

# FB PAGER: ANDROID APP FOR FACEBOOK

Dr. Ajay Mathur, Shubham Mathur

Head of Dept., Student, B.Tech.Final (Computer Sc. & Engg.) Ramaiah Inst.of Tech.,Bangalore  
Computer Sc. & Engg,  
Govt. Polytechnic College, Jodhpur (Rajasthan), India

**Abstract :** FB Pager is an app which is helpful for user as well as the public. In the current system the user have to download and check the post manually. It is very time consuming and costly. Our FB Pager deals with the various activities related to the Facebook post. The FB Pager downloads all the post from the Facebook page with images and likes. The user has to add a page using id to app. The app will be developed on Android platform. Android is a most popular mobile devices software platform and operating system. It based on the Linux kernel, and developed by Google.

**IndexTerms - facebook, Android, fb lite, FB Pager**

## I. INTRODUCTION

Facebook is not only a website but it is now a platform to stay connected. Therefore we chosen our project related to facebook. It is to be developed on android platform. Android is a most popular mobile devices software platform and operating system. It based on the Linux kernel, and developed by Google. Managed code can be written by the developers using Java language, that allows controlling of the device via Google-developed Java libraries. The FB Pager downloads all the post from the facebook page with images and likes. The user has to add a page using id to app. The app will be developed on Android platform. Android is a open source software. Google has made the entire source code (including network and telephony stacks) open that were not available previously, under an Apache license. Certain parts can't be made open (related to a specific hardware) and are not considered part of the Android platform. Proprietary extensions may be added freely with Apache License without submitting those back to the open source community. The Google's contributions to android may remain open-sourced, while the branches could impose varieties of licenses.

## II. LITERATURE SURVEY

The app is very useful as it provides the facility to get contents without logging in every time. The app will fetch the images, post and likes from the previously added pages. The user will be able to download and share these things. The facebook website and facebook lite both requires the user to log in. These two provides all the functionalities. The proposed FB Pager can be used without logging in. The FB Pager is only deals with images and likes. The user has to add the page using Id or username. After that the FB Pager will automatically download all the images and likes[1].

In Joinson's study, the facebook's frequency of use - that is, returning back to Facebook multiple times a day - was found to be bounded with *surveillance gratifications*. This involves various things such as looking at user-generated content, including photographs and status updates. Similarly, Hart reported that the significant variable in a model predicting the frequency of Facebook use in both undergraduate and high school students is entertainment gratification. These results implies that passively engaging with social or entertainment-related content on Facebook can motivate users to return to the site frequently.

Interestingly, Papacharassi and Mendelson found that a greater affinity with the site is developed for people who used Facebook more frequently especially when they used it to escape from negative emotions.

Joinson also found that the time duration spent on Facebook on a day was predicted by what a user referred to as *content gratifications*, which involve engaging in non-socially oriented Facebook activities (i.e., playing games, searching applications, and completing quizzes).

## III. SOFTWARE REQUIREMENT SPECIFICATIONS

### 3.1 Product Overview

The app is very useful as it provides the facility to get contents without logging in every time. The app will fetch the images, post and likes from the previously added pages. The user will be able to download and share these things. The Facebook website and Facebook lite both requires the user to log in. These two provides all the functionalities. The proposed FB Pager can be used without logging in. The FB Pager is only deals with images and likes. The user has to add the page using Id or username. After that the FB Pager will automatically download all the images and likes.

### 3.2 External Interface Requirements

The Google API is used for this application, which make it mandatory that it follows the terms of use specified by Google. The software should be designed in such a way that it never disclose any personal information of any users, and also should collect no personal information from its own users. This software is proposed to be utilized and will be designed to run on any Android operating system version 4.1 or higher. The software will provide the concept of forward compatible for all currently released Android operating system versions

### 3.3 User Interface

The Layout Editor provide the facility to quickly build layouts by dragging UI elements into a visual design editor instead of writing the layout XML by hand. The design editor allows the user to preview the layout on different Android devices and versions, and it also provides the dynamically resize the layout to be sure it works well on differently screen sizes. The Layout Editor is much powerful when building a new layout with Constraint Layout—a layout manager provided in a support library that's compatible with Android 2.3 (API level 9) and higher (Figure 3.1).

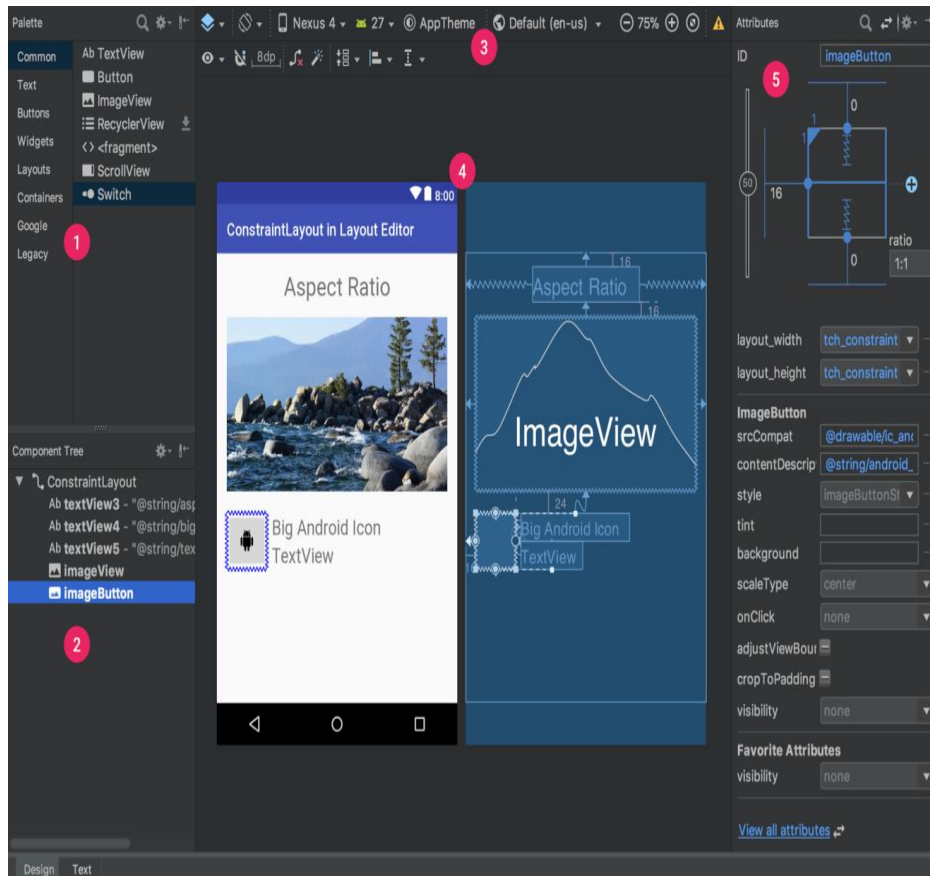


Figure 3.1: UI layout

1. Palette: This allows the user to view list of views and view groups that we can drag into layout.
2. Component Tree: It shows the View hierarchy for layout.
3. Toolbar: It consists of Buttons to configure the layout appearance in the editor and allows to change some layout attributes.
4. Design editor: Layout in Design or Blueprint view, or both.
5. Attributes: These are Controls for the selected view's attributes.

### 3.4 Hardware Interfaces

The FB Pager is intended as a mobile application for the Android platform and hence is solely supported on Android-powered devices. This produces the illusion of peer-to-peer interactivity between mobile and Facebook website. FB Pager is being developed specifically for Android 8.1 (Oreo) and all versions released after it. Information will be sent using TCP/IP and the HTTP protocol. All the network communication interfaces are provided as abstractions in the Android platform.

### 3.5 Software Interfaces

- 1 Operating System: Windows 7 or above
- 2 Kotlin: As the Android provides supports for Kotlin, the application was built on it.
- 3 Google Android SDK 3.0: The Android platform provides a software stack for mobile devices that includes an operating system, middleware and key applications. The Developers can easily create applications for the platform using the Android SDK. It also allows the applications to be written using Java programming language.
- 4 XML: The XML is a simple, easy, and very flexible text format which is designed only to carry data, not to display data.

### 3.6 Communication Interfaces

Android system's framework includes Handler as a part for managing threads. A handler object gets messages and executes code to handle the messages. It allows to create a handler for a new thread, but it also allows to create a handler that's connected to an existing thread. When a user application connect a handler to its UI thread, the code that handles messages runs on the UI thread.

### 3.7 Functional Requirements

FB Pager is an app which is helpful for user as well as the public. In the current system the user have to download and check the post manually. It is very time consuming and costly. Our FB Pager deals with the various activities related to the Facebook post. This following list illustrates different functions provided by the application:

1. **User page adding:** The user must be able to add a new page.
2. **Sharing media:** The user must be able to share the images and other media.
3. **Downloading Media:** The user must be able to share the images and other media.
4. **Page organization:** The user must be organizing the various portion of screen.
5. **Theme:** The must be able to apply various themes.

## IV. METHODOLOGY

### Software Process Models

There are mainly two well known methodology followed by the developers for Mobile App development: Waterfall and Agile.

**4.1 Waterfall Methodology:** – It is a sequential design process. In this development a steadily downwards approach like a waterfall is followed. It starts from the conception till the conclusion of the software. It is a traditional approach in which the product is developed exactly as per the customer expectations.

**4.2 Agile Approach:** –The Agile software development is fast, quick and iterative. In this the tasks are subdivided into small phases of work, and frequent assessment and adaptation to plans is done.

This project was developed using the Agile Approach, because mobile apps have shorter development life cycles, need frequent changing demands, need frequent updates, and provides the ability to quick download.

Agile methodologies have is efficient and helpful to mobile app development environment. It provides all these characteristics appropriately as it is more flexible, while traditional methods are costly and there is very less scope of change. The agile methodology is very adaptive, because it helps user to create apps that are seamless, very quick, very small in size and easy to work upon. The Agile development allows the app more stable with less no. of errors, thus increasing the overall quality.

## V. DESIGN

There will be three modules

1. **Splash:** In splash screen we hit a api that give us access\_token we require in future, that access\_token is stored in shared preferences.
2. **Home Screen:** In this we opening fragment that having the page info. In this page we hit a api to load side navigation pages list icon and names based on their username. There is a plus icon in toolbar & navigation by that we storing user page data in shared preferences.
3. **Detail Page:**The user can download or share image. The shared image is first download then ready to share. The Image will be downloading in FB Pager folder in internal Storage.

### 5.1 Architecture Design

The Android apps have a very much complex program structure as compared traditional desktop counterparts. In traditional desktop application the majority of cases have a single entry point from the launcher shortcut and run as a single monolithic process but a typical Android app is build using multiple app components that includes various activities, services, content providers, fragments, and broadcast receivers.

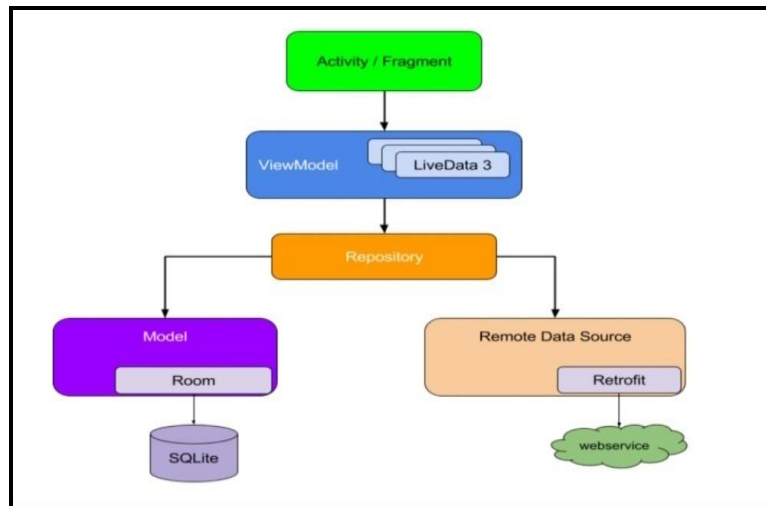


Figure: 5.1 Architecture Design

5.2 Class Diagram

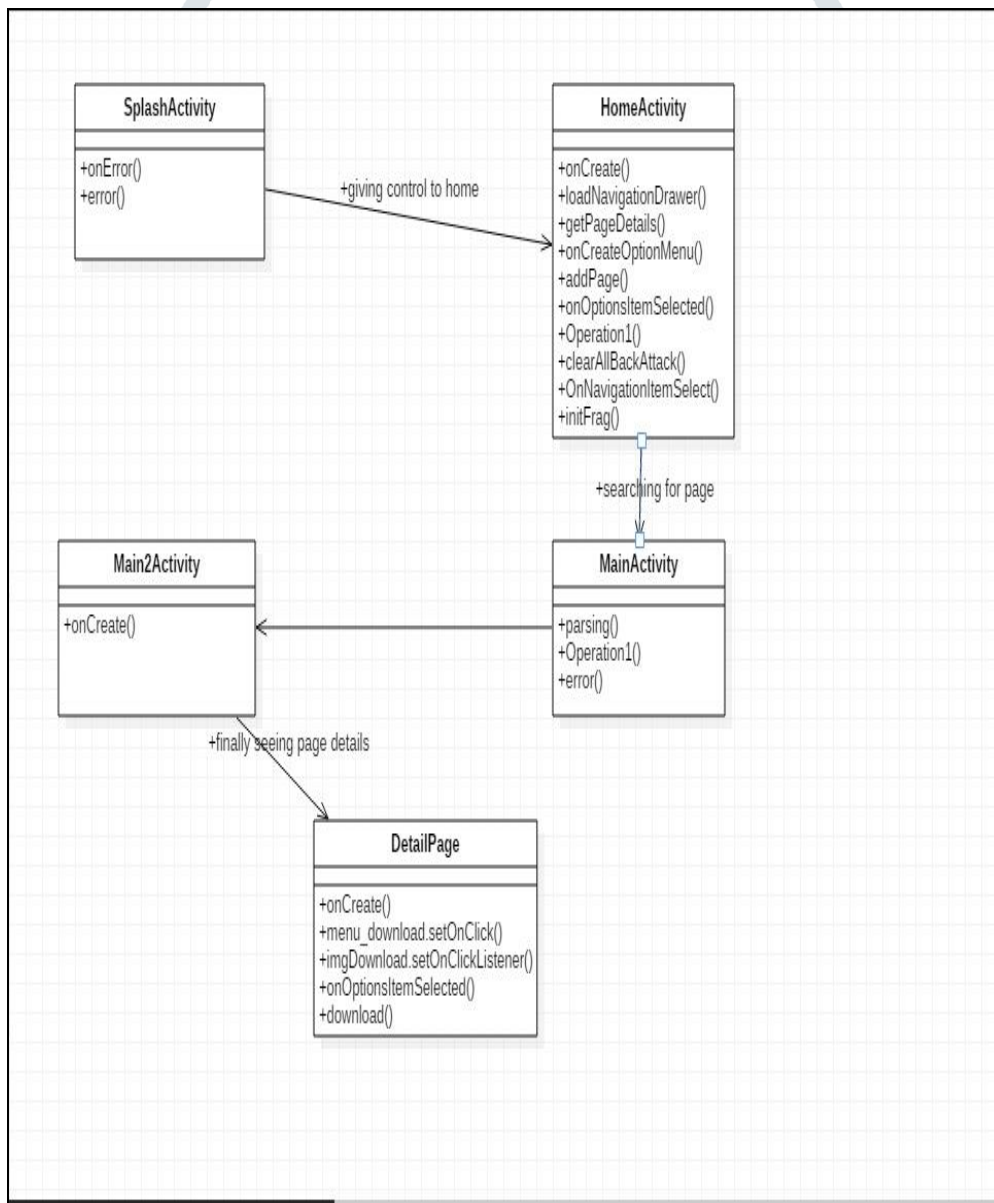


Figure: 5.2 Class Design

### 5.3 Sequence Diagram

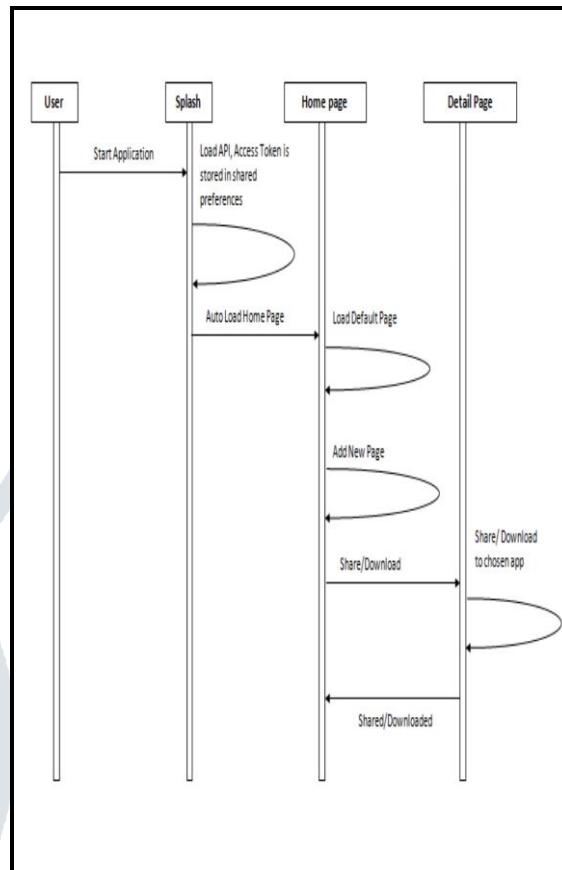


Figure: 5.3 Sequence Diagram

### 5.4 Data flow diagram

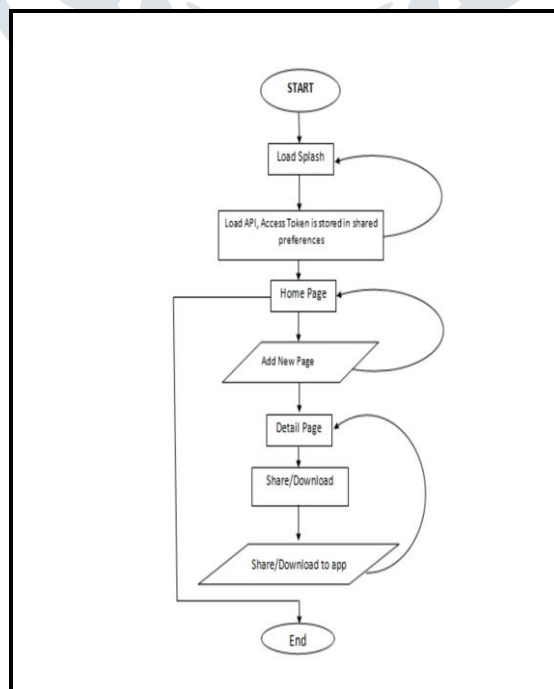


Figure 5.4 Data flow diagram



## VI. IMPLEMENTATION

### 6.1 Tools Introduction

**Android Studio:-** Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on INTELLIJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps.

**Kotlin:-** The official language on Android is Kotlin. The best feature is that it is interoperable with the existing Android languages and runtime. It's other feature includes, expressive, concise, and powerful language.

#### *Overall view of the project in terms of implementation*

The third phase of this project is the implementation phase. In this phase all the planned work is performed. The main things needed during implementation are to maintain control and communication. Our project is implemented on android studio. We have used kotlin as language as it is much better integrate with android studio. Continuous monitoring and appropriate adjustments are made during the project and same are recorded as variances from the original plan. This is the step where a project manager spends most of the total time. During project implementation, tasks are carrying out the by the team members, and progress information is being reported through timely team meetings. The information is used by project manager to maintain control over the direction of the project by making a comparison of the progress reports with the project plan to measure the performance of the project activities and take corrective action as needed. The first action is always to bring the project back on time (i.e., to return it to the original plan). If this cannot be done, the team should record variations from the original plan and record and publish modifications to the plan.

### 6.2 Algorithm

1. In splash screen we hit a API that give us access\_token we require in future, that access\_token is stored in shared preferences.
2. Home Screen we opening fragment that having the page info.
3. In this page I hit a API to load side navigation pages list icon and names based on their username.
4. In the Fragment page We hit 2 API using that access\_token we stored in shared preferences.
5. 1st API give us the page info that we code in Fragment.
6. 2nd API give us the page's photos list that We code in Fragment.
7. We load images in recycler view Fragment.
8. On click of any image it navigate to detail page.
9. There is a plus icon in toolbar & navigation by that we storing user page data in shared preferences.
10. The user can download or share image.
11. The shared image is first download then ready to share.
12. The Image downloads in FB Pager folder in internal Storage.

## VII. TESTING

Testing is a process to identify and verify the correctness, completeness and quality of developed computer software. The main aim of testing is to uncover the hidden error which are not found until the software is run against the environment.

### 7.1 Testing Tools and Environment

**1. Build Local Unit Test:-** For this purpose we have first tested our app on our local machine with the help of android studio. android studio provides full android environment on which we can test our app.

**2. Build Instrumental Unit Test:-** For this purpose we have tested our app on our android phone . The Android Testing Support Library was used for testing. This library provides APIs that allow developer to quickly build and run instrumented test code for apps. The Testing Support Library includes APIs for functional UI tests.

## VIII. CONCLUSION & SCOPE FOR FUTURE WORK

The project is implemented successfully on the platform “android studio”. Our kotlin plugin is working fine ,and all the API’s used in this project are working perfectly fine.

We have implemented our project on our personal machine via emulator as well as on a android phone. It is working perfectly fine.

The hypothesis which was that Energizer of this project is last the longest in all of the devices tested. The results we got also do support our hypothesis.

In today’s information and digital world

- The era of mobile technology is widely based on the android app .
- The smart phones are emerging The websites are now vanishing. This is time to change from conventional websites to apps(mobile app) which has become the part of daily life .
- We are introducing “FBPager.apk” (a miniature of our Facebook) the android application software.

We think the tests we did went smoothly and had no problems.

The user has to add the page using Id or username. After that the FB Pager will automatically download all the images and likes

An interesting future study might involve adding more features like chat option, to enhance the app and make it faster

## REFERENCES

- [1] <https://code.facebook.com/posts/1365439333482197/how-we-built-facebook-lite-for-every-android-phone-and-network>
- [2] [http://www.tutorialspoint.com/android/android\\_sqlite\\_database.html](http://www.tutorialspoint.com/android/android_sqlite_database.html)
- [3] <http://www.androidhive.info/2011/11/android-sqlite-database-tutorial/>
- [4] <https://developer.android.com/training/index.html>
- [5] <https://developer.android.com/studio/index.html>
- [6] <https://developer.android.com/topic/libraries/architecture/guide.html>

