LNMIIT Outpass App

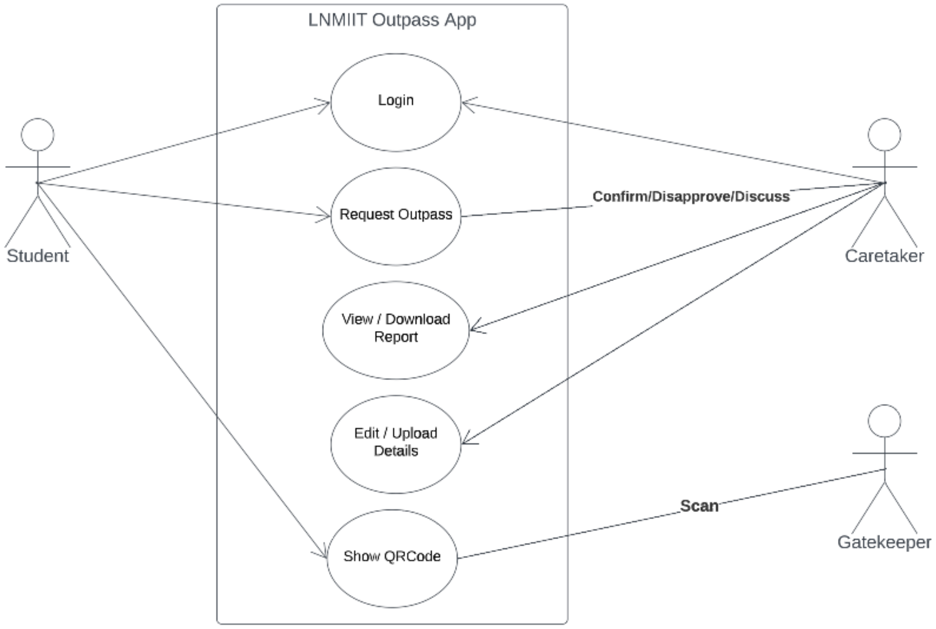
Documentation

LNMIIT

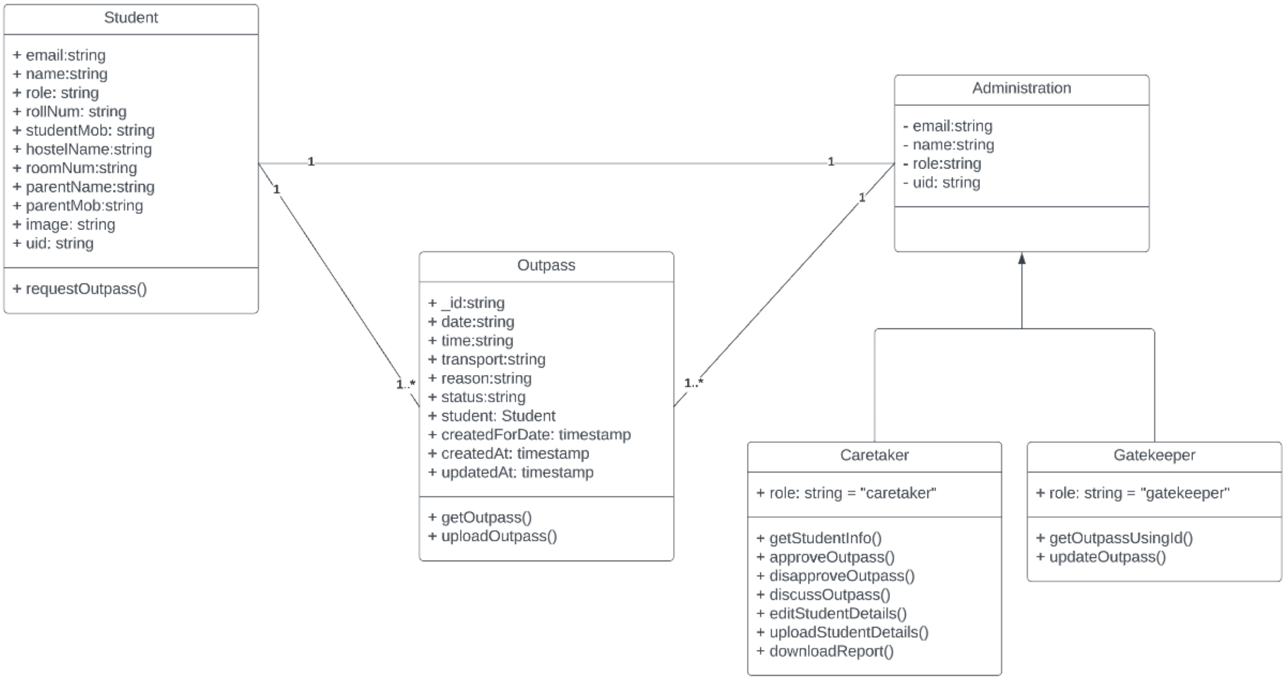
23rd February 2022

Table of Contents

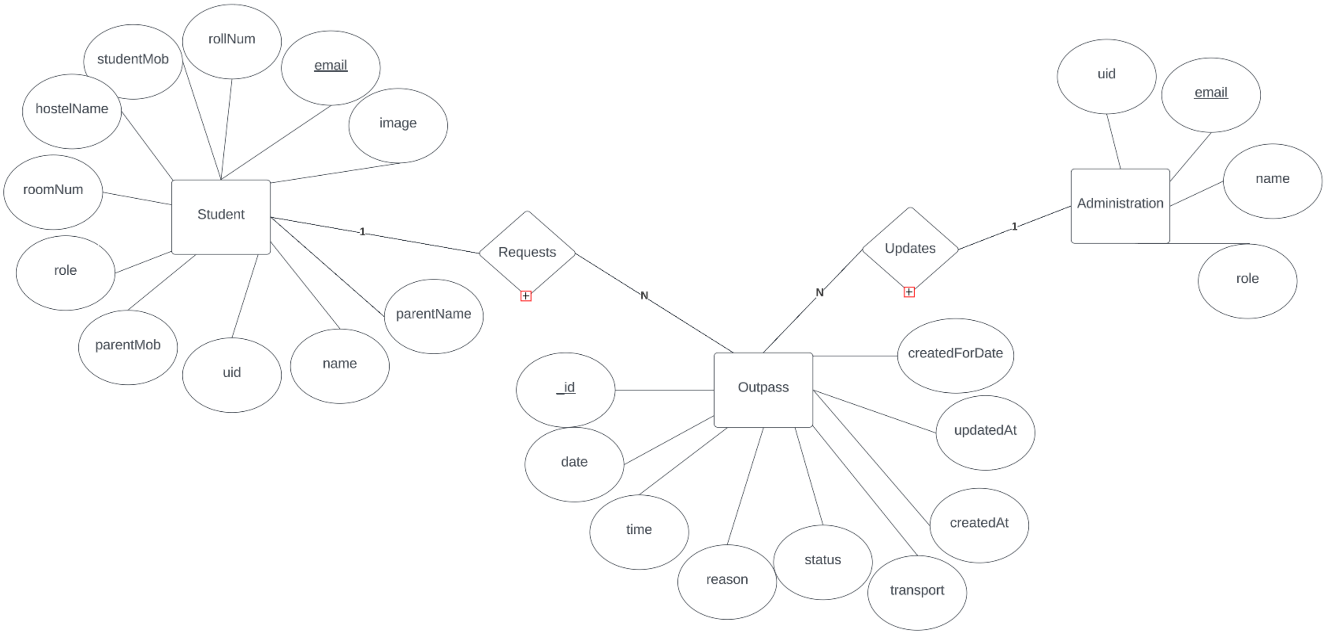
Use Case Diagram



Class Diagram



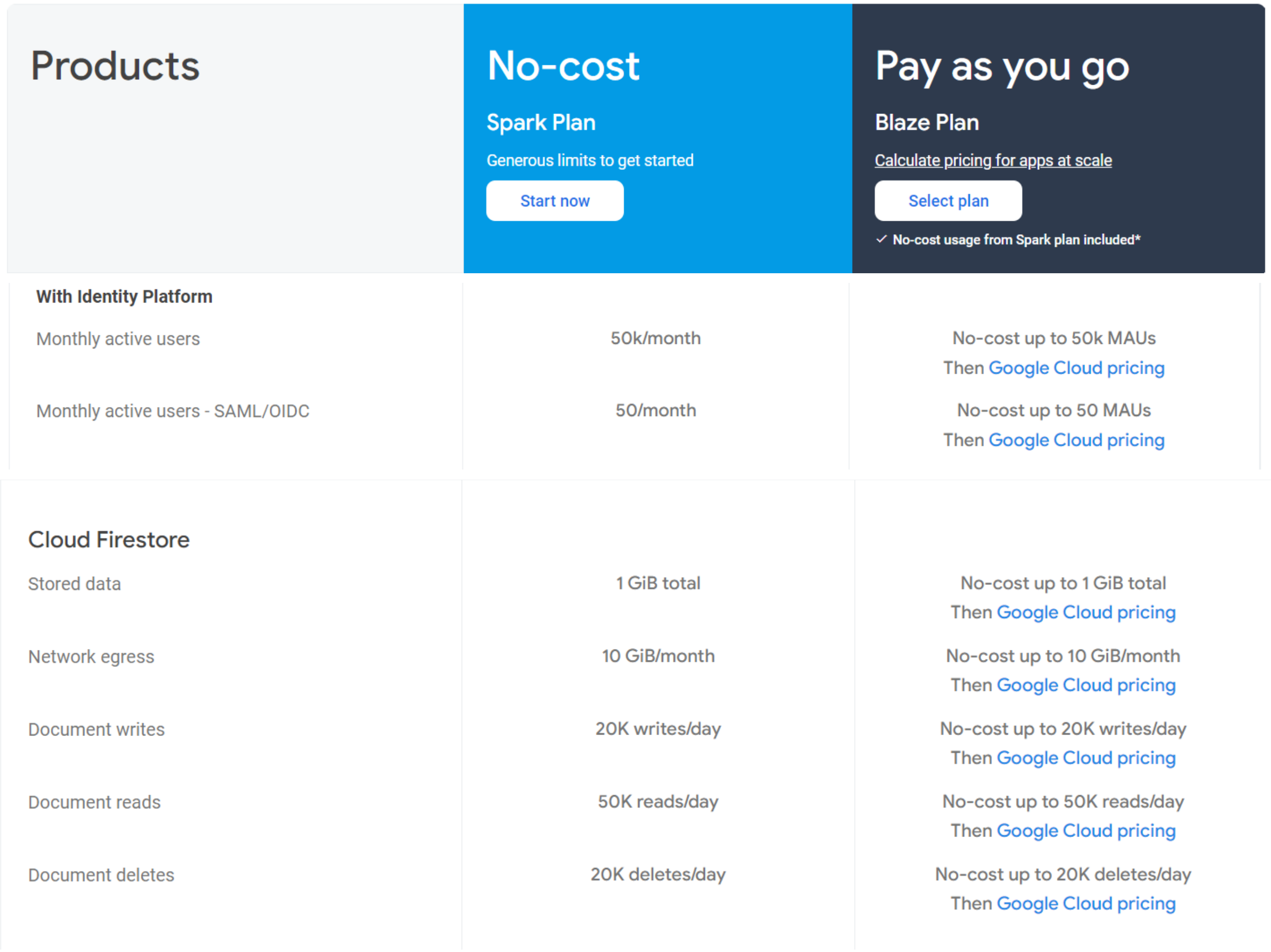
ER Diagram


studentMob
rollNum
email
hostelName
image
roomNum
Student
role
parentMob
uid
name
parentName
1
Requests
N
_id
date
time
reason
status
transport
Outpass
createdForDate
updatedAt
createdAt
N
Updates
1
uid
email
name
Administration
role

Firebase

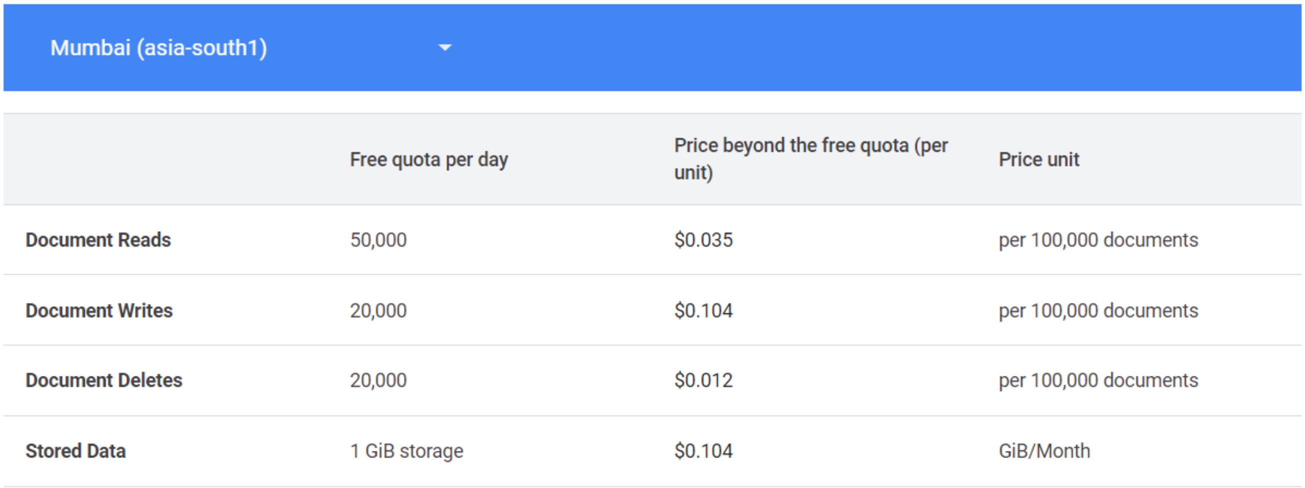
a) Pricing of Firebase:

Source:- https://firebase.google.com/pricing



Products	No-cost Spark Plan Generous limits to get started Start now	Pay as you go Blaze Plan Calculate pricing for apps at scale Select plan ✓ No-cost usage from Spark plan included*
With Identity Platform		
Monthly active users	50k/month	No-cost up to 50k MAUs Then Google Cloud pricing
Monthly active users - SAML/OIDC	50/month	No-cost up to 50 MAUs Then Google Cloud pricing
Cloud Firestore		
Stored data	1 GiB total	No-cost up to 1 GiB total Then Google Cloud pricing
Network egress	10 GiB/month	No-cost up to 10 GiB/month Then Google Cloud pricing
Document writes	20K writes/day	No-cost up to 20K writes/day Then Google Cloud pricing
Document reads	50K reads/day	No-cost up to 50K reads/day Then Google Cloud pricing
Document deletes	20K deletes/day	No-cost up to 20K deletes/day Then Google Cloud pricing

Source:- https://cloud.google.com/firestore/pricing



Mumbai (asia-south1)

	Free quota per day	Price beyond the free quota (per unit)	Price unit
Document Reads	50,000	$0.035	per 100,000 documents
Document Writes	20,000	$0.104	per 100,000 documents
Document Deletes	20,000	$0.012	per 100,000 documents
Stored Data	1 GiB storage	$0.104	GiB/Month

b) App Usage:

(i) Monthly active users -

Around 2000

Status -> No problem (Max limit - 50,000)

(ii) Write Operations -

Worst Case -> 8000/day

Status -> No problem (Max limit - 20,000/day)

(iii) Delete Operations -

No delete operation

Status -> No problem

(iv) Storage -

1GB -> Can store outpass of more than 1 year

(v) Read Operations -

On average -> 20,000 to 30,000 / day

Status -> May exceed on some busy day (Max limit - 50,000/day)

c) Setting Firebase:

i) Creating Firebase Project -

Documentation - [Setting Up a Firebase Project for React Native]

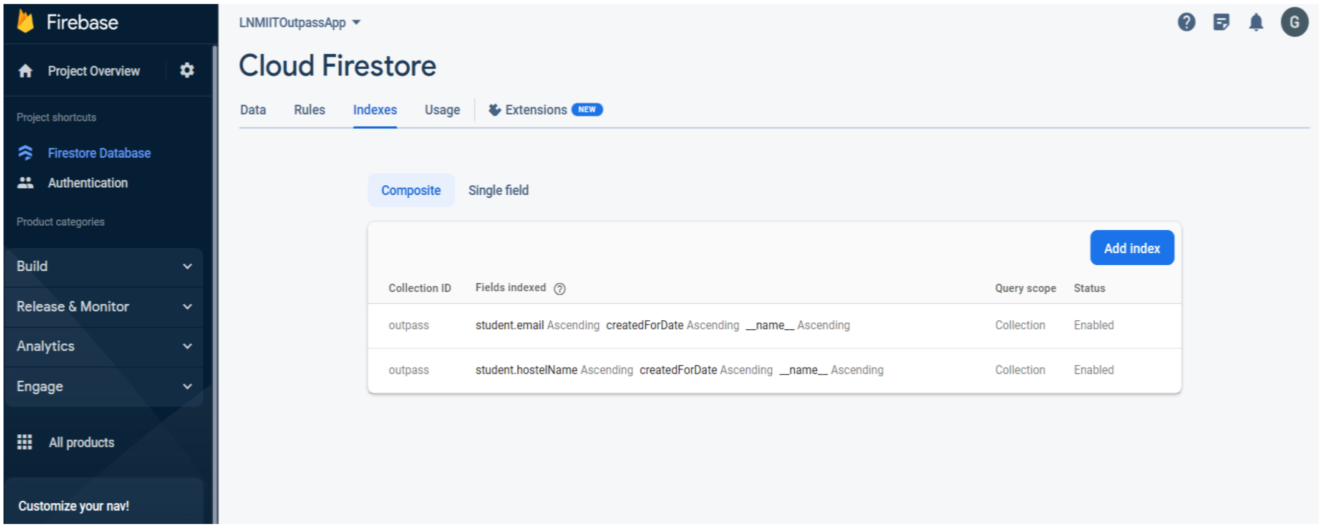
Check document from -> Creating a Firebase Project

ii) Setting Firebase Google Signin -

Documentation - [How to Set Up Google Login in React Native & Firebase]

iii) Setting Index for query –

Documentation - [Add and edit indexes - Firebase Help (google.com)]



Collection ID	Fields indexed	Query scope	Status
outpass	student.email Ascending createdForDate Ascending __name__ Ascending	Collection	Enabled
outpass	student.hostelName Ascending createdForDate Ascending __name__ Ascending	Collection	Enabled

iv)Building the app for publishing(Android) –

Documentation - [Publishing to Google Play Store · React Native]

Check ->

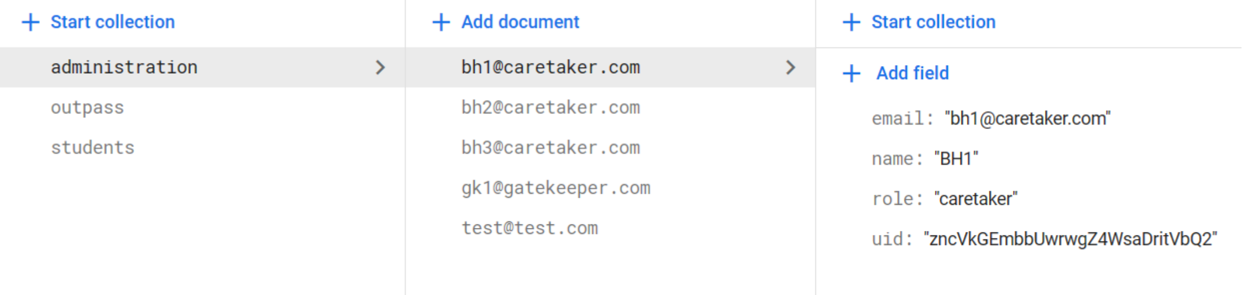
- Generating an upload key (Android)
- Setting up Gradle variables
- Adding signing config to your app's Gradle config
- Generating the release AAB

v) Building the app for publishing (IOS) –

Documentation - [Publishing to Apple App Store · React Native]

d) Collections in Firebase:

(i) administration-

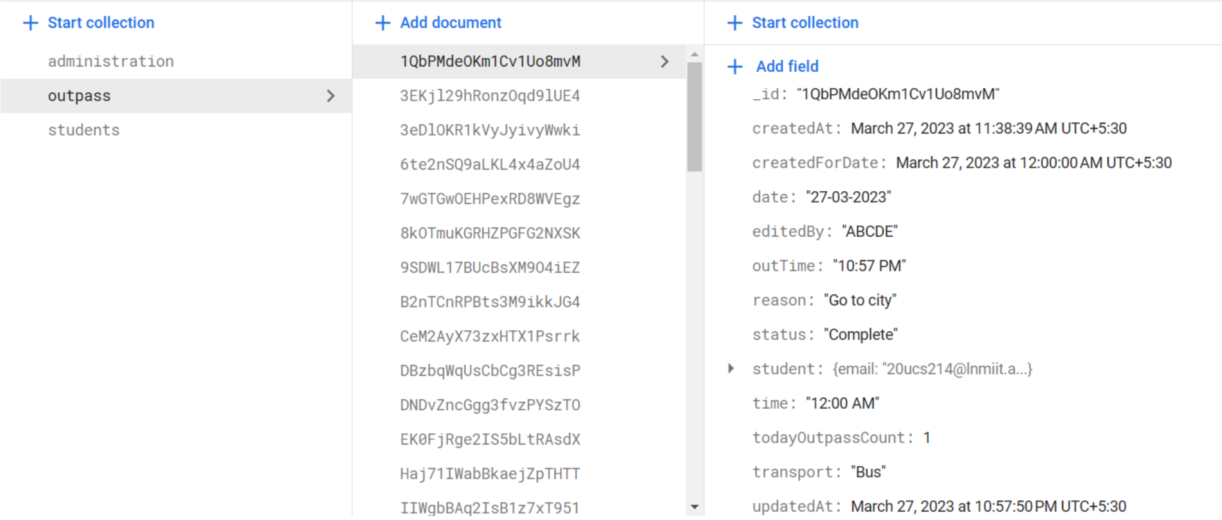


```
administration
outpass
students

bh1@caretaker.com
bh2@caretaker.com
bh3@caretaker.com
gk1@gatekeeper.com
test@test.com

email: "bh1@caretaker.com"
name: "BH1"
role: "caretaker"
uid: "zncVkGEmbbUwrwgZ4WsaDritVbQ2"
```

(ii) outpass-

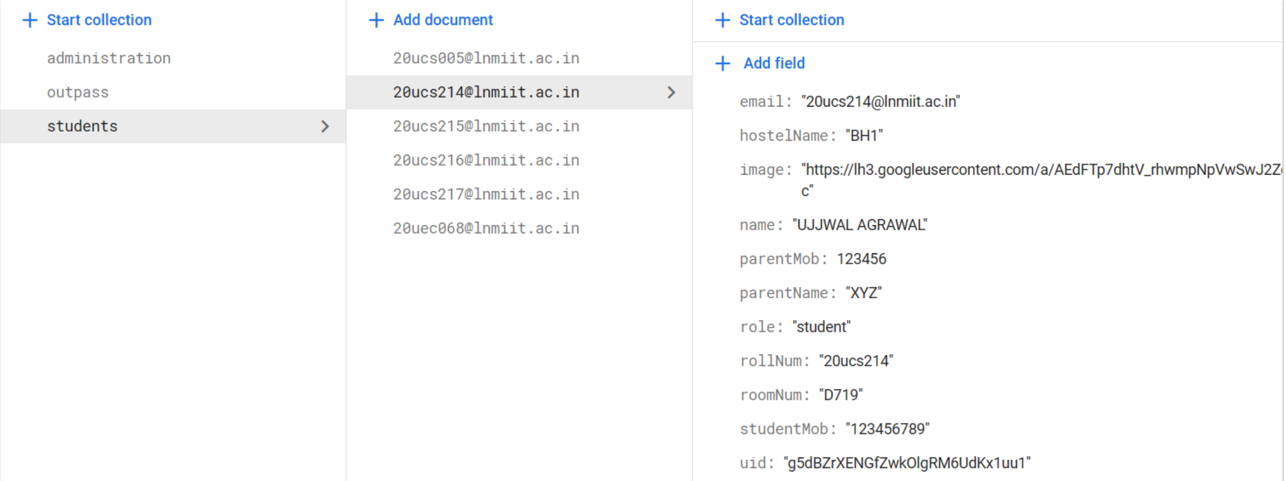


```
administration
outpass
students

1QbPMdeOKm1Cv1Uo8mvM
3EKjl29hRonzOqd9lUE4
3eDlOKR1kVyJyivyWwki
6te2nSQ9aLKL4x4aZoU4
7wGTGwOEHPexRD8WVEgz
8kOTmuKGRHZPGFG2NXSK
9SDWL17BUcBsXM9O4iEZ
B2nTCnRPBts3M9ikkJG4
CeM2AyX73zxHTX1Psrrk
DBzbqWqUsCbCg3REsisP
DNDvZncGgg3fvzPYSzTO
EK0FjRge2IS5bLtRAsdX
Haj71IWabBkaejZpTHTT
IIWgbBAq2IsB1z7xT951

_id: "1QbPMdeOKm1Cv1Uo8mvM"
createdAt: March 27, 2023 at 11:38:39 AM UTC+5:30
createdForDate: March 27, 2023 at 12:00:00 AM UTC+5:30
date: "27-03-2023"
editedBy: "ABCDE"
outTime: "10:57 PM"
reason: "Go to city"
status: "Complete"
student: {email: "20ucs214@lnmiit.a...}
time: "12:00 AM"
todayOutpassCount: 1
transport: "Bus"
updatedAt: March 27, 2023 at 10:57:50 PM UTC+5:30
```

(iii) students-



```
administration
outpass
students

20ucs005@lnmiit.ac.in
20ucs214@lnmiit.ac.in
20ucs215@lnmiit.ac.in
20ucs216@lnmiit.ac.in
20ucs217@lnmiit.ac.in
20uec068@lnmiit.ac.in

email: "20ucs214@lnmiit.ac.in"
hostelName: "BH1"
image: "https://lh3.googleusercontent.com/a/AEdFTp7dhtV_rhwmpNpVwSwJ2Z
        c"
name: "UJJWAL AGRAWAL"
parentMob: 123456
parentName: "XYZ"
role: "student"
rollNum: "20ucs214"
roomNum: "D719"
studentMob: "123456789"
uid: "g5dBZrXENGfZwkOlgRM6UdKx1uu1"
```

Packages

```
1) npm uninstall -g react-native-cli @react-native-community/cli
    npx react-native init LNMOutpassApp

2) npm install @react-navigation/native
    npm install react-native-screens react-native-safe-area-context
    npm install @react-navigation/native-stack

3)Firebase
   npm install --save @react-native-firebase/app
   npm install @react-native-firebase/auth
   npm install @react-native-google-signin/google-signin
   npm install @react-native-firebase/firestore

4) Redux
    npm install redux
    npm install redux-thunk
    npm install react-redux

5) npm install @react-navigation/bottom-tabs
    npm i react-native-vector-icons

6) npm install moment
7) npm install react-native-form-component
      npm i react-native-modal-datetime-picker @react-native-community/datetimepicker
```

8) npm install @react-native-picker/picker

9) npm install react-native-document-picker

10) npm install react-native-fs

11) npm install xlsx

12) npm i react-native-toast-message

13) npm i react-native-splash-screen --save

```
npm i lottie-react-native
```

14) For QR Code -

```
npm i -S react-native-svg react-native-qrcode-svg
npm i js-base64
npm install react-native-qrcode-scanner --save
npm install react-native-camera --save
npm install react-native-permissions
```

Permissions

1) Internet Permission:

- Primary requirement for an app to work with the internet.

2) Read External Storage Permission:

- This permission is required by only the caretaker.
- While uploading the excel sheet to the application to update student data.

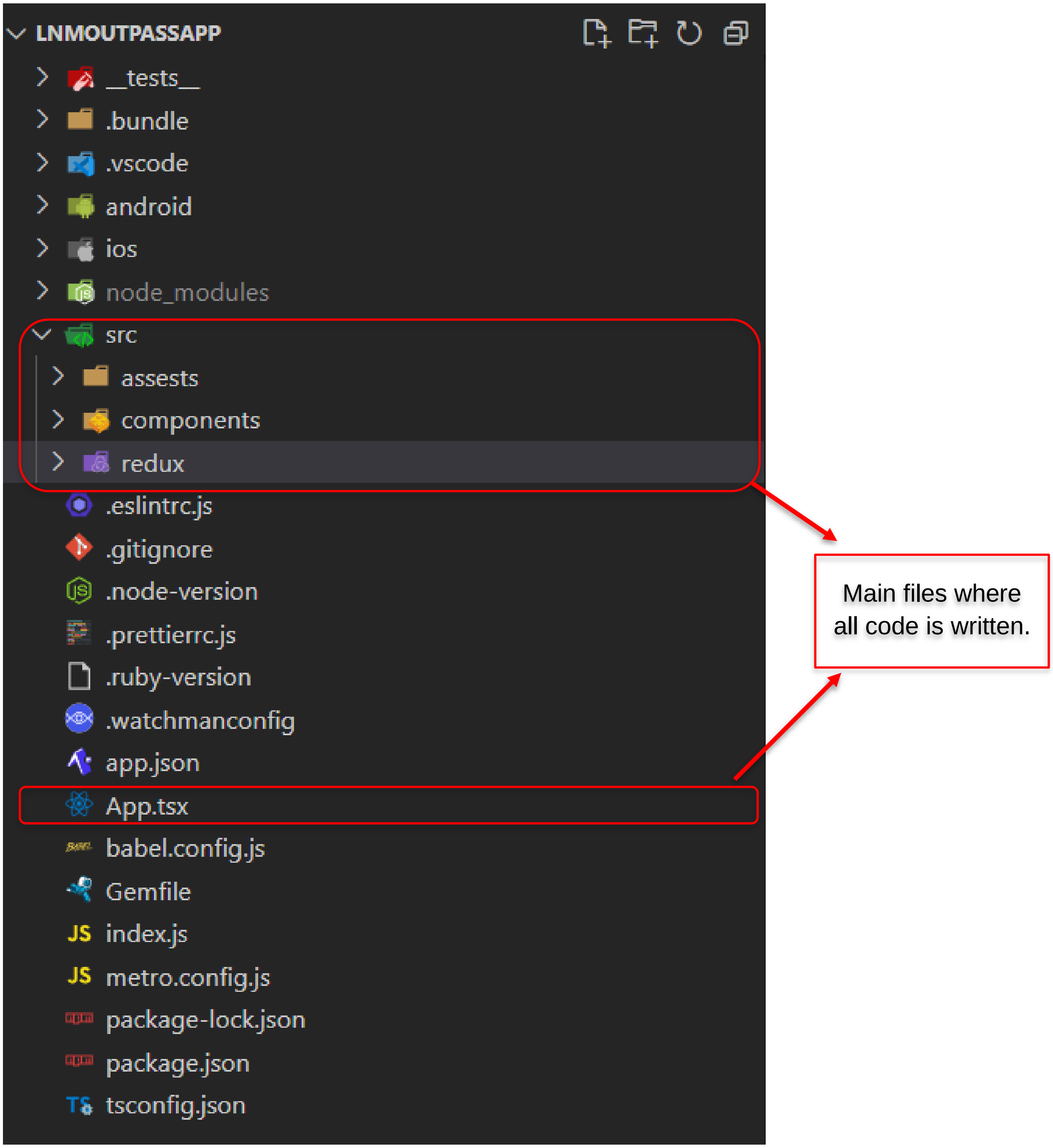
3) Write External Storage Permission:

- This permission is required by only the caretaker.
- While downloading the excel sheet to the application to the student's outpass report.

4) Camera Permission:

- This permission is required by only the gatekeeper.
- While scanning the QRCode gatekeeper has to allow the camera to open.

Folder Structure



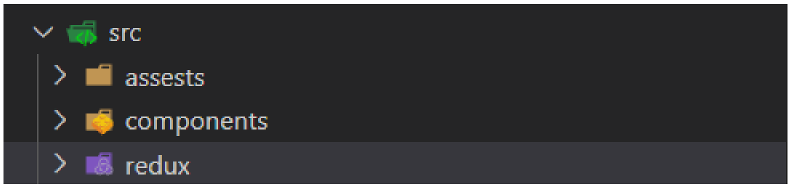
```
LNMOUTPASSAPP
  > __tests__
  > .bundle
  > .vscode
  > android
  > ios
  > node_modules
  v src
      > assests
      > components
      > redux
    .eslintrc.js
    .gitignore
    .node-version
    .prettierrc.js
    .ruby-version
    .watchmanconfig
    app.json
    App.tsx
    babel.config.js
    Gemfile
    index.js
    metro.config.js
    package-lock.json
    package.json
    tsconfig.json
```

Main files where all code is written.

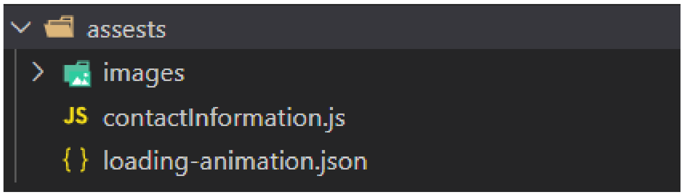
1) App.tsx:

- After splash screen app moves to this component which first hide the splash screen and then moves the screen to main app.
- It creates stack where it put the MainScreen component.
- It also create header for the app.

2) Src folder:

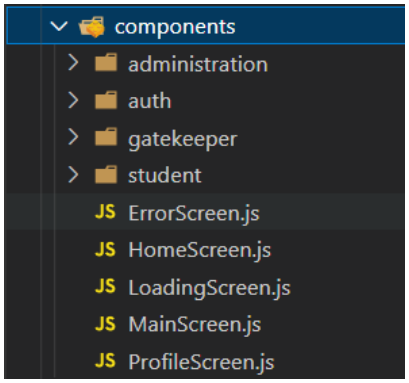


a) Assests folder-



- All the extra resource used is kept in this folder.
- Contains –
  - All the images in the images folder which are used in the app,
  - Contact information of the administration, and
  - Loading Animation file.

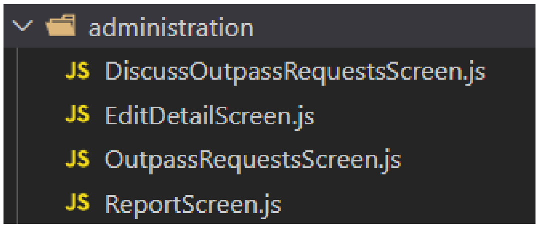
b) Components folder-



```
components
  administration
  auth
  gatekeeper
  student
  ErrorScreen.js
  HomeScreen.js
  LoadingScreen.js
  MainScreen.js
  ProfileScreen.js
```

- This is the main code of the app.
- Starts with the MainScreen which then pass the app either to HomeScreen if the login is there otherwise to the LoginScreen in Auth folder.
- MainScreen forms the BottomNavbar and is responsible for redirecting the user according to his role in the User data.
- ProfileScreen shows the details of the user.
- LoadingScreen show the loading animation.
- ErrorScreen shows the error if any major error occurs while accessing data.

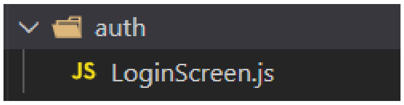
- Administration Folder :-



```
administration
  DiscussOutpassRequestsScreen.js
  EditDetailScreen.js
  OutpassRequestsScreen.js
  ReportScreen.js
```

  - Contains all the code files corresponding to administration users.
  - OutpassRequestsScreen – All the outpass requests the visible with <u>pending status</u>.
  - DiscussOutpassRequestsScreen - All the outpass requests the visible with <u>discuss status</u>.

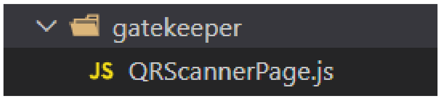
- **Auth folder :-**



```
auth
  LoginScreen.js
```

  - Only contains LoginScreen which helps user to login through two options –
    - Email and Password (For Administration),
    - Google Login (For Students)

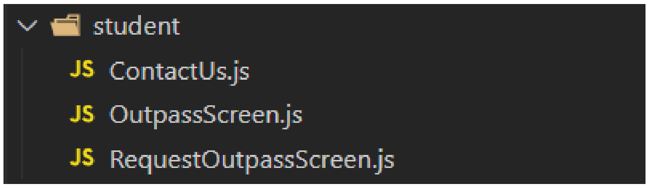
- **Gatekeeper folder :-**



```
gatekeeper
  QRScannerPage.js
```

  - QRScannerPage – Opens QRCode Scanner to scan QRCode of outpass of the student.

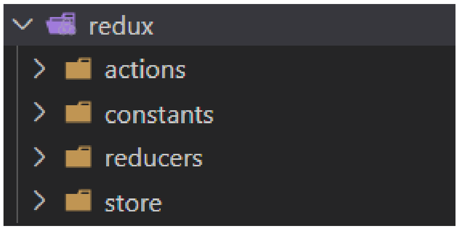
- **Student folder :-**



```
student
  ContactUs.js
  OutpassScreen.js
  RequestOutpassScreen.js
```

  - RequestOutpassScreen – Opens form for student to request for outpass.
  - Outpass Screen – Shows all the outpass requested by student.
  - ContactUs page shows the information of the administration of their respective hostel.

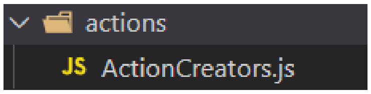
3) Redux folder:



```
redux
  actions
  constants
  reducers
  store
```

- It is mainly the backend part of the app.
- It uses react-redux for the proper data flow in the app.
- Takes data from the firebase using functions in actions and then passing it to their respective reducers and moving to the app using store.

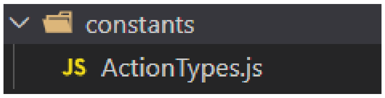
a) Actions folder-



```
actions
  JS ActionCreators.js
```

- Contains all the functions to fetch, modify and add data to or from firebase.
- These functions are imported by the Screens to perform the required operations.

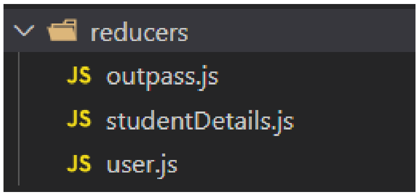
b) Constants folder-



```
constants
  JS ActionTypes.js
```

- Contains all the types of actions that can be created to perform on the state of particular data.

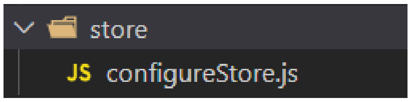
c) Reducers folder-



reducers
JS outpass.js
JS studentDetails.js
JS user.js

- ○ Contains files which stores state of the different data and different actions that can be taken on it.
- ○ Outpass -> Store state for outpass data (used by student as well as administration).
- ○ User -> Store state for user data(used by student as well as administration).
- ○ StudentDetails -> Store state for studentDetails data (used by caretaker to edit details of student).

d) Store folder-



store
JS configureStore.js

- ○ It configures the store which contain all the data got from the reducers.
- ○ This store is then imported by the Screens to use the data when required.