HACKHUNT SOCIAL APP

This is a social media app that is used to let people share their life hack ideas with other users. Initially the users have to login to their accounts. Users can share pictures of their life hack ideas into their profiles and have description for their posts. Other users can view them and leave their comments for every life hack post.

Group 2
Ying Deng, Jin Hung Chin, Sadaf Riazi

Link to video presentation:
https://youtu.be/bblrNrwcaxo

Contribution:

Ying Deng (300340489): launching page, main activity(menu & ActionBar), login & signup, user's posts activity(posts with delete and edit features), view post activity, recommendation page (40%)

Jin Hung Chin (300338902): importing images, new post uploading, comment section (leave and display comments), bottom navigation bar (35%)

Sadaf Riazi (300342525): home page(layout and display all posts), edit profile(update user information), project report (25%)

About HackHunt

HackHunt is a social media app that is used by people to share their lifehack ideas with other users. When the user launches the app, the app initially shows the launching page and then takes the user to login page. If the user has an account with HackHunt, then they can just simply login using their username and password. Otherwise, they can sign up as a new user. Each user has a profile in which shows all their picture posts that they have shared with others. Users can enlarge their pictures in their profiles to view the description and comments of each post or delete any post they don't like. Users can share a new image post with description at any time. They simply need to use their photo gallery to upload an image post into their profile. Using the bottom navigation bar, the user can navigate through different tabs of the app. In the home page of the app, the user can view other users' life hack posts based on the date posted and leave comments for them. Users can edit their profiles to change their password or add an email to their account. In the recommendation tab, users can enjoy finding different useful lifehack videos from YouTube which are recommended by the app.

Motivation

The main motivation of the app was to create a unique social media app for people to share their interesting and quick life hack ideas. We live in a world where time is very essential. Some quick tips and ideas to make life easier and quicker is very useful and interesting to majority of people. Many people enjoyed TikTok for the lifehacks that the users were sharing there. Now HackHunt can give this opportunity to people to have a platform to share and receive useful life hacks to make their lives faster and easier.

Setting up Parse and AWS

We chose Parse to start our App. Parse is an open-source framework to develop application backends. Using Parse technology to develop social network apps is handy for beginners, so we used some basic functions of Parse technology and we use the Parse server of Amazon AWS to store data.

First of all, we set up an instance in AWS, which is quite complicated and time-consuming for a beginner. After that, we added dependencies in build.gradle and write a StarterApplication in our project to make it work.

MainActivity

The main activity includes the menu and the ActionBar. The menu on the top right is for edit profile and logout functions. We created a menu xml file under resource folder and then we have onCreateOptionsMenu() and onOptionsItemSelected() methods in MainActivity to make it work.

The ActionBar is mostly what we've been taught in class but the extension here is that we use setDisplayHomeAsUpEnabled(true) and then we modify the manifest to add parentActivityName in each activity tag. In this way, when the user click the arrow on the top left, they can go back to the parent activity of the current activity.

The fun part is that the most of other activities extend MainActivity so they can have the menu and ActionBar functions without the hassle of copying and pasting the code.

Bottom Navigation Bar

The bottom navigation bar is there to allow user to visit main different activities of our app which includes home page activity, recommendation page activity, user's personal profile page activity and creating new post activity. We are using Switch Case method to perform the task of taking user to each tab that they click on. Navigation bar is implemented all the main 4 activities. To create our bottom navigation bar, we will first add the BottomNavigationView in most of out XML layouts, then create a menu_nav xml under the resource folder. After that we assigned four vector icon for each menu items and place

them in the bottom navigation view to be displayed out. Each of them will then have an on-click listener which start different activities accordingly.

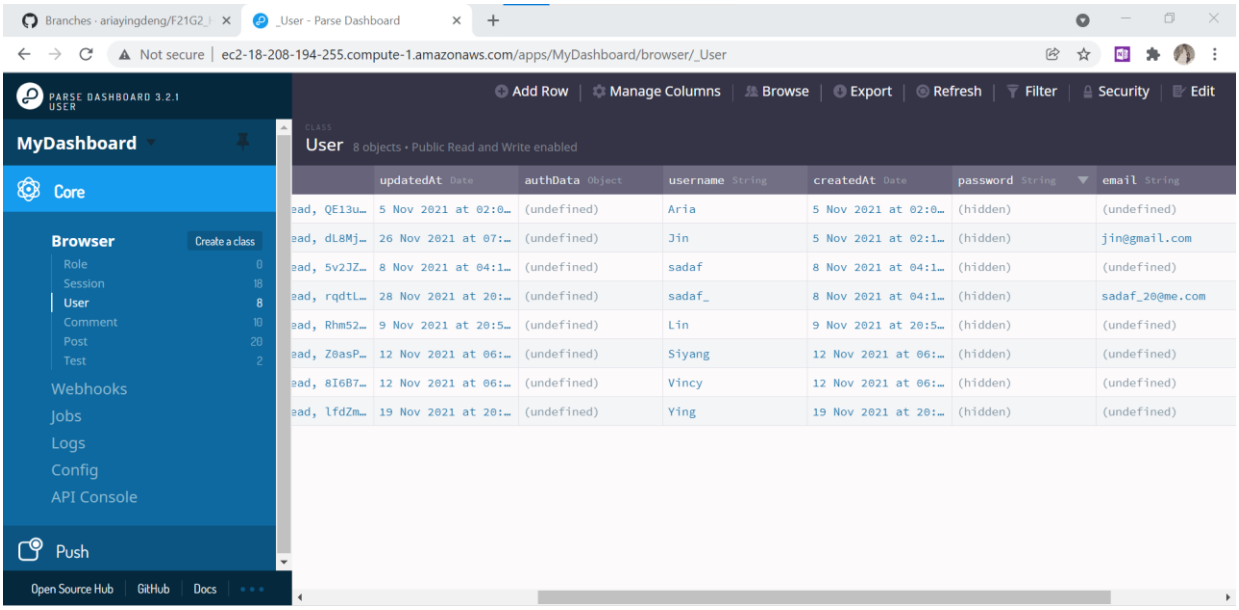
Launching Activity

Initially when the user launches the app, a welcoming page displays for a few seconds. In the launching activity we used an Image view to welcome users and display name and logo of our app. In this page we used TimerTask and Timer utility classes to perform the task of taking user to login page after showing the welcoming page for 3000 milliseconds.

In this page, ImageView and TextView are used on a constraint layout to display logo and welcoming phrase.

Login and Sign-up Activity

For this section, we are using parse server to connect to the database for Users table. The Users table includes ID (String), username (String), time that user account was created (String), password (Hidden String) and email (String).

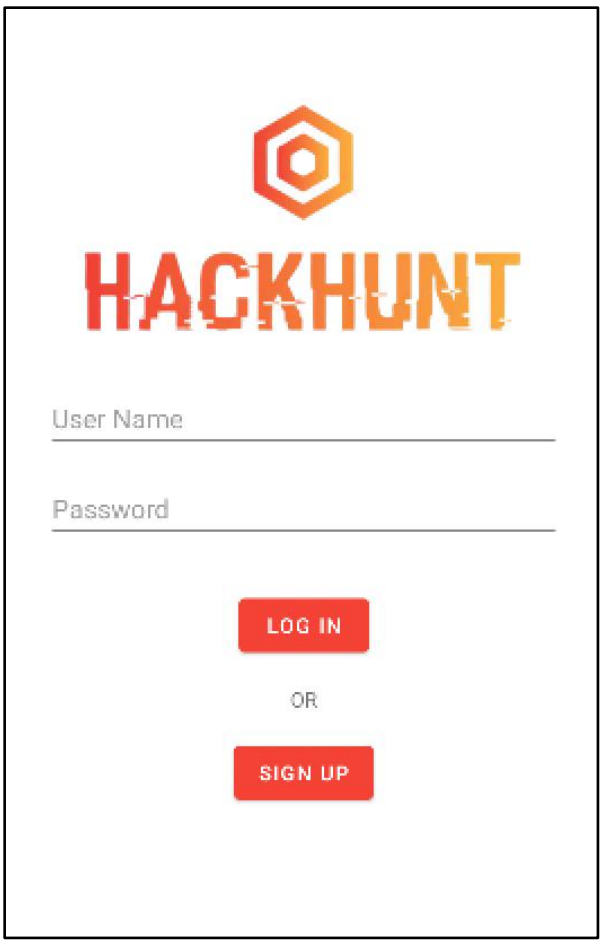


- Above is the User table created in Parse Database.

In this activity, we used two EditText views to take user's username and password. Users can either click on login button to sign into their account or click on the sign-up button to create a new account with HackHunt.

For Login, if the information is not missing, it will take the username and password to parse server database to verify if they exist. If the information is valid and exists in database user table, then the user is logged in and is taken to the Recommendation page. Otherwise, there will be a toast for invalid login information.

For sign-up, if the information is not missing, it will create new ParseUser object in user table with the new user's username and password. Once the object of new user is created in database, there will be a toast that the sign up was successful and user is ready to login. Otherwise, there will be an error.

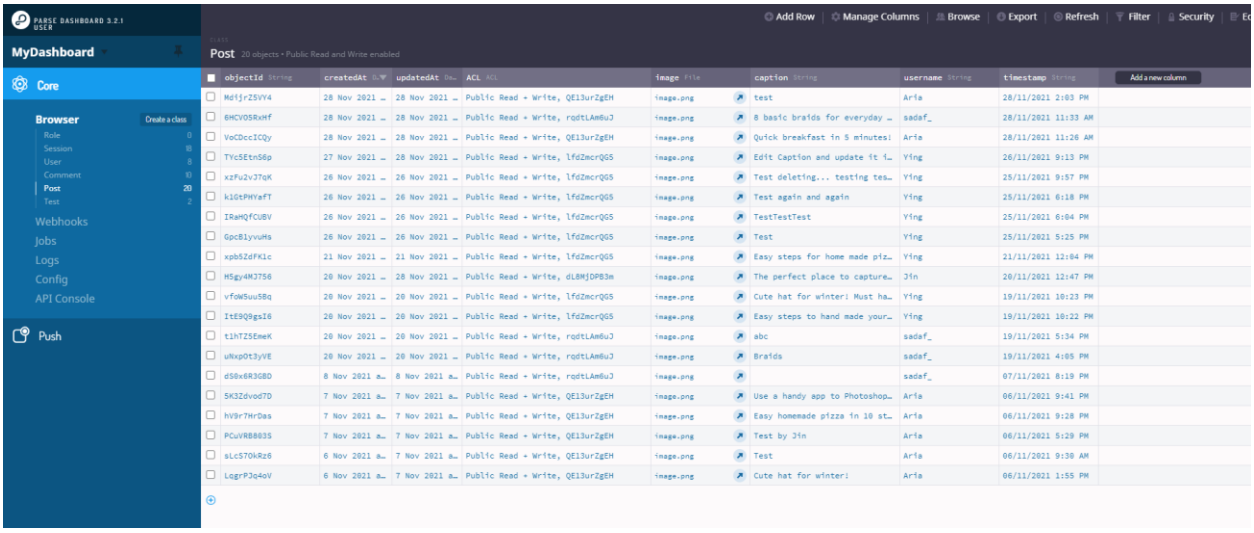


In this page, an ImageView, 2 TextView with hints and 2 Buttons are used on a constraint layout for users to login or sign-up into an account.

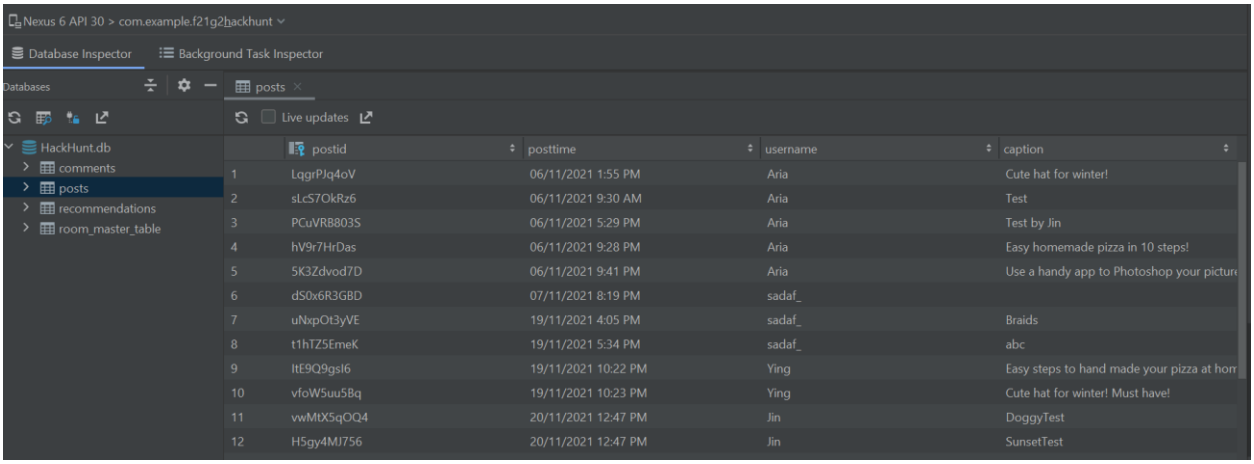
New Post Activity

To create a new life hack post, the user first needs to pick and image from their gallery. Initially the user must click on the create new post image to choose an image from gallery. To make this possible, we are using ActivityResultLauncher to start an intent which will access the device's local photo gallery. Users will then pick the image they wished to be posted. Once the image is selected and the caption is written, user can click on the post button to post the new life hack to their account. Once the post button is clicked, if the image and caption are not missing, the caption is set to a string and the image is converted from Drawable to BitmapDrawable to Bitmap and finally into Parse file. Next we get the username of current user through ParseUser, create timestamp for the post to be created using Calendar class in Java and lastly we will create new parse object for the

to create the Post table in the database and save all the previous post's details in it. The Post table will have timestamp (String), username (String), caption (String), image (Parse File). Interestingly, for every new post instead of just storing the post on Parse database, we added the functionality to stored the post in SQL database as well. So the storage of a new post was done simultaneously on Parse database and SQL database. Once the process is successful, user can will get a toast noticing them that their post have been shared. However if there is an error, and a Toast is given to user that there is an issue sharing the post.

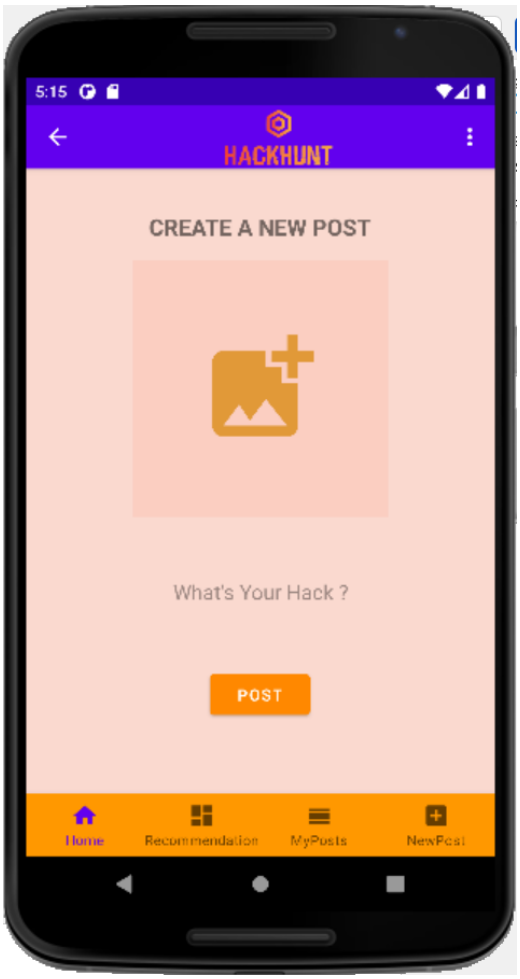


- Above is the Post table created in Parse Database.



	postid	posttime	username	caption
1	LqgrPJq4oV	06/11/2021 1:55 PM	Aria	Cute hat for winter!
2	sLcS7OkRz6	06/11/2021 9:30 AM	Aria	Test
3	PCuVRB803S	06/11/2021 5:29 PM	Aria	Test by Jin
4	hV9r7HrDas	06/11/2021 9:28 PM	Aria	Easy homemade pizza in 10 steps!
5	5K3Zdvod7D	06/11/2021 9:41 PM	Aria	Use a handy app to Photoshop your pictur
6	dS0x6R3GBD	07/11/2021 8:19 PM	sadaf_	
7	uNxpOt3yVE	19/11/2021 4:05 PM	sadaf_	Braids
8	t1hTZ5EmeK	19/11/2021 5:34 PM	sadaf_	abc
9	ItE9Q9gsI6	19/11/2021 10:22 PM	Ying	Easy steps to hand made your pizza at hom
10	vfoW5uu5Bq	19/11/2021 10:23 PM	Ying	Cute hat for winter! Must have!
11	vwMtX5qOQ4	20/11/2021 12:47 PM	Jin	DoggyTest
12	H5gy4MJ756	20/11/2021 12:47 PM	Jin	SunsetTest

- Above is the Posts table created in SQLite Database.



In this activity layout, A Text-View is used to show the header, a clickable ImageView is used to get image and a Text-View with hint to get the caption. Then there is button to create the post at the bottom. Everything is set on a Constraint Layout.

User's Posts Activity

We use ScrollView with a vertical layout inside it for user's post page and we use ParseQuery to get all the posts of the logged user and put the data including the username, picture, caption and date into the post layout and add the post into the scroll view.

For the image of each post, we retrieve it from database as a ParseFile and get the data to convert it into a Bitmap and use the method setImageBitmap to make it viewable.

We use the object Id for each image post to delete, edit and save them if needed. If the user clicks on DELETE, the post and all of its comments are deleted from both of our two database servers using the post Id.

Once the user clicks on EDIT, the text view for caption and save become visible for user to edit the caption and save it once done. When the user clicks on SAVE after editing the caption, the text is saved as a String and updated into database using the postID for the newCaption and the text views for edit and save become hidden.

Two methods of deleteComment and editPostCaption are implemented to help these operations. The deleteComment method, query the postId from the Comment table and if it exists delete the object of the post comments from database. The editPostCaption, query the object of the post from Post table and update the caption for the newCaption column.

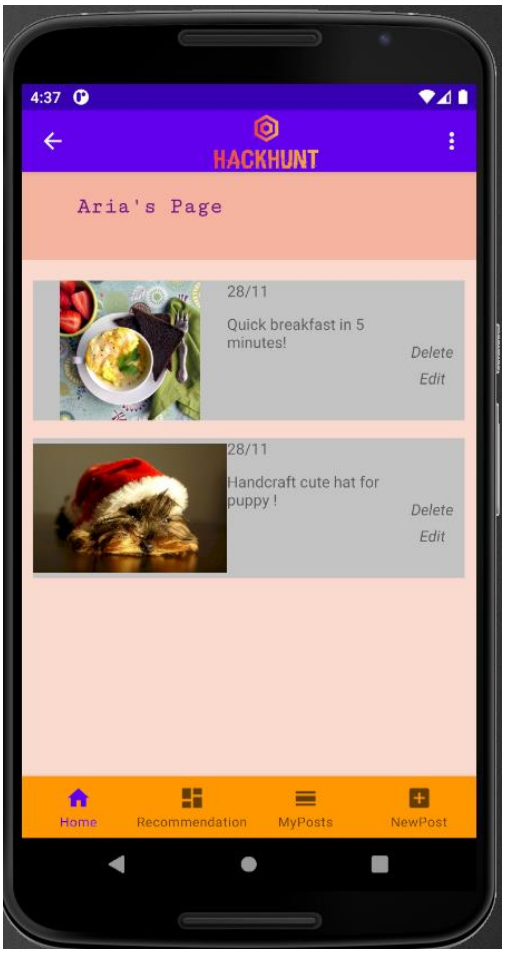
The posts are clickable for users to open them and start the new activity of view post and view the comments.

We also migrate data related to posts and comments from Parse to SQLite database. We exported all the existing posts and comments in the Parse as CSV files and insert the lists into the SQLite.

existingpostslist.csv | existingcomments.csv

```
objectId,caption,username,timestamp
LqgrPJq4oV,Cute hat for winter!,Aria,06/11/2021 1:55 PM
sLcS7OkRz6,Test,Aria,06/11/2021 9:30 AM
PCuVRB803S,Test by Jin,Aria,06/11/2021 5:29 PM
hV9r7HrDas,Easy homemade pizza in 10 steps!,Aria,06/11/2021 9:28 PM
5K3Zdvod7D,Use a handy app to Photoshop your pictures!,Aria,06/11/2021 9:41 PM
dS0x6R3GBD,,sadaf_,07/11/2021 8:19 PM
uNxpOt3yVE,Braids,sadaf_,19/11/2021 4:05 PM
t1hTZ5EmeK,abc,sadaf_,19/11/2021 5:34 PM
ItE9Q9gsI6,Easy steps to hand made your pizza at home! ,Ying,19/11/2021 10:22 PM
vfoW5uu5Bq,Cute hat for winter! Must have!,Ying,19/11/2021 10:23 PM
vwMtX5qOQ4,DoggyTest,Jin,20/11/2021 12:47 PM
H5gy4MJ756,SunsetTest,Jin,20/11/2021 12:47 PM
utYjIGJidy,Edit Bridge Test ,Jin,20/11/2021 12:48 PM
iB27sXlRFR,FieldTest,Jin,20/11/2021 12:48 PM
lWhyMcSgHm,SceneTest,Jin,20/11/2021 12:49 PM
xpb5ZdFK1c,Easy steps for home made pizzas!,Ying,21/11/2021 12:04 PM
GpcB1yvuHs,Test,Ying,25/11/2021 5:25 PM
IRaHQfCUBV,TestTestTest,Ying,25/11/2021 6:04 PM
k1GtPHYafT,Test again and again,Ying,25/11/2021 6:18 PM
```

- Above is the CSV file used for migration of data

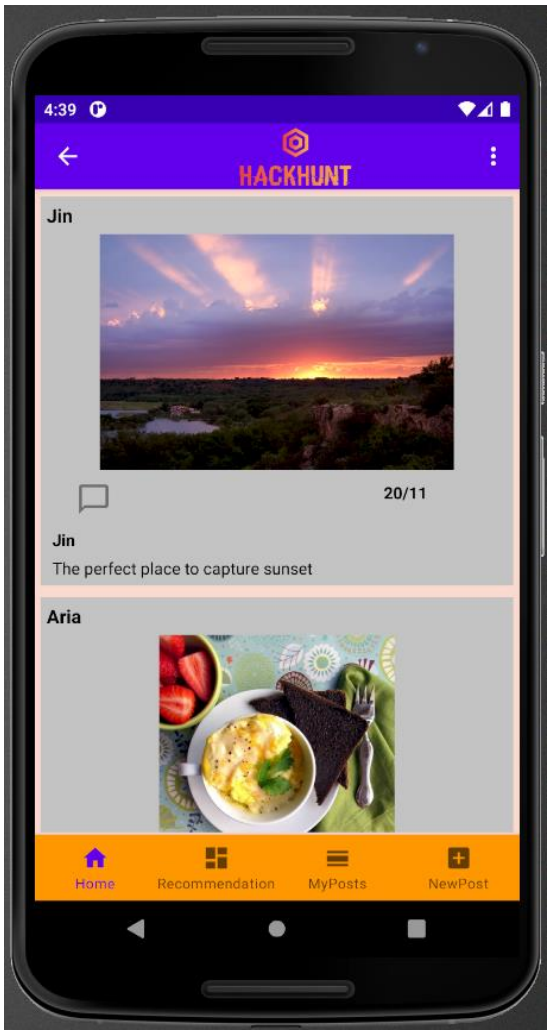


In this page, we are having a horizontal linear layout for each post to display the post image, caption an date, delete and edit clickable texts. The user profile has a constraint layout to view the username on top of the page and vertical scroll view to scroll up and down to view different posts. There is Bottom navigation toolbar at bottom of the page to navigate through different tabs.

Home Activity

In the home activity, the user can view all the posts created by every user. The home activity is ordered based on the time the post was updated. Again, we are using parse server to get all the posts data from the database. We implemented a method to retrieve the object for each ParseFile image from our database and then we query username, caption and timestamp based on each image. We decode the ByteArray for the image to become viewable by users. All the fetched data from the database is set to our layouts

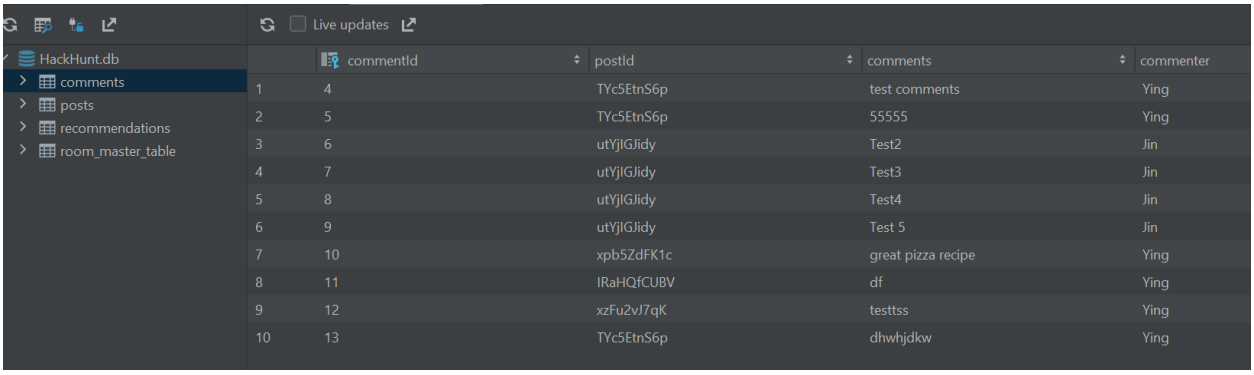
for display. The posts are clickable for users to open them and start the new activity of view post and view the comments.



This page has 4 text views to display username twice, the date and the caption for each post. There is a clickable image view for comments to be clicked and the comment section to view for each post. There is an image view to display the image of each post. Each post is set on a relative layout. The home page is set on a vertical scroll view to scroll up and down to view different posts. There is Bottom navigation toolbar at bottom of the page to navigate through different tabs.

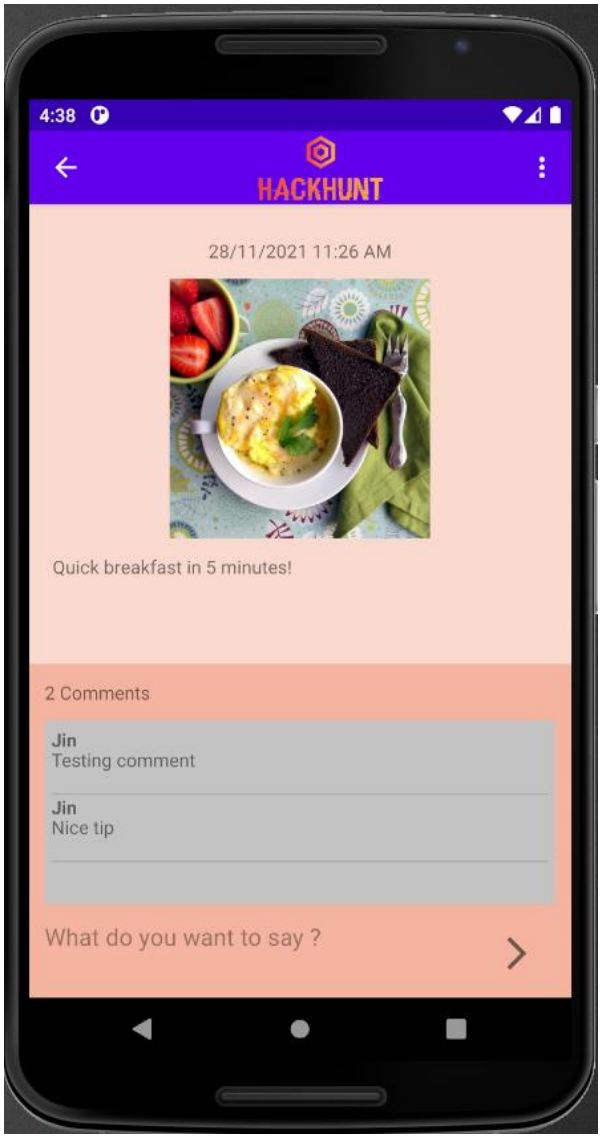
View Post Activity

When the user clicks a particular post, the ViewPost Activity will show the details including the larger picture and the comments down below. We use bundle to pass information we need such as username, picture, caption, date and the post id into the view post activity. There is also a comment section where users can leave their comments for the post. In a technical perspective, leaving comment for a particular post will require a main key which is the postId. The postId came from the query through Post table in parse database and then put into bundle and carried over to View Post Activity to be used. After having the postId, we will just get the name of current user as commenter, the user's input as comment and saved them into sql database. If the comment is stored successfully in the database, a toast will notify the user that their comment is shared successfully. Otherwise, an error message will occur saying issue sharing your comment. To display the comments out from database, we need will need to first query the database with the commentDao. Then, we create a comment list and populate the list with the comments from the database. The list will then go through an adapter and finally displayed in the list view.



	commentId	postId	comments	commenter
1	4	TYc5EtnS6p	test comments	Ying
2	5	TYc5EtnS6p	55555	Ying
3	6	utYjlGJidy	Test2	Jin
4	7	utYjlGJidy	Test3	Jin
5	8	utYjlGJidy	Test4	Jin
6	9	utYjlGJidy	Test 5	Jin
7	10	xpb5ZdFK1c	great pizza recipe	Ying
8	11	IRaHQfCUBV	df	Ying
9	12	xzFu2vJ7qK	testtss	Ying
10	13	TYc5EtnS6p	dhwhjdkw	Ying

- Above is the Comments table created id SQLite database

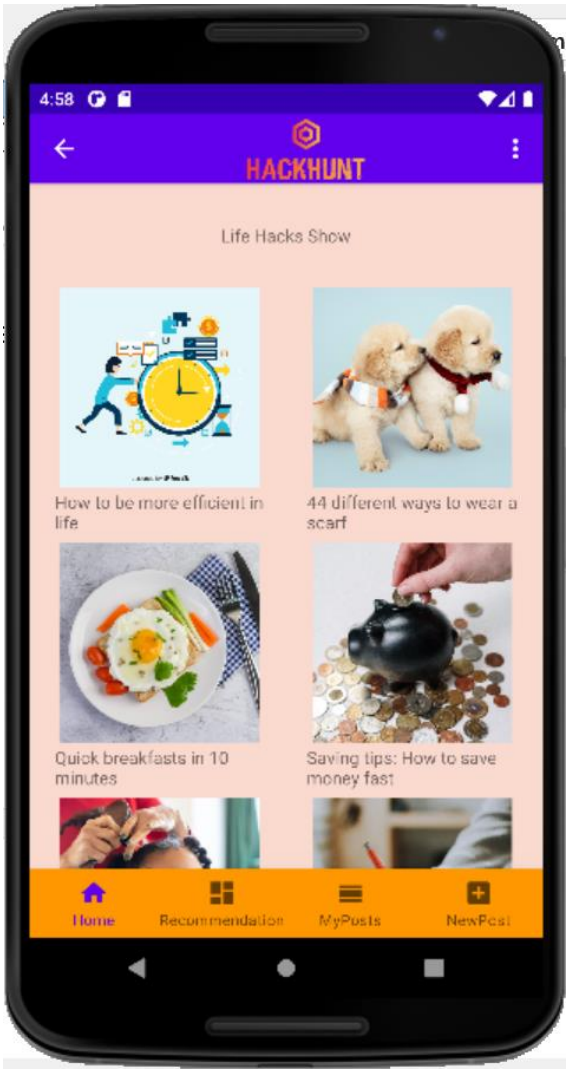


In this post layout, we are using vertical linear layout to include our text views for date and caption and image view for our image. The linear layout and comment section are included under a constraint layout. For the comment section, there is a text view for comment title, and a list view to display all the comments for each post. Users can write their comment and use the clickable post image on the right side to post share their comments for the life hack post.

Recommendation Activity

Our recommendation page is a GridView that displays the contents that our app provide. We store the pictures, descriptions and video links in the Recommendation entity of our

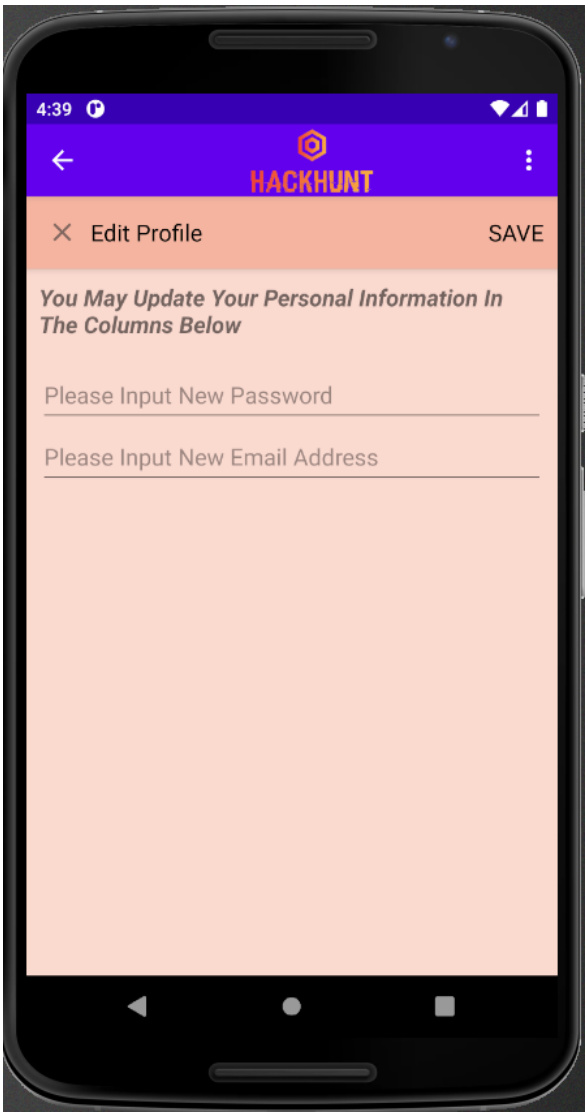
SQLite database. When the user clicks a recommendation post, it will go to the particular YouTube link to play the video.



Edit Profile Activity

In the edit profile activity, the user can change their password and set email to their account. The user is asked to enter their password and email. Once the user hits the save button, if the information is not missing, the parse server database is updated with new password and email for the user and the Toast is given that the information is successfully updated. Otherwise, there will be an error. The user has an option of closing this tab if they do not wish to change their personal information on their account. Normal parse query cannot be used because the user information is critical and protected in the

database so order to update information, we had to authenticate the user with parse user and then we could do the update through parse user object.



This page has a Linear Layout with a toolbar on top of the page to display, clickable close image, edit profile text view and clickable SAVE text view. There is a linear vertical layout for the user to view the text views for title, edit text for password with hint, edit text for email with hint.