Capstone_Stage1_Summary

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)
- [Home Screen](#)
- [Search Results Screen](#)
- [Details Screen](#)
- [Favorites Screen](#)

[Key Considerations](#)
- [How will your app handle data persistence?](#)
- [Describe any corner cases in the UX.](#)
- [Describe any libraries you'll be using and share your reasoning for including them.](#)
- [Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)
- [Task 1: Project Setup](#)
- [Task 2: Implement Code to fetch data and verify](#)
- [Task 3: Implement UI for Main Activity and Fragments](#)
- [Task 4: Implement Detail Activity](#)
- [Task 5: Implement Database and Content Provider for Favorites](#)
- [Task 6: Implement Widget and Use Material Design](#)
- [Task 7: Implement Google Play Services](#)
- [Task 8: Generate,Test and Deploy](#)

GitHub Username: paramrajpura

ArxivXplorer

Description

Explore the world of arXiv.org from Cornell University Libray with open access to 1-million+ e-prints in Physics, Mathematics, Computer Science, Quantitative Biology, Quantitative Finance and Statistics.

Intended User

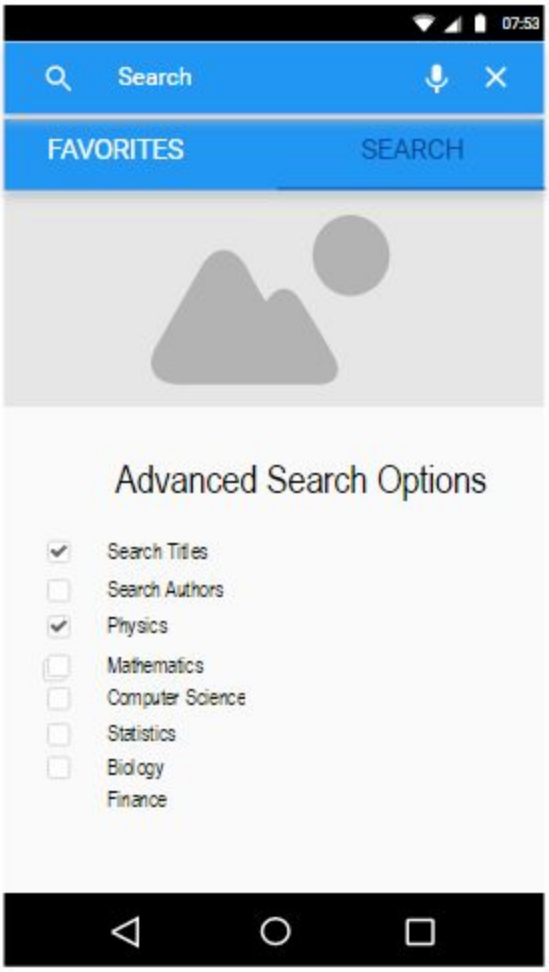
Students, Researchers. Scientists & Academicians from Physics, Mathematics, Computer Science, Quantitative Biology, Quantitative Finance and Statistics background.

Features

- Search options to explore published works on arxiv.org
- Advanced Search for shortlisting based on authors, categories, published date etc.
- Download and view e-paper PDF
- Tag favorites to view and search later.
- Widget for quick access
- Voice to text for search

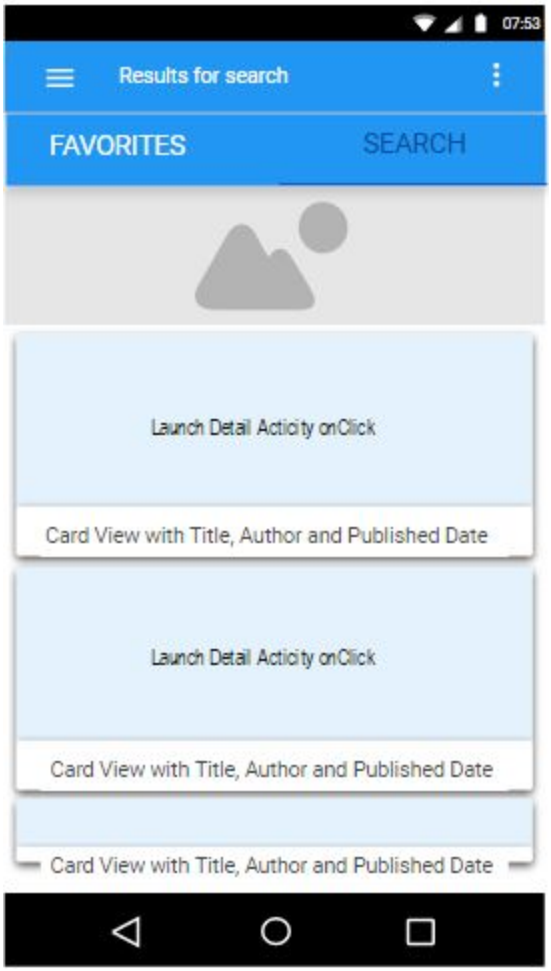
User Interface Mocks

Home Screen



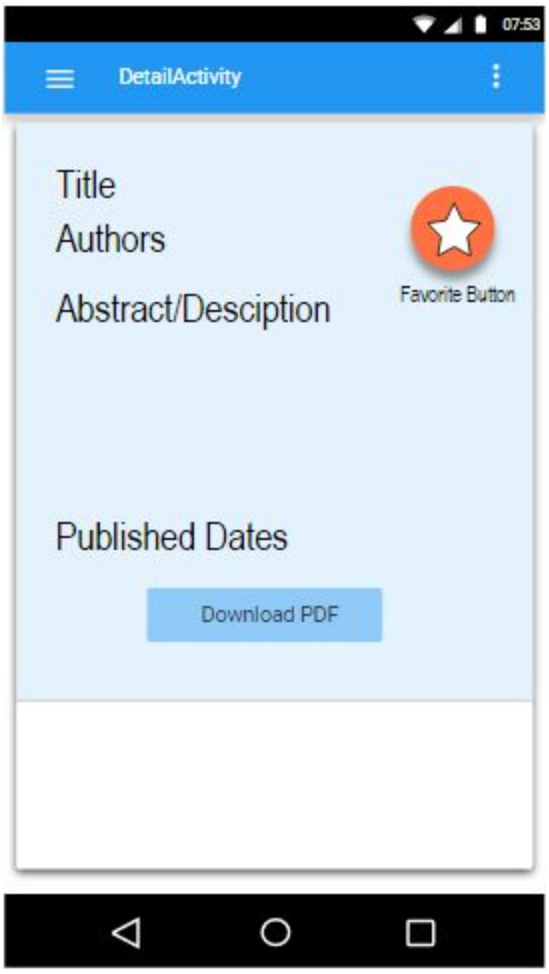
The App's home screen accommodates a search bar to explore the Arxiv repository and the advanced search options to filter the search results with desired requirements.
The Search bar also supports speech to text conversion for inputting search keywords.
Home Screen also accommodates 2 tabs namely Search and Favorites(explained later).

Search Results Screen



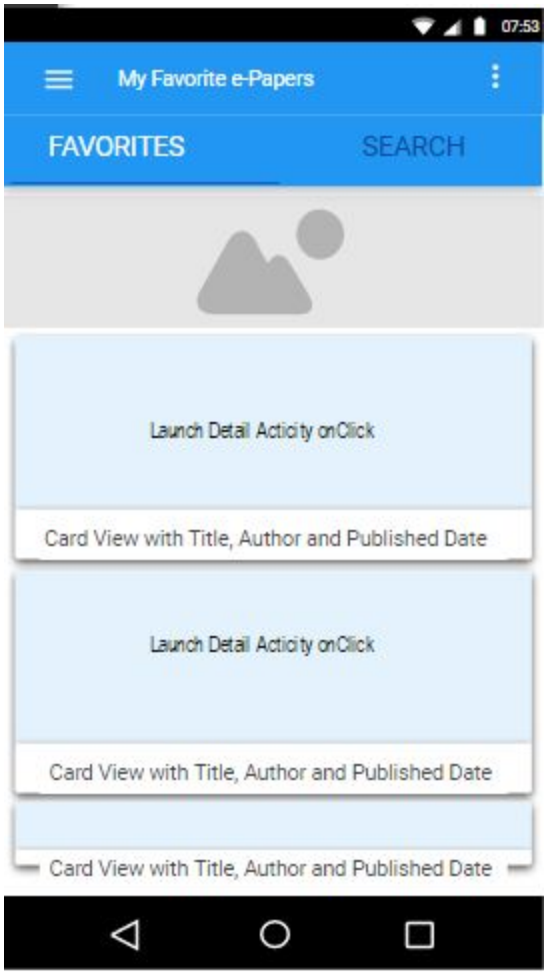
Search Results displays a list of Card-View with details like Title, Author and Published Date of the e-Paper.
The user if interested, clicks on the Card-View to move to the detailed description of the e-Paper.

Details Screen



Detail Screen displays the Description/Abstract of the work published. And accommodates the button to Download and view the complete PDF of e-Paper published on Arxiv.
User may tag the work to ‘My Favorites’ to maintain a list of interesting e-Papers.

Favorites Screen



Favorites screen similar to the Search Result Screen displays the list of User's Favorite tagged e-Papers. Clicking the card-view loads the Detail Screen for the corresponding e-Paper.

Key Considerations

How will your app handle data persistence?

The favorites section requires data to be stored on the device. Content provider with SQL Database will handle the favorites data. Other User settings if required will be stored with Shared Preferences.

Describe any corner cases in the UX.

- E-paper Category: User experience will be defined by how easily user is able to locate the desired e-paper being searched. Reading a category name sounds easiest but not the most user-friendly. Color coding for each category or an image background for card-view shall be visually appealing. Realizing the same is one corner case to handle.
- Widget: This app has “My Favorites” section for the user. Hence a collection widget is an ideal option for user experience. Designing its functionality and usability is a case to ponder over at current stage.

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

- ButterKnife - Easy and clean code development via Annotations for binding views
- Retrofit- safe HTTP client service with easy interface
- Retrofit-convertor-gson & simplexml - Conversion of xml data from Arxiv xml format
- Schematic - Helper library for Content Providers

Describe how you will implement Google Play Services.
Possible Usage:

- Google Analytics: To study the user activities with app and analyze data for further improvement is app.
- Google Identity: To link the user via Google credentials and allow multiple devices for access.

Next Steps: Required Tasks

Task 1: Project Setup

- Setup project and configure libraries
- Design the icons and add to app assets
- Configure devices/emulators to test for various categories (Eg: Tablet,different API levels etc.)

Task 2: Implement Code to fetch data and verify

- Before going to UI, it becomes important to be familiar with the data format and available resources to fetch the xml data from Arxiv. Implement basic API/query, data parsing with manual search keywords using the Retrofit library and simpleframework xml convertor.
- Double check on formats of data to be retrieved and displayed on UI.

Task 3: Implement UI for Main Activity and Fragments

- Add search bar with speech to text, verify with unit test and integrate to fetch and parse data code.
- Implement card view and adapter to display parsed data.
- Setup listeners for the user interaction activities like clicks, scroll.
- Add swipe screen for two tabs on the main activity.

Task 4: Implement Detail Activity

- Implement UI for the detail activities with appropriate buttons and views.
- Handle the inter-activity data transfer via singleton/parcelable based on size of data.
- Set up listeners to user interactions for favorite and read PDF buttons.

Task 5: Implement Database and Content Provider for Favorites

- Setup database and content provider with library to store user Favorites entries.
- Populate the favorites tab on main activity with tagged entries and implement similar behaviours for detail activity.

Task 6: Implement Widget and Use Material Design

- Create collection widget and link it to detail activity for favorites entry.
- Use material design guidelines for the app and add modifications for tablet version.
- Add accessibility features and RTL support.

Task 7: Implement Google Play Services

- Use a sign in option to sync the data with Google id with Identity services.
- Use Analytics to study user interactions with app.

Task 8: Generate,Test and Deploy

- Generate App flavours and required build versions.
- Test the app by debugging and final validation.
- Double check the requirements for Capstone.
- Generate APK and complete procedure for deploying to Play Store