2017

SCHOOL DIARIES

DEVELOPED BY:
RAM TADUVAI & AKHIL GADIPUDI

SYRACUSE UNIVERSITY | Computer Engineering Fall 2017

App Idea :

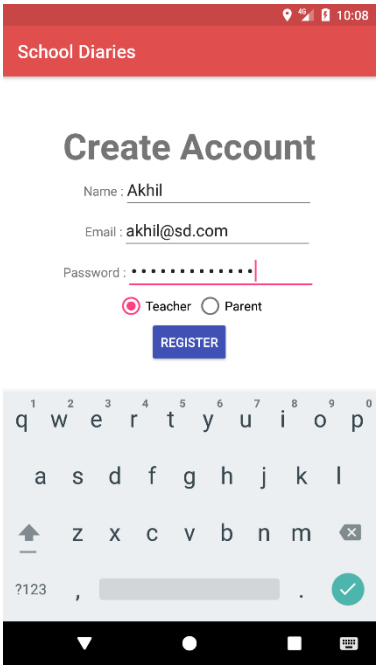
The main use case of our application is in daycares are preschools. At that stage kids a too young to be able to communicate everything that goes on in their class, but parent want to be updated on everything that goes on in the class. And other solutions like, teacher talking to parents to give updates is not instant and not practical to talk to every single parent every day. This app connects teacher with all the parents at the same time. Makes a world of difference for the parent who can stay updated on everything their child is doing at that moment and also for the teacher to organize, keep track of all the activities and last but not least communicate with all parents easily.

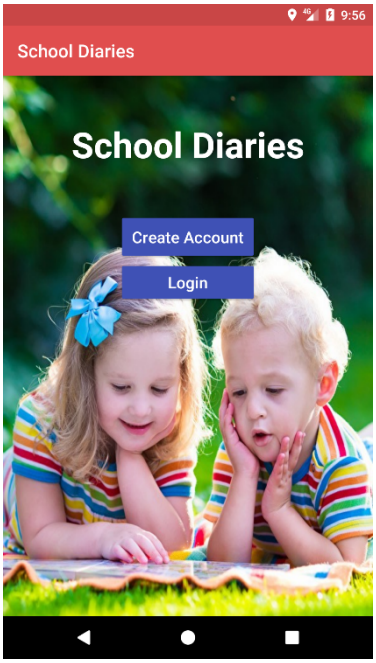
App Usage:

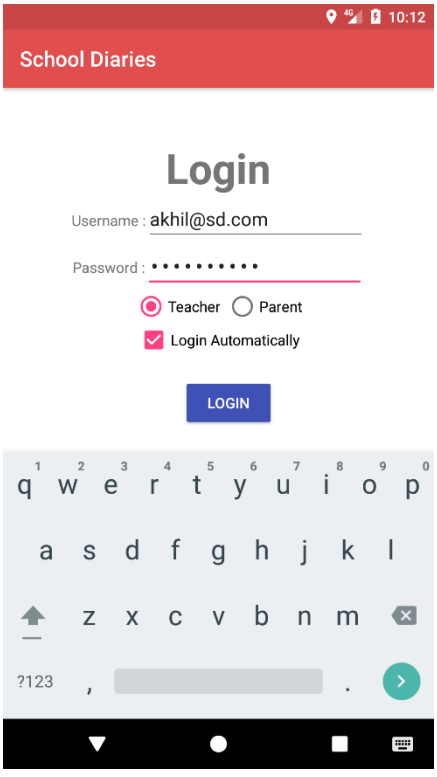
Creating Accounts and Logins :

The moment you open the app for the first time, you are given the option to login or create account. Lets go through creating account first.

As shown in the screenshot, it provides you option to create a teacher account or parent account. We will see how their features differ further in this report. And also, as you would expect from a good login forum. This page won't let you leave any field empty.

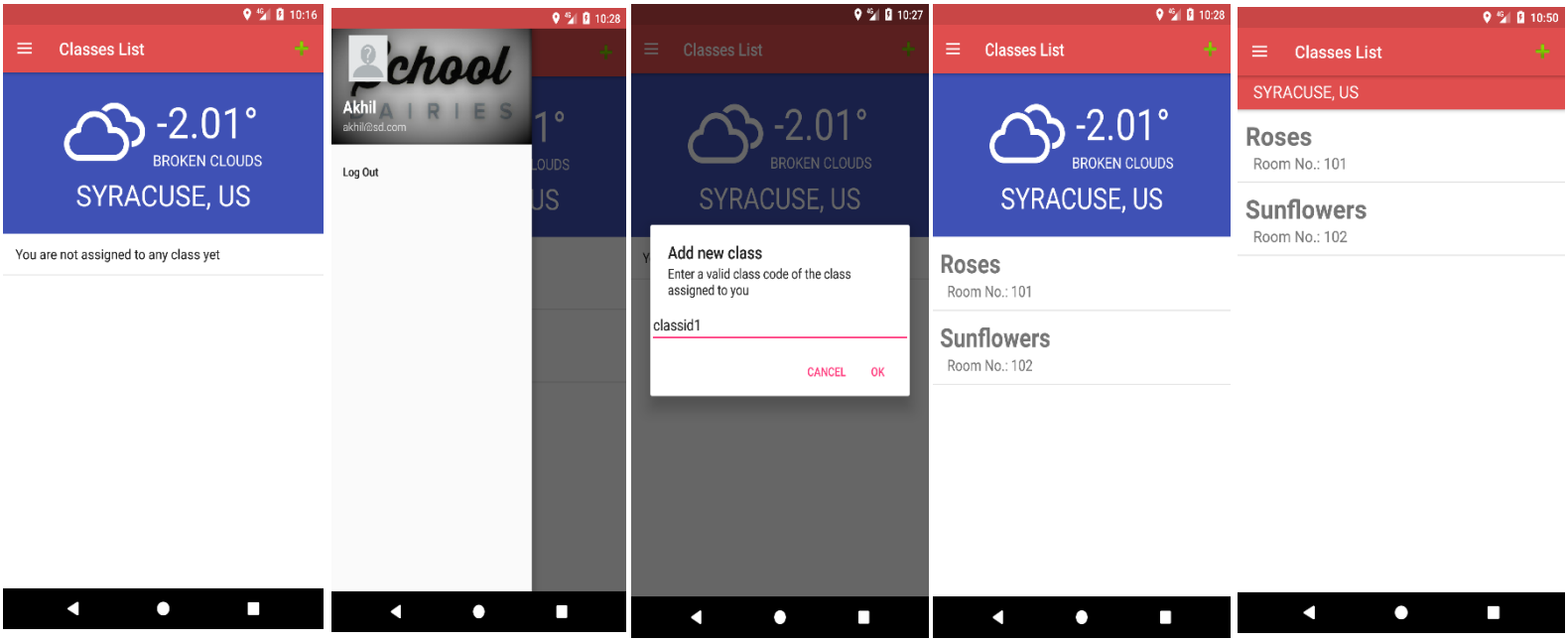






Now, lets login into the account we just created. I created a teacher account. Here I am choosing to login automatically. The Username and Password are stored using ***Preferences.*** All the ***authentication features*** are provided using ***Firebase***. ***All data is hosted on cloud.***

After you login for the first time with a teacher account, this is what you see. Both teacher and parent side of the app have navigation drawer. The plus button in top right allows teacher to add classes to their account, lets add 2 class to this new account using the id's of the classes.



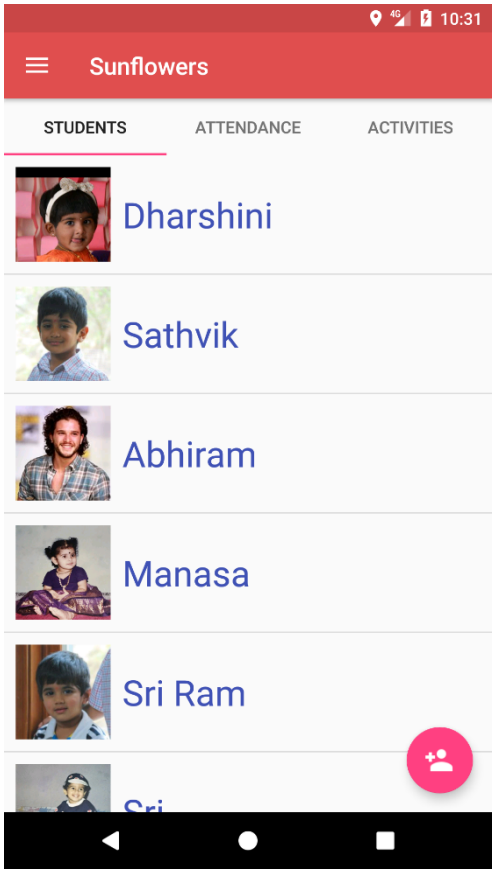
The weather card slide up to shrink to just show the location, this done using ***advanced Coordinator layout***. The weather data is obtained ***in real-time from 3rd party API***. The location details required for weather and in some other features of the app that will be discussed further in the report is obtained using the phones ***GPS***, for this, a ***Service*** is being used to constantly get location data and provide it to fragments that require the data using ***Broadcast Receivers.*** The screenshots below show the service and broadcast receivers used.

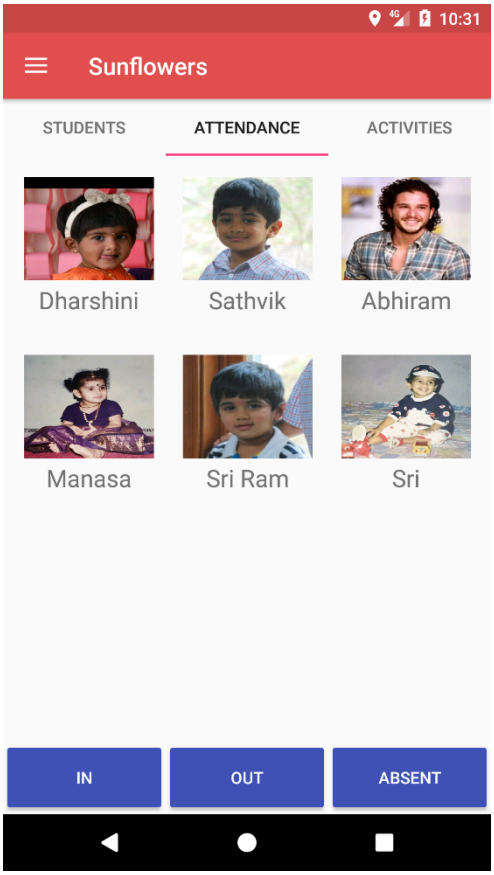
```
serviceIntent = new Intent(getApplicationContext(), LocationHelper.class);
startService(serviceIntent);
```

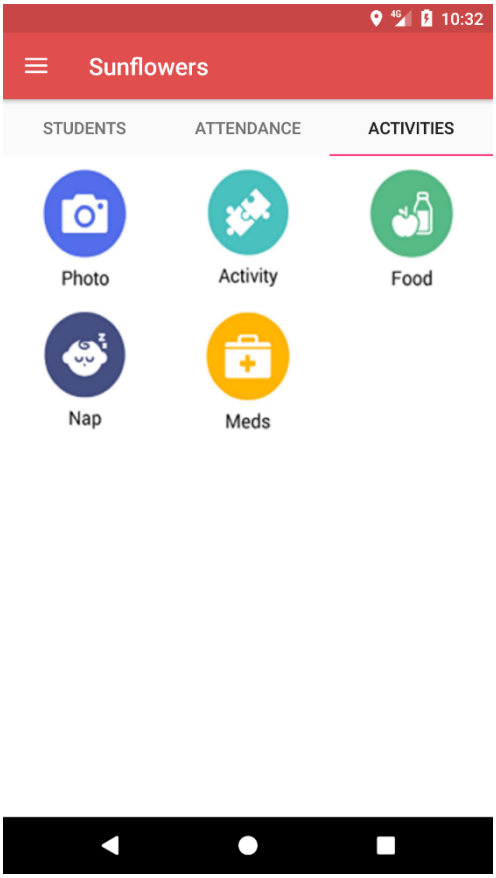
```
<service android:name=".LocationHelper" android:process=":my_service" />
```

```
@Override
public void onResume() {
    super.onResume();
    IntentFilter filter = new IntentFilter();
    filter.addAction("GPS_COORDINATES");
    getActivity().registerReceiver(mMessageReceiver, filter);
}
private BroadcastReceiver mMessageReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        if(intent.getStringExtra("status").equalsIgnoreCase("fail"))
        {
            Intent newintent = new Intent();
            intent.setAction("GPS_REQUEST");
            getActivity().sendBroadcast(newintent);
        }
        logitude = intent.getDoubleExtra("long",0f);
        latitude = intent.getDoubleExtra("lat",0f);
        SetWeather();
    }
};
```

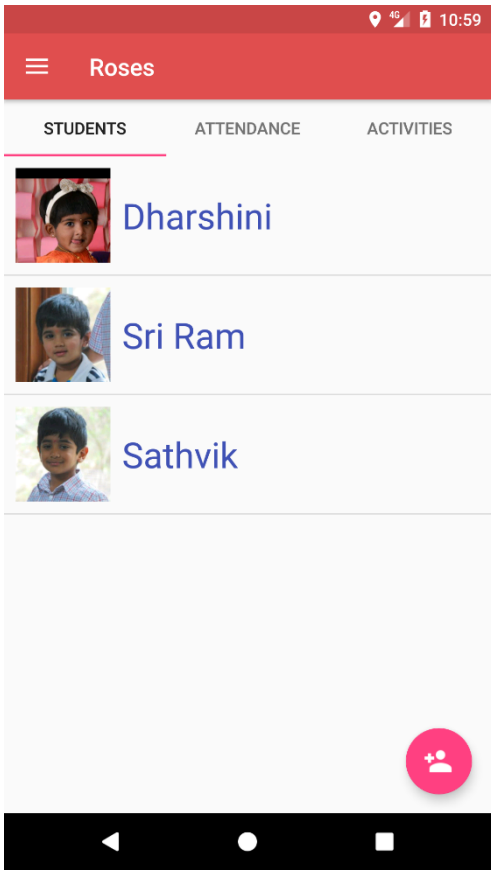
After you select one of the classes. You load into the class fragment which has a ***ViewPager*** with ***multiple 3rd party advanced viewpager animation that are selected randomly*** every time you open the class fragment. Let's talk about each page in the view pager.

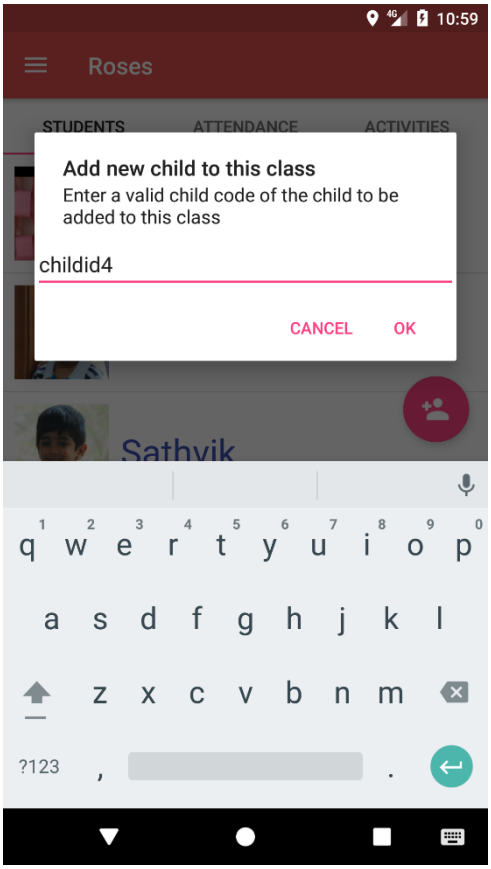




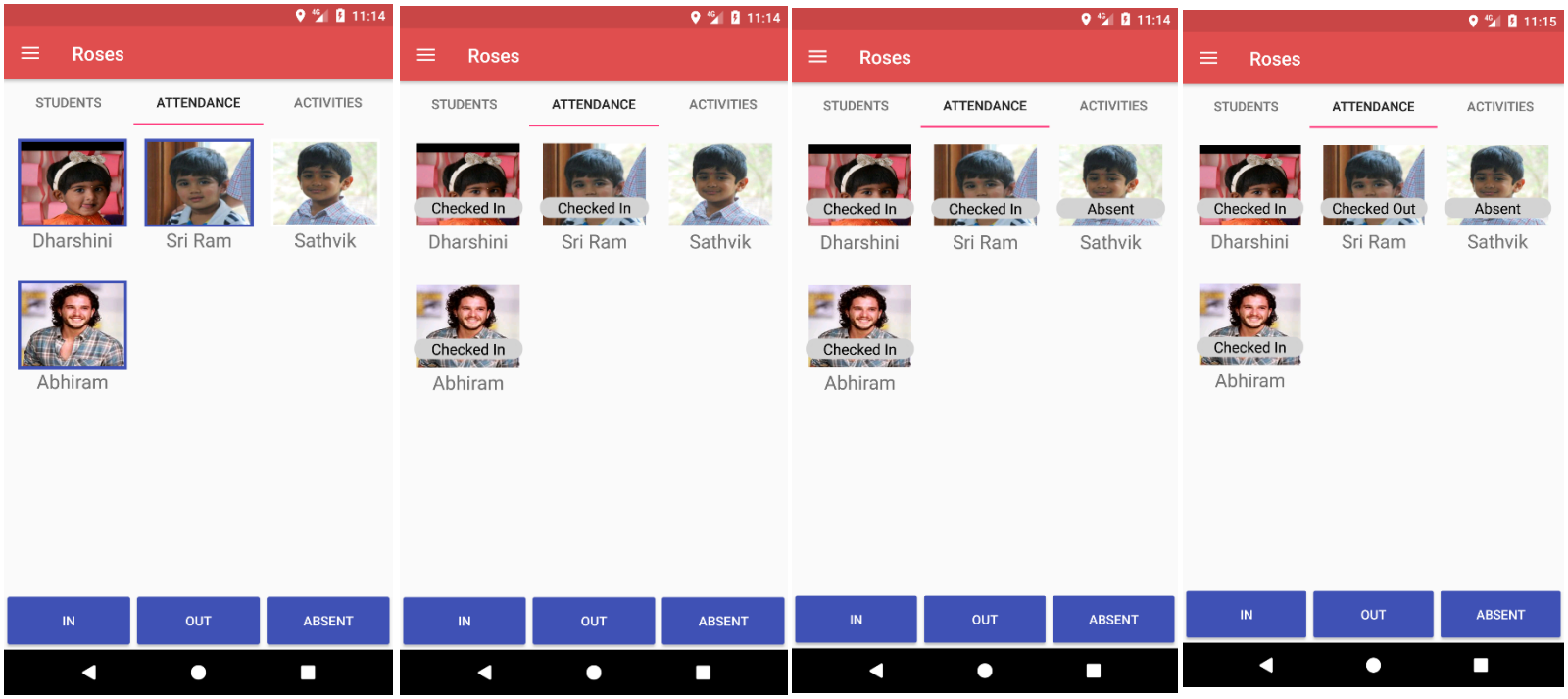


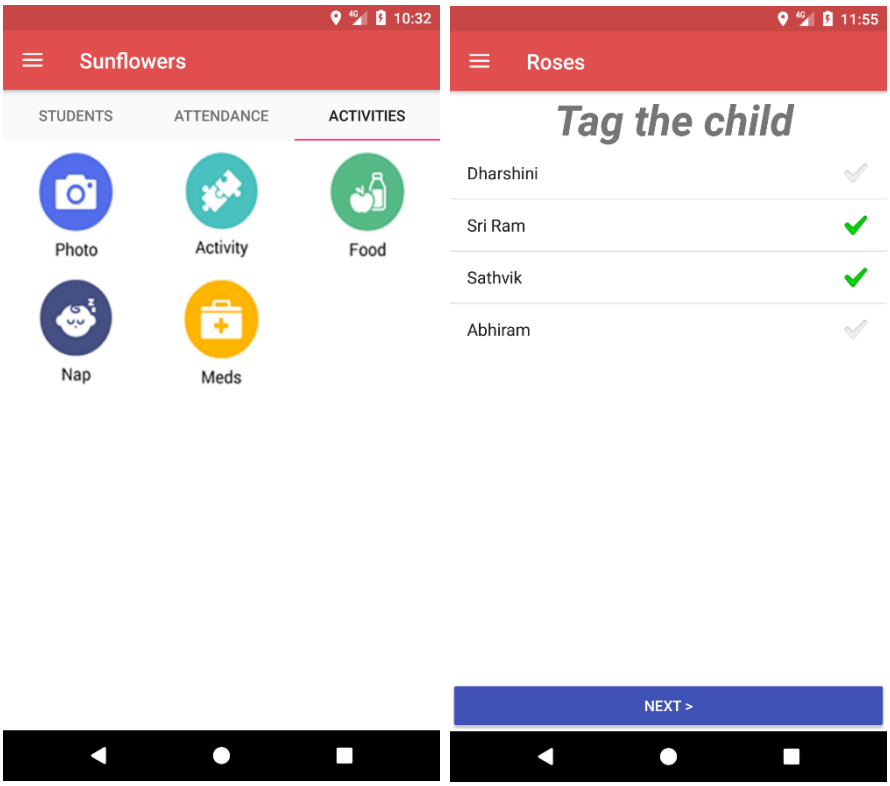
The first page is STUDENTS page, which contains a ***RecyclerView*** that displays all the children that were added to this class. The recyclerview used ***3rd party animations from wasabeef***. Its shows student's image and their name. All images of students and other images that will be shown in other features of the app are stored in ***Firebase storage.*** At the bottom of this page is a ***Floating Action Button*** to add children to this class. Let's go back and change class ROSES that has less children and add new children to it using the childcode assigned while creating child account. Clicking on a child in this list bring up their Activity feed. We will talk about is after we talk about all 3 pages in the view pager because the feed is affected by actions in all 3 of these pages.



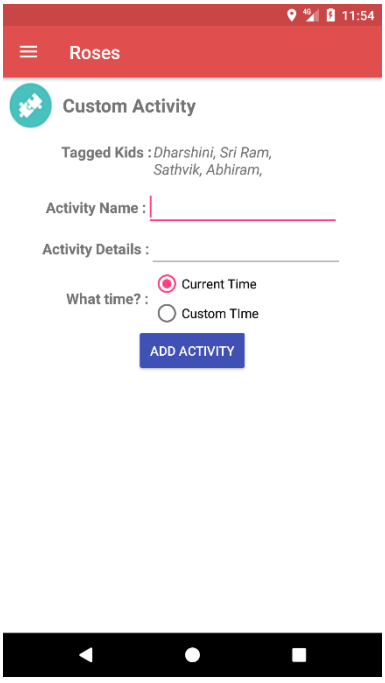


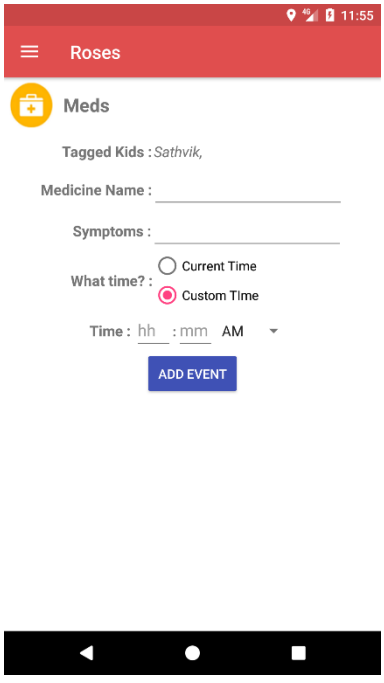
Coming to second page, the ATTENDANCE page, allows teacher to mark students of the class as checked in as soon as they enter the class, mark them checked out when they leave the class and make them absent if they don't attend the class that day. Initially all student are unmarked. This ***Grid RecyclerView*** allows to ***select any number of students*** and change their status to any of the 3 states mentioned. Let's mark some of the student as checked in and some as absent, checkout 1 student to show checking out.

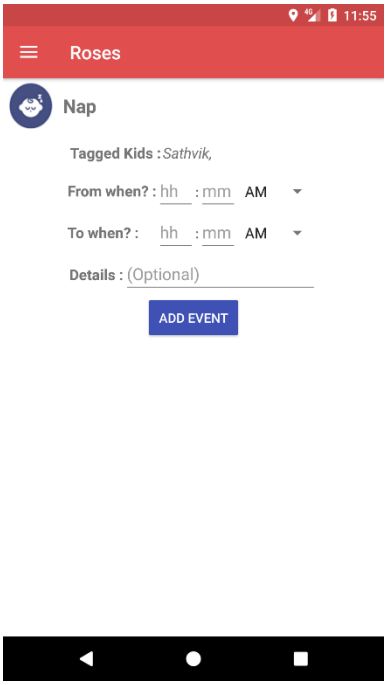


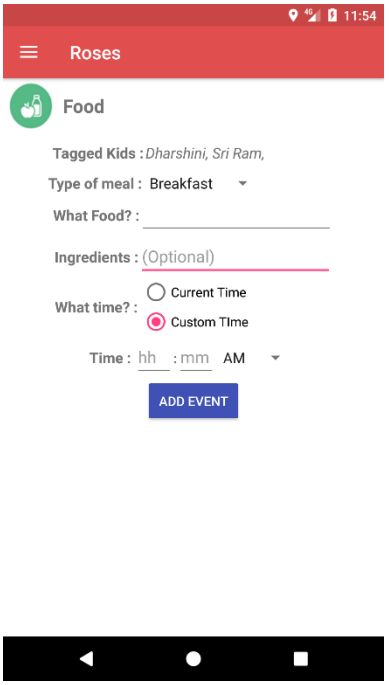


Coming to the last page of the viewpager, which shows all the different kinds of activities that cant be added to students. We have already seen the page to select the activity, now lets see each activity and what it adds. When you select which activity to add, you are taken to a fragment where you tag the children, all the kids in the class are shown here, only the kids selected here will have their feed updated with that activity.









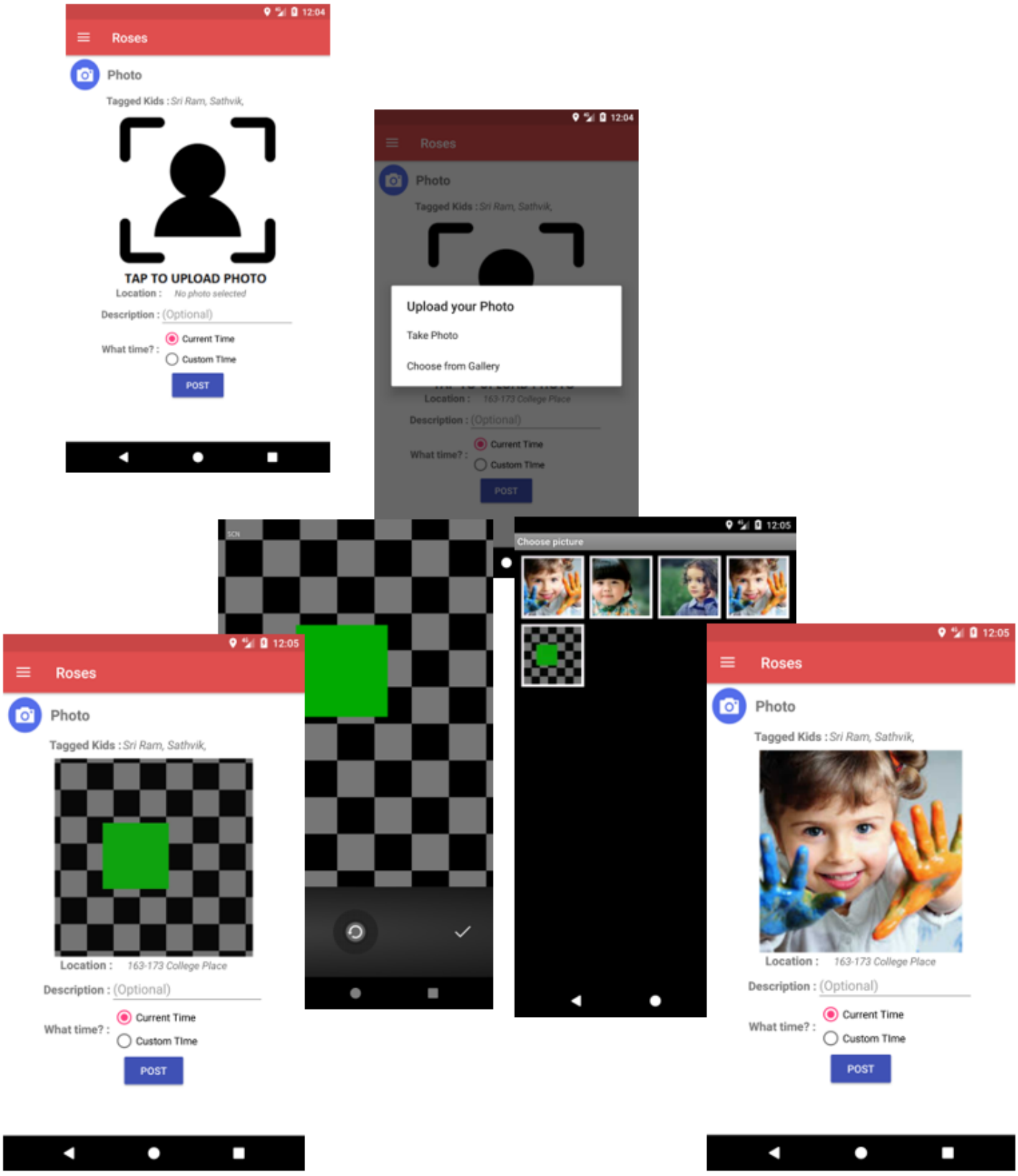
The above 4 activities are technically quite similar.

You fill out the details of the activity as required and select to post it with either current time or a custom time. And the activity is added to the feed of all the children that are tagged in the activity. The fragments that asked you to tag student is shown below. The photo activity is quite different from the rest of them so let look at it below.

The photo activity allows the teacher to create activity containing a photo, for this purpose a new photo can be take using the ***Camera API*** or uploaded using ***Galery API***. The photo is then uploaded to ***Firebase storage*** when the post button is clicked.

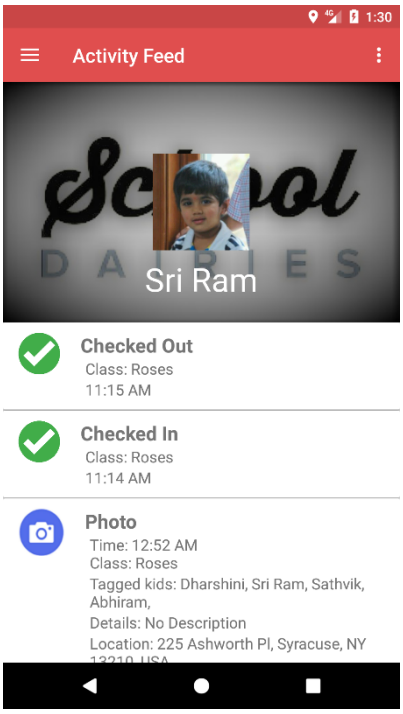
```
progressDialog.setMessage("Uploading image to cloud");
String ImageName = "Image_" + MyActId;
databaseReference.child(MyActId).child("photo_name").setValue(ImageName);

StorageReference storagePref = FirebaseStorage.getInstance().getReference().child("PostImages").child(ImageName);
storagePref.putFile(CurrImageURI).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {
        if(task.isSuccessful())
        {
            progressDialog.dismiss();
            Toast.makeText(getContext(), "Added Photo activity to the feed", Toast.LENGTH_SHORT).show();
        }
    }
});
```

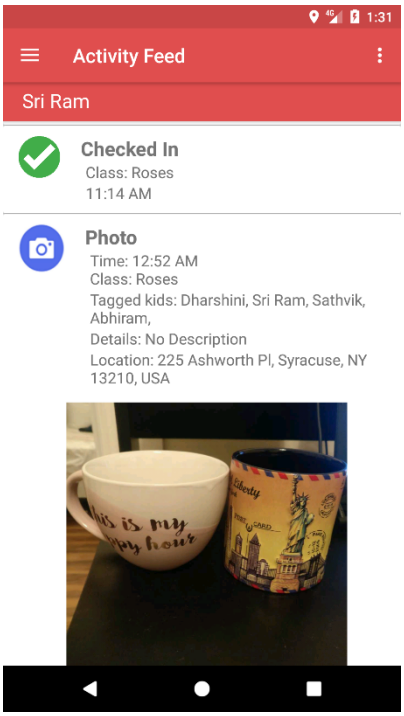
The screenshots below show different stages of adding a photo activity

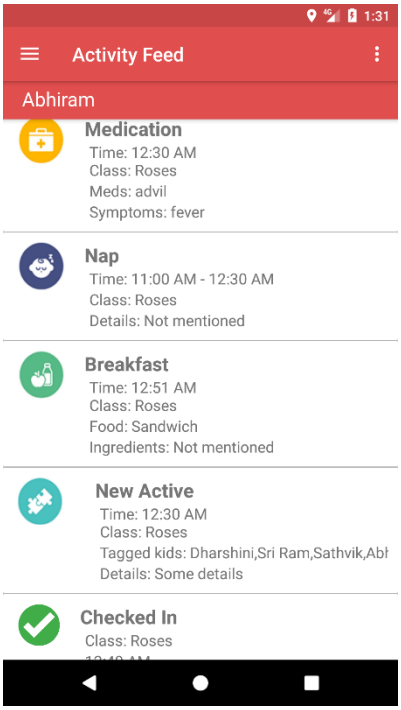


Now lets look at activity feeds, these are like news feeds on social media. Here, when a kid is tagged in an activity that activity appears on their activity feed. On the teacher side of the a[[, you get to a child's activity feed by clicking on a child in the list in the first page of the viewpager which has the list of all children in the class.

When you click on a child in the children list, screen changes to child's activity feed with ***Shared element Transition*** with the child's picture as the shared element.

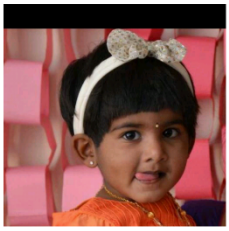






2:10

Dharshini's Profile



Personal Details

Dharshini

Date of Birth

Parent Contact

Harish

1234567890

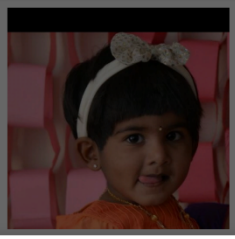
Health Details

Medication

Allergies

2:10

Dharshini's Profile



Upload your Photo

Take Photo

Choose from Gallery

rent Contact

rish

34567890

alth Details

dication

rgies

2:10

Dharshini's Profile

Personal Details

Dharshini

Date of Birth

Parent Contact

Harish

1234567890

Health Details

Medication

Allergies

Notes

Save Profile

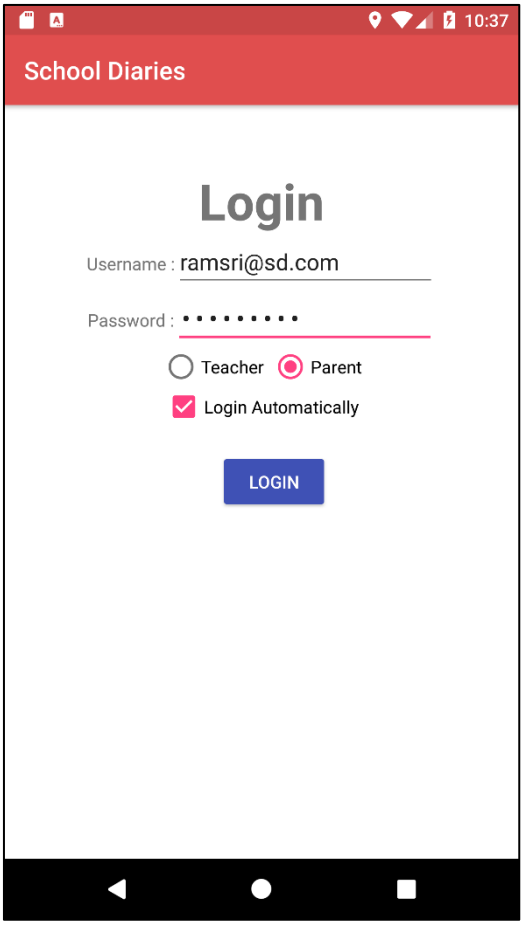
Delete Child Profile

Parent Side

The parent side of the app is basically for viewing the activities of their kid at the school. It is not just for the parents of the kid, infact even their grandparents and other family members can see what their lovely kid is doing at school.

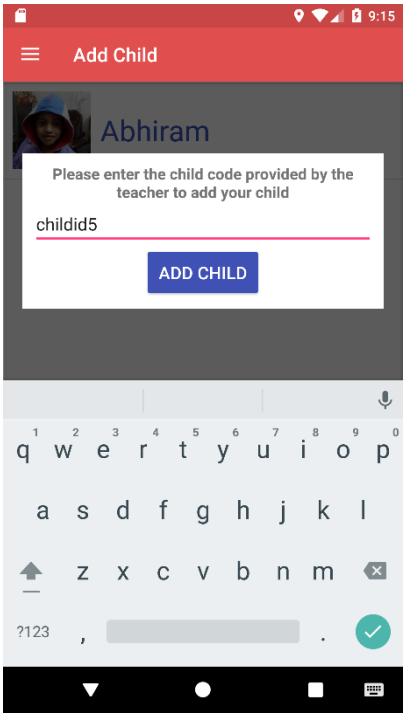
App Login:

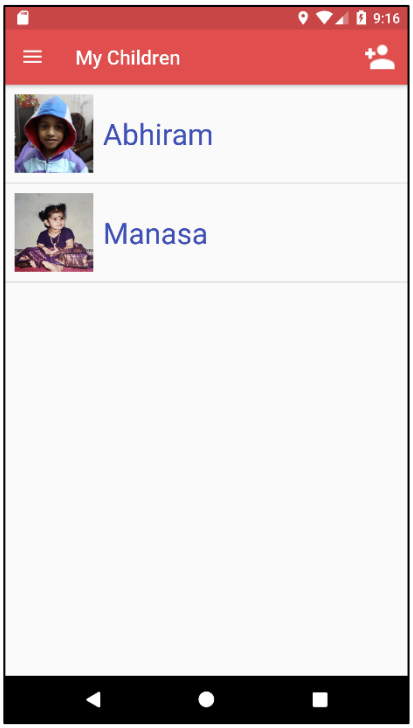
Creating account and logging into the account is the similar to that of the teacher side of this app. All they got to do is user valid credinetials to create their account. They just need to check the “Parent” radio button indicating that you need to register to a Parent type account. All this data will be saved in the ***Firebase Cloud storage***. And also, they can use the “Login Automatically” check box to remember their credentials to login automatically when they reopen their app. For this, we used ***Preferences*** to store the Username and Password of the current logged in user.



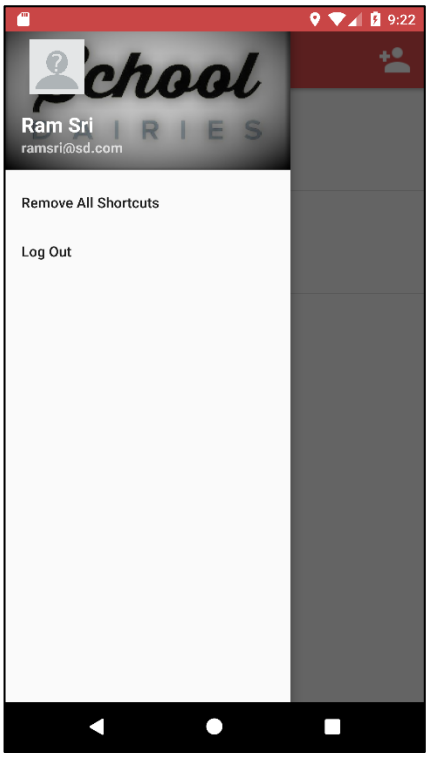
Once the parent logs into his account, he gets the list of his children he added to this account. All the names of the kids get loaded in a **Recycler View**. So, it gets dynamically updated whenever a new child is added to this account.

Speaking of Adding a new child to their account, the can use the “Add Child” button in the top right of the toolbar on this page to get this done. On clicking this button, the user gets prompted to enter the valid child code to add a particular kid to the list. We have provided enough checks so that the child gets add only if a correct Child code is entered.





In the parent side, we also included “***Navigation Drawer”*** which provides user to quickly **Logout** of his account and also to create ***Shortcuts*** which he/she will be able to create. Along with these features, it also loads the current user’s Name, email id and the profile picture.

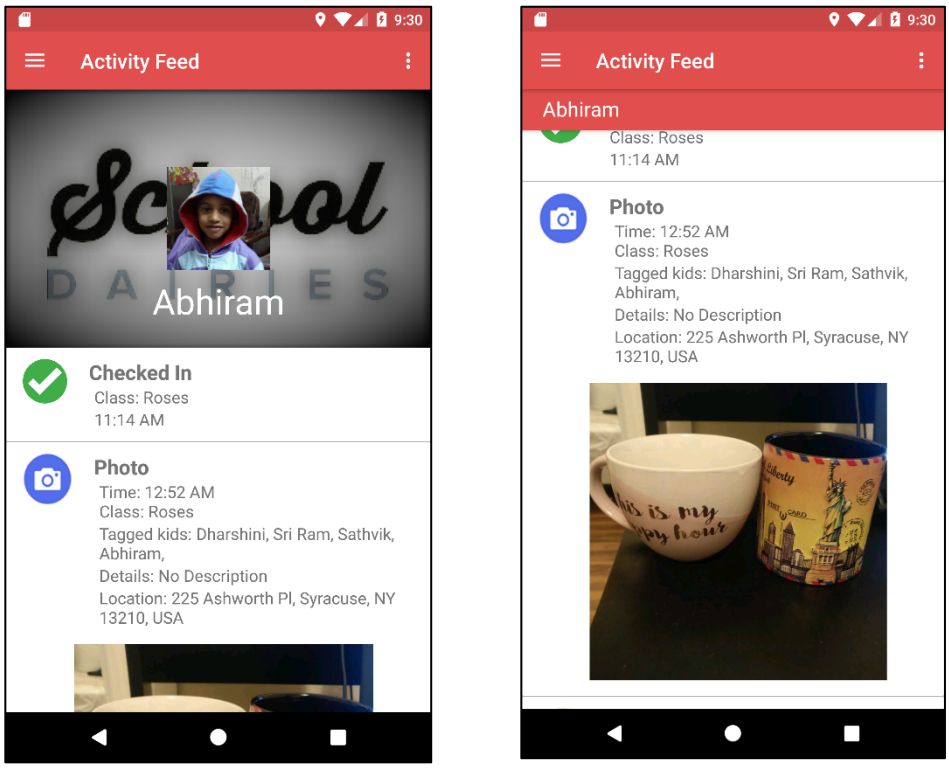


Coming back to the child list, on clicking the child name, it takes to the Activity feed of that particular child. In this feed, only those activities in which this kid has been tagged will be appeared here.

Let's talk about the different features this feed has got.

1) **Coordinator Layout:**

The parent layout of this fragment is a Coordinator layout. It loads the profile picture of the kid which is stored in the ***Firebase Storage*** and the name which is stored in the ***Firebase database.*** This entire view is implemented inside the collapsible toolbar, so it gets collapsed when the user scrolls down to look into the feed.



2) **Recycler View with different Card Layouts:**

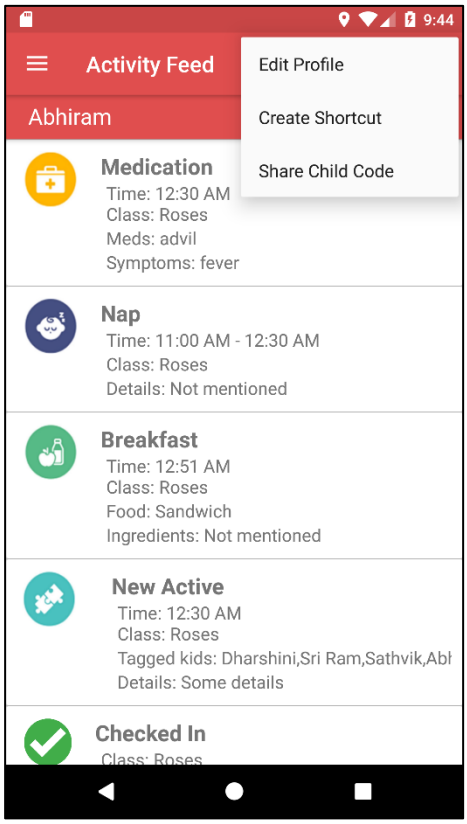
All the activities that have been pushed by the teacher on to Abhiram's (kid) will be displayed in a recycler view. However, each activity has different details that are to be displayed, so we used different card view layouts to display them.

This feed gets updated dynamically, that is, any new activity pushed by teacher gets updated to this feed dynamically. And also, we put ***Advanced Animations effects*** to make it fun while loading a new activity on to the feed.

3) **Shared Element Transition:**

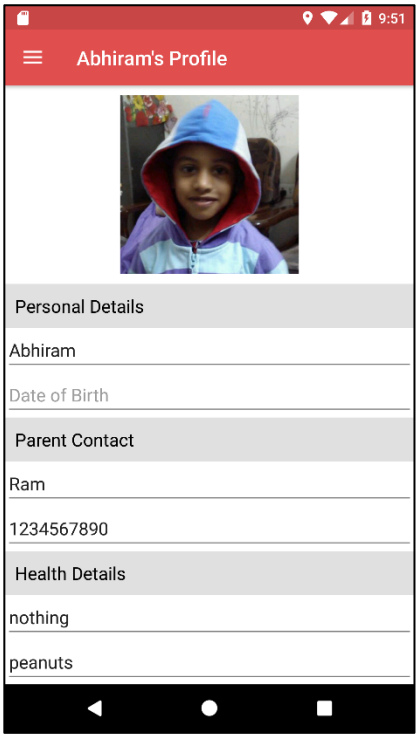
Speaking about transitions how could we miss the amazing ***Shared Element transition.*** In our app, we attacked the shared element transition to the profile picture of the kid. So, whenever user clicks on a kid in the Child list fragment, it shares the image from that fragment to the profile picture of the kid in this Activity feed fragment.

In the same Activity Feed fragment, we also have a ***Toolbar with multiple items*** which provides couple of important options for the user.



1) **Edit Profile:**

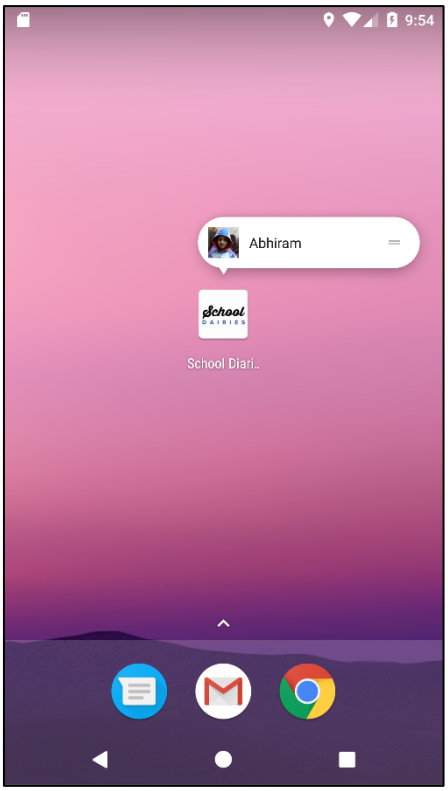
This provides the user to edit the profile of this current kid. The user can change the profile picture of their kid. For this we provided user with two options, they can either capture photo from the camera, or upload it directly from the gallery. We user ***Camera API*** for capturing photo and ***Gallery API*** to upload the photo. We also included image compression feature in the entire app where we used this feature to compress the image, crop the image into square for better performance of the app. Once all the changes have been made, the image gets uploaded to the ***Firebase storage*** and all the data gets saved into the ***Firebase database***.



2) Create Shortcut

This button creates a shortcut to the current child's Activity Feed. So from now on whenever the user clicks on that shortcut, he gets landed directly in the activity feed fragment.

In our case we used "Dynamic Shortcut" to create this shortcut. We have also attached the profile picture and the name of the current child for this shortcut.

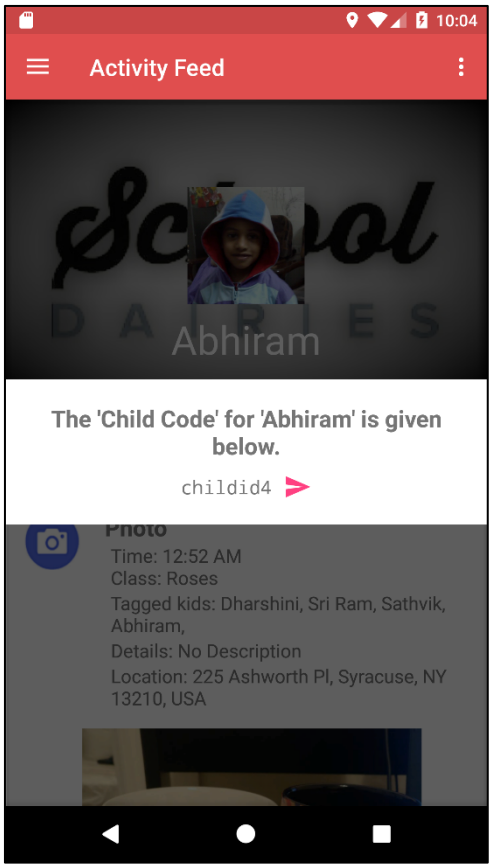


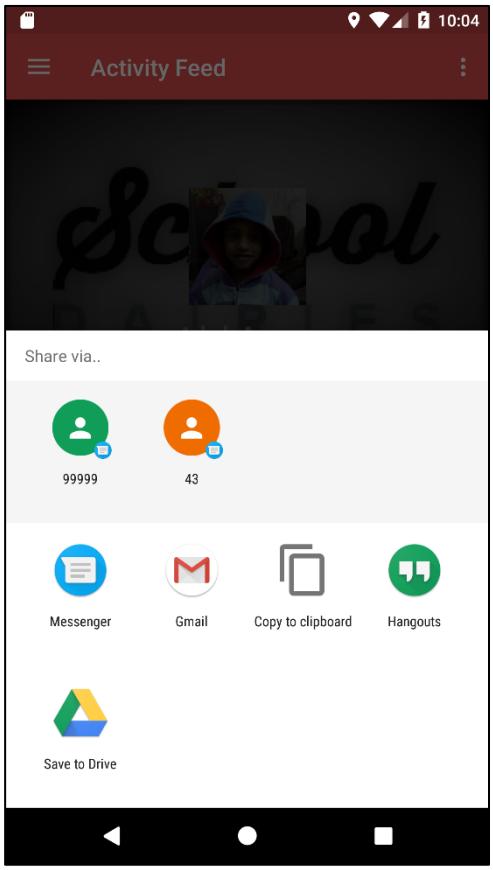

```
public void onResourceReady(Bitmap resource, GlideAnimation glideAnimation) {
    ShortcutManager shortcutManager = getActivity().getSystemService(ShortcutManager.class);
    Intent intent =  new Intent(getActivity().getApplicationContext(), ParentMain.class);
    intent.setAction("SHORTCUT");
    intent.putExtra("childid",childid);
    intent.putExtra("childName",childName);
    intent.putExtra("childPic",childPic);
    ShortcutInfo shortcut = new ShortcutInfo.Builder(getActivity().getApplicationContext(), childid)
            .setShortLabel(childName)
            .setLongLabel("Open "+childName+ " Activity Feed")
            .setIcon(Icon.createWithBitmap(resource))
            .setIntent(intent)
            .build();
```

3) Share Child Code

As we mentioned earlier, this app can also used by other family member to view the activities of this kid. So, the child code of this current kid can be sent to other people via SMS or other data sharing apps like whatsapp. For this feature, we used the ***Content Provider*** to send a message that contains the child code that needs to be put in while adding a child to their list.

```
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, myMessage);
sendIntent.setType("text/plain");
startActivity(Intent.createChooser(sendIntent, "Share via.."));
```



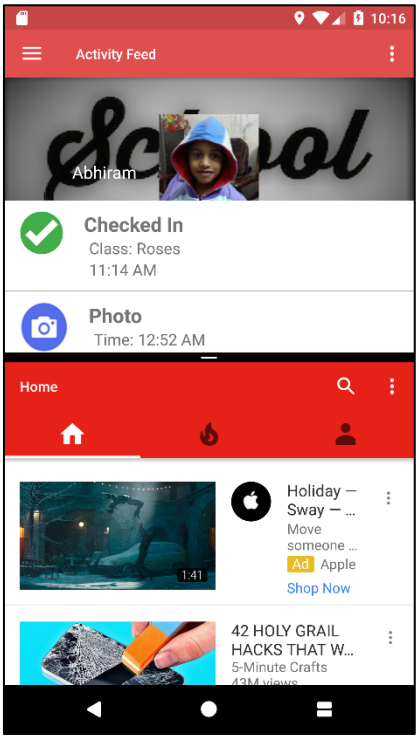


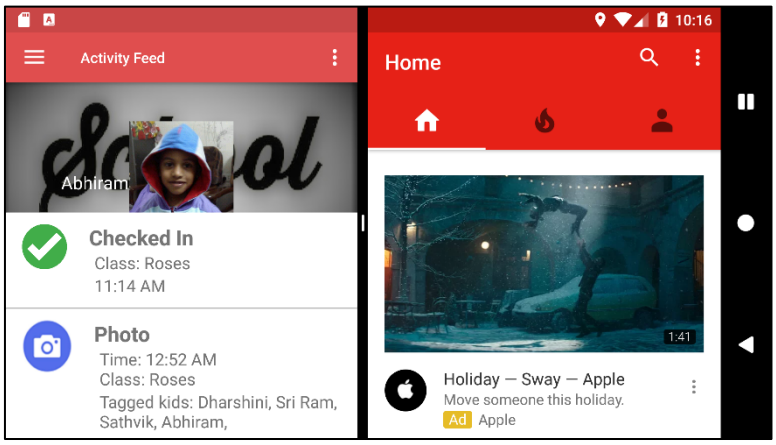
For the purpose of ***Data Security,*** we included another feature to our app such that whenever the parent logs out of this profile, all his shortcuts will be deleted. And also, we have provided a button in ***Navigation Drawer*** which deletes all the created shortcut buttons.

Other Features:

Multi Window Support:

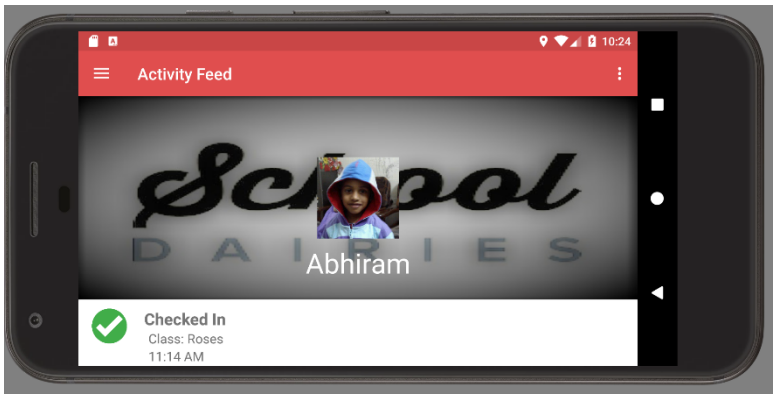
Our app supports multi window feature so that you go multi-tasking with our app. Just drag our app onto a multi window and enjoy multi-tasking.

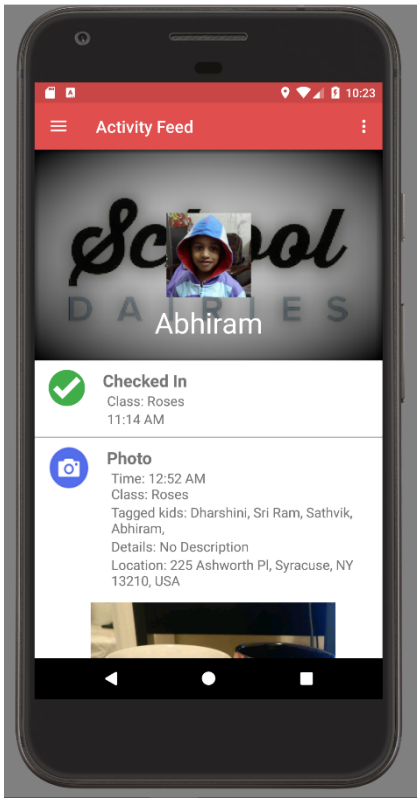


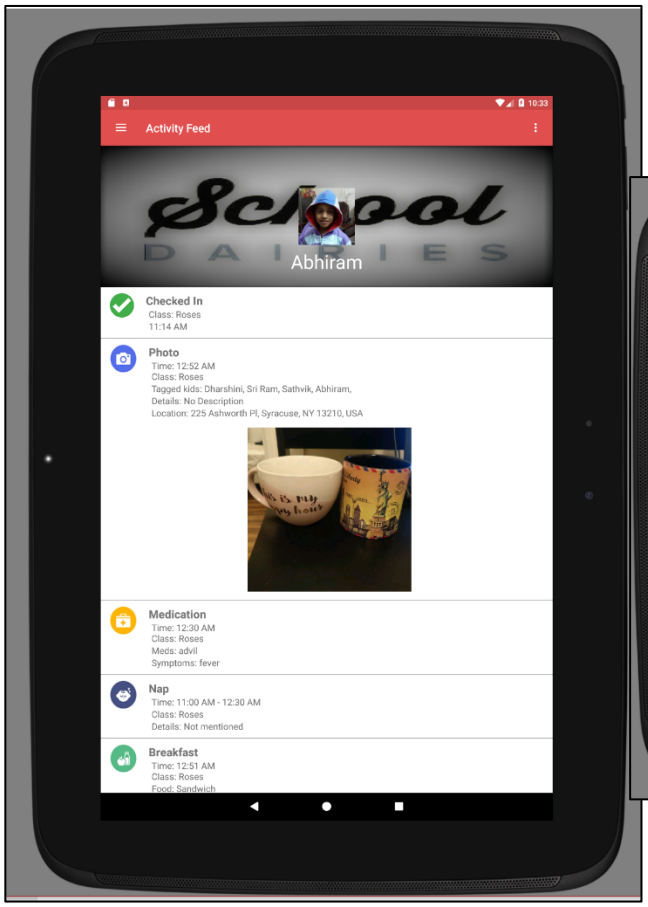


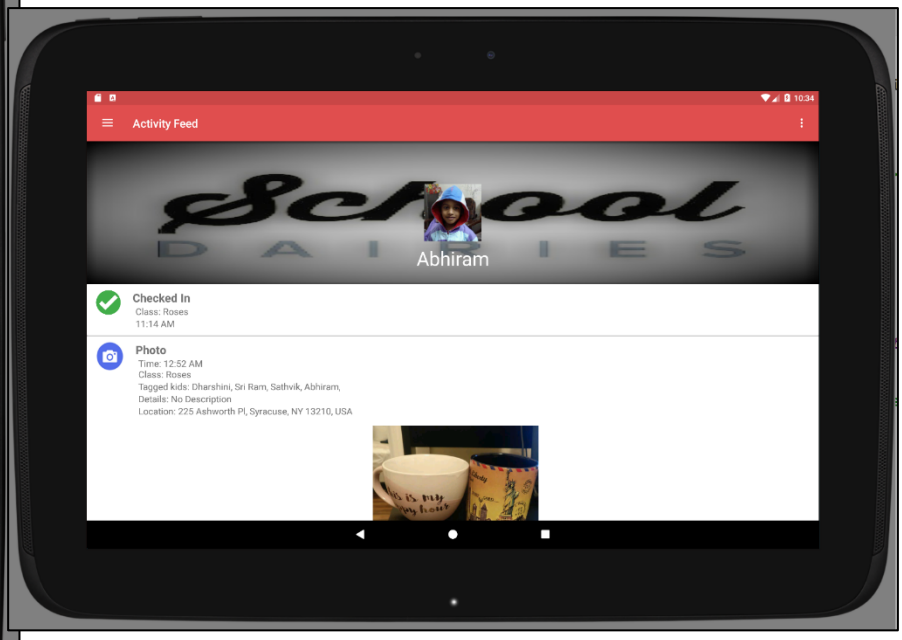
Multiple Device Support:

Our app is also designed to support multiple screen sizes. It works on all android phones and tablets of any size like a charm. It also supports **Orientation changes** by adjusting the screen accordingly for landscape and portrait mode of your device. And also, no data gets lost due to the orientation changes.






10:33
Activity Feed
School
DAIRIES
Abhiram
Checked In
Class: Roses
11:14 AM
Photo
Time: 12:52 AM
Class: Roses
Tagged kids: Dharshini, Sri Ram, Sathvik, Abhiram,
Details: No Description
Location: 225 Ashworth Pl, Syracuse, NY 13210, USA
Medication
Time: 12:30 AM
Class: Roses
Meds: advil
Symptoms: fever
Nap
Time: 11:00 AM - 12:30 AM
Class: Roses
Details: Not mentioned
Breakfast
Time: 12:51 AM
Class: Roses
Food: Sandwich


10:34
Activity Feed
School
DAIRIES
Abhiram
Checked In
Class: Roses
11:14 AM
Photo
Time: 12:52 AM
Class: Roses
Tagged kids: Dharshini, Sri Ram, Sathvik, Abhiram,
Details: No Description
Location: 225 Ashworth Pl, Syracuse, NY 13210, USA