**Android Manifest File:**

In his file all activities are written and also written that which activity will run first. In the Manifest file there is also a provider tag that is for picking the files from the gallery and sharing to the other applications from our application. Application icon, rounded icon, name and theme of the application.

**Package EditProfile:**

In this activity .kt file **updateProfile()** method will get the currently logged in and get data from the Edit texts and update the profile of the person that is logged in

In the same activity **getData()** data will be retrieved from the firebase according to the currently logged in person and set into the image view and edit texts.

Then in the **onCreate()** method set on click action the update button so when the button pressed it will update the profile.

**Package messages:**

**ChatLogActivity:**

In the this package in the ChatLogActivity .kt file i perform send and receive messages functionality also time functionality.

In the **onCreate()**  method I change the name of the title bar with the user name to whom the message. Then set actions on the buttons like send messages and pick media.

In the **listenForMessages()** method I get the id of the current user and target user id and using that id’s i retrieve the messages of the both users that send each other. And set the messages to the TextVIew in the recyclerView if there is an image that they share i set that image into the ImageView and set the time also.

In the **performSendMessage()** i get the id’s of both users and make a reference in the firebase in the very organized way like sending messages folder/who is sending / who is receiving. And push the message that is written inside the editText if the edit text is empty then there should be an image so the image will upload on the firebase storage and the link will be pushed here. Make a CHatMessage object and pass it the user id’s image reference and time.

In **onActivityResult()** i write code for the uploading the image into the firebase storage and also write the code for the picking image from the gallery and save the download link into the string so we can upload the on tot the message in firebase.

**LatestMessagesActivity:**

**listenForLatestestMessage()** method i retrieved the latest message that any user sent and showed in the recycler view.

**verifyUserLoggedIn()**  In this method i check if the user is logged in or not in not then take the user to the login screen.

**onItemMenuSelected()** in this method I set an action on the menu and take the user to the activity according to the user menu selected.

**NewMessageActivity:**

**fetchUsers()** in this method i fetch all the users that are signup on my applications and show in the recyclerview.

**UserItem()** sets the image of the user in the image view of that user.

**Package model:**

Just two simple kotlin classes

**Package registerLogin:**

**LoginActivity**  in this activity in the **Login()** method I get the email and password entered in the editText and compare with the existing one and make the user logged in.

**RegisterActivity** in this activity in **onCreate()** method i write code for the google sign in

And get user id and make users able to pick images from the gallery.

**onActivityResult()** in this method i write code for the user to pick image and set into the imageVIew button and also write cod the upload google user info to the firebase

**firebaseAuthWithGoolge()** in this method makes the user login into the app and upload his complete information to the firebase database and firebase auth.

**updateUI()** in this method fetch the google signed in user complete information.

**performRegister())** this method will run if the user doesn't want to use google sign in method then he can put his information and can be registered to our app.

**uploadImageToFirebaseStorage()** in this method the user who manually fills his information and chooses his profile picture. His picture will upload into the firebase storage so it can be fetech in the application for further use.

**saveUserToFirebaseDatabase()** In this method the user's information will upload to the firebaseDatabase by getting from the editTexts and when pressing the **register** button.

**SplashScreen** in this activity splash screen time in handout will be managed in the **onCreate()** method using simple animation and handler and time delay for the logo that appears in the splash screen.

**Package ShareFiles**

**ShareFiles** in this activity user will be able to share files to the other applications in the **init()**  method and get the id’s from the xml files.

**setListeners()** in this method pass the click listeners method to the images and buttons.

**onActivityResult()** in this method if the user picks an image from the gallery it will open the gallery and the picture that the user picked will be set on the imageView.

**shareImage()** this method will perform when the user clicks the share button in this method the selected image will share to the other applications it will show the other applications for sharing.

**shareText()** this method will run when user write some text in the editText and press shareText button this method will open list of applications to which he can share that text.

**Utils**  this is basically java class in which the image will decoded into bitmap for sharing into the other applications i pass and image uri to this image and it decode into the bitmap.

**Package views**

**ChatItems** in this files there are two classes **chatFromItem** in this class th messages will handle which is sended from the user all the message data like user id message image time will be handle here and set in the **chatFromItem** xml file.

**chatToItems** i

**latestMessageRow** this class basically the handler of the recycler view which set new row of the new user message when the message arrived.

**Drawbale folder:**

In this folder we keep all the images and styling files that will be use in our xml files and the application icon will also stored here.

**Layout:**

In this folder all the xml files (Screens of the our application) will be store here.

**Activity\_chat\_log.xml**  in this file i use constraints layout and at the bottom a editText and two buttons and give them style from the drawable folder. And place a recycler view in the rest of the screen for messages.

**Activity\_edit\_profile.xml** in this xml file make screen for the edit profile i use relative layout and place rounded image view and edit texts according ot user information in the firebase and a buttons at the end of the layout.

**Activity\_latest\_message.xml** in this file only place the recycler view for the latest messages.

**Activity\_login.xml** in this file we place only a relative layout and edit text for the email and password and button for the login in the middle of the screen.

**Activity\_new\_message.xml**  in this file in the constraints layout place only ythe recyclerView for the all users registered in the application.

**Activity\_register.xml** in this file make a rounded button for set image and username,email,and password and a button and at the end of the layout there is a google button if the user want to sign up using his google account.

**Activity\_share\_files.xml**  in this file make scroll view and inside it make imagesview and buttons for sharing image and a edittext and button for the share text that is written in the edit text.

**Activity\_splash\_screen.xml** in this files make relative layout and a place a icon of firechat in the middle of the screen.

**chat\_from\_row .xml** and **chat\_to\_row.xml** both files are same place constraints layout and profile image view , textView, imageVIew for sending and received images, and time for the message sending and receiving.r

**Activity\_latest\_message\_row.xml** in this file just use a simple row a constraint layout and image view, text for the username and text for the latest message.

**User\_new\_row\_message.xml** in this file make a row for the user info that signed up in the application.

**Menu:**

**Nav\_menu.xml**  in this file make menu items according to our application screens .

**Values:**

**Styles,strings and colors:**

These files are the build in by the android like in the styles files there are themes of the applications and in the strings file all the hard coded string values that are used in the out applications. And the color diles the colors of the applications are placed.

**Xml:**

**File\_path.xml** in this files gte the permission of the internal and external storage like we place provide in the manifest file.  
  
  
  
  
**אופק**להשלים לפה את שיעור שעבר   
  
  
-יש לחדד כאשר אני נשאל על עיצוב אני מרחיב בדפי העיצוב וה xml  
כאשר אני נשאל לגבי קוד, תכנות, פונקציונאליות אני אגש ל activity  
**Java שאינם activity  
Models**-chatMassage **מחלקת עזר ששומרת את האלמנטים של ההודעות.  
  
  
-data  
מחלקה ששומרת את כל האלמנטים  
של משתמש .   
User -   
מחלקה השומרת על פרטי המשתמש.**  
**Notifications  
Client-**יצור לנו קובץ עזר לשיחת ה כתובת אל ה api  
  
  
  
**-MyFirebaseInstanceId**קובץ עזר לשמירת המפתח של המשתמש לאחר התחברות וזיהוי **.**   
  
**-MyResponse  
האם התגובה הצליחה.**  
**-OerNotification  
  
יציג את את ההתראות**  
  
  
**-Sender**מה הדאטה ולאן היא נשלחת  
**-Token  
מחלקת עזר של כתובות להתראות**  
  
**shareFiles  
  
-utils  
מתודה שמבצעת פעולה שהופכת uri  
לביטמאפ.**   
  
  
**Views  
-ChatItems.kt**  
-LatestMessageRow  
  
  
**Non activity xmls**chat\_from\_row.xml

chat\_to\_row.xml

latest\_message\_row.xml

user\_row\_new\_message.xml  
  
  
כעת נקח רק את קבצי הגאווה שאינם אקטיביטי  
נרחיב על כל אחד בנפרד,

קבצי אקסאמאל שאינם אקטיביטי.  
  
  
**Chat\_from\_row.xml**קובץ עזר לעיצוב הודעת טקסט הנשלחת למשתמש  
במסך הצאט לוג אקטיביטי, המתקבלת מהמשתמש.

**Chat\_to\_row.xml**קובץ עזר לעיצוב הודעת טקסט הנשלחת למשתמש  
במסך הצאט לוג אקטיביטי.

**Latest\_message\_row.xml  
-תפריט הצאטים .**

**User\_row\_new\_message.xml**תפריט בחירת המשתמש לשליחת ההודעה חדשה.