GitHub Username: alfianlosari

PixaNote

Description

Pixanote is an app where you can snap a picture of a note with your camera and extract text from the picture and share it with other users.

Intended User

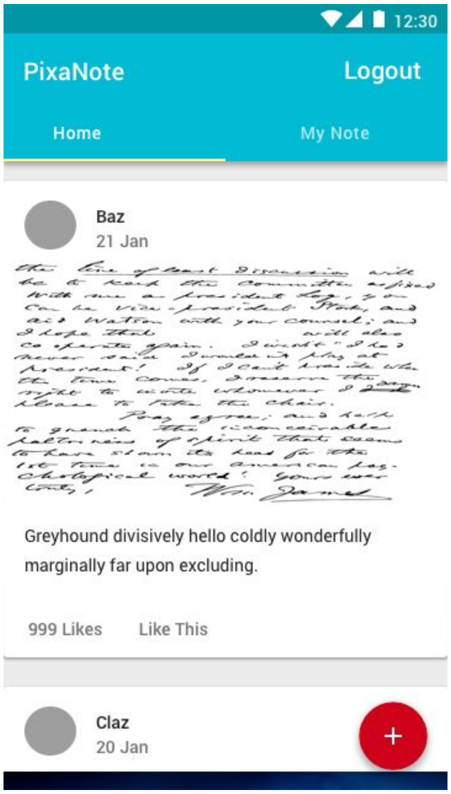
People that want to share notes with others easily by just snapping a picture of the notes

Features

- Snap picture of note and extract text from the note
- Share note with other users
- Like other user notes

User Interface Mocks

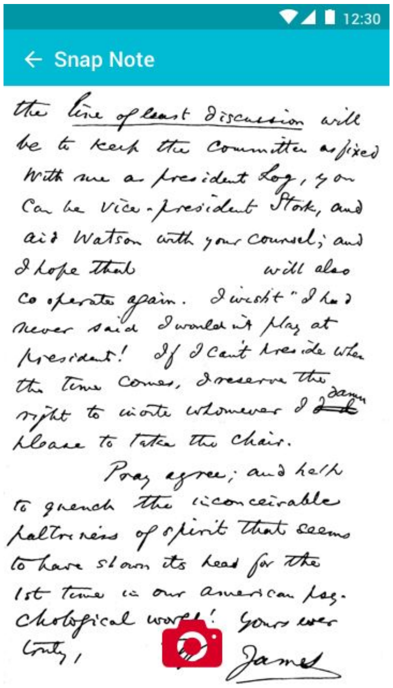
Screen 1 - Main Screen



Greyhound divisively hello coldly wonderfully marginally far upon excluding.

Main Screen that is divided by 2 tabs, home is all the latest notes by other users, my note is the notes taken by the logged in user himself. User can like other people notes by tapping on like this. Like count will also displayed under each PixaNote.

Screen 2 - Snapshot Screen

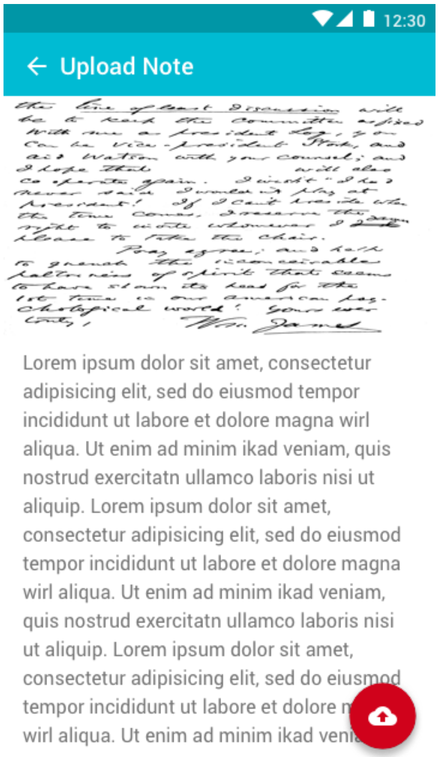


the line of least discussion will
be to keep the Committee as fixed
With me a president Log, you
Can be Vice-president Stork, and
aid Watson with your counsel; and
I hope that will also
Co operate again. I wish't "I had
never said I wouldn't play at
president! If I can't preside when
the time comes, I reserve the
right to invite whomever I damn
please to take the chair.

Pray agree; and help
to quench the inconceivable
paltriness of spirit that seems
to have shown its head for the
1st time in our american psy-
chological world! Yours ever
truly,
James

Screen where user will use the camera display preview to snap their note

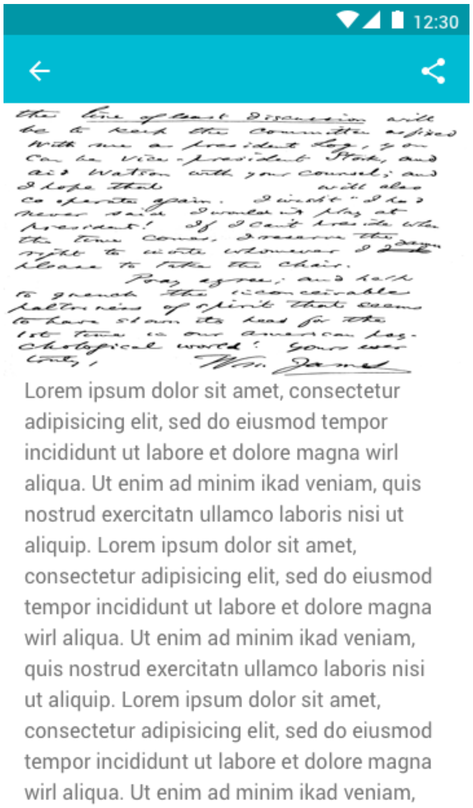
Screen 3 - Upload Preview Screen



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna wirl aliqua. Ut enim ad minim ikad veniam, quis nostrud exercitatn ullamco laboris nisi ut aliquip. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna wirl aliqua. Ut enim ad minim ikad veniam, quis nostrud exercitatn ullamco laboris nisi ut aliquip. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore m wirl aliqua. Ut enim ad minim ikad veni

Display the preview of the image and the result of the extracted characters from the notes. User can cancel or upload their note to share

Screen 4 - Preview Screen

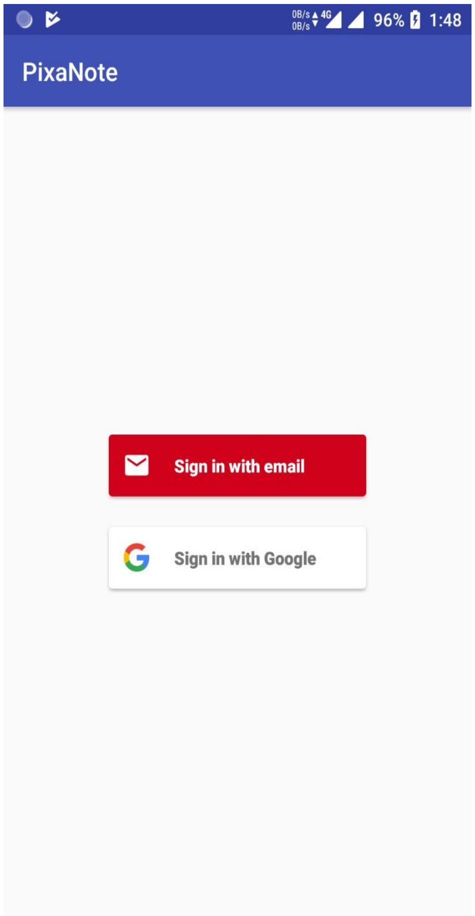


the line of least discussion will
be to keep the Committee as fixed
with me as president &c., you
can be vice-president Stark, and
aid Watson with your counsel; and
I hope that   will also
co-operate again. I wish't "I ha'd
never said I would n't play at
president! If I can't preside when
the time comes, I reserve the
right to write whomever I [illegible]
please to take the chair.
Pray agree; and help
to quench the inconceivable
paltriness of spirit that seems
to have shown its head for the
1st time in our American psy-
chological world! Yours ever
truly,
Wm. James

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna wirl aliqua. Ut enim ad minim ikad veniam, quis nostrud exercitatn ullamco laboris nisi ut aliquip. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna wirl aliqua. Ut enim ad minim ikad veniam, quis nostrud exercitatn ullamco laboris nisi ut aliquip. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna wirl aliqua. Ut enim ad minim ikad veniam,

Display the image of the note shared by the PixaNote user and all the extracted character from the notes, user can share the note with Android Share Intent

Screen 5 - Login Screen



Authenticate the user by using email or Google Sign In as authentication mechanism

Key Considerations

How will your app handle data persistence?

The App used the Firebase Realtime Database persist data online and locally

Describe any edge or corner cases in the UX.

When user want to cancel to upload the PixaNote taken, an alert dialog will be displayed to ask them if they want to cancel, if yes they will be taken back to main screen

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

Picasso - For handling background image loading and caching
Firebase Realtime Database - For Handling Database persistence online and locally
Firebase Authentication - Authenticate and manage users
Firebase Storage - For handling uploading the blob of the image to the Google Cloud Platform
Firebase UI - For displaying authentication screen and binding database to the android adapter view
Google Mobile Vision - For handling extracting text from the notes
ButterKnife - For Binding Views to the activity

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

The app will use the Google Firebase Service to persist data online using Firebase Realtime Database Service and Firebase Storage to retrieve and upload blobs of image. Firebase Authentication will be used to authenticate and manage users using email and Google SignIn

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Create new project in firebase.google.com
- Design the schema that will be used to handle users and picture taken inside Firebase Realtime Database
- Generate Google-Services.json in Firebase and copy file to the root of the project folder
- Add Firebase SDK (Database, Storage, and Authentication Dependency)
- Add Google Mobile Service Dependency
- Add Firebase UI Dependency
- Add Picasso Dependency
- Add Butterknife Dependency

Task 2: Implement UI for Each Activity and Fragment

List of UI

- Integrate Firebase UI Authentication
- Build UI for MainActivity
- Build UI for NoteListFragment

- Build UI for SnapActivity
- Build UI for UploadPreviewActivity
- Build UI for NoteDetailActivity
- Build UI For PixaNoteWidget

Task 3: Main Activity UI Layout

- Implement CoordinatorLayout combined with ViewPager to handle switching tabs for all note and main note as the layout for the main activity

Task 4: Note List Layout For RecyclerView and FragmentPager

- Implement the note list layout that will display image view, text, like count text, and like button
- Create a recyclerview adapter that will use the note list layout as the viewholder layout
- Create a new fragment that will use the recycler view as the root layout
- Setup the fragment pager that the view pager will use inside the main activity

Task 5: SnapActivity And Google Mobile Vision Integration

- Create layout for snap activity using GMS vision Camera Source preview
- Implement the extract text method using GMS Vision library from the image after snap button is pressed

Task 6: UploadPreviewActivity Layout

- Create Layout using CoordinatorLayout, ImageView, and TextView to display the image and extracted Text

Task 7: DetailNoteActivity Layout

- Create Layout using CoordinatorLayout, ImageView, and TextView to display the image and extracted Text
- Add TextView and ImageView to display the author of the note, date taken

Task 8: Implement Firebase Database inside the MainActivity

- Setup firebase database to observe the database to get the collection of all latest notes, and list of current authenticated user note
- Bind the data to the recyclerview
- Handle action to write to the database when user like an activity

Task 9: Implement Firebase Database and Firebase Storage inside the UploadPreviewActivity

- Handle Image uploading using Firebase Storage
- Save the location of the image and other attributes such as author, timestamp to the Firebase Database.

Task 10: Implement PixaNoteWidget in HomeScreen

- Setup the widget layout to display the selected note image and extracted text
- Setup the widget provider and update mechanism

Task 11: Implement Firebase Database and Intent Sharing inside the DetailNoteActivity

- Implement Firebase database observer and bind ui
- Use share compat intent builder so the text of the notes can be shared using intent

Task 12: Stylize the App Theme to follow the Material Design Guidelines

- Setup Application Color according to material design guidelines
- Setup margin and padding according to material design guidelines

Task 13: Configure Signing Configuration

- Equip app with keystore and password that are included in repository, refer the keystore using relative path