GitHub Username: imabhishekkumar

Back 2 College

Description

An app that bridges the gap between the teacher and the students in a college. It can be used to share important announcements, documents, images to the students and students can pin important once in their home-screen.

Intended User

Students and Techers of a college.

Features

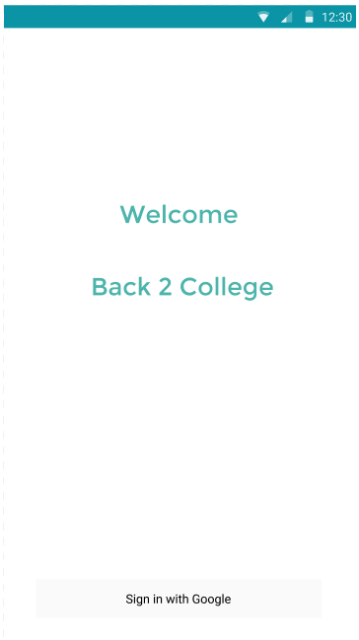
- Allows users to sign in via Google
- Allows teachers to share important announcement, documents, images, etc. to students grouped by department of study
- Notifies users when a new information is shared.
- Allow users to save the document/ image

- Announcement can be pinned to home screen as a widget.

User Interface Mocks

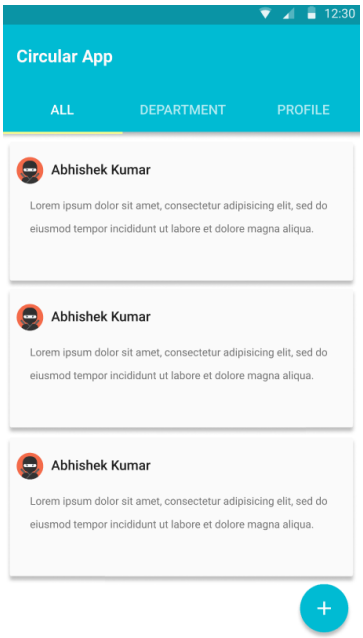
These mockups were created on Figma.

Screen 1



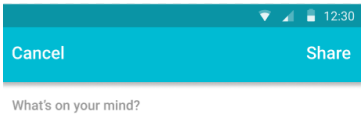
This is the login screen where the users can sign in to the app using Google Sign-In.

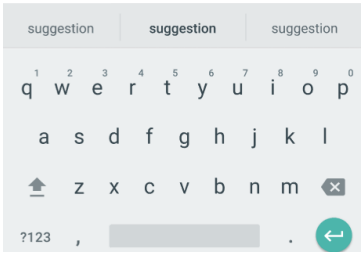
Screen 2



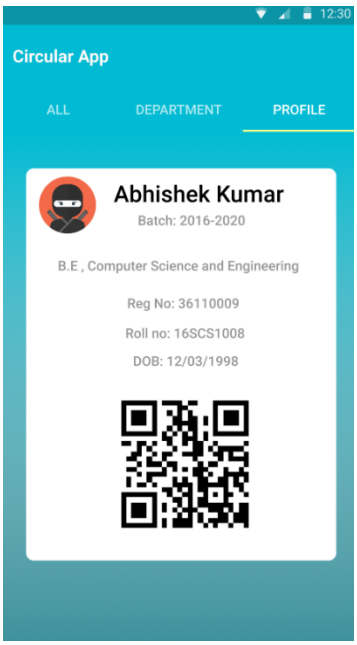
This is the screen where all the announcements, pictures, documents will be displayed once the user is logged in after registering to the app.

Screen 3



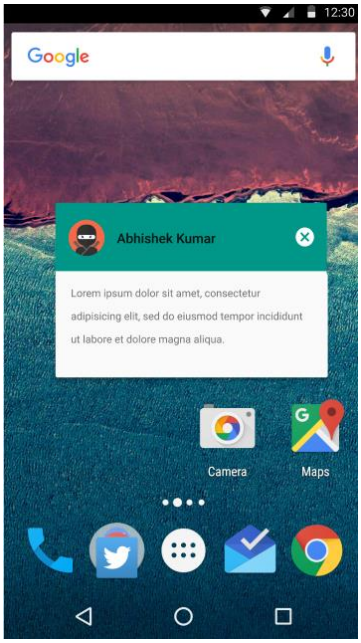


Screen where teachers, college authorities, etc. can add important information for all users to see.



Each user will have a profile where their information can been seen and a unique QR code is generated using register number of the user, which can be used to retrieve information(if needed for attendance purpose, event registration ,etc.)

Screen 4



User can place widgets to their homescreen for a particular post.

Key Considerations

How will your app handle data persistence?

Data persistence will be handled using Firebase Firestore and the Google SignIn using Firebase Authetication. Data will be fetched from Firestore using Firestore queries and will be populated using a recyclerview. MVVM architecture will be used in this application. Fetched details will be stored locally in the SQLLite database to support offline services.

Describe any edge or corner cases in the UX.

- User will be able to access the app offline. He/She will be notified in cases where app does not display real-time data.
- It will be ensured that the app does not consume too much battery.

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

- Picasso to handle the loading and caching of images.

- Various Firebase libraries like Firebase Firestore, Firebase Auth, Firebase Cloud Messaging (for notifications), Firebase Cloud Functions (To trigger notification when any information is added to Firestore).
- Expandable TextView – to show/hide long texts in user's feed.
- Circular ImageView – in profile avatars and post avatars.

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

Google Play Services will be used to implement Google Sign-in feature to the app.

Common project specifications

- App will be solely written in Java programming language
- App will use stable versions for all libraries, Gradle plugin and Android Studio IDE

Library/Tool used	Version
Android Studio	3.5.0
Gradle	5.4.1
Butterknife	10.2.0
Firebase Firestore	21.1.1
Firebase Messaging	20.0.0
Firebase Storage	19.0.1
Firebase Authentication	19.0.0
Gilde	4.9.0

- App keeps all strings in a strings.xml file, dimensions in dimens.xml and supports RTL layout switching on all layouts for accessibility.
- App will conform to material design guidelines

Next Steps: Required Tasks

Task 1: Project Setup

- List required dependecies
- Configure libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Login Activity
- Build UI for MainActivity with tapped fragments and design each fragment.
- Build UI for AddPost Activity
- Build UI for widget

Task 3: Implementing Google Sign In

- Implement google sign in and out feature using google services and firebase.
- Handle authentication check in each activity.

Task 4: Implement Firestore and Firebase Cloud Storage

- When the user add any post it should be stored in firestore and cloud storage(if required)
- The data will be fetched using queries and will be displayed in the MainActivity using RecyclerView.
- Notification will be sent to users about a new post using Firebase Cloud Functions and Firebase Cloud Messaging.
- Users will be able share posts using the IntentService.

Task 5: Create build variants

- Create required build variants
- Create keystore for release apk