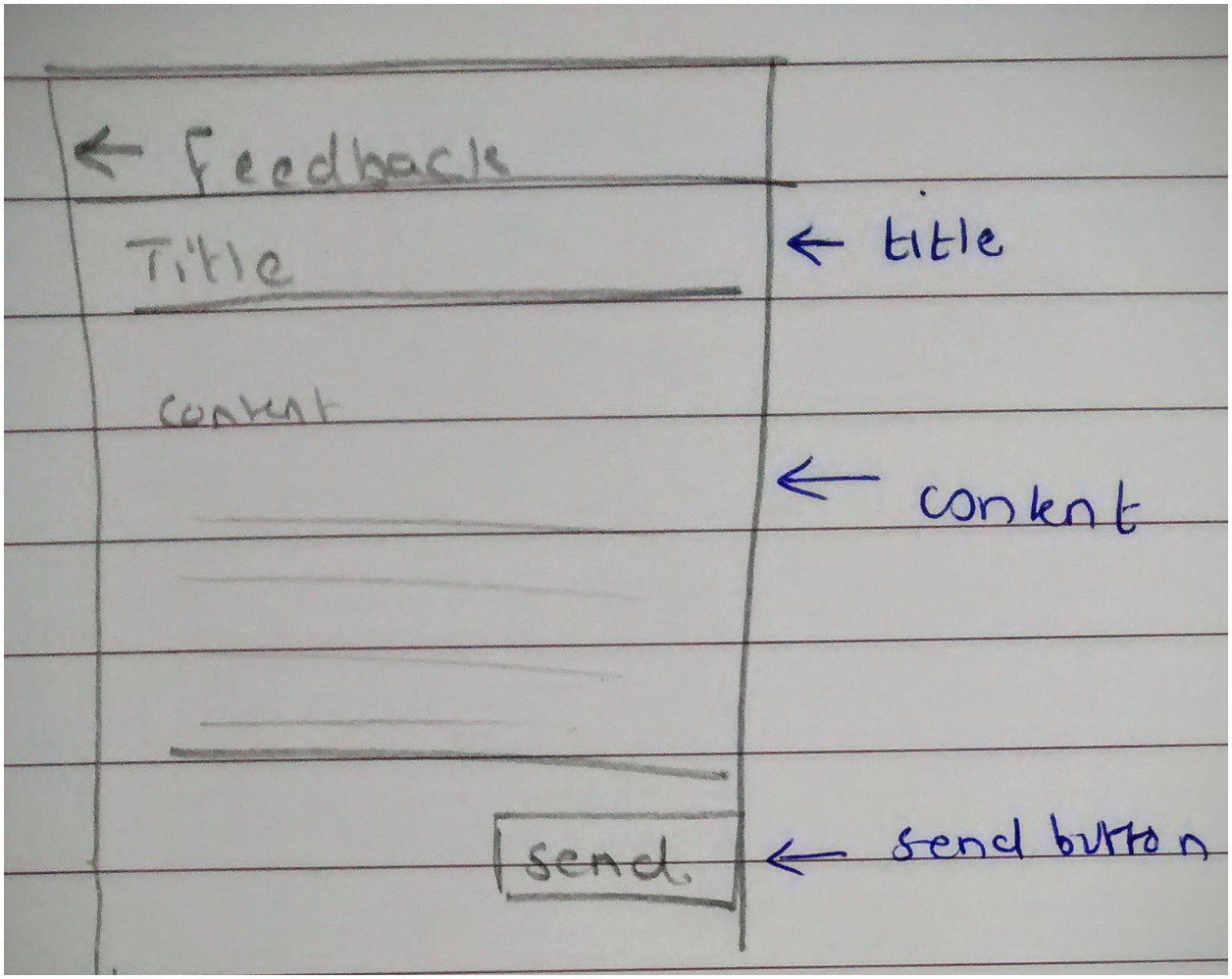
Search screen will have search box, which will allow entering criteria and clearing the same. Below the box there will be basic filter options for types like quotes, kirtans etc, followed by recently searched for items.

Screen 4: Search results and Favorites screen



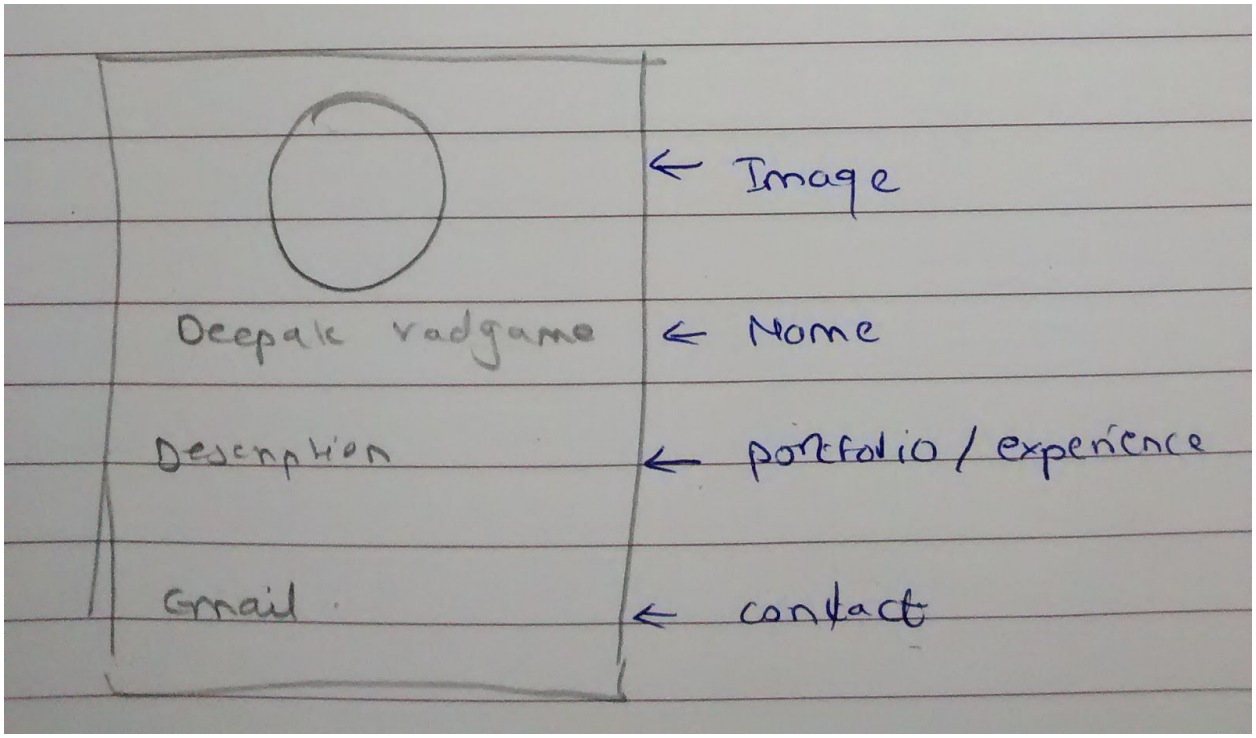
Search Results and Favorites screen will be very similar to home page screen with card layouts. Only the title of the screen will be different.

Screen 5: Feedback screen



Feedback screen will be very basic, with title field, description field and send button.

Screen 6: About screen



About Page will have image of the developer, name, description and ways to contact (email and social media links).

Key Considerations

How will your app handle data persistence?

App will store data in SQL database and use the same using Cursor loaders and Content provider.

Describe any corner cases in the UX.

- Different types of content and images will mean different aspect ratios for the card view. This will cause issue in Tablet (landscape) layout which should be solved using StaggeredLayout.
- YouTube player full screen API's full-screen mode, pause and back buttons need to feel native to the app.
- The details screens have 4 types of content which need to look consistent. A story details screen with scrollable text needs to look and behave (title, fab placements etc) similar to a quote details screen which has too little content.

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

- Retrofit + OkHttp - for retrieving and parsing data from server
- Glide - for loading & displaying images into views with appropriate aspect ratios.
- Butterknife - for reducing the boilerplate code required for binding views.
- Leakcanary - for checking memory leaks in application.
- Android design support - for including material design.
- Google Identity - for using user's identity to save favorites and sync across devices.
- Google Analytics - for analytics
- YouTube - to embed YouTube videos in the 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

Setup the project and configure

- Create skeleton app and link to Github repo
- Configure libraries.
- Configure gradle for instant run / improved speed.

Task 2: Create API on server side

- Create REST APIs
    - To get stream given latest id available with user.
    - To add favorite against a user's gmail id.
    - To get favorites for a user's gmail id.
- Filter to check if gmail id in the requests belong to same device requests are coming from (for security).

Task 3: Implement Content Provider

- Create DB structure
- Create Content Provider

- Create queries to
    - Get the stream
    - Filter based on type of content
    - Search based on criteria
    - Search based on quotes (to be used in Widget)

Task 4: Implement CursorLoader and SyncAdaptor

- Create SyncAdaptor which refreshes the data at regular intervals.
- Implement CursorLoader to load the data.

Task 5: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for Detail activity
- Build UI for favorites
- Build UI for search
- Build UI for Tablets

Task 6: Implement Activities and Fragments

- Code for MainActivity
- Code for selecting user account.
- Code for saving to favorites
- Code for retrieving favorites
- Code for search

Task 7: YouTube

Setup and implement YouTube API

- Register and get developer key
- Import YouTube library and use the API for the player
- Test sample video

Task 8: Material design

- Choose palette, font sizes, paddings etc.

- Update code with AppCompat and respective XMLs.
- Introduce motion between shared elements.
- Add transitions between activities.

Task 9: Settings

Implement settings API to

- Change Google account
- Change notification rate

Task 10: Widget & Notifications

- Create Widget for quotes
- Setup scheduler to change it every day.
- Add notifications to the app

Task 11: Google Analytics

- Setup google analytics
- Add analytics events for all user interactions.

Task 12: Productionize

- Production build variant
- Install release task
- Check memory leaks using leak canary
- Test accessibility and RTL