

אפליקציית רשת חברתית - SocialMid

שם התלמיד/ה: רועי חומסקי

ת.ז התלמיד/ה: 213639438

שם המנחה: שרה בן יוסף

שם החלופה: עבודת גמר תכנון ותכנות מערכות - טלפונים חכמים באנדרואיד

שם בית ספר: מקיף ו' "יגאל אלון"

תאריך הגשה: מאי 2021

**תוכן עניינים**

**הקדמה.................................................................................................................3**

**מבנה / ארכיטקטורה של הפרויקט........................................................................4-14**

מטרות ויעדי המערכת.............................................................................................4

תרשים מבנה ארגוני............................................................................................... 5

מטרת האפליקציה וקהל יעד....................................................................................6

אילוצים ומגבלות והרשאות.....................................................................................6

תיחום המערכת......................................................................................................8

טבלת מסכים......................................................................................................6-9

תרשים זרימה של מסכים........................................................................................9

טבלת מחלקות................................................................................................10-11

תרשים UML.......................................................................................................12

שימוש באבני יסוד...........................................................................................13-14

**מדריך למשתמשים...........................................................................................15-18**

**בסיס נתונים ואחסון..........................................................................................19-22**

**רפלקציה.............................................................................................................23**

**נספחים........................................................................................................24-135**

**הקדמה**

הספר מכיל פרטים בעבור אפליקציה שעשיתי במהלך השנה העוסקת באפליקציית רשת חברתית, מסכים המופיעים באפליקציה עצמה, בסיס הנתונים שהשתמשתי בו וכו'.

הרקע לפרויקט הינו, לאפשר לאנשים להעלות תמונות ולשתף אותם עם חברים והעולם.

האתגר שנתקלתי בו הוא עבודה עם FIREBASE מכיוון שהapi של הבסיס הנתונים החינמי טיפה מסורבל וזאת גם הפעם הראשונה שאני עובד איתו.

הסיבה שבגללה בחרתי את הנושא מכיוון שכיום רוב האנשים בעולם מחזיקים בטלפון נייד ורוצים לשתף את חייהם עם קרובים ועם שאר העולם, SocialMid היא פלטפורמת תקשורת חברתית בה אפשר לשתף תמונות וצ'אט פרטי.

המטרה העיקרית של הפרויקט הזה הינה לאפשר לאנשים לשתף את מחשבותיהם וחייהם לכלל העולם בלחיצת כפתור ולתת פלטפורמה לתקשורת בין כלל החברה.

**מבנה / ארכיטקטורה של הפרויקט**

**מטרות ויעדי המערכת:**

|  |  |  |  |
| --- | --- | --- | --- |
| **שם היעד** | **מטרה** | **קריטריון** | **עדיפות** |
| קליטת פרטי המשתמש והצגתם | אפשרות לייצוג דינמי של נתונים תוך קליטה מהירה ופשוטה | תפעולי | גבוהה |
| הוספה ומחיקה של פוסטים | אפשרות לבחירה ועיצוב חופשי של פוסטים | תפעולי | גבוהה |
| ארגון הפוסטים בפיד ובפרופיל האישי | צורך סדר וארגון, סדר כרונולוגי | ארגוני | גבוהה |
| אפשרות של עדכון ועריכה של פוסטים ופרטים אישיים | עדכון יעיל ודינמי על לתקן טעויות או עדכון של מידע לא רלוונטי או עדכני | ניהולי | גבוהה |
| שמירת המידע במקום מאובטח מאורגן וזמין בכל עת | אחסון וארגון מסודר של הנתונים במערכת כך שתתאפשר גישה נוחה לנתונים כלשהם במערכת וניתן יהיה בקלות לאתר בה מידע מסוים. | תפעולי | גבוהה |
| התכתבות בין משתמשים | שמירה על קשר חברתי בין משתמשים לצורך קיום קשר בין אישי | תפעולי | בינונית |

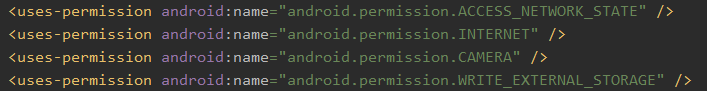
**מטרת האפליקציה:**

לאפשר לאנשים בעלי היכרות מוקדמת או אשר הכירו דרך האפליקציה לתקשר ולשתף לכלל החברה את תחומי ענייניו, פעולותיו או דעותיו בצורה חופשית.

**קהל יעד:**

האפליקציה אינה מיועדת לקהל ספציפי למרות שתחום זה יותר פעיל בקרב צעירים אך האפליקציה מונגשת לכולם.

**הרשאות:**

לכלל המשתמשים יש גישה לכל התוכן והם רשאים לתקשר עם כל אדם אשר נמצא באפליקציה ולהתכתב עם אנשים אשר מוגדרים כ"חברים" של המשתמש, לכל פוסט שמועלה רק לכותב הפוסט יש אפשרות לערוך אותו מלבד זאת כל משתמש רשאי להגיב לראות כל פוסט.

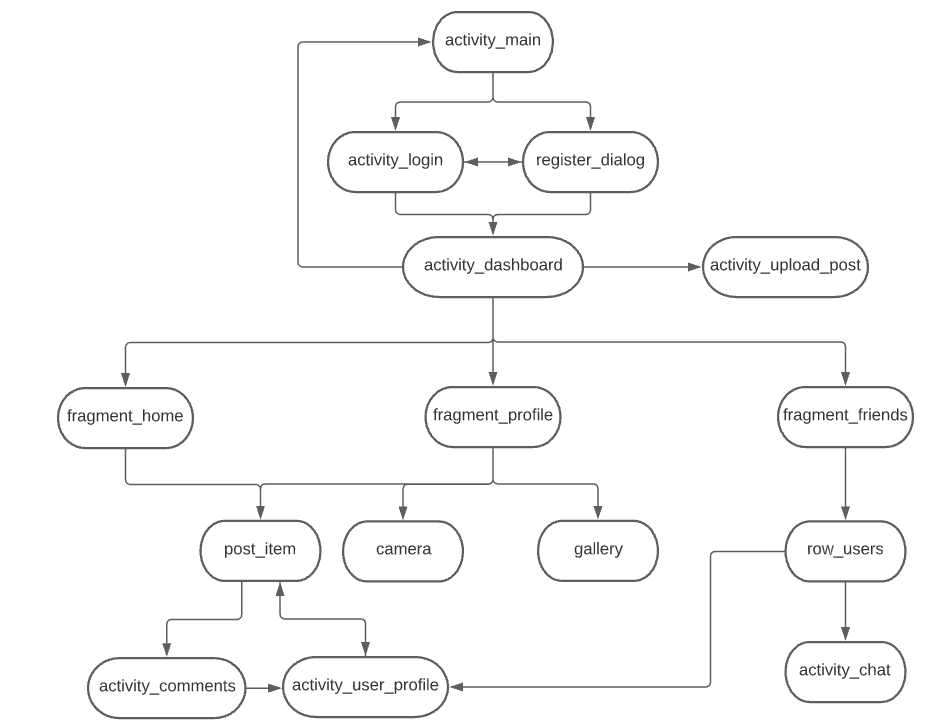
האפליקציה מבקשת השראה מן המשתמש בעת העלאת תמונה מהגלריה או מהמצלמה על מנת לגשת למצלמה או לאחסון התמונות בטלפון, בקשה זו קוראת בפעם הראשונה שבא מתבקשים להעלות תמונה בפוסט, כמו כן ישנה גם הרשאה לשימוש במצב אינטרנט ולשימוש באינטרנט והרשאה לכתיבה לזיכרון חיצוני.

**אילוצים ומגבלות:**

|  |  |
| --- | --- |
| **שם האילוץ** | **תיאור האילוץ** |
| חיבור אינטרנט | האפליקציה דורשת חיבור אינטרנט על מנת לתקשר עם השרתים |
| חשבון דואר אלקטרוני | האפליקציה דורשת דואר אלקטרוני כדי ליצור משתמש בצורה מאובטחת המאפשרת שיחזור סיסמא בעת הצורך |
| מצלמה | על מנת להעלות פוסטים ישנו צורך בתמונה את התמונה אפשר לצלם או להעלות מהגלריה על כן ישנה דרישה למצלמה |
| מיועדת למכשירי אנדרואיד בלבד | האפליקציה מיועדת למכשירי אנדרואיד בלבד (בעלי API level 23 – Android 6.0  ומעלה) כמו כן, המידע מאוחסן בשרת התומך מגוון רחב של מערכות הפעלה. |

**טבלת מסכים:**

|  |  |  |
| --- | --- | --- |
| **שם המסך** | **תכולה** | **תפקיד** |
| activity\_main | מכיל תמונת לוגו של האפליקציה ושני כפתורים להרשמה/כניסה. | כאשר אין חשבון מחובר במכשיר מסך זה יפתח בהתחלה ותפקידו להעביר את המשתמש למסך ההתחברות או למסך ההרשמה. |
| register\_dialog | מסך הרשמה מוצג בdialog, מכיל תיבות טקסט להכנסת פרטים: דוא"ל, סיסמא וnickname- כפתור שליחה והרשמה וכפתור להתחברות משתמש במידה וכבר יש. | הרשמת משתמשים חדשים לאפליקציה. |
| activity\_login | מסך התחברות, מכיל תיבות טקסט להכנסת פרטים: דוא"ל וסיסמא של המשתמש  כפתור לכניסה, כפתור "שכחתי סיסמא" לשחזור סיסמא וכפתור להרשמת משתמש במידה ואין. | כניסת משתמשים לאפליקציה. |
| activity\_dashboard | מסך המכיל את הfragments וסרגל ניווט בין הfragments. | מחזיק את הfragments זהו המסך הראשי אליו נכנסים כאשר ישנו משתמש מחובר, דרך מסך זה ניתן גם לנווט בין הfragments באמצעות סרגל הניווט. |
| activity\_upload\_post | מסך המכיל תמונת preview לפני העלאה, שני כפתורים לבחירה האם לבחור תמונה מהמצלמה או מהגלריה, תיבת טקסט להכנסת תיאור לפוסט וכפתור העלאה. | מסך בו ניתן להעלות תמונה לאפליקציה, ניתן לבחור האם להעלות את התמונה מהמצלמה או מהגלריה ולהוסיף תיאור לפוסט, כמו כן לפני העלאה ניתן לראות כיצד התמונה תראה. |
| fragment\_home | במסך זה ישנה רשימת גלילה מחזורית של פוסטים (recycleView). | במסך זה ניתן לגלול לאורך כל הפוסטים שהעלו לאפליקציה. |
| fragment\_profile | מסך המכיל תמונה פרופיל ותמונת רקע וגם שם משתמש ואימייל של המשתמש, במסך זה ישנו גם כפתור "מעופף" בוא ניתן ללחוץ על מנת לשנות תמונת רקע/פרופיל ולשנות את שם המשתמש, כמו כן במסך זה ניתן לראות ברשימת גלילה מחזורית את כל הפוסטים שהמשתמש העלה. במידה והמשתמש לא העלה פוסטים ניתן ללחוץ על הכפתור שיוצג על מנת להעלות פוסט חדש. | מסך זה הוא מסך הפרופיל של המשתמש בוא ניתן לראות את כל המידע של אותו המשתמש, מסך זה מציג רק את המידע של המשתמש המחובר. דרך מסך זה ניתן לעדכן את תמונת הפרופיל והרקע כמו גם לשנות את ה-nickname.  מסך זה משמש לתצוגה של כל הפוסטים שאותו משתמש העלה. או להעלות פוסט ראשון במידה ואין. |
| fragment\_friends | במסך זה ישנה רשימת גלילה מחזורית שבה ניתן לראות את כל החברים של אותו משתמש (recycleView) ובעת לחיצה על "חבר" ניתן להיכנס לצ'אט איתו. | מסך זה נועד להציג את רשימת החברים של המשתמש המחובר ודרכו ניתן גם לגשת לצ'אט עם אחד "החברים". |
| post\_item | מסך זה מכיל את כל המידע של פוסט אחד, שם המשתמש שהעלה את הפוסט וניתן ללחוץ על השם משתמש כדי להגיע לפרופיל שלו ביחד עם התמונת פרופיל של אותו משתמש, תמונת הפוסט, שני לחצנים של לייק או לכתוב תגובה, תיאור הפוסט, מספר הלייקים והתגובות של הפוסט, תאריך העלאת הפוסט ולחצן שמוביל למסך בוא יש את כל התגובות, במידה והינך יוצר הפוסט ניתן גם לערוך אותו ולמחוק אותו. | מסך זה מציג את הפוסט שהועלה, לכל פוסט יש אותה תבנית שמתאימה למסך זה, מסך זה מוביל למספר פונקציות שניתן לעשות למשל, כניסה לפרופיל של משתמש אחר, כניסה לתגובות וכתיבת תגובה, סימון לייק ועריכה של הפוסט. |
| activity\_comments | מסך זה מכיל את תיאור הפוסט עם שם ותמונת הפרופיל של המשתמש שהעלה את הפוסט כמו גם רשימת גלילה מחזורית של תגובות כאשר בכל תגובה יש את שם ותמונת הפרופיל של המשתמש שכתב אותה ביחד עם התגובה עצמה ובמידה ואתה כותב התגובה ניתן למחוק אותה באמצעות כפתור. | מסך זה מציג את כל התגובות של הפוסט ביחד עם תיאור הפוסט ומידע על כותבי התגובות. |
| activity\_user\_profile | מסך הדומה ל- fragment\_profile, מסך המכיל תמונה פרופיל ותמונת רקע וגם שם משתמש ואימייל של המשתמש, כפתור הוספה כחבר, כמו כן במסך זה ניתן לראות ברשימת גלילה מחזורית את כל הפוסטים שהמשתמש העלה. | מסך זה מציג את הפרופיל של משתמש מסוים כמו fragment\_profile, ניתן להוסיפו כחבר או לבטל את החברות, במסך זה יש את כל המידע עליו והפוסטים שהעלה. |
| camera | מסך המצלמה בו ניתן לצלם תמונה – נוצר על ידי המערכת הפעלה. | מסך בו ניתן לצלם תמונה לפוסט. |
| gallery | מסך הגלריה בו ניתן לבחור תמונה – נוצר על ידי המערכת הפעלה. | מסך בו ניתן להעלות תמונה מהגלריה לפוסט. |
| row\_users | מסך זה מכיל מעין כרטיס מידע של המשתמש (כאשר הוא חבר) – תמונת פרופיל, שם ואימייל של המשתמש וכפתור הסרה כחבר, כמו כן ניתן ללחוץ ולהגיע לצ'אט עם אותו משתמש או ללחוץ על התמונה ולהגיע לפרופיל של אותו משתמש. | מסך זה מייצג כרטיס מידע של משתמש בעמוד החברים של המשתמש המחובר, דרך מסך זה ניתן להיכנס לצ'אט או לפרופיל של המשתמש, מסך זה מוכל בfragment\_profile כאשר הוא חלק מהרשימה של החברים. |
| activity\_chat | מסך זה מכיל את שם המשתמש ותמונת הפרופיל שלו, תיבת טקסט לכתיבת הודעה וכפתור שליחה, ישנה גם רשימה של ההודעות כאשר לכל הודעה יש את הטקסט שנשלח עם התאריך והצבע בהתאם למי ששולח, ישנה גם אופציה לראות האם נשלח ההודעה והאם היא נקראה. | במסך זה ניתן לראות את הצ'אט של המשתמש עם אותו חבר ולשלוח הודעה. |

**תרשים זרימה של מסכים:**

**נתונים**

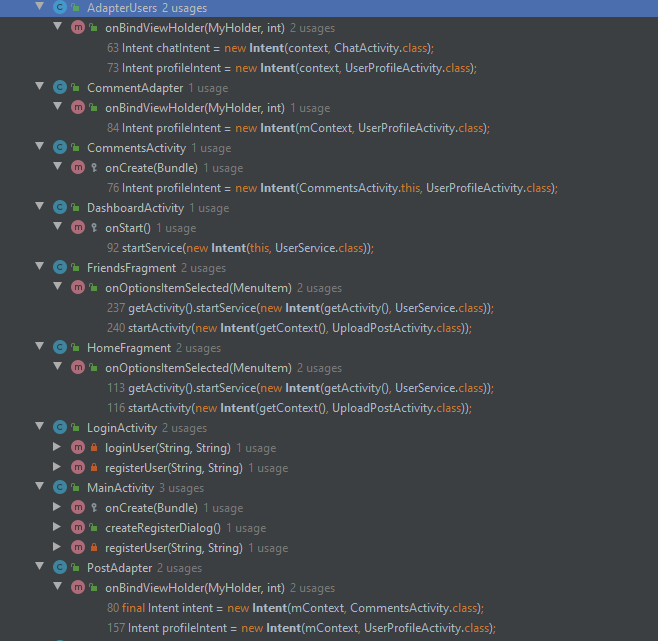
**טבלת מחלקות:**

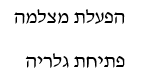
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **שם המחלקה** | **תפקיד** | **סוג** | **תכונות** | **פעולות** | **שם בתוכנה** |
| Message | מחלקה המתארת את העצם – הודעה כאשר שולחים הודעת צ'אט זוהי התבנית. | public class | שולח, מקבל, חותמת זמן, תוכן ההודעה, ראה. | getter, setter, constructor | public class Message |
| User | מחלקה המתארת את העצם – משתמש כאשר נוצר משתמש או שמתחבר אחד או כאשר יש שימוש ברשימה של משתמשים זוהי התבנית שמשתמשים. | public class | מזהה משתמש, שם, אימייל, תמונת פרופיל, תמונת רקע. | getter, setter, constructor | public class User |
| Like | מחלקה המתארת את העצם – לייק כאשר מקבלים את רשימת הלייקים של פוסט זוהי התבנית. | public class | כמות לייקים, רשימה מי עשה לייק, מזהה פוסט. | getter, setter, constructor | public class Like |
| Post | מחלקה המתארת את העצם – פוסט כאשר מקבלים רשימה של פוסטים זוהי התבנית. | public class | מזהה פוסט, תמונת הפוסט, תיאור הפוסט, מזהה של יוצר הפוסט, חותמת זמן. | getter, setter, constructor | public class Post |
| Comment | מחלקה המתארת את העצם – תגובה כאשר מקבלים רשימה של תגובה זוהי התבנית. | public class | מזהה כותב התגובה, תוכן, מזהה התגובה. | getter, setter, constructor | public class Comment |
| DoubleClickListener | מחלקה המתארת עצם של לחיצה כפולה. | public abstract class | הפרש הזמן בין הלחיצות. | constructor, onClick, onDoubleClick | public abstract class DoubleClickListener implements View.OnClickListener |

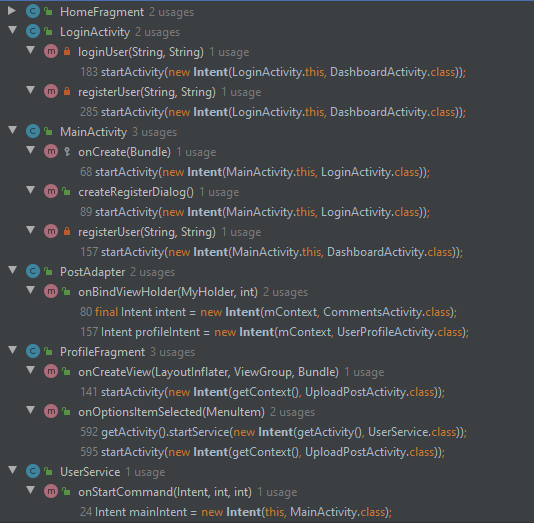
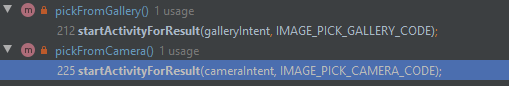
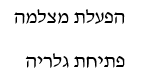
**תרשים UML**

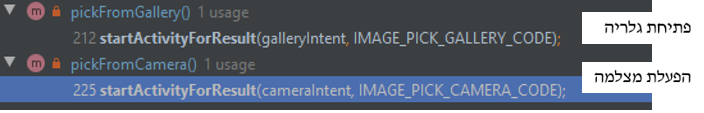


**שימוש באבני יסוד**

**א. :Intent**





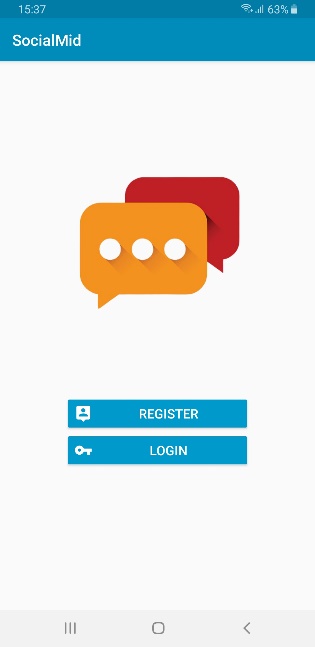
**ב. Activity :**

**ג. Service :**

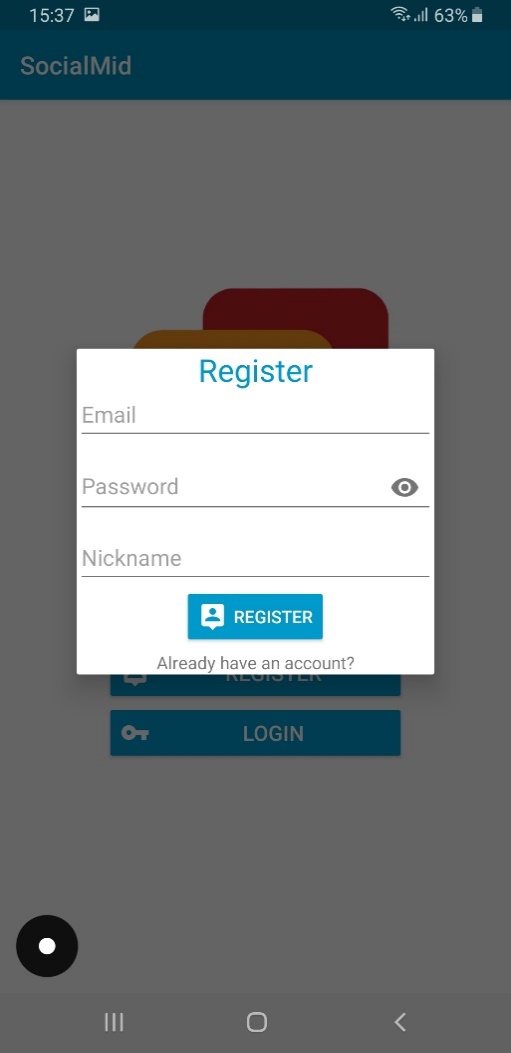
שירות שבודק ברקע האם ישנו משתמש מחובר כבר במכשיר במידה ואין המשתמש יועבר למסך MainActivity כדי להירשם או להתחבר לחשבון קיים, השירות פועל בעת הפעלת מסך DashboardActivity, FriendsFragment, HomeFragment, ProfileFragment.

**ד.** **BrodcastReciever :**

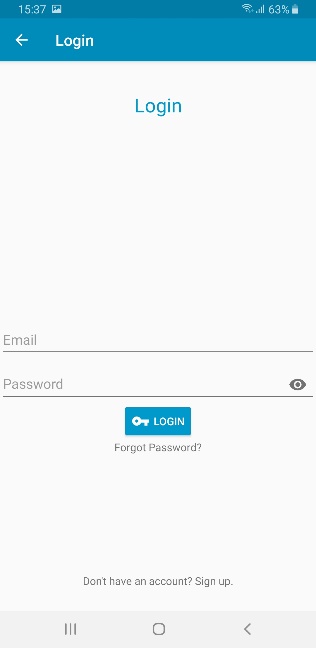
כאשר הסוללה נמוכה המשתמש מקבל הודעת toast שהסוללה נמוכה, ישנו שימוש ב BrodcastReciever מובנה.

**מדריך למשתמש**

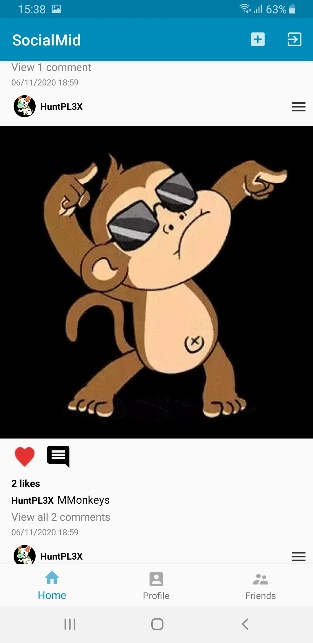
MainActivity

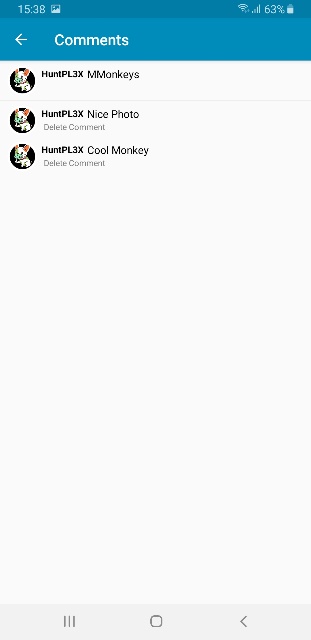
המסך הראשי שנפתח בעת התקנת האפליקציה ואו כאשר אין משתמש מחובר במכשיר, דרך מסך זה ניתן להירשם באמצעות חלונית ההרשמה שנפתחת או באמצעות מעבר למסך ההתחברות על ידי לחיצה על אחד מהכפתורים.

חלונית ההרשמה דרכה ניתן ליצור משתמש לאפליקציה, על מנת להירשם יש להכניס את הפרטים הרצויים (אימייל, סיסמא, שם משתמש/כינוי) בהתאם לשדות המתאימים, במידה ויש כבר משתמש ניתן ללחוץ על השורה מתחת ללחצן הרשמה שתוביל את המשתמש למסך ההתחברות, כמו כן ניתן גם ללחוץ בשדה הסיסמא על לחצן העיו המאפשר לראות את הסיסמא המוסתרת, לאחר שהמשתמש מילא את הפרטים בצורה נכונה עליו ללחוץ על לחצן ההתחברות במידה ויש תקלה באחד השדות תופיע הודעה בהתאם.

LoginActivity

מסך ההתחברות דרכו ניתן להתחבר לאפליקציה דרך משתמש רשום על ידי הכנסת אימייל וסיסמא של משתמש ולחיצה על כפתור ההתחברות, במידה ואין לבעל המכשיר משתמש הוא יכול ללחוץ על השורה התחתונה ביותר במסך ולפתוח את חלונית ההרשמה דרכה ניתן להירשם, במידה ומשתמש שכח את סיסמתו ניתן לשחזר אותה באמצעות לחיצה על השורה מתחת ללחצן ההתחברות ותפתח חלונית בה המשתמש יכניס את האימייל המתאים ויקבל בתיבת הדואר הנכנס שלו קישור לשחזור סיסמא, כמו כן כמו בחלונית ההתחברות ניתן ללחוץ על לחצן העין ולראות את הסיסמא המוסתרת.

HomeFragment

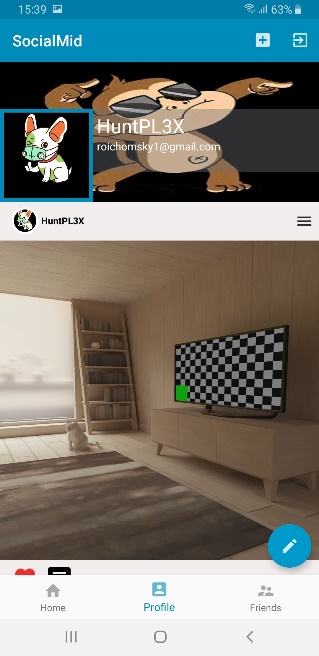
המסך הראשי – מסך "הפיד" דרכו ניתן לראות את כל הפוסטים שהועלו לאפליקציה כאשר בכל פוסט ניתן ללחוצץ לחיצה כפולה על התמונה כדי לעשות לייק או על ידי כפתור הלב, ניתן לכתוב תגובה על ידי לחיצה על הכפתור הסמוך לכפתור הלייק אשר יפתח חלונית לכתוב בה את התגובה, כמו כן ניתן לראות את כל התגובות של הפוסט, לראות את התאריך והשעה בה הועלה הפוסט ומספר הלייקים שלו, בכל פוסט ישנו תיאור הפוסט תמונת הפוסט ושם ותמונת יוצר הפוסט, ניתן לערוך את הפוסט באמצעות לחיצה על כפתור ה"המבורגר" (כפתור בעל שלוש פסים אופקיים) שם ניתן לשנות את תיאור הפוסט או למחוק אותו לגמרי, מלבד הפוסטים דרך מסך זה ניתן לעבור למסכים אחרים, דרך תפריט הניווט התחתון ניתן לעבור לפרופיל של המשתמש ולרשימת החברים שלו, ועל ידי התפריט העליון ניתן להתנתק מהמשתמש או להעלות פוסט חדש.

CommentsActivity

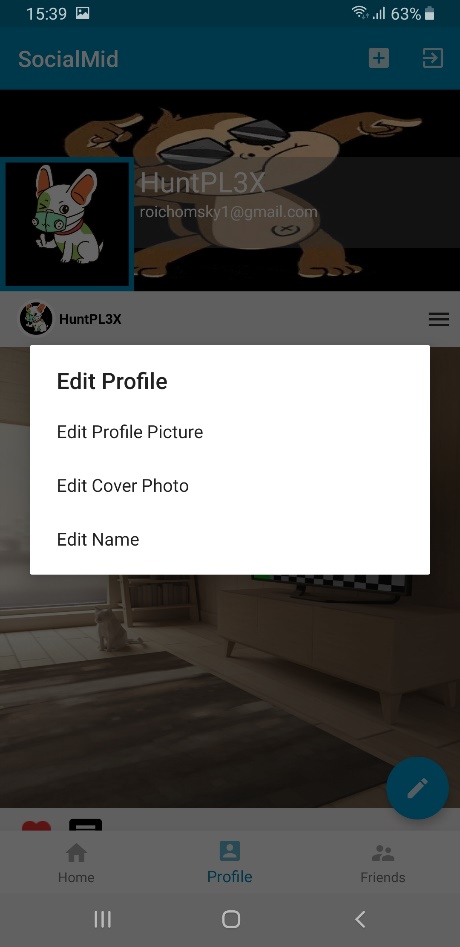
מסך התגובות בו ניתן לראות את כל התגובות של הפוסט ומי כתב אותם ובמידה והמשתמש כתב תגובה הוא יכול למחוק אותה.

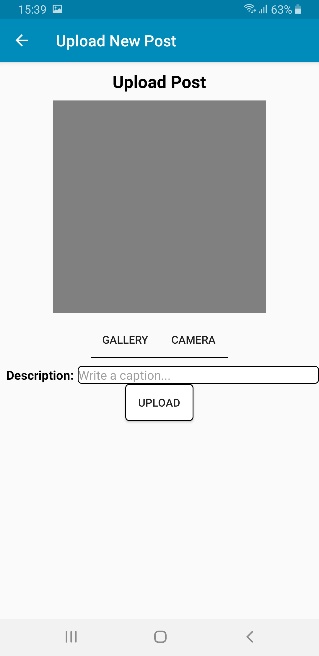
UserProfileActivity

מסך הפרופיל של משתמש בו ניתן לראות את תמונת הרקע והפרופיל של משתמש מסוים, כמו כן גם את השם משתמש שלו והאימייל והאם הוא חבר של המשתמש המחובר, ניתן גם לראות את כל הפוסטים שהעלה המשתמש וניתן גם להוסיף או להוריד אותו מרשימת החברים על ידי לחיצה על השורה השחורה מתחת לאימייל.

ProfileFragment

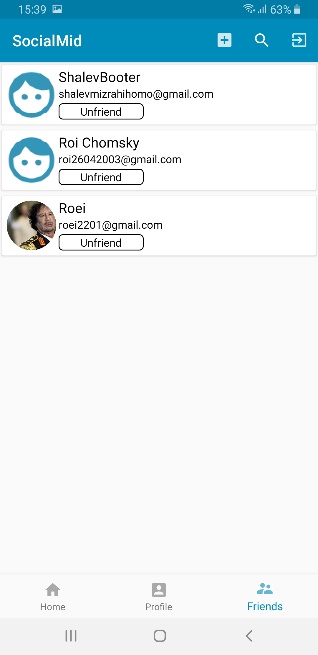
מסך הפרופיל של המשתמש המחובר בו ניתן לראות את תמונת הרקע והפרופיל של המשתמש, כמו כן גם את השם משתמש שלו והאימייל, וגם לראות את כל הפוסטים שהעלה המשתמש. בנוסף, ניתן לערוך את הפרופיל של המשתמש המחובר על ידי לחיצה על לחצן העט הכחול שבצידו הימני התחתון של המסך בו תפתח חלונית לעריכת הפרופיל.

בחלונית העריכה ניתן לערוך את תמונת הרקע ואת תמונת הפרופיל כמו גם את שם המשתמש על ידי לחיצה על הכפתור המתאים.

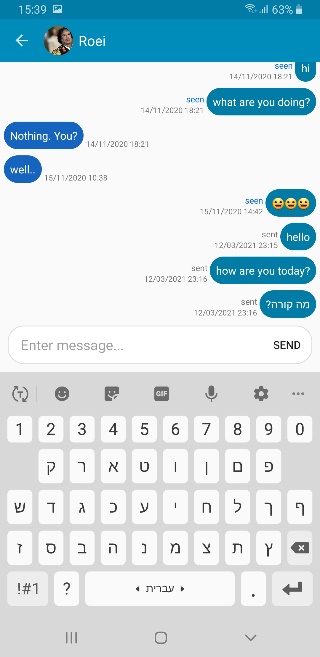


UploadPostActivity

דרך מסך זה ניתן להעלות פוסט חדש למסך זה מגיעים באמצעות כל חלון באפליקציה (דרך הפרופיל, הבית, או מסך החברים), במסך זה ישנה אופציה להעלות תמונה דרך הגלריה או המצלמה, שדה לכתיבת פרופיל, ומסך שבו ניתן לראות כיצד התמונה תראה לפני שמעלים אותה וכפתור העלאה.

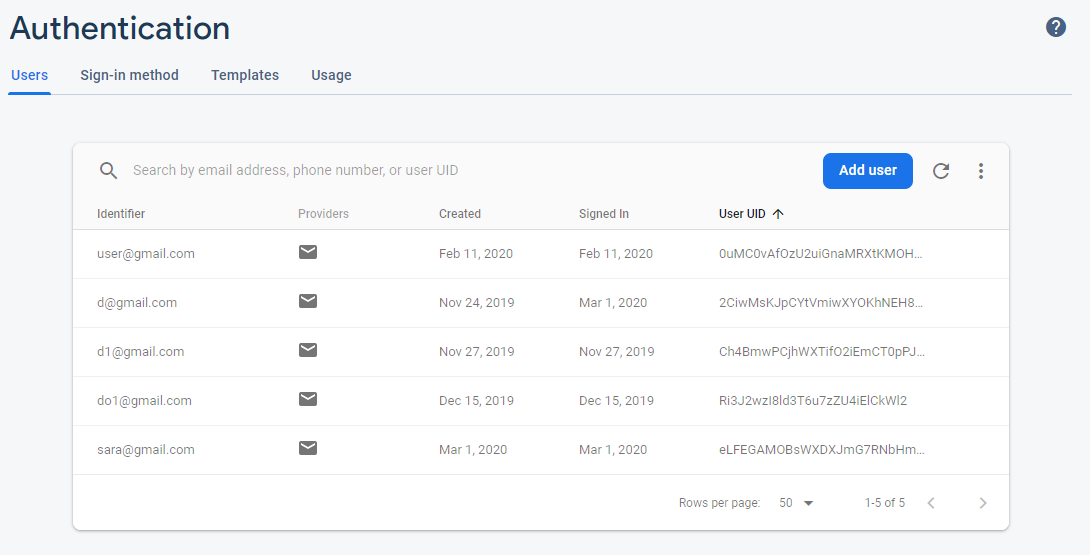
FriendsFramgent

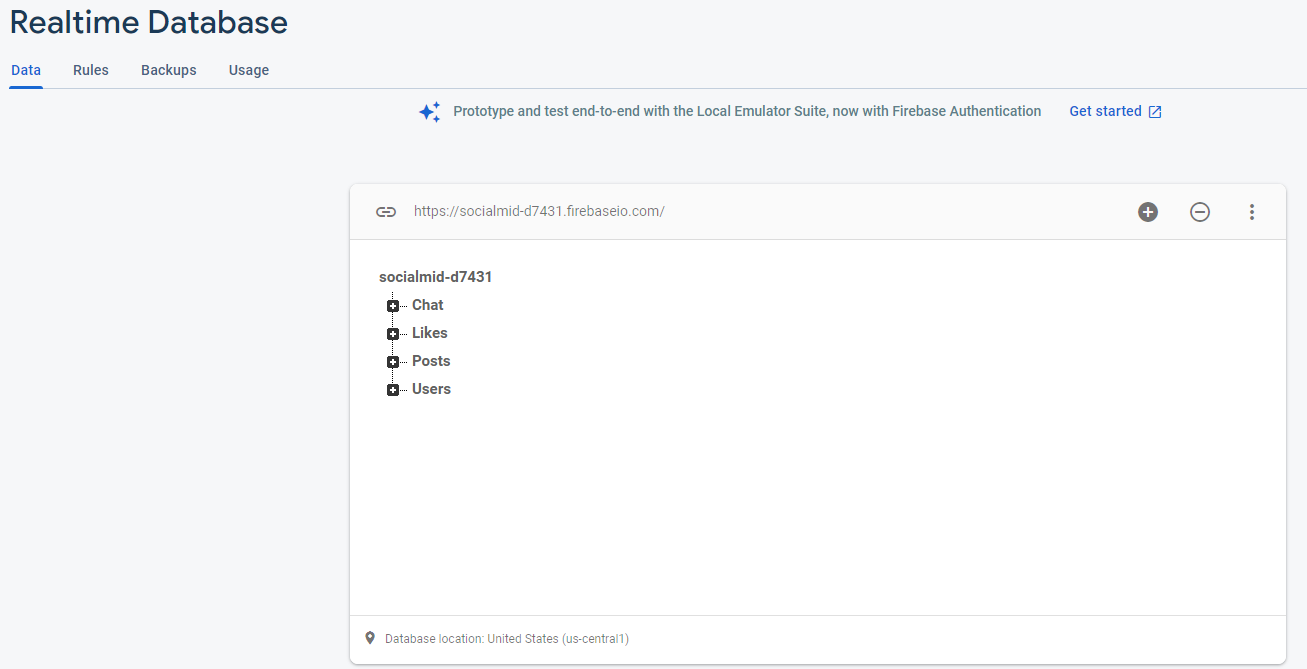
מסך זה הוא מסך רשימת החברים של המשתמש המחובר דרכו ניתן להיכנס לצ'אט עם אותו חבר על ידי לחיצה על המלבן ועל ידי לחיצה על התמונת פרופיל ניתן להיכנס לפרופיל של החבר, כמו כן ניתן גם להוציא את אותו משתמש מרשימת החברים על ידי לחיצה על הלחצן שמתחת לאימייל, באמצעות כפתור זכוכית המגדלת שבתפריט העליון ניתן לחפש משתמש מסוים ברשימת החברים.

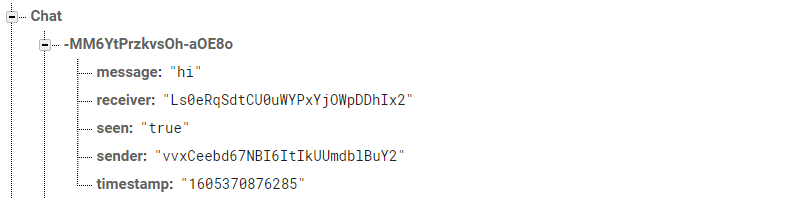
ChatActivity

מסך זה הוא מסך הצ'אט דרכו ניתן להתכתב עם חבר ההודעות הכהות יותר הן של החבר והבהירות יותר הן של המשתמש המחובר במכשיר, בכל הודעה ניתון לראות האם נשלחה או נראתה על ידי החבר, כמו כן ישנו גם את שדה הכתיבה של ההודעה וכפתור שליחה של הודעה, בתפריט העליון ניתן לראות את שם המשתמש של החבר ואת תמונת הפרופיל שלו.

**בסיס נתונים ואחסון**

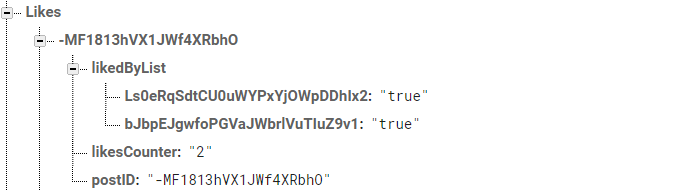
בפרוייקט שלי השתמשתי בבסיס הנתונים החינמי של גוגל .FIREBASE

על מנת להתחבר לאפליקציה השתמשתי ב-Authentication של פיירבייס דרכו ניתן להירשם באמצעות אימייל וסיסמא והוא יוצר מספר מזהה לכל משתמש אוטומטית.

שמירת הנתונים התבצעה באמצעות Realtime Database של פיירבייס, כאשר השמירה מתבצעת במבנה של עץ שדות המורכב ממפתח וערך. במבנה הנתונים שלי השתמשתי בארבעה טבלאות – צ'אט, לייקים, פוסטים, משתמשים, כאשר לכל טבלה יש קשרים וערכים מטבלאות אחרות, כמו כן גם כל טבלה מייצגת עצם בקוד.

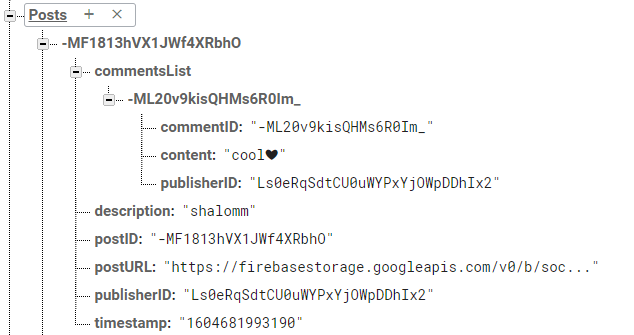
טבלת הצ'אט בנויה כך שלכל תגובה יש תוכן, שולח, מקבל, נראות וחותמת זמן, כל אובייקט מייצג תגובה באפליקציה, כל הטבלה מורכבת ממפתחות זרים והיא למעשה רשימה של תגובות.

|  |  |  |  |
| --- | --- | --- | --- |
| **רשימת עמודות** | **תפקיד** | **טיפוס הנתונים** | **חובה או לא** |
| תוכן | זר | מחרוזת - String | חובה |
| מזהה שולח | זר | מחרוזת - String | חובה |
| מזהה מקבל | זר | מחרוזת - String | חובה |
| נראות | זר | מחרוזת - String | חובה |
| חותמת זמן | זר | מחרוזת - String | חובה |



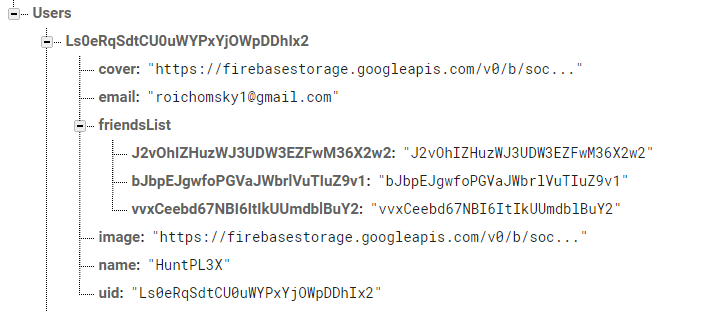
טבלת הלייקים מייצגת את אובייקט הלייק בקוד והיא מורכבת מרשימה של כל מי שעשה לייק, מספר לייקים ומזהה פוסט.

|  |  |  |  |
| --- | --- | --- | --- |
| **רשימת עמודות** | **תפקיד** | **טיפוס הנתונים** | **חובה או לא** |
| רשימת לייקים | זר | רשימה של מזהה משתמש - אמת | חובה |
| מזהה פוסט | מפתח ראשי | מחרוזת - String | חובה |
| מספר לייקים | זר | מחרוזת - String | חובה |



טבלת הפוסטים מייצגת את אובייקט הפוסט וכל פוסט מורכב מרשימת התגובות ולכל תגובה יש מזהה, תוכן ומזהה של כותב התגובה, לפוסט יש גם תיאור, מזהה פוסט, קישור לתמונת הפוסט (מאוחסנת גם בפיירבייס), מזהה של מעלה הפוסט וחותמת זמן.

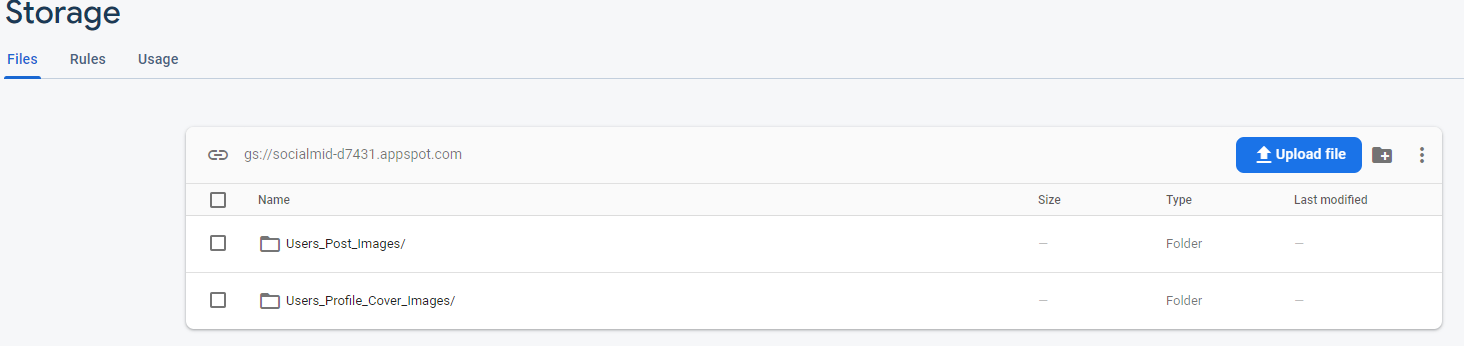
|  |  |  |  |
| --- | --- | --- | --- |
| **רשימת עמודות** | **תפקיד** | **טיפוס הנתונים** | **חובה או לא** |
| רשימת תגובות | זר | רשימה של תגובות (מזהה תגובה, תוכן, מזהה כותב) | חובה |
| מזהה פוסט | מפתח ראשי | מחרוזת - String | חובה |
| תיאור הפוסט | זר | מחרוזת - String | חובה |
| קישור לתמונת הפוסט | זר | מחרוזת - String | חובה |
| מזהה כותב הפוסט | זר | מחרוזת - String | חובה |
| חותמת זמן | זר | מחרוזת - String | חובה |



טבלת המשתמשים מייצגת את המשתמש הפרטי, חלק מהמידע שהיא מאחסנת נלקח מה-Authentication והיא מורכבת מקישורים לאחסון של תמונת פרופיל ורקע, רשימת חברים (מזהה חבר – מזהה חבר), אימייל, שם/כינוי, מזהה משתמש.

|  |  |  |  |
| --- | --- | --- | --- |
| **רשימת עמודות** | **תפקיד** | **טיפוס הנתונים** | **חובה או לא** |
| רשימת חברים | זר | רשימה של מזהה חבר – מזהה חבר | לא חובה |
| מזהה משתמש | מפתח ראשי | מחרוזת - String | חובה |
| קישור לתמונת פרופיל | זר | מחרוזת - String | לא חובה |
| קישור לתמונת הרקע | זר | מחרוזת - String | לא חובה |
| שם משתמש/כינוי | זר | מחרוזת - String | חובה |
| אימייל | זר | מחרוזת - String | חובה |

בפרוייקט ישנו שימוש גם באחסון של פיירבייס, האחסון מאחסן את התמונות שמועלות לאפליקציה כמו למשל תמונת פרופיל או רקע ותמונות לפוסטים.



התמונות של הפוסטים נשמרות לפי המיקום שלהן בטלפון וPATH באחסון הוא דומה רק שהתיקייה ההתחלתית היא של הפוסטים, שיטת השמירה הזאת בא מתאמי נוחות, את תמונות הרקע והפרופיל נשמרות כולן בתיקייה המיועדת באחסון.

**רפלקציה**

הצדדים החזקים של הפרויקט זה החשיבה של המאחורי הקלעים והתכנון שלו, פתרון בעיות שבפרוייקט.

הצד החלש של הפרויקט הינו המראה החיצוני שלו, המראה מושך את העניין מאורגן בצורה הטובה ביותר לצורך ארגון המידה בצורה מסודרת ביותר, ואני לא כלכך חזק בFrontEnd ולא מאוד טוב בעיצוב.

בפרויקט שלי יכולתי להוסיף שאפשר לעלות תמונות כהודעות בצ'אט אך לא היה לי מספיק זמן והרגשתי שהפרוייקט גדול מספיק.

מהפרויקט למדתי להכיר את המערכת עצמה ואת שורות הקוד ולהעשיר את הידע שלי בתחום, יצא לי להתנסות בfrontEnd משהו שלא כלכך יצא לי לעשות. הבאגים בפרוייקט עזרו לי בתחום הPentesting שאני לומד, וגרמו לי לחשוב כיצד אפשר לנצל את הבאגים האלה לטובת הפורצים.

**נספחים**

AndroidManifest.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.roichomsky.socialmid"**>  
 *<!-- Internet Permission -->* <**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"** />  
 <**uses-permission android:name="android.permission.INTERNET"** />  
 <**uses-permission android:name="android.permission.CAMERA"** />  
 <**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"** />  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@drawable/icon"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".ChatActivity"  
 android:theme="@style/AppThemeChat"**/>  
 <**activity android:name=".UserProfileActivity"** />  
 <**activity android:name=".CommentsActivity"** />  
 <**activity android:name=".UploadPostActivity"** />  
 <**activity android:name=".LoginActivity"** />  
 <**activity android:name=".DashboardActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 <**activity android:name=".MainActivity"** />  
  
 <**service android:name=".UserService"** />  
  
 <**receiver android:name=".BatteryBroadcastReceiver"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.BATTERY\_LOW"**></**action**>  
 </**intent-filter**>  
 </**receiver**>  
 </**application**>  
  
</**manifest**>

AdapterUsers.java:

**package** com.roichomsky.socialmid;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.squareup.picasso.Picasso;  
  
**import** java.util.List;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**public class** AdapterUsers **extends** RecyclerView.Adapter<AdapterUsers.MyHolder> {  
  
 **private** Context **context**;  
 **private** List<User> **userList**;  
  
 *//Constructor* **public** AdapterUsers(Context context, List<User> userList) {  
 **this**.**context** = context;  
 **this**.**userList** = userList;  
 }  
  
 @NonNull  
 @Override  
 **public** MyHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  
 *//Inflate layout(row\_users.xml)* View view = LayoutInflater.*from*(**context**).inflate(R.layout.***row\_users***, parent, **false**);  
  
 **return new** MyHolder(view);  
 }  
  
 @Override  
 **public void** onBindViewHolder(@NonNull **final** MyHolder holder, **final int** position) {  
 *//Get data* String userImage = **userList**.get(position).getImage();  
 String userName = **userList**.get(position).getName();  
 **final** String userEmail = **userList**.get(position).getEmail();  
  
 *//set data* holder.**mNameTv**.setText(userName);  
 holder.**mEmailTv**.setText(userEmail);  
 **try** {  
 Picasso.*get*().load(userImage)  
 .placeholder(R.drawable.***ic\_default\_img***)  
 .into(holder.**mAvatarIv**);  
 }  
 **catch** (Exception e){  
 }  
  
 holder.**itemView**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent chatIntent = **new** Intent(**context**, ChatActivity.**class**);  
 chatIntent.putExtra(**"uid"**, **userList**.get(position).getUid());  
 **context**.startActivity(chatIntent);  
 }  
 });  
  
 *//handle item click* holder.**mAvatarIv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent profileIntent = **new** Intent(**context**, UserProfileActivity.**class**);  
 profileIntent.putExtra(**"uid"**, **userList**.get(position).getUid());  
 **context**.startActivity(profileIntent);  
 }  
 });  
  
 holder.**unFriendTv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**).child(FirebaseAuth.*getInstance*().getCurrentUser().getUid());  
 reference.child(**"friendsList"**).child(**userList**.get(position).getUid()).removeValue();  
 Toast.*makeText*(**context**, **"Unfriended..."**, Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return userList**.size();  
 }  
  
 *//view holder class* **class** MyHolder **extends** RecyclerView.ViewHolder{  
  
 ImageView **mAvatarIv**;  
 TextView **mNameTv**, **mEmailTv**, **unFriendTv**;  
  
 **public** MyHolder(@NonNull View itemView){  
 **super**(itemView);  
  
 *//init views* **unFriendTv** = itemView.findViewById(R.id.***unfriendTv***);  
 **mAvatarIv** = itemView.findViewById(R.id.***avatarIv***);  
 **mNameTv** = itemView.findViewById(R.id.***nameTv***);  
 **mEmailTv** = itemView.findViewById(R.id.***emailTv***);  
 }  
  
  
 }  
}

BatteryBroadcastReceiver.java:

**package** com.roichomsky.socialmid;  
  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.widget.ProgressBar;  
**import** android.widget.Toast;  
  
**import** java.sql.Time;  
**import** java.text.SimpleDateFormat;  
**import** java.util.Date;  
**import** java.util.Locale;  
  
**public class** BatteryBroadcastReceiver **extends** BroadcastReceiver {  
  
 @Override  
 **public void** onReceive(**final** Context context, **final** Intent intent) {  
 Toast.*makeText*(context,**"Battery level is low"**,Toast.***LENGTH\_SHORT***).show();  
 }  
}

ChatActivity.java:

**package** com.roichomsky.socialmid;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.ActionBar;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.appcompat.widget.Toolbar;  
**import** androidx.recyclerview.widget.LinearLayoutManager;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**import** android.os.Bundle;  
**import** android.text.TextUtils;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.tasks.OnFailureListener;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
**import** com.like.Utils;  
**import** com.squareup.picasso.Picasso;  
  
**import** java.sql.Timestamp;  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
**import** java.util.List;  
  
**public class** ChatActivity **extends** AppCompatActivity {  
  
 **private** RecyclerView **recyclerView**;  
 **private** ChatAdapter **mChatAdapter**;  
 **private** List<Message> **messageList**;  
  
 **public** String **friendUID**, **myUID**;  
  
 ActionBar **actionBar**;  
 Toolbar **toolbar**;  
  
 TextView **nameTv**;  
 ImageView **avatarIv**;  
 EditText **messageEt**;  
 Button **sendBtn**;  
  
 DatabaseReference **refForSeen**;  
 ValueEventListener **seenListener**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_chat***);  
  
 **toolbar** = findViewById(R.id.***chat\_toolbar***);  
 setSupportActionBar(**toolbar**);  
 **toolbar**.setNavigationIcon(getResources().getDrawable(R.drawable.***ic\_action\_back***));  
 **toolbar**.setNavigationOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 finish();  
 }  
 });  
  
 **avatarIv** = findViewById(R.id.***avatarIv***);  
 **nameTv** = findViewById(R.id.***toolbar\_name***);  
 **messageEt** = findViewById(R.id.***chatboxEt***);  
 **sendBtn** = findViewById(R.id.***chatbox\_sendBtn***);  
  
 **friendUID** = getIntent().getStringExtra(**"uid"**);  
 **myUID** = FirebaseAuth.*getInstance*().getCurrentUser().getUid();  
  
 getUserData(**friendUID**,**nameTv**,**avatarIv**);  
  
  
 **recyclerView** = findViewById(R.id.***recyclerview\_message\_list***);  
 **recyclerView**.setHasFixedSize(**true**);  
 **recyclerView**.setLayoutManager(**new** LinearLayoutManager(**this**));  
  
 **messageList** = **new** ArrayList<>();  
  
 readMessages();  
  
 seenMessages();  
  
 **sendBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!TextUtils.*isEmpty*(**messageEt**.getText().toString())) {  
 sendMessage(**messageEt**.getText().toString());  
 }  
 **else**{  
 Toast.*makeText*(getApplicationContext(), **"Cannot send empty message..."**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 });  
  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == android.R.id.***home***){  
 finish();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **void** getUserData(String uid, TextView nameTv, ImageView avatarIv){  
 *//get path of database named "Users" containing users info* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**).child(uid);  
 *//get all data from path* reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 User user = dataSnapshot.getValue(User.**class**);  
 **if** (user.getImage() != **null**){  
 **try**{  
 Picasso.*get*().load(user.getImage()).into(avatarIv);  
 }  
 **catch** (Exception e){ }  
 }  
 nameTv.setText(user.getName());  
  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
  
 **void** sendMessage(String message){  
 Timestamp timestamp = **new** Timestamp(System.*currentTimeMillis*());  
  
 HashMap<String, Object> results = **new** HashMap<>();  
 results.put(**"sender"**, FirebaseAuth.*getInstance*().getCurrentUser().getUid());  
 results.put(**"receiver"**, **friendUID**);  
 results.put(**"timestamp"**, Long.*toString*(timestamp.getTime()));  
 results.put(**"message"**, message);  
 results.put(**"seen"**, **"false"**);  
  
 DatabaseReference ref = FirebaseDatabase.*getInstance*().getReference(**"Chat"**);  
 ref.push().setValue(results);  
 **messageEt**.setText(**""**);  
 }  
  
 **void** readMessages() {  
 DatabaseReference ref = FirebaseDatabase.*getInstance*().getReference(**"Chat"**);  
 ref.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **messageList**.clear();  
 **for** (DataSnapshot ds : dataSnapshot.getChildren()){  
 Message message = ds.getValue(Message.**class**);  
 **if** ((message.getReceiver().equals(**friendUID**) && message.getSender().equals(**myUID**)) ||  
 (message.getReceiver().equals(**myUID**) && message.getSender().equals(**friendUID**))){  
 **messageList**.add(message);  
 }  
  
 **mChatAdapter** = **new** ChatAdapter(getApplicationContext(), **messageList**);  
 **mChatAdapter**.notifyDataSetChanged();  
  
 **recyclerView**.setAdapter(**mChatAdapter**);  
 **recyclerView**.scrollToPosition(**messageList**.size()-1);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 **void** seenMessages(){  
 **refForSeen** = FirebaseDatabase.*getInstance*().getReference(**"Chat"**);  
 HashMap<String, Object> hasSeen = **new** HashMap<>();  
 hasSeen.put(**"seen"**, **"true"**);  
 **seenListener** = **refForSeen**.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **for** (DataSnapshot ds : dataSnapshot.getChildren()){  
 Message message = ds.getValue(Message.**class**);  
 **if** (message.getReceiver().equals(**myUID**) && message.getSender().equals(**friendUID**)){  
 ds.getRef().updateChildren(hasSeen);  
 }  
  
 **mChatAdapter** = **new** ChatAdapter(getApplicationContext(), **messageList**);  
 **mChatAdapter**.notifyDataSetChanged();  
  
 **recyclerView**.setAdapter(**mChatAdapter**);  
 **recyclerView**.scrollToPosition(**messageList**.size()-1);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 @Override  
 **protected void** onPause() {  
 **super**.onPause();  
 **refForSeen**.removeEventListener(**seenListener**);  
 }  
}

ChatAdapter.java:

**package** com.roichomsky.socialmid;  
  
**import** android.content.Context;  
**import** android.graphics.Color;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.TextView;  
  
**import** com.google.firebase.auth.FirebaseAuth;  
  
**import** java.sql.Timestamp;  
**import** java.text.SimpleDateFormat;  
**import** java.util.Calendar;  
**import** java.util.Date;  
**import** java.util.List;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**public class** ChatAdapter **extends** RecyclerView.Adapter{  
  
 **private static final int *VIEW\_TYPE\_MESSAGE\_SENT*** = 1;  
 **private static final int *VIEW\_TYPE\_MESSAGE\_RECEIVED*** = 2;  
  
 **public** Context **mContext**;  
 **public** List<Message> **mMessage**;  
  
 **public** ChatAdapter(Context context , List<Message> message){  
 **this**.**mContext** = context;  
 **this**.**mMessage** = message;  
 }  
  
 @Override  
 **public int** getItemViewType(**int** position) {  
 Message message = **mMessage**.get(position);  
 **if** (message.getSender().equals(FirebaseAuth.*getInstance*().getCurrentUser().getUid())) {  
 *// If the current user is the sender of the message* **return *VIEW\_TYPE\_MESSAGE\_SENT***;  
 } **else** {  
 *// If some other user sent the message* **return *VIEW\_TYPE\_MESSAGE\_RECEIVED***;  
 }  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return mMessage**.size();  
 }  
  
 @NonNull  
 @Override  
 **public** RecyclerView.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  
 View view;  
  
 **if** (viewType == ***VIEW\_TYPE\_MESSAGE\_SENT***) {  
 view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.***item\_message\_sent***, parent, **false**);  
 **return new** SentHolder(view);  
 } **else if** (viewType == ***VIEW\_TYPE\_MESSAGE\_RECEIVED***) {  
 view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.***item\_message\_received***, parent, **false**);  
 **return new** ReceivedHolder(view);  
 }  
  
 **return null**;  
 }  
  
 @Override  
 **public void** onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, **int** position) {  
 Message message = **mMessage**.get(position);  
  
 **switch** (holder.getItemViewType()) {  
 **case *VIEW\_TYPE\_MESSAGE\_SENT***:  
 ((SentHolder) holder).bind(message);  
 **break**;  
 **case *VIEW\_TYPE\_MESSAGE\_RECEIVED***:  
 ((ReceivedHolder) holder).bind(message);  
 }  
 }  
  
 **public class** ReceivedHolder **extends** RecyclerView.ViewHolder{  
  
 TextView **timestampTv**, **messageTv**;  
  
 **public** ReceivedHolder(@NonNull View itemView) {  
 **super**(itemView);  
 **messageTv** = itemView.findViewById(R.id.***text\_message\_body***);  
 **timestampTv** = itemView.findViewById(R.id.***text\_message\_time***);  
 }  
  
 **void** bind(Message message) {  
 **messageTv**.setText(message.getMessage());  
  
 *// Format the stored timestamp into a readable String using method.* Calendar c = Calendar.*getInstance*();  
 c.setTimeInMillis(Long.*parseLong*(message.getTimestamp()));  
 Date d = c.getTime();  
 SimpleDateFormat sdf = **new** SimpleDateFormat(**"dd/MM/yyyy HH:mm"**);  
 **timestampTv**.setText(sdf.format(d));  
 }  
  
 }  
 **public class** SentHolder **extends** RecyclerView.ViewHolder{  
  
 TextView **timestampTv**, **messageTv**, **isSeenTv**;  
  
 **public** SentHolder(@NonNull View itemView) {  
 **super**(itemView);  
 **messageTv** = itemView.findViewById(R.id.***text\_message\_body***);  
 **timestampTv** = itemView.findViewById(R.id.***text\_message\_time***);  
 **isSeenTv** = itemView.findViewById(R.id.***text\_message\_seen***);  
 }  
  
 **void** bind(Message message) {  
 **messageTv**.setText(message.getMessage());  
  
 System.***out***.println(message.getSeen());  
  
 **if**(message.getSeen().equals(**"true"**)){  
 **isSeenTv**.setText(**"seen"**);  
 **isSeenTv**.setTextColor(Color.*parseColor*(**"#0066ff"**));  
 }  
  
 *// Format the stored timestamp into a readable String using method.* Calendar c = Calendar.*getInstance*();  
 c.setTimeInMillis(Long.*parseLong*(message.getTimestamp()));  
 Date d = c.getTime();  
 SimpleDateFormat sdf = **new** SimpleDateFormat(**"dd/MM/yyyy HH:mm"**);  
 **timestampTv**.setText(sdf.format(d));  
 }  
 }  
}

Comment.java:

**package** com.roichomsky.socialmid;  
  
**public class** Comment {  
 **private** String **publisherID**;  
 **private** String **content**;  
 **private** String **commentID**;  
  
 **public** Comment(){}  
  
 **public** Comment(String publisherID, String content, String commentID) {  
 **this**.**publisherID** = publisherID;  
 **this**.**content** = content;  
 **this**.**commentID** = commentID;  
 }  
  
 **public** String getPublisherID() {  
 **return publisherID**;  
 }  
  
 **public void** setPublisherID(String publisherID) {  
 **this**.**publisherID** = publisherID;  
 }  
  
 **public** String getContent() {  
 **return content**;  
 }  
  
 **public void** setContent(String content) {  
 **this**.**content** = content;  
 }  
  
 **public** String getCommentID() {  
 **return commentID**;  
 }  
  
 **public void** setCommentID(String commentID) {  
 **this**.**commentID** = commentID;  
 }  
}

CommentAdapter.java:

**package** com.roichomsky.socialmid;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.tasks.OnFailureListener;  
**import** com.google.android.gms.tasks.OnSuccessListener;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
**import** com.squareup.picasso.Picasso;  
  
**import** java.util.List;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**public class** CommentAdapter **extends** RecyclerView.Adapter<CommentAdapter.MyHolder> {  
  
 **public** Context **mContext**;  
 **public** List<Comment> **mComment**;  
 **public** String **profileUserUID**;  
 **public** String **postID**;  
  
 **public** CommentAdapter(Context mContext, List<Comment> mComment, String profileUserUID, String postID) {  
 **this**.**mContext** = mContext;  
 **this**.**mComment** = mComment;  
 **this**.**profileUserUID** = profileUserUID;  
 **this**.**postID** = postID;  
 }  
  
 @NonNull  
 @Override  
 **public** CommentAdapter.MyHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  
 View view = LayoutInflater.*from*(**mContext**).inflate(R.layout.***comment\_item***, parent, **false**);  
 **return new** CommentAdapter.MyHolder(view);  
 }  
  
 @Override  
 **public void** onBindViewHolder(@NonNull **final** CommentAdapter.MyHolder holder, **int** position) {  
  
 **final** FirebaseUser fUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 **final** Comment comment = **mComment**.get(position);  
 publisherInfo(holder.**avatarIv**, holder.**usernameTv**, comment.getPublisherID());  
 holder.**commentTv**.setText(comment.getContent());  
  
 **if** (comment.getPublisherID().equals(fUser.getUid())){  
 holder.**dltTv**.setVisibility(View.***VISIBLE***);  
 }  
 holder.**dltTv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 FirebaseDatabase.*getInstance*().getReference(**"Posts"**).child(**postID**).child(**"commentsList"**).child(comment.getCommentID()).removeValue()  
 .addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 Toast.*makeText*(**mContext**, **"Comment deleted..."**, Toast.***LENGTH\_SHORT***).show();  
 }  
 })  
 .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 Toast.*makeText*(**mContext**, **"Comment has not been deleted..."**, Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 }  
 });  
  
 holder.**usernameTv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!fUser.getUid().equals(comment.getPublisherID()) &&  
 !**profileUserUID**.equals(comment.getPublisherID())) {  
 Intent profileIntent = **new** Intent(**mContext**, UserProfileActivity.**class**);  
 profileIntent.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 profileIntent.putExtra(**"uid"**, comment.getPublisherID());  
 **mContext**.startActivity(profileIntent);  
 }  
 }  
 });  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return mComment**.size();  
 }  
  
 **class** MyHolder **extends** RecyclerView.ViewHolder{  
  
 *//Views - layout view holder* ImageView **avatarIv**;  
 TextView **usernameTv**, **commentTv**, **dltTv**;  
  
 **public** MyHolder(@NonNull View itemView) {  
 **super**(itemView);  
  
 *//init* **dltTv** = itemView.findViewById(R.id.***dltTv***);  
 **avatarIv** = itemView.findViewById(R.id.***avatarIv***);  
 **usernameTv** = itemView.findViewById(R.id.***usernameTv***);  
 **commentTv** = itemView.findViewById(R.id.***commentTv***);  
 }  
  
 }  
  
 **private void** publisherInfo(**final** ImageView avatarIv, **final** TextView usernameTv, String uid){  
 *//get path of database named "Users" containing users info* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**).child(uid);  
 *//get all data from path* reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 User user = dataSnapshot.getValue(User.**class**);  
 **if** (user.getImage() != **null**){  
 **try**{  
 Picasso.*get*().load(user.getImage()).into(avatarIv);  
 }  
 **catch** (Exception e){ }  
 }  
 usernameTv.setText(user.getName());  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
}

CommentsActivity.java:

**package** com.roichomsky.socialmid;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.ActionBar;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.recyclerview.widget.LinearLayoutManager;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
**import** com.squareup.picasso.Picasso;  
  
**import** java.util.ArrayList;  
  
**public class** CommentsActivity **extends** AppCompatActivity {  
  
 RecyclerView **recyclerView**;  
 ImageView **avatarIv**;  
 TextView **usernameTv**, **descriptionTv**;  
  
 ActionBar **actionBar**;  
  
 ArrayList<Comment> **commentsList**;  
 CommentAdapter **commentAdapter**;  
  
 DatabaseReference **databaseReference**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_comments***);  
  
 **final** FirebaseUser fUser = FirebaseAuth.*getInstance*().getCurrentUser();  
  
 *//support bar* **actionBar** = getSupportActionBar();  
 **actionBar**.setTitle(**"Comments"**);  
 *//set back button enabled* **actionBar**.setDisplayShowHomeEnabled(**true**);  
 **actionBar**.setDisplayHomeAsUpEnabled(**true**);  
  
 *//init* **avatarIv** = findViewById(R.id.***avatarIv***);  
 **usernameTv** = findViewById(R.id.***usernameTv***);  
 **descriptionTv** = findViewById(R.id.***descriptionTv***);  
  
 **recyclerView** = findViewById(R.id.***comments\_recyclerView***);  
 *//properties* **recyclerView**.setHasFixedSize(**true**);  
 **recyclerView**.setLayoutManager(**new** LinearLayoutManager(getApplicationContext()));  
  
 **commentsList** = **new** ArrayList<>();  
  
 String postID = getIntent().getStringExtra(**"postID"**);  
 **final** String publisherID = getIntent().getStringExtra(**"publisherID"**);  
 String description = getIntent().getStringExtra(**"description"**);  
  
 publisherInfo(**avatarIv**, **usernameTv**, publisherID);  
 **usernameTv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!fUser.getUid().equals(publisherID) &&  
 !getIntent().getStringExtra(**"baseClass"**).equals(**"UserProfileActivity"**)) {  
 Intent profileIntent = **new** Intent(CommentsActivity.**this**, UserProfileActivity.**class**);  
 profileIntent.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 profileIntent.putExtra(**"uid"**, publisherID);  
 startActivity(profileIntent);  
 }  
 }  
 });  
 **descriptionTv**.setText(description);  
 getAllComments(postID, publisherID);  
 }  
  
 **private void** publisherInfo(**final** ImageView avatarIv, **final** TextView usernameTv, String uid){  
 *//get path of database named "Users" containing users info* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**).child(uid);  
 *//get all data from path* reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 User user = dataSnapshot.getValue(User.**class**);  
 **if** (user.getImage() != **null**){  
 **try**{  
 Picasso.*get*().load(user.getImage()).into(avatarIv);  
 }  
 **catch** (Exception e){ }  
 }  
 usernameTv.setText(user.getName());  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 **private void** getAllComments(**final** String postID, **final** String publisherID){  
 *//get path of database named "Comments" containing comments info* **databaseReference** = FirebaseDatabase.*getInstance*().getReference(**"Posts"**).child(postID).child(**"commentsList"**);  
 *//get all comments info related to the post* **databaseReference**.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **commentsList**.clear();  
 **for** (DataSnapshot ds : dataSnapshot.getChildren()){  
 Comment comment = ds.getValue(Comment.**class**);  
 **commentsList**.add(comment);  
 }  
 **if** (getIntent().getStringExtra(**"baseClass"**).equals(**"UserProfileActivity"**)){  
 **commentAdapter** = **new** CommentAdapter(CommentsActivity.**this**, **commentsList**, publisherID, postID);  
 }  
 **else**{  
 **commentAdapter** = **new** CommentAdapter(CommentsActivity.**this**, **commentsList**, FirebaseAuth.*getInstance*().getCurrentUser().getUid(), postID);  
 }  
 **recyclerView**.setAdapter(**commentAdapter**);  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == android.R.id.***home***){  
 finish();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
}

DashboardActivity.java:

**package** com.roichomsky.socialmid;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.ActionBar;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.fragment.app.FragmentTransaction;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.MenuItem;  
  
**import** com.google.android.material.bottomnavigation.BottomNavigationView;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
  
**public class** DashboardActivity **extends** AppCompatActivity {  
  
 *//views* FirebaseAuth **firebaseAuth**;  
  
 BottomNavigationView **navigationView**;  
  
 ActionBar **actionBar**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_dashboard***);  
  
 *// Actionbar and its title* **actionBar** = getSupportActionBar();  
 **actionBar**.setTitle(**"SocialMid"**);  
  
 *//init firebaseAuth* **firebaseAuth** = FirebaseAuth.*getInstance*();  
  
  
 *// Bottom navigation* **navigationView** = findViewById(R.id.***navigation***);  
 **navigationView**.setOnNavigationItemSelectedListener(**selectedListener**);  
  
 *// Home fragment transaction (default, on start)  
 //actionBar.setTitle("Home"); // Change actionbar title* HomeFragment homeFragment = **new** HomeFragment();  
 FragmentTransaction ftl = getSupportFragmentManager().beginTransaction();  
 ftl.replace(R.id.***content***, homeFragment, **""**);  
 ftl.commit();  
  
 *//init views* }  
  
 **private** BottomNavigationView.OnNavigationItemSelectedListener **selectedListener** =  
 **new** BottomNavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 **public boolean** onNavigationItemSelected(@NonNull MenuItem menuItem) {  
 *// Handle item clicks* **switch** (menuItem.getItemId()){  
 **case** R.id.***nav\_home***:  
 *// Home fragment transaction  
 //actionBar.setTitle("Home"); // Change actionbar title* HomeFragment homeFragment = **new** HomeFragment();  
 FragmentTransaction ftl = getSupportFragmentManager().beginTransaction();  
 ftl.replace(R.id.***content***, homeFragment, **""**);  
 ftl.commit();  
 **return true**;  
 **case** R.id.***nav\_profile***:  
 *// Profile fragment transaction  
 //actionBar.setTitle("Profile"); // Change actionbar title* ProfileFragment profileFragment = **new** ProfileFragment();  
 FragmentTransaction ftl2 = getSupportFragmentManager().beginTransaction();  
 ftl2.replace(R.id.***content***, profileFragment, **""**);  
 ftl2.commit();  
 **return true**;  
 **case** R.id.***nav\_friends***:  
 *// Friends fragment transaction  
 //actionBar.setTitle("Friends"); // Change actionbar title* FriendsFragment friendsFragment = **new** FriendsFragment();  
 FragmentTransaction ftl3 = getSupportFragmentManager().beginTransaction();  
 ftl3.replace(R.id.***content***, friendsFragment, **""**);  
 ftl3.commit();  
 **return true**;  
 }  
  
 **return false**;  
 }  
 };  
  
 @Override  
 **protected void** onStart() {  
 *//checks on start of app* startService(**new** Intent(**this**, UserService.**class**));  
 **super**.onStart();}}

DoubleClickListener.java:

**package** com.roichomsky.socialmid;  
  
**import** android.os.SystemClock;  
**import** android.view.View;  
  
**public abstract class** DoubleClickListener **implements** View.OnClickListener {  
  
 *// The time in which the second tap should be done in order to qualify as  
 // a double click* **private static final long *DEFAULT\_QUALIFICATION\_SPAN*** = 200;  
 **private long doubleClickQualificationSpanInMillis**;  
 **private long timestampLastClick**;  
  
 **public** DoubleClickListener() {  
 **doubleClickQualificationSpanInMillis** = ***DEFAULT\_QUALIFICATION\_SPAN***;  
 **timestampLastClick** = 0;  
 }  
  
 **public** DoubleClickListener(**long** doubleClickQualificationSpanInMillis) {  
 **this**.**doubleClickQualificationSpanInMillis** = doubleClickQualificationSpanInMillis;  
 **timestampLastClick** = 0;  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 **if**((SystemClock.*elapsedRealtime*() - **timestampLastClick**) < **doubleClickQualificationSpanInMillis**) {  
 onDoubleClick();  
 }  
 **timestampLastClick** = SystemClock.*elapsedRealtime*();  
 }  
  
 **public abstract void** onDoubleClick();  
  
}

FriendsFragment.java:

**package** com.roichomsky.socialmid;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.annotation.Nullable;  
**import** androidx.core.view.MenuItemCompat;  
**import** androidx.fragment.app.Fragment;  
**import** androidx.recyclerview.widget.LinearLayoutManager;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**import** android.text.TextUtils;  
**import** android.view.LayoutInflater;  
**import** android.view.Menu;  
**import** android.view.MenuInflater;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.SearchView;  
  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
  
**import** java.util.ArrayList;  
  
**public class** FriendsFragment **extends** Fragment {  
  
 RecyclerView **recyclerView**;  
  
 FirebaseAuth **firebaseAuth**;  
  
 ArrayList<User> **friendsList**;  
 ArrayList<String> **uidList**;  
 AdapterUsers **adapterUsers**;  
  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.***fragment\_friends***, container, **false**);  
 *//init recyclerView* **recyclerView** = view.findViewById(R.id.***users\_recyclerView***);  
 *//set it's properties* **recyclerView**.setHasFixedSize(**true**);  
 **recyclerView**.setLayoutManager(**new** LinearLayoutManager(getActivity()));  
  
 *//init firebaseAuth* **firebaseAuth** = FirebaseAuth.*getInstance*();  
  
 *//init user list* **friendsList** = **new** ArrayList<>();  
 **uidList** = **new** ArrayList<>();  
  
 *//Get all FriendsUID* getAllFriendsUID();  
 getAllFriends();  
 **return** view;  
 }  
  
 **private void** searchUsers(**final** String query, **final** ArrayList<String> uidList) {  
 *//get current user* **final** FirebaseUser fUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 *//get path of database named "Users" containing users info* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**);  
 *//get all data from path* reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **friendsList**.clear();  
 **for** (DataSnapshot ds : dataSnapshot.getChildren()) {  
 User user = ds.getValue(User.**class**);  
  
 *//get all searched users except currently signed in* **if** (uidList.contains(user.getUid())) {  
  
 **if** (user.getName().toLowerCase().contains(query.toLowerCase()) ||  
 user.getEmail().toLowerCase().contains(query.toLowerCase())) {  
 **friendsList**.add(user);  
 }  
 }  
  
 *//adapter* **adapterUsers** = **new** AdapterUsers(getActivity(), **friendsList**);  
 *//refresh adapter after search* **adapterUsers**.notifyDataSetChanged();  
 *//set adapter to recyclerView* **recyclerView**.setAdapter(**adapterUsers**);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 **private void** getAllFriends(){  
 *//get current user* **final** FirebaseUser fUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 *//get path of database named "Users" containing users info* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**);  
 *//get all data from path* reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **friendsList**.clear();  
 **for** (DataSnapshot ds : dataSnapshot.getChildren()) {  
 User user = ds.getValue(User.**class**);  
 *//get all friends* **if** (**uidList**.contains(user.getUid())) {  
 **friendsList**.add(user);  
 }  
  
 *//adapter* **adapterUsers** = **new** AdapterUsers(getActivity(), **friendsList**);  
 *//set adapter to recyclerView* **recyclerView**.setAdapter(**adapterUsers**);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 **private void** getAllFriendsUID() {  
 *//get current user* **final** FirebaseUser fUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 *//get path of database named "Users" containing users info* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**).child(fUser.getUid()).child(**"friendsList"**);  
 DatabaseReference reference2nd = FirebaseDatabase.*getInstance*().getReference(**"Users"**);  
 *//get all data from path* reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **uidList**.clear();  
 **for** (DataSnapshot ds : dataSnapshot.getChildren()) {  
 String uid = ds.getValue(String.**class**);  
 **uidList**.add(uid);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 @Override  
 **public void** onCreate(@Nullable Bundle savedInstanceState) {  
 setHasOptionsMenu(**true**); *// show menu in fragment* **super**.onCreate(savedInstanceState);  
 }  
  
 **public void** likesNotification(){  
 DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Likes"**);  
 reference.addListenerForSingleValueEvent(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **for** (DataSnapshot ds : dataSnapshot.getChildren()){  
 Like like = ds.getValue(Like.**class**);  
 **if** (Integer.*parseInt*(like.getLikesCounter()) >= 10){  
  
 }  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
  
 *//inflate option menu* @Override  
 **public void** onCreateOptionsMenu(Menu menu, MenuInflater inflater) {  
 *//inflating menu* inflater.inflate(R.menu.***menu\_main***, menu);  
  
 *//SearchView* MenuItem item = menu.findItem(R.id.***action\_search***);  
 **final** SearchView searchView = (SearchView) MenuItemCompat.*getActionView*(item);  
  
 *//search listener* searchView.setOnQueryTextListener(**new** SearchView.OnQueryTextListener() {  
 @Override  
 **public boolean** onQueryTextSubmit(String s) {  
 *//called when user presses the search button from keyboard  
 //if search query is not empty then search* **if** (!TextUtils.*isEmpty*(s.trim())){  
 *//search text contains text, search it* searchUsers(s, **uidList**);  
 }  
 **return false**;  
 }  
  
 @Override  
 **public boolean** onQueryTextChange(String s) {  
 *//called when user changes the search text  
 //if search query is not empty then search* **if** (!TextUtils.*isEmpty*(s.trim())){  
 *//search text contains text, search it  
 //searchUsers(s);* }  
 **else** {  
 *//search text empty get all users* getAllFriendsUID();  
 getAllFriends();  
 }  
 **return false**;  
 }  
 });  
 **super**.onCreateOptionsMenu(menu, inflater);  
 }  
  
 *//handle menu item click* @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 *//get item id* **int** id = item.getItemId();  
 **if** (id==R.id.***action\_logout***){  
 **firebaseAuth**.signOut();  
 getActivity().startService(**new** Intent(getActivity(), UserService.**class**));  
 }  
 **if** (id==R.id.***action\_upload***){  
 startActivity(**new** Intent(getContext(), UploadPostActivity.**class**));  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
}

HomeFragment.java:

**package** com.roichomsky.socialmid;  
  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.annotation.Nullable;  
**import** androidx.fragment.app.Fragment;  
**import** androidx.recyclerview.widget.LinearLayoutManager;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**import** android.view.LayoutInflater;  
**import** android.view.Menu;  
**import** android.view.MenuInflater;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
  
**import** java.util.ArrayList;  
**import** java.util.Collections;  
  
**public class** HomeFragment **extends** Fragment {  
  
 **private** FirebaseAuth **firebaseAuth**;  
 **private** RecyclerView **recyclerView**;  
 **private** ArrayList<Post> **postList**;  
 **private** PostAdapter **postAdapter**;  
  
 **public** HomeFragment() {  
 *// Required empty public constructor* }  
  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 *// Inflate the layout for this fragment* View view = inflater.inflate(R.layout.***fragment\_home***, container, **false**);  
  
 *//init firebaseAuth* **firebaseAuth** = FirebaseAuth.*getInstance*();  
  
 *//init post list* **postList** = **new** ArrayList<>();  
  
 *//init recyclerView* **recyclerView** = view.findViewById(R.id.***posts\_recyclerView***);  
 *//set it's properties* **recyclerView**.setLayoutManager(**new** LinearLayoutManager(getActivity()));  
 **recyclerView**.setHasFixedSize(**true**);  
 getAllPosts();  
 **return** view;  
 }  
 @Override  
 **public void** onCreate(@Nullable Bundle savedInstanceState) {  
 setHasOptionsMenu(**true**); *// show menu in fragment* **super**.onCreate(savedInstanceState);  
 }  
  
 **private void** getAllPosts() {  
 *//get path of database named "Posts" containing posts info* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Posts"**);  
 *//get all data from path* reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **postList**.clear();  
 **for** (DataSnapshot ds : dataSnapshot.getChildren()) {  
 Post post = ds.getValue(Post.**class**);  
 **postList**.add(post);  
  
 Collections.*reverse*(**postList**);  
  
 *//adapter* **postAdapter** = **new** PostAdapter(getActivity(), **postList**, FirebaseAuth.*getInstance*().getCurrentUser().getUid(), **"HomeClass"**);  
 **recyclerView**.setAdapter(**postAdapter**);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 *//inflate option menu* @Override  
 **public void** onCreateOptionsMenu(Menu menu, MenuInflater inflater) {  
 *//inflating menu* inflater.inflate(R.menu.***menu\_main***, menu);  
 MenuItem item = menu.findItem(R.id.***action\_search***);  
 item.setVisible(**false**);  
 **super**.onCreateOptionsMenu(menu, inflater);  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 *//get item id* **int** id = item.getItemId();  
 **if** (id==R.id.***action\_logout***){  
 **firebaseAuth**.signOut();  
 getActivity().startService(**new** Intent(getActivity(), UserService.**class**));  
 }  
 **if** (id==R.id.***action\_upload***){  
 startActivity(**new** Intent(getContext(), UploadPostActivity.**class**));  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
}

Like.java:

**package** com.roichomsky.socialmid;  
  
**import** java.util.HashMap;  
  
**public class** Like {  
 **private** String **likesCounter**;  
 **private** HashMap<String, Object> **likedByList**;  
 **private** String **postID**;  
  
 **public** Like(){}  
  
 **public** Like(String likesCounter, HashMap<String, Object> likedByList, String postID) {  
 **this**.**likesCounter** = likesCounter;  
 **this**.**likedByList** = likedByList;  
 **this**.**postID** = postID;  
 }  
  
 **public** String getLikesCounter() {  
 **return likesCounter**;  
 }  
  
 **public void** setLikesCounter(String likesCounter) {  
 **this**.**likesCounter** = likesCounter;  
 }  
  
 **public** HashMap<String, Object> getLikedByList() {  
 **return likedByList**;  
 }  
  
 **public void** setLikedByList(HashMap<String, Object> likedByList) {  
 **this**.**likedByList** = likedByList;  
 }  
  
 **public** String getPostID() {  
 **return postID**;  
 }  
  
 **public void** setPostID(String postID) {  
 **this**.**postID** = postID;  
 }  
}

LoginActivity.java:

**package** com.roichomsky.socialmid;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.ActionBar;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.app.AlertDialog;  
**import** android.app.Dialog;  
**import** android.app.ProgressDialog;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.text.InputType;  
**import** android.util.Patterns;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.tasks.OnCompleteListener;  
**import** com.google.android.gms.tasks.OnFailureListener;  
**import** com.google.android.gms.tasks.Task;  
**import** com.google.firebase.auth.AuthResult;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
  
**import** java.util.HashMap;  
  
**public class** LoginActivity **extends** AppCompatActivity {  
  
 Dialog **registerDialog**;  
 TextView **mNoAccountTv**, **mForgotPassTv**, **mHaveAccountTv**;  
 EditText **mEmailEt**, **mPasswordEt**, **mNameEt**, **mPasswordEt2**, **mEmailEt2**;  
 Button **mLoginBtn**, **mRegisterBtN2**;  
 **private** FirebaseAuth **mAuth**, **mAuth2**;  
  
 *//progressBar to display while logging user and registering one* ProgressDialog **progressDialog**, **progressDialog2**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_login***);  
  
 *// Actionbar and its title* ActionBar actionBar = getSupportActionBar();  
 actionBar.setTitle(**"Login"**);  
 *// Enable back button* actionBar.setDisplayHomeAsUpEnabled(**true**);  
 actionBar.setDisplayShowHomeEnabled(**true**);  
  
 *// Initialize Firebase Auth* **mAuth** = FirebaseAuth.*getInstance*();  
  
 *//init* **mEmailEt** = findViewById(R.id.***emailEt***);  
 **mPasswordEt** = findViewById(R.id.***passwordEt***);  
 **mLoginBtn** = findViewById(R.id.***loginBtn***);  
 **mNoAccountTv** = findViewById(R.id.***noAccountTv***);  
 **mForgotPassTv** = findViewById(R.id.***forgotPassTv***);  
 **progressDialog** = **new** ProgressDialog(**this**);  
  
 *//forgot password TextView click* **mForgotPassTv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 showRecoverPasswordDialog();  
 }  
 });  
  
 *//don't have an account TextView click open register dialog* **mNoAccountTv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 createRegisterDialog();  
 }  
 });  
  
 *//login button click* **mLoginBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//input data* String email = **mEmailEt**.getText().toString().trim();  
 String password = **mPasswordEt**.getText().toString().trim();  
 *//validate* **if** (!Patterns.***EMAIL\_ADDRESS***.matcher(email).matches()){  
 *//set error and focus to email editText* **mEmailEt**.setError(**"Invalid Email"**);  
 **mEmailEt**.setFocusable(**true**);  
 }  
 **else**{  
 loginUser(email, password); *// login in the user* }  
 }  
 });  
 }  
  
 **private void** showRecoverPasswordDialog() {  
 *//AlertDialog* AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 builder.setTitle(**"Recover Password"**);  
  
 *//set layout - LinearLayout* LinearLayout linearLayout = **new** LinearLayout(**this**);  
 *//views to set in dialog* **final** EditText emailEt = **new** EditText(**this**);  
 emailEt.setHint(**"Email"**);  
 emailEt.setInputType(InputType.***TYPE\_TEXT\_VARIATION\_EMAIL\_ADDRESS***);  
  
 *//sets the mid EditText width to fit n amount of 'M' letters* emailEt.setMinEms(16);  
  
 linearLayout.addView(emailEt);  
 linearLayout.setPadding(10,10,10,10);  
  
 builder.setView(linearLayout);  
  
 *//buttons recover* builder.setPositiveButton(**"Recover"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *//input email* String email = emailEt.getText().toString().trim();  
 beginRecovery(email);  
 }  
 });  
  
 *//buttons cancel* builder.setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *//dismiss dialog* dialog.dismiss();  
 }  
 });  
  
 *//show dialog* builder.create().show();  
 }  
  
 **private void** beginRecovery(String email) {  
 *//displays the progressBar* **progressDialog**.show();  
 **progressDialog**.setMessage(**"Sending email..."**);  
 **mAuth**.sendPasswordResetEmail(email).addOnCompleteListener(**new** OnCompleteListener<Void>() {  
 @Override  
 **public void** onComplete(@NonNull Task<Void> task) {  
 **progressDialog**.dismiss();  
 **if**(task.isSuccessful()){  
 Toast.*makeText*(LoginActivity.**this**, **"Check email for reset link"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else**{  
 Toast.*makeText*(LoginActivity.**this**, **"Failed"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 *//get and show proper error message* **progressDialog**.dismiss();  
 Toast.*makeText*(LoginActivity.**this**, **""**+e.getMessage(), Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 }  
  
 **private void** loginUser(String email, String password) {  
 *//displays the progressBar* **progressDialog**.setMessage(**"Logging In..."**);  
 **progressDialog**.show();  
 **mAuth**.signInWithEmailAndPassword(email, password)  
 .addOnCompleteListener(**this**, **new** OnCompleteListener<AuthResult>() {  
 @Override  
 **public void** onComplete(@NonNull Task<AuthResult> task) {  
 **if** (task.isSuccessful()) {  
 *//dismisses the progressBar* **progressDialog**.dismiss();  
 *//user is logged in, so start DashboardActivity* startActivity(**new** Intent(LoginActivity.**this**, DashboardActivity.**class**));  
 finish();  
 } **else** {  
 *//dismisses the progressBar* **progressDialog**.dismiss();  
 *// If sign in fails, display a message to the user.* Toast.*makeText*(LoginActivity.**this**, **"Login failed."**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 });  
 }  
  
 **public boolean** onSupportNavigateUp(){  
 onBackPressed(); *// go previous activity* **return super**.onSupportNavigateUp();  
 }  
  
 **public void** createRegisterDialog()  
 {  
 *// Creates dialog and sets the layout and flags* **registerDialog** = **new** Dialog(**this**);  
 **registerDialog**.setCancelable(**true**);  
 **registerDialog**.setContentView(R.layout.***register\_dialog***);  
  
 *// Init* **mEmailEt2** = **registerDialog**.findViewById(R.id.***emailEt***);  
 **mPasswordEt2** = **registerDialog**.findViewById(R.id.***passwordEt***);  
 **mNameEt** = **registerDialog**.findViewById(R.id.***nameEt***);  
 **mRegisterBtN2** = **registerDialog**.findViewById(R.id.***registerBtn***);  
 **mHaveAccountTv** = **registerDialog**.findViewById(R.id.***haveAccountTv***);  
 **mHaveAccountTv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **registerDialog**.dismiss();  
 }  
 });  
  
 *// Handle register button click* **mRegisterBtN2**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//input email, password* String email = **mEmailEt2**.getText().toString().trim();  
 String password = **mPasswordEt2**.getText().toString().trim();  
 String name = **mNameEt**.getText().toString().trim();  
 *//validate* **if** (!Patterns.***EMAIL\_ADDRESS***.matcher(email).matches()){  
 *//set error and focus to email editText* **mEmailEt2**.setError(**"Invalid Email"**);  
 **mEmailEt2**.setFocusable(**true**);  
 }  
 **else if**(password.length()<6){  
 *//set error and focus to password editText* **mPasswordEt2**.setError(**"Password length at least 6 characters"**);  
 **mPasswordEt2**.setFocusable(**true**);  
 }  
 **else if**(name.length()<2){  
 *//set error and focus to password editText* **mNameEt**.setError(**"Invalid nickname"**);  
 **mNameEt**.setFocusable(**true**);  
 }  
 **else**{  
 registerUser(email, password); *// Registers the user* }  
 }  
 });  
  
 **registerDialog**.show();  
 }  
  
 **private void** registerUser(String email, String password) {  
 *//email and password pattern is valid, show progressDialog and start registering user* **progressDialog2**.show();  
  
 **mAuth**.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(**this**, **new** OnCompleteListener<AuthResult>() {  
 @Override  
 **public void** onComplete(@NonNull Task<AuthResult> task) {  
 **if** (task.isSuccessful()) {  
 *// Sign up success, dismiss dialog and start register activity.* **progressDialog2**.dismiss();  
  
 FirebaseUser user = **mAuth2**.getCurrentUser();  
 *// Get user email and uid from auth* String email = user.getEmail();  
 String uid = user.getUid();  
 String name = **mNameEt**.getText().toString().trim();  
 *// When user is registered, store user info in firebase realtime database too  
 // using HashMap* HashMap<Object , String> hashMap = **new** HashMap<>();  
 hashMap.put(**"email"**, email);  
 hashMap.put(**"uid"**, uid);  
 hashMap.put(**"name"**, name);  
 hashMap.put(**"image"**, **""**);  
 *// firebase database instance* FirebaseDatabase database = FirebaseDatabase.*getInstance*();  
 *// Path to store user data named "Users"* DatabaseReference reference = database.getReference(**"Users"**);  
 *// Put data withing hashmap in database* reference.child(uid).setValue(hashMap);  
  
 Toast.*makeText*(LoginActivity.**this**, **"Registered...\n"**+user.getEmail(), Toast.***LENGTH\_SHORT***).show();  
 startActivity(**new** Intent(LoginActivity.**this**, DashboardActivity.**class**));  
 finish();  
 } **else** {  
 *// If sign in fails, display a message to the user.* **progressDialog2**.dismiss();  
 Toast.*makeText*(LoginActivity.**this**, **"Registration failed."**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 **progressDialog2**.dismiss();  
 Toast.*makeText*(LoginActivity.**this**, **""**+e.getMessage(), Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 }  
}

MainActivity.java:

**package** com.roichomsky.socialmid;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.app.Dialog;  
**import** android.app.ProgressDialog;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.util.Patterns;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.tasks.OnCompleteListener;  
**import** com.google.android.gms.tasks.OnFailureListener;  
**import** com.google.android.gms.tasks.Task;  
**import** com.google.firebase.auth.AuthResult;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 *//3310 JAVA CODING LINES  
  
 //views* EditText **mEmailEt**, **mPasswordEt**, **mNameEt**;  
 TextView **mHaveAccountTv**;  
 ProgressDialog **progressDialog**;  
 FirebaseAuth **mAuth**;  
 Dialog **registerDialog**;  
 Button **mRegisterBtn**, **mLoginBtn**, **mRegisterBtN2**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 *// Init views* **progressDialog** = **new** ProgressDialog(**this**);  
 **progressDialog**.setMessage(**"Registering User..."**);  
 **mAuth** = FirebaseAuth.*getInstance*();  
 **mRegisterBtn** = findViewById(R.id.***registerBtn***);  
 **mLoginBtn** = findViewById(R.id.***loginBtn***);  
  
 *// Handle register button click* **mRegisterBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//Start RegisterActivity* createRegisterDialog();  
 }  
 });  
 *//handle login button click* **mLoginBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 startActivity(**new** Intent(MainActivity.**this**, LoginActivity.**class**));  
 }  
 });  
 }  
  
 **public void** createRegisterDialog()  
 {  
 **registerDialog** = **new** Dialog(**this**);  
 **registerDialog**.setCancelable(**true**);  
 **registerDialog**.setContentView(R.layout.***register\_dialog***);  
  
 *// Init* **mEmailEt** = **registerDialog**.findViewById(R.id.***emailEt***);  
 **mPasswordEt** = **registerDialog**.findViewById(R.id.***passwordEt***);  
 **mRegisterBtN2** = **registerDialog**.findViewById(R.id.***registerBtn***);  
 **mHaveAccountTv** = **registerDialog**.findViewById(R.id.***haveAccountTv***);  
 **mNameEt** = **registerDialog**.findViewById(R.id.***nameEt***);  
  
 **mHaveAccountTv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 startActivity(**new** Intent(MainActivity.**this**, LoginActivity.**class**));  
 }  
 });  
  
 *// Handle register button click* **mRegisterBtN2**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//input email, password* String email = **mEmailEt**.getText().toString().trim();  
 String password = **mPasswordEt**.getText().toString().trim();  
 String name = **mNameEt**.getText().toString().trim();  
 *//validate* **if** (!Patterns.***EMAIL\_ADDRESS***.matcher(email).matches()){  
 *//set error and focus to email editText* **mEmailEt**.setError(**"Invalid Email"**);  
 **mEmailEt**.setFocusable(**true**);  
 }  
 **else if**(password.length()<6){  
 *//set error and focus to password editText* **mPasswordEt**.setError(**"Password length at least 6 characters"**);  
 **mPasswordEt**.setFocusable(**true**);  
 }  
 **else if**(name.length()<2){  
 *//set error and focus to password editText* **mNameEt**.setError(**"Invalid nickname"**);  
 **mNameEt**.setFocusable(**true**);  
 }  
 **else**{  
 registerUser(email, password); *// Registers the user* }  
 }  
 });  
  
 **registerDialog**.show();  
 }  
 **private void** registerUser(String email, String password) {  
 *//email and password pattern is valid, show progressDialog and start registering user* **progressDialog**.show();  
  
 **mAuth**.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(**this**, **new** OnCompleteListener<AuthResult>() {  
 @Override  
 **public void** onComplete(@NonNull Task<AuthResult> task) {  
 **if** (task.isSuccessful()) {  
 *// Sign in success, dismiss dialog and start register activity.* **progressDialog**.dismiss();  
  
 FirebaseUser user = **mAuth**.getCurrentUser();  
 *// Get user email and uid from auth* String email = user.getEmail();  
 String uid = user.getUid();  
 String name = **mNameEt**.getText().toString().trim();  
 *// When user is registered store user info in firebase realtime database too  
 // using HashMap* HashMap<Object , String> hashMap = **new** HashMap<>();  
 hashMap.put(**"email"**, email);  
 hashMap.put(**"uid"**, uid);  
 hashMap.put(**"name"**, name);  
 *// firebase database instance* FirebaseDatabase database = FirebaseDatabase.*getInstance*();  
 *// Path to store user data named "Users"* DatabaseReference reference = database.getReference(**"Users"**);  
 *// Put data withing hashmap in database* reference.child(uid).setValue(hashMap);  
  
 Toast.*makeText*(MainActivity.**this**, **"Registered...\n"**+user.getEmail(), Toast.***LENGTH\_SHORT***).show();  
 startActivity(**new** Intent(MainActivity.**this**, DashboardActivity.**class**));  
 finish();  
 } **else** {  
 *// If sign in fails, display a message to the user.* **progressDialog**.dismiss();  
 Toast.*makeText*(MainActivity.**this**, **"Registration failed."**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 **progressDialog**.dismiss();  
 Toast.*makeText*(MainActivity.**this**, **""**+e.getMessage(), Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 }  
}

Message.java:

**package** com.roichomsky.socialmid;  
  
**public class** Message {  
  
 **private** String **sender**, **receiver**, **timestamp**, **message**, **seen**;  
 **public** Message(){}  
  
 **public** Message(String sender, String receiver, String timestamp, String message, String seen) {  
 **this**.**sender** = sender;  
 **this**.**receiver** = receiver;  
 **this**.**timestamp** = timestamp;  
 **this**.**message** = message;  
 **this**.**seen** = seen;  
 }  
  
 **public** String getSeen() {  
 **return seen**;  
 }  
  
 **public void** setSeen(String seen) {  
 **this**.**seen** = seen;  
 }  
  
 **public** String getMessage() {  
 **return message**;  
 }  
  
 **public void** setMessage(String message) {  
 **this**.**message** = message;  
 }  
  
 **public** String getSender() {  
 **return sender**;  
 }  
  
 **public void** setSender(String sender) {  
 **this**.**sender** = sender;  
 }  
  
 **public** String getReceiver() {  
 **return receiver**;  
 }  
  
 **public void** setReceiver(String receiver) {  
 **this**.**receiver** = receiver;  
 }  
  
 **public** String getTimestamp() {  
 **return timestamp**;  
 }  
  
 **public void** setTimestamp(String timestamp) {  
 **this**.**timestamp** = timestamp;}}

Post.java:

**package** com.roichomsky.socialmid;  
  
**public class** Post {  
 **private** String **postID**;  
 **private** String **postURL**;  
 **private** String **description**;  
 **private** String **publisherID**;  
 **private** String **timestamp**;  
  
 **public** Post(){ }  
  
 **public** Post(String description,String postID, String postURL, String publisherID, String likesCounter, String timestamp) {  
 **this**.**postID** = postID;  
 **this**.**postURL** = postURL;  
 **this**.**description** = description;  
 **this**.**publisherID** = publisherID;  
 **this**.**timestamp** = timestamp;  
 }  
  
 **public** String getPostID() {  
 **return postID**;  
 }  
  
 **public void** setPostID(String postID) {  
 **this**.**postID** = postID;  
 }  
  
 **public** String getPostURL() {  
 **return postURL**;  
 }  
  
 **public void** setPostURL(String postURL) {  
 **this**.**postURL** = postURL;  
 }  
  
 **public** String getDescription() {  
 **return description**;  
 }  
  
 **public void** setDescription(String description) {  
 **this**.**description** = description;  
 }  
  
 **public** String getPublisherID() {  
 **return publisherID**;  
 }  
  
 **public void** setPublisherID(String publisherID) {  
 **this**.**publisherID** = publisherID;  
 }  
  
 **public** String getTimestamp() {  
 **return timestamp**;  
 }  
  
 **public void** setTimestamp(String timestamp) {  
 **this**.**timestamp** = timestamp;  
 }  
}

PostAdapter.java:

**package** com.roichomsky.socialmid;  
  
**import** android.app.AlertDialog;  
**import** android.app.ProgressDialog;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.text.TextUtils;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.EditText;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.RelativeLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.tasks.OnFailureListener;  
**import** com.google.android.gms.tasks.OnSuccessListener;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
**import** com.like.LikeButton;  
**import** com.like.OnLikeListener;  
**import** com.squareup.picasso.Picasso;  
  
**import** java.text.SimpleDateFormat;  
**import** java.util.Calendar;  
**import** java.util.Date;  
**import** java.util.HashMap;  
**import** java.util.List;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**public class** PostAdapter **extends** RecyclerView.Adapter<PostAdapter.MyHolder>{  
  
 **public** Context **mContext**;  
 **public** List<Post> **mPost**;  
 **public** String **profileUserUID**;  
 **public** String **baseClass**;  
  
 FirebaseUser **firebaseUser**;  
  
 ProgressDialog **pd**;  
  
 **public** PostAdapter(Context mContext, List<Post> mPost, String profileUserUID, String baseClass){  
 **this**.**mContext** = mContext;  
 **this**.**mPost** = mPost;  
 **this**.**profileUserUID** = profileUserUID;  
 **this**.**baseClass** = baseClass;  
 }  
  
 @NonNull  
 @Override  
 **public** MyHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  
 View view = LayoutInflater.*from*(**mContext**).inflate(R.layout.***post\_item***, parent, **false**);  
 **return new** PostAdapter.MyHolder(view);  
 }  
  
 @Override  
 **public void** onBindViewHolder(@NonNull **final** MyHolder holder, **final int** position) {  
 **firebaseUser** = FirebaseAuth.*getInstance*().getCurrentUser();  
 **final** Post post = **mPost**.get(position);  
  
 **if** (post.getDescription().equals(**""**)){  
 holder.**descriptionTv**.setVisibility(View.***GONE***);  
 } **else** {  
 holder.**descriptionTv**.setVisibility(View.***VISIBLE***);  
 holder.**descriptionTv**.setText(post.getDescription());  
 }  
  
 **pd** = **new** ProgressDialog(**mContext**);  
  
 **final** Intent intent = **new** Intent(**mContext**, CommentsActivity.**class**);  
 intent.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 intent.putExtra(**"postID"**, post.getPostID());  
 intent.putExtra(**"publisherID"**, post.getPublisherID());  
 intent.putExtra(**"description"**, post.getDescription());  
 intent.putExtra(**"baseClass"**, **baseClass**);  
  
 holder.**commentsTv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **mContext**.startActivity(intent);  
 }  
 });  
  
 holder.**commentIv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **firebaseUser**.getUid();  
 addComment(**firebaseUser**.getUid(), post.getPostID(), intent);  
 }  
 });  
  
 holder.**postIv**.setOnClickListener(**new** DoubleClickListener() {  
 @Override  
 **public void** onDoubleClick() {  
 alreadyLiked(**firebaseUser**.getUid(), post.getPostID(), holder.**likeBtn**);  
  
 **if** (!holder.**likeBtn**.isLiked()){  
 **final int**[] currentLikes = {0};  
 **final** DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Likes"**).child(post.getPostID());  
 **final** DatabaseReference reference2 = reference.child(**"likedByList"**);  
 reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 Like likes = dataSnapshot.getValue(Like.**class**);  
  
 **if** (likes != **null**) {  
 holder.**likesTv**.setText(likes.getLikesCounter() + **" likes"**);  
 currentLikes[0] = Integer.*parseInt*(likes.getLikesCounter());  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
  
 holder.**likeBtn**.setLiked(**true**);  
 currentLikes[0] = currentLikes[0]+1;  
 HashMap<String, Object> results = **new** HashMap<>();  
 results.put(**"likesCounter"**, Integer.*toString*(currentLikes[0]));  
 results.put(**"postID"**, post.getPostID());  
 reference.updateChildren(results)  
 .addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 holder.**likesTv**.setText(currentLikes[0]+**" likes"**);  
 HashMap<String, Object> results2 = **new** HashMap<>();  
 results2.put(**firebaseUser**.getUid(), **"true"**);  
 reference2.updateChildren(results2);  
 }  
 })  
 .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 holder.**likeBtn**.setLiked(**false**);  
 }  
 });  
 }  
 }  
 });  
  
 holder.**usernameTv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!**firebaseUser**.getUid().equals(post.getPublisherID()) &&  
 !**profileUserUID**.equals(post.getPublisherID())){  
 Intent profileIntent = **new** Intent(**mContext**, UserProfileActivity.**class**);  
 profileIntent.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 profileIntent.putExtra(**"uid"**, post.getPublisherID());  
 **mContext**.startActivity(profileIntent);  
 }  
 }  
 });  
  
 **if** (post.getPublisherID().equals(**firebaseUser**.getUid())){  
 holder.**optionIv**.setVisibility(View.***VISIBLE***);  
 holder.**optionIv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 String[] options = {**"Edit Description"**, **"Delete Post"**};  
  
 *//alert dialog* AlertDialog.Builder builder = **new** AlertDialog.Builder(**mContext**);  
 *//set title* builder.setTitle(**"Post Options"**);  
  
 builder.setItems(options, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **if** (which==0){  
 EditDescription(post);  
 }  
 **else if** (which==1){  
 FirebaseDatabase.*getInstance*().getReference(**"Likes"**).child(post.getPostID()).removeValue();  
 FirebaseDatabase.*getInstance*().getReference(**"Posts"**).child(post.getPostID()).removeValue().addOnSuccessListener(  
 **new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 Toast.*makeText*(**mContext**, **"Post has been deleted..."**, Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 ).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 Toast.*makeText*(**mContext**, **"Failed to delete..."**, Toast.***LENGTH\_LONG***).show();  
 }  
 });  
 }  
 }  
 });  
 builder.create().show();  
 }  
 });  
 }  
  
 Calendar c = Calendar.*getInstance*();  
 c.setTimeInMillis(Long.*parseLong*(post.getTimestamp()));  
 Date d = c.getTime();  
 SimpleDateFormat sdf = **new** SimpleDateFormat(**"dd/MM/yyyy HH:mm"**);  
 holder.**timeTv**.setText(sdf.format(d));  
  
 getCommentsAmount(post.getPostID(), holder.**commentsTv**, holder.**timeTv**);  
 publisherInfo(holder.**avatarIv**, holder.**usernameTv**, holder.**username2Tv**, post.getPublisherID());  
 getPostImage(holder.**postIv**, post.getPostID());  
 addLikesToPost(post, holder.**likeBtn**, holder.**likesTv**, **firebaseUser**.getUid());  
 }  
  
 @Override  
 **public int** getItemCount() {  
 *//returns the posts list size* **return mPost**.size();  
 }  
  
 **class** MyHolder **extends** RecyclerView.ViewHolder{  
  
 *//layout view holder  
  
 //Views* LikeButton **likeBtn**;  
 ImageView **avatarIv**, **postIv**, **commentIv**, **optionIv**;  
 TextView **usernameTv**, **username2Tv**, **likesTv**, **descriptionTv**, **commentsTv**, **timeTv**;  
  
 **public** MyHolder(@NonNull View itemView){  
 **super**(itemView);  
  
 *//init* **optionIv** = itemView.findViewById(R.id.***optionIv***);  
 **avatarIv** = itemView.findViewById(R.id.***avatarIv***);  
 **postIv** = itemView.findViewById(R.id.***postIv***);  
 **likeBtn** = itemView.findViewById(R.id.***likeBtn***);  
 **commentIv** = itemView.findViewById(R.id.***commentIv***);  
 **usernameTv** = itemView.findViewById(R.id.***usernameTv***);  
 **likesTv** = itemView.findViewById(R.id.***likesTv***);  
 **descriptionTv** = itemView.findViewById(R.id.***descriptionTv***);  
 **commentsTv** = itemView.findViewById(R.id.***commentsTv***);  
 **username2Tv** = itemView.findViewById(R.id.***username2Tv***);  
 **timeTv** = itemView.findViewById(R.id.***timeTv***);  
 }  
 }  
  
 **private void** publisherInfo(**final** ImageView avatarIv, **final** TextView usernameTv, **final** TextView username2Tv, String uid){  
 *//get path of database named "Users" containing users info* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**).child(uid);  
 *//get all data from path* reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 User user = dataSnapshot.getValue(User.**class**);  
 **if** (user.getImage() != **null**){  
 **try**{  
 Picasso.*get*().load(user.getImage()).placeholder(R.drawable.***ic\_default\_img***).into(avatarIv);  
 }  
 **catch** (Exception e){ }  
 }  
 usernameTv.setText(user.getName());  
 username2Tv.setText(user.getName());  
  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 **private void** getPostImage(**final** ImageView postIv, String postID){  
 DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Posts"**).child(postID);  
 reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 Post post = dataSnapshot.getValue(Post.**class**);  
 **try** {  
 Picasso.*get*().load(post.getPostURL()).into(postIv);  
 }  
 **catch** (Exception e){  
 Picasso.*get*().load(R.drawable.***ic\_add\_image***).into(postIv);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 **private void** addLikesToPost(**final** Post post, **final** LikeButton likeBtn, **final** TextView likesTv, **final** String uid){  
 **final int**[] currentLikes = {0};  
 **final** DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Likes"**).child(post.getPostID());  
 **final** DatabaseReference reference2 = reference.child(**"likedByList"**);  
 reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 Like likes = dataSnapshot.getValue(Like.**class**);  
  
 **if** (likes != **null**) {  
 likesTv.setText(likes.getLikesCounter() + **" likes"**);  
 currentLikes[0] = Integer.*parseInt*(likes.getLikesCounter());  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
  
 alreadyLiked(uid, post.getPostID(), likeBtn);  
  
 likeBtn.setOnLikeListener(**new** OnLikeListener() {  
 @Override  
 **public void** liked(**final** LikeButton likeButton) {  
 likeButton.setLiked(**true**);  
 currentLikes[0] = currentLikes[0]+1;  
 HashMap<String, Object> results = **new** HashMap<>();  
 results.put(**"likesCounter"**, Integer.*toString*(currentLikes[0]));  
 results.put(**"postID"**, post.getPostID());  
 reference.updateChildren(results)  
 .addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 likesTv.setText(currentLikes[0]+**" likes"**);  
 HashMap<String, Object> results2 = **new** HashMap<>();  
 results2.put(uid, **"true"**);  
 reference2.updateChildren(results2);  
 }  
 })  
 .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 likeButton.setLiked(**false**);  
 }  
 });  
 }  
  
 @Override  
 **public void** unLiked(**final** LikeButton likeButton) {  
 likeButton.setLiked(**false**);  
 HashMap<String, Object> results = **new** HashMap<>();  
 currentLikes[0] = currentLikes[0]-1;  
 results.put(**"likesCounter"**, Integer.*toString*(currentLikes[0]));  
 results.put(**"postID"**, post.getPostID());  
 reference.updateChildren(results)  
 .addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 likesTv.setText(currentLikes[0]+**" likes"**);  
 reference2.child(uid).removeValue();  
 }  
 })  
 .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 likeButton.setLiked(**true**);  
 }  
 });  
 }  
 });  
  
  
 }  
  
 **private void** alreadyLiked(**final** String uid, String postID, **final** LikeButton likeButton){  
 **final boolean**[] has = {**false**};  
 **final** DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Likes"**).child(postID).child(**"likedByList"**).child(uid);  
 reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 has[0] = dataSnapshot.exists();  
 **if** (has[0]){  
 likeButton.setLiked(**true**);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 }  
  
 **private void** addComment(**final** String uid, String postID, **final** Intent intent){  
 *//creates the branch for the comment* **final** DatabaseReference databaseReference = FirebaseDatabase.*getInstance*().getReference(**"Posts"**).child(postID).child(**"commentsList"**);  
 **final** String key = databaseReference.push().getKey();  
 **assert** key != **null**;  
 databaseReference.child(key);  
  
 *//creates the dialog view* AlertDialog.Builder builder = **new** AlertDialog.Builder(**mContext**);  
 builder.setTitle(**"Add a comment..."**);  
 *//set layout of dialog* LinearLayout linearLayout = **new** LinearLayout(**mContext**);  
 linearLayout.setOrientation(LinearLayout.***VERTICAL***);  
 linearLayout.setPadding(10,10,10,10);  
 *//add edit text* **final** EditText editText = **new** EditText(**mContext**);  
 editText.setHint(**"Add a comment..."**);  
 linearLayout.addView(editText);  
  
 **pd**.setMessage(**"Adding the comment..."**);  
  
 builder.setView(linearLayout);  
  
 builder.setPositiveButton(**"Add"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *//input text from edit text* String value = editText.getText().toString().trim();  
 **if** (!TextUtils.*isEmpty*(value)){  
 **pd**.show();  
 HashMap<String, Object> result = **new** HashMap<>();  
 result.put(**"publisherID"**, uid);  
 result.put(**"content"**, value);  
 result.put(**"commentID"**, key);  
 databaseReference.child(key).updateChildren(result)  
 .addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 *//added, dismiss progress bar* **pd**.dismiss();  
 Toast.*makeText*(**mContext**, **"Added..."**, Toast.***LENGTH\_SHORT***).show();  
 **mContext**.startActivity(intent);  
 }  
 })  
 .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 *//failed, dismiss progress bar, get and show error message* **pd**.dismiss();  
 Toast.*makeText*(**mContext**, **""**+e.getMessage(), Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 }  
 **else**{  
 Toast.*makeText*(**mContext**, **"Please enter validate input"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 });  
 *//add button in dialog to cancel* builder.setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.dismiss();  
 }  
 });  
 *//create and show dialog* builder.create().show();  
 }  
  
 **private void** getCommentsAmount(String postID, **final** TextView commentsTv, **final** TextView timeTv){  
 **final long**[] currentComments = {0};  
 DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Posts"**).child(postID).child(**"commentsList"**);  
 reference.addListenerForSingleValueEvent(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 currentComments[0] = dataSnapshot.getChildrenCount();  
 String commentsTvText = **"View all "**+ currentComments[0] +**" comments"**;  
 **if** (currentComments[0] == 1){  
 commentsTvText = **"View 1 comment"**;  
 }  
 **else if**(currentComments[0] == 0){  
 commentsTv.setVisibility(View.***INVISIBLE***);  
 RelativeLayout.LayoutParams params1 = **new** RelativeLayout.LayoutParams(  
 ViewGroup.LayoutParams.***WRAP\_CONTENT***, ViewGroup.LayoutParams.***WRAP\_CONTENT***);  
  
 params1.addRule(RelativeLayout.***ALIGN\_PARENT\_TOP***);  
 timeTv.setLayoutParams(params1);  
 }  
 **else** {  
 commentsTvText = **"View all "**+ currentComments[0] +**" comments"**;  
 }  
 commentsTv.setText(commentsTvText);  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
 commentsTv.setVisibility(View.***INVISIBLE***);  
 }  
 });  
 }  
  
 **private void** EditDescription(**final** Post post) {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**mContext**);  
 builder.setTitle(**"Edit Description"**);  
 *//set layout of dialog* LinearLayout linearLayout = **new** LinearLayout(**mContext**);  
 linearLayout.setOrientation(LinearLayout.***VERTICAL***);  
 linearLayout.setPadding(10,10,10,10);  
 *//add edit text* **final** EditText editText = **new** EditText(**mContext**);  
 editText.setHint(**"Enter Description"**);  
 linearLayout.addView(editText);  
  
 builder.setView(linearLayout);  
  
 builder.setPositiveButton(**"Edit"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *//input text from edit text* String value = editText.getText().toString().trim();  
 DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Posts"**).child(post.getPostID()).child(**"description"**);  
 **if** (!TextUtils.*isEmpty*(value)){  
 **pd**.setMessage(**"Updating..."**);  
 **pd**.show();  
 reference.setValue(value).addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 **pd**.dismiss();  
 }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 **pd**.dismiss();  
 Toast.*makeText*(**mContext**, **"Failed to update..."**, Toast.***LENGTH\_LONG***).show();  
 }  
 });  
 }  
 **else**{  
 Toast.*makeText*(**mContext**, **"Please enter validate input"**, Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 });  
  
 *//add button in dialog to cancel* builder.setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.dismiss();  
 }  
 });  
  
 builder.create().show();  
 }  
}

ProfileFragment.java:

**package** com.roichomsky.socialmid;  
  
  
**import** android.Manifest;  
**import** android.app.AlertDialog;  
**import** android.app.ProgressDialog;  
**import** android.content.ContentValues;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.annotation.Nullable;  
**import** androidx.core.content.ContextCompat;  
**import** androidx.fragment.app.Fragment;  
**import** androidx.recyclerview.widget.LinearLayoutManager;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**import** android.provider.MediaStore;  
**import** android.text.TextUtils;  
**import** android.view.LayoutInflater;  
**import** android.view.Menu;  
**import** android.view.MenuInflater;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.EditText;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.tasks.OnFailureListener;  
**import** com.google.android.gms.tasks.OnSuccessListener;  
**import** com.google.android.gms.tasks.Task;  
**import** com.google.android.material.floatingactionbutton.FloatingActionButton;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
**import** com.google.firebase.storage.FirebaseStorage;  
**import** com.google.firebase.storage.StorageReference;  
**import** com.google.firebase.storage.UploadTask;  
**import** com.squareup.picasso.Picasso;  
  
**import** java.util.ArrayList;  
**import** java.util.Collections;  
**import** java.util.HashMap;  
  
**import static** android.app.Activity.***RESULT\_OK***;  
  
**public class** ProfileFragment **extends** Fragment {  
  
 *//firebase* FirebaseAuth **firebaseAuth**;  
 FirebaseUser **user**;  
 FirebaseDatabase **firebaseDatabase**;  
 DatabaseReference **databaseReference**;  
  
 *//storage* StorageReference **storageReference**;  
 *//path where images of user profile and cover will be stored* String **storagePath** = **"Users\_Profile\_Cover\_Images/"**;  
  
 ArrayList<Post> **postList**;  
 PostAdapter **postAdapter**;  
  
 *//views from xml* FloatingActionButton **fab**;  
 ImageView **avatarIv**, **coverIv**;  
 TextView **nameTv**, **emailTv**, **noPostsBtn**;  
 LinearLayout **noPostsLl**;  
 RecyclerView **recyclerView**;  
  
 *//progress dialog* ProgressDialog **pd**;  
  
 *//permissions constants* **private static final int *CAMERA\_REQUEST\_CODE*** = 100;  
 **private static final int *STORAGE\_REQUEST\_CODE*** = 200;  
 **private static final int *IMAGE\_PICK\_GALLERY\_CODE*** = 300;  
 **private static final int *IMAGE\_PICK\_CAMERA\_CODE*** = 400;  
 *//arrays of permissions to be requested* String[] **cameraPermissions**;  
 String[] **storagePermissions**;  
  
 *//uri of picked image* Uri **image\_uri**;  
  
 *//for checking profile or cover photo* String **profileOrCoverPhoto**;  
  
 **final** Boolean[] **avatarImage** = {**false**};  
 **final** Boolean[] **coverImage** = {**false**};  
  
 **public** ProfileFragment() {  
 *// Required empty public constructor* }  
  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 *// Inflate the layout for this fragment* View view = inflater.inflate(R.layout.***fragment\_profile***, container, **false**);  
  
  
 *//init firebase* **firebaseAuth** = FirebaseAuth.*getInstance*();  
 **user** = **firebaseAuth**.getCurrentUser();  
 **firebaseDatabase** = FirebaseDatabase.*getInstance*();  
 **databaseReference** = **firebaseDatabase**.getReference(**"Users"**);  
 **storageReference** = FirebaseStorage.*getInstance*().getReference();  
  
 *//init arrays of permissions* **cameraPermissions** = **new** String[]{Manifest.permission.***CAMERA***, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***};  
 **storagePermissions** = **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***};  
  
 *//init views* **avatarIv** = view.findViewById(R.id.***avatarIv***);  
 **fab** = view.findViewById(R.id.***fab***);  
 **coverIv** = view.findViewById(R.id.***coverIv***);  
 **nameTv** = view.findViewById(R.id.***nameTv***);  
 **emailTv** = view.findViewById(R.id.***emailTv***);  
 **noPostsBtn** = view.findViewById(R.id.***noPostsBtn***);  
 **noPostsLl** = view.findViewById(R.id.***noPostsLl***);  
  
 **postList** = **new** ArrayList<>();  
  
 *//init progress dialog* **pd** = **new** ProgressDialog(getActivity());  
  
 **noPostsBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 startActivity(**new** Intent(getContext(), UploadPostActivity.**class**));  
 }  
 });  
  
 **avatarIv**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!**avatarImage**[0]){  
 **avatarIv**.setClickable(**false**);  
 *//Edit Profile Picture* **pd**.setMessage(**"Updating Profile Picture"**);  
 **profileOrCoverPhoto** = **"image"**;  
 showImagePicDialog();  
 }  
 }  
 });  
  
 *//fab on click* **fab**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 ShowEditProfileDialog();  
 }  
 });  
  
 *//init recyclerView* **recyclerView** = view.findViewById(R.id.***posts\_recyclerView***);  
 *//set it's properties* **recyclerView**.setLayoutManager(**new** LinearLayoutManager(getActivity()));  
 **recyclerView**.setHasFixedSize(**true**);  
 retrieveUserData();  
 getAllPosts();  
 **return** view;  
 }  
  
 **private void** retrieveUserData() {  
 *//retrieve user details using email* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**).child(**user**.getUid());  
 reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 User userProfile = dataSnapshot.getValue(User.**class**);  
 **nameTv**.setText(userProfile.getName());  
 **emailTv**.setText(userProfile.getEmail());  
 **if** (userProfile.getImage()!= **null**){  
 Picasso.*get*().load(userProfile.getImage()).into(**avatarIv**);  
 **avatarImage**[0] = **true**;  
 }  
 **if** (userProfile.getCover()!= **null**){  
 Picasso.*get*().load(userProfile.getCover()).into(**coverIv**);  
 **coverImage**[0] = **true**;  
 }  
 }  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 }  
  
 **private void** ShowEditProfileDialog() {  
 */\*Show dialog containing options  
 - Edit Profile Picture  
 - Edit Cover Photo  
 - Edit Name\*/  
  
 // Options to show in dialog* String[] options = {**"Edit Profile Picture"**, **"Edit Cover Photo"**, **"Edit Name"**};  
 *//alert dialog* AlertDialog.Builder builder = **new** AlertDialog.Builder(getActivity());  
 *//set title* builder.setTitle(**"Edit Profile"**);  
 *//set items to dialog* builder.setItems(options, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *//handle dialog item click* **if** (which==0){  
 *//Edit Profile Picture* **pd**.setMessage(**"Updating Profile Picture"**);  
 **profileOrCoverPhoto** = **"image"**;  
 showImagePicDialog();  
 }  
 **else if** (which==1){  
 *//Edit Cover Picture* **pd**.setMessage(**"Updating Cover Picture"**);  
 **profileOrCoverPhoto** = **"cover"**;  
 showImagePicDialog();  
 }  
 **else if** (which==2){  
 *//Edit Profile Name* **pd**.setMessage(**"Updating Profile Name"**);  
 *//calling method and pass key "name" as param to update it's value in database* showNameUpdateDialog(**"name"**);  
 }  
 }  
 });  
 *//create and show dialog* builder.create().show();  
  
 }  
  
 **private void** showNameUpdateDialog(**final** String key) {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(getActivity());  
 builder.setTitle(**"Update "** + key);  
 *//set layout of dialog* LinearLayout linearLayout = **new** LinearLayout(getActivity());  
 linearLayout.setOrientation(LinearLayout.***VERTICAL***);  
 linearLayout.setPadding(10,10,10,10);  
 *//add edit text* **final** EditText editText = **new** EditText(getActivity());  
 editText.setHint(**"Enter "** + key);  
 linearLayout.addView(editText);  
  
 builder.setView(linearLayout);  
  
 *//add buttons in dialog to update* builder.setPositiveButton(**"Update"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *//input text from edit text* String value = editText.getText().toString().trim();  
 **if** (!TextUtils.*isEmpty*(value)){  
 **pd**.show();  
 HashMap<String, Object> result = **new** HashMap<>();  
 result.put(key, value);  
  
 **databaseReference**.child(**user**.getUid()).updateChildren(result)  
 .addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 *//updated, dismiss progress bar* **pd**.dismiss();  
 Toast.*makeText*(getActivity(), **"Updated..."**, Toast.***LENGTH\_LONG***).show();  
  
 }  
 })  
 .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 *//failed, dismiss progress bar, get and show error message* **pd**.dismiss();  
 Toast.*makeText*(getActivity(), **""**+e.getMessage(), Toast.***LENGTH\_LONG***).show();  
 }  
 });  
 }  
 **else**{  
 Toast.*makeText*(getActivity(), **"Please enter validate input"**, Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 });  
 *//add button in dialog to cancel* builder.setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.dismiss();  
 }  
 });  
 *//create and show dialog* builder.create().show();  
 }  
  
 **private void** showImagePicDialog(){  
 *//shows dialog containing options Camera and Gallery to pick image  
 // Options to show in dialog* String[] options = {**"Camera"**, **"Gallery"**};  
 String[] options2 = {**"Camera"**, **"Gallery"**, **"Remove"**};  
 **if** (**avatarImage**[0] && **profileOrCoverPhoto**.equals(**"image"**)){  
 options = options2;  
 }  
 **if** (**coverImage**[0] && **profileOrCoverPhoto**.equals(**"cover"**)){  
 options = options2;  
 }  
 *//alert dialog* AlertDialog.Builder builder = **new** AlertDialog.Builder(getActivity());  
 *//set title* builder.setTitle(**"Pick Image Option"**);  
 *//set items to dialog* builder.setItems(options, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *//handle dialog item click* **if** (which==0){  
 *//Camera* **if** (!checkCameraPermission()){  
 requestCameraPermission();  
 }  
 **else**{  
 pickFromCamera();  
 }  
 } **else if** (which==1){  
 *//Gallery* **if** (!checkStoragePermission()){  
 requestStoragePermission();  
 }  
 **else**{  
 pickFromGallery();  
 }  
 }  
 **else if** (which==2){  
 FirebaseDatabase.*getInstance*().getReference(**"Users"**).child(**user**.getUid()).child(**profileOrCoverPhoto**).removeValue();  
 **if** (**profileOrCoverPhoto**.equals(**"image"**)){  
 Picasso.*get*().load(R.drawable.***ic\_add\_image***).into(**avatarIv**);  
 }  
 **if** (**profileOrCoverPhoto**.equals(**"cover"**)){  
 **coverIv**.setImageResource(R.color.***colorPrimaryDark***);  
 }  
 }  
 }  
 });  
 *//create and show dialog* builder.create().show();  
 }  
  
 **private boolean** checkStoragePermission(){  
 */\*check if storage permission is enabled or not  
 \* return trie of enabled  
 \* return false if not enabled\*/* **boolean** result = ContextCompat.*checkSelfPermission*(getActivity(), Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***)  
 == (PackageManager.***PERMISSION\_GRANTED***);  
 **return** result;  
 }  
  
 **private void** requestStoragePermission(){  
 *//request runtime storage permission* requestPermissions(**storagePermissions**, ***STORAGE\_REQUEST\_CODE***);  
  
 }  
  
 **private boolean** checkCameraPermission(){  
 */\*check if camera permission is enabled or not  
 \* return trie of enabled  
 \* return false if not enabled\*/* **boolean** result = ContextCompat.*checkSelfPermission*(getActivity(), Manifest.permission.***CAMERA***)  
 == (PackageManager.***PERMISSION\_GRANTED***);  
  
 **boolean** result1 = ContextCompat.*checkSelfPermission*(getActivity(), Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***)  
 == (PackageManager.***PERMISSION\_GRANTED***);  
 **return** result && result1;  
 }  
  
 **private void** requestCameraPermission(){  
 *//request runtime storage permission* requestPermissions(**cameraPermissions**, ***CAMERA\_REQUEST\_CODE***);  
  
 }  
  
 @Override  
 **public void** onRequestPermissionsResult(**int** requestCode, @NonNull String[] permissions, @NonNull **int**[] grantResults) {  
 */\*This method called when user press Allow or Deny from permission request dialog  
 \* here we will handle permissions cases (allowed & denied)\*/* **switch** (requestCode){  
 **case *CAMERA\_REQUEST\_CODE***:{  
 *//picking from camera, first check if camera and storage permissions allowed or not* **if** (grantResults.**length** > 0){  
 **boolean** cameraAccepted = grantResults[0] == PackageManager.***PERMISSION\_GRANTED***;  
 **boolean** writeStorageAccepted = grantResults[1] == PackageManager.***PERMISSION\_GRANTED***;  
 **if** (cameraAccepted && writeStorageAccepted) {  
 *//permissions enabled* pickFromCamera();  
 }  
 **else**{  
 *//permissions denied* Toast.*makeText*(getActivity(), **"Please enable camera & storage permission"**, Toast.***LENGTH\_LONG***).show();  
  
 }  
 }  
 }  
 **case *STORAGE\_REQUEST\_CODE***:{  
 *//picking from gallery, first check if storage permission allowed or not* **if** (grantResults.**length** > 0){  
 **boolean** writeStorageAccepted = grantResults[1] == PackageManager.***PERMISSION\_GRANTED***;  
 **if** (writeStorageAccepted) {  
 *//permissions enabled* pickFromGallery();  
 }  
 **else**{  
 *//permissions denied* Toast.*makeText*(getActivity(), **"Please enable storage permission"**, Toast.***LENGTH\_LONG***).show();  
  
 }  
 }  
 }  
 }  
  
 **super**.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 }  
  
 **private void** pickFromCamera() {  
 ContentValues values = **new** ContentValues();  
 values.put(MediaStore.Images.Media.***TITLE***, **"Temp Pic"**);  
 values.put(MediaStore.Images.Media.***DESCRIPTION***, **"Temp Description"**);  
 *//put image uri* **image\_uri** = getActivity().getContentResolver().insert(MediaStore.Images.Media.***EXTERNAL\_CONTENT\_URI***, values);  
  
 *//intent to start camera* Intent cameraIntent = **new** Intent(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  
 cameraIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***, **image\_uri**);  
 startActivityForResult(cameraIntent, ***IMAGE\_PICK\_CAMERA\_CODE***);  
 }  
  
 **private void** pickFromGallery() {  
 *//pick from gallery* Intent galleryIntent = **new** Intent(Intent.***ACTION\_PICK***);  
 galleryIntent.setType(**"image/\*"**);  
 startActivityForResult(galleryIntent, ***IMAGE\_PICK\_GALLERY\_CODE***);  
 }  
  
 @Override  
 **public void** onActivityResult(**int** requestCode, **int** resultCode, @Nullable Intent data) {  
 */\*This method weill be called after picking image from camera or gallery\*/* **if** (resultCode == ***RESULT\_OK***){  
 **if** (requestCode == ***IMAGE\_PICK\_GALLERY\_CODE***){  
 *//image is picked from gallery, get uri of image* **image\_uri** = data.getData();  
 uploadProfileCoverPhoto(**image\_uri**);  
 }  
 **if** (requestCode == ***IMAGE\_PICK\_CAMERA\_CODE***){  
 *//image is picked from camera, get uri of image* uploadProfileCoverPhoto(**image\_uri**);  
 }  
  
 }  
 **super**.onActivityResult(requestCode, resultCode, data);  
 }  
  
 **private void** uploadProfileCoverPhoto(Uri uri) {  
 */\* instead of creating separate functions for Profile Picture and Profile Cover Photo  
 I've assigned to each option a sting value: cover, image\*/  
  
 //show progress* **pd**.show();  
  
 String filePathAndName = **storagePath** + **""** + **profileOrCoverPhoto** + **"\_"** + **user**.getUid();  
  
 StorageReference storageReference2nd = **storageReference**.child(filePathAndName);  
 storageReference2nd.putFile(uri)  
 .addOnSuccessListener(**new** OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 **public void** onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
 Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();  
 **while**(!uriTask.isSuccessful());  
 Uri downloadUri = uriTask.getResult();  
 *//check if image is uploaded or not and url is received* **if** (uriTask.isSuccessful()){  
 *//image uploaded  
 //add/update url in user's database* HashMap<String, Object> results = **new** HashMap<>();  
 *//first parameter is the key: "image" or "cover"  
 //second parameter is the value of the image url stored in the firebase storage* results.put(**profileOrCoverPhoto**, downloadUri.toString());  
 **databaseReference**.child(**user**.getUid()).updateChildren(results)  
 .addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 *//url in database of user is added successfully  
 //dismiss progress bar* **pd**.dismiss();  
 Toast.*makeText*(getActivity(), **"Image updated..."**, Toast.***LENGTH\_LONG***).show();  
 }  
 })  
 .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 *//error adding url in database of user  
 //dismiss progress bar* **pd**.dismiss();  
 Toast.*makeText*(getActivity(), **"Error updating Image..."**, Toast.***LENGTH\_LONG***).show();  
 }  
 });  
 }  
 **else**{  
 *//error* **pd**.dismiss();  
 Toast.*makeText*(getActivity(), **"Some error occurred"**, Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 })  
 .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 *//there were some errors, show error and dismiss the dialog* **pd**.dismiss();  
 Toast.*makeText*(getActivity(), e.getMessage(), Toast.***LENGTH\_LONG***).show();  
 }  
 });  
  
 }  
  
 @Override  
 **public void** onCreate(@Nullable Bundle savedInstanceState) {  
 setHasOptionsMenu(**true**); *// show menu in fragment* **super**.onCreate(savedInstanceState);  
 }  
  
 **private void** getAllPosts() {  
 *//get current user* **final** FirebaseUser fUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 *//get path of database named "Posts" containing posts info* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Posts"**);  
 *//get all data from path* reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **postList**.clear();  
 **for** (DataSnapshot ds : dataSnapshot.getChildren()) {  
 Post post = ds.getValue(Post.**class**);  
 **if** (post.getPublisherID().equals(fUser.getUid())) {  
 **postList**.add(post);  
 }  
 }  
  
 **if** (**postList**.isEmpty()){  
 **noPostsLl**.setVisibility(View.***VISIBLE***);  
 }  
  
 Collections.*reverse*(**postList**);  
  
 *//adapter* **postAdapter** = **new** PostAdapter(getActivity(), **postList**, fUser.getUid(), **"ProfileClass"**);  
 **recyclerView**.setAdapter(**postAdapter**);  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 *//inflate option menu* @Override  
 **public void** onCreateOptionsMenu(Menu menu, MenuInflater inflater) {  
 *//inflating menu* inflater.inflate(R.menu.***menu\_main***, menu);  
 MenuItem item = menu.findItem(R.id.***action\_search***);  
 item.setVisible(**false**);  
 **super**.onCreateOptionsMenu(menu, inflater);  
 }  
  
  
  
 *//handle menu item click* @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 *//get item id* **int** id = item.getItemId();  
 **if** (id==R.id.***action\_logout***){  
 **firebaseAuth**.signOut();  
 getActivity().startService(**new** Intent(getActivity(), UserService.**class**));  
 }  
 **if** (id==R.id.***action\_upload***){  
 startActivity(**new** Intent(getContext(), UploadPostActivity.**class**));  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
}

UploadPostActivity.java:

**package** com.roichomsky.socialmid;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.annotation.Nullable;  
**import** androidx.appcompat.app.ActionBar;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.core.content.ContextCompat;  
  
**import** android.Manifest;  
**import** android.app.ProgressDialog;  
**import** android.content.ContentValues;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.provider.MediaStore;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.ImageView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.tasks.OnFailureListener;  
**import** com.google.android.gms.tasks.OnSuccessListener;  
**import** com.google.android.gms.tasks.Task;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.storage.FirebaseStorage;  
**import** com.google.firebase.storage.StorageReference;  
**import** com.google.firebase.storage.UploadTask;  
  
**import** java.sql.Timestamp;  
**import** java.util.HashMap;  
  
**public class** UploadPostActivity **extends** AppCompatActivity {  
  
 *//views* Button **galleryBtn**, **cameraBtn**, **uploadBtn**;  
 ImageView **previewIv**;  
 EditText **descriptionEt**;  
  
 FirebaseAuth **firebaseAuth**;  
 FirebaseDatabase **firebaseDatabase**;  
 FirebaseUser **user**;  
  
 ActionBar **actionBar**;  
  
 *//progress dialog* ProgressDialog **pd**;  
  
 *//permissions constants* **private static final int *CAMERA\_REQUEST\_CODE*** = 100;  
 **private static final int *STORAGE\_REQUEST\_CODE*** = 200;  
 **private static final int *IMAGE\_PICK\_GALLERY\_CODE*** = 300;  
 **private static final int *IMAGE\_PICK\_CAMERA\_CODE*** = 400;  
 *//arrays of permissions to be requested* String[] **cameraPermissions**;  
 String[] **storagePermissions**;  
  
 DatabaseReference **databaseReference**;  
  
 *//storage* StorageReference **storageReference**;  
 String **storagePath** = **"Users\_Post\_Images/"**;  
  
 *//uri of picked image* Uri **image\_uri**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_upload\_post***);  
  
 **actionBar** = getSupportActionBar();  
 **actionBar**.setTitle(**"Upload New Post"**);  
 *//enable back button in actionbar* **actionBar**.setDisplayShowHomeEnabled(**true**);  
 **actionBar**.setDisplayHomeAsUpEnabled(**true**);  
  
 **firebaseAuth** = FirebaseAuth.*getInstance*();  
 **user** = **firebaseAuth**.getCurrentUser();  
 **firebaseDatabase** = FirebaseDatabase.*getInstance*();  
 **databaseReference** = **firebaseDatabase**.getReference(**"Posts"**);  
 **storageReference** = FirebaseStorage.*getInstance*().getReference();  
 *//init arrays of permissions* **cameraPermissions** = **new** String[]{Manifest.permission.***CAMERA***, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***};  
 **storagePermissions** = **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***};  
  
 **galleryBtn** = findViewById(R.id.***galleryBtn***);  
 **cameraBtn** = findViewById(R.id.***cameraBtn***);  
 **uploadBtn** = findViewById(R.id.***uploadBtn***);  
 **previewIv** = findViewById(R.id.***previewIv***);  
 **descriptionEt** = findViewById(R.id.***descriptionEt***);  
  
 **pd** = **new** ProgressDialog(**this**);  
 **pd**.setMessage(**"Uploading Post..."**);  
  
 **galleryBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if**(!checkStoragePermission()){  
 requestStoragePermission();  
 }  
 **else**{  
 pickFromGallery();  
 }  
 }  
 });  
  
 **cameraBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!checkCameraPermission()){  
 requestCameraPermission();  
 }  
 **else**{  
 pickFromCamera();  
 }  
 }  
 });  
  
 **uploadBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 uploadPostPhoto(**image\_uri**);  
 }  
 });  
 }  
  
 **private boolean** checkStoragePermission(){  
 */\*check if storage permission is enabled or not  
 \* return trie of enabled  
 \* return false if not enabled\*/* **boolean** result = ContextCompat.*checkSelfPermission*(**this**, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***)  
 == (PackageManager.***PERMISSION\_GRANTED***);  
 **return** result;  
 }  
  
 **private void** requestStoragePermission(){  
 *//request runtime storage permission* requestPermissions(**storagePermissions**, ***STORAGE\_REQUEST\_CODE***);  
  
 }  
  
 **private boolean** checkCameraPermission(){  
 */\*check if camera permission is enabled or not  
 \* return trie of enabled  
 \* return false if not enabled\*/* **boolean** result = ContextCompat.*checkSelfPermission*(**this**, Manifest.permission.***CAMERA***)  
 == (PackageManager.***PERMISSION\_GRANTED***);  
  
 **boolean** result1 = ContextCompat.*checkSelfPermission*(**this**, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***)  
 == (PackageManager.***PERMISSION\_GRANTED***);  
 **return** result && result1;  
 }  
  
 **private void** requestCameraPermission(){  
 *//request runtime storage permission* requestPermissions(**cameraPermissions**, ***CAMERA\_REQUEST\_CODE***);  
  
 }  
  
 @Override  
 **public void** onRequestPermissionsResult(**int** requestCode, @NonNull String[] permissions, @NonNull **int**[] grantResults) {  
 */\*This method called when user press Allow or Deny from permission request dialog  
 \* here we will handle permissions cases (allowed & denied)\*/* **switch** (requestCode){  
 **case *CAMERA\_REQUEST\_CODE***:{  
 *//picking from camera, first check if camera and storage permissions allowed or not* **if** (grantResults.**length** > 0){  
 **boolean** cameraAccepted = grantResults[0] == PackageManager.***PERMISSION\_GRANTED***;  
 **boolean** writeStorageAccepted = grantResults[1] == PackageManager.***PERMISSION\_GRANTED***;  
 **if** (cameraAccepted && writeStorageAccepted) {  
 *//permissions enabled* pickFromCamera();  
 }  
 **else**{  
 *//permissions denied* Toast.*makeText*(**this**, **"Please enable camera & storage permission"**, Toast.***LENGTH\_LONG***).show();  
  
 }  
 }  
 }  
 **case *STORAGE\_REQUEST\_CODE***:{  
 *//picking from gallery, first check if storage permission allowed or not* **if** (grantResults.**length** > 0){  
 **boolean** writeStorageAccepted = grantResults[1] == PackageManager.***PERMISSION\_GRANTED***;  
 **if** (writeStorageAccepted) {  
 *//permissions enabled* pickFromGallery();  
 }  
 **else**{  
 *//permissions denied* Toast.*makeText*(**this**, **"Please enable storage permission"**, Toast.***LENGTH\_LONG***).show();  
  
 }  
 }  
 }  
 }  
  
 **super**.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 }  
  
 **private void** pickFromGallery() {  
 *//pick from gallery* Intent galleryIntent = **new** Intent(Intent.***ACTION\_PICK***);  
 galleryIntent.setType(**"image/\*"**);  
 startActivityForResult(galleryIntent, ***IMAGE\_PICK\_GALLERY\_CODE***);  
 }  
  
 **private void** pickFromCamera() {  
 ContentValues values = **new** ContentValues();  
 values.put(MediaStore.Images.Media.***TITLE***, **"Temp Pic"**);  
 values.put(MediaStore.Images.Media.***DESCRIPTION***, **"Temp Description"**);  
 *//put image uri* **image\_uri** = **this**.getContentResolver().insert(MediaStore.Images.Media.***EXTERNAL\_CONTENT\_URI***, values);  
  
 *//intent to start camera* Intent cameraIntent = **new** Intent(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  
 cameraIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***, **image\_uri**);  
 startActivityForResult(cameraIntent, ***IMAGE\_PICK\_CAMERA\_CODE***);  
 }  
  
 **private void** uploadPostPhoto(Uri uri) {  
 *//show progress* **pd**.show();  
  
 String filePathAndName = **storagePath** + **""** + **"post"** + **"\_"** + uri.toString();  
 Timestamp timestamp = **new** Timestamp(System.*currentTimeMillis*());  
 StorageReference storageReference2nd = **storageReference**.child(filePathAndName);  
 storageReference2nd.putFile(uri)  
 .addOnSuccessListener(**new** OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 **public void** onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
 Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();  
 Uri downloadUri = uriTask.getResult();  
 *//check if image is uploaded or not and url is received* **if** (uriTask.isSuccessful()){  
 *//image uploaded  
 //add url in post's database* HashMap<String, Object> results = **new** HashMap<>();  
 *//first parameter is the key - post  
 //second parameter is the value of the image url stored in the firebase storage* results.put(**"postURL"**, downloadUri.toString());  
 results.put(**"description"**, **descriptionEt**.getText().toString().trim());  
 results.put(**"publisherID"**, **user**.getUid());  
 String postID = **databaseReference**.push().getKey();  
 results.put(**"postID"**, postID);  
 results.put(**"timestamp"**, Long.*toString*(timestamp.getTime()));  
 **databaseReference**.child(postID).updateChildren(results)  
 .addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 *//url in database of post is added successfully  
 //dismiss progress bar* **pd**.dismiss();  
 Toast.*makeText*(getApplicationContext(), **"Image uploaded..."**, Toast.***LENGTH\_LONG***).show();  
 finish();  
 }  
 })  
 .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 *//error adding url in database of posts  
 //dismiss progress bar* **pd**.dismiss();  
 Toast.*makeText*(UploadPostActivity.**this**, **"Error uploading Image..."**, Toast.***LENGTH\_LONG***).show();  
 }  
 });  
 }  
 **else**{  
 *//error* **pd**.dismiss();  
 Toast.*makeText*(UploadPostActivity.**this**, **"Some error occurred"**, Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 })  
 .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 *//there were some errors, show error and dismiss the dialog* **pd**.dismiss();  
 Toast.*makeText*(UploadPostActivity.**this**, e.getMessage(), Toast.***LENGTH\_LONG***).show();  
 }  
 });  
  
 }  
  
 @Override  
 **public void** onActivityResult(**int** requestCode, **int** resultCode, @Nullable Intent data) {  
 */\*This method will be called after picking image from camera or gallery\*/* **if** (resultCode == ***RESULT\_OK***){  
 **if** (requestCode == ***IMAGE\_PICK\_GALLERY\_CODE***){  
 *//image is picked from gallery, get uri of image* **image\_uri** = data.getData();  
 **if** (**image\_uri** != **null**) {  
 **previewIv**.setImageURI(**image\_uri**);  
 }  
 **else**{  
 Toast.*makeText*(getApplicationContext(), **"Please choose a photo"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 **if** (requestCode == ***IMAGE\_PICK\_CAMERA\_CODE***) {  
 *//image is picked from camera, get uri of image* **if** (**image\_uri** != **null**) {  
 **previewIv**.setImageURI(**image\_uri**);  
 }  
 **else**{  
 Toast.*makeText*(getApplicationContext(), **"Please choose a photo"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 }  
 **super**.onActivityResult(requestCode, resultCode, data);  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == android.R.id.***home***){  
 finish();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
}

User.java:

**package** com.roichomsky.socialmid;  
  
**public class** User {  
  
 *//use same name as in firebase database* **private** String **uid**;  
 **private** String **name**;  
 **private** String **email**;  
 **private** String **image**;  
 **private** String **cover**;  
  
 **public** User(){  
  
 }  
  
 **public** User(String uid, String name, String email, String search, String image, String cover) {  
 **this**.**uid** = uid;  
 **this**.**name** = name;  
 **this**.**email** = email;  
 **this**.**image** = image;  
 **this**.**cover** = cover;  
 }  
  
 **public** String getUid() {  
 **return uid**;  
 }  
  
 **public void** setUid(String uid) {  
 **this**.**uid** = uid;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
 **public** String getEmail() {  
 **return email**;  
 }  
  
 **public void** setEmail(String email) {  
 **this**.**email** = email;  
 }  
  
 **public** String getImage() {  
 **return image**;  
 }  
  
 **public void** setImage(String image) {  
 **this**.**image** = image;  
 }  
  
 **public** String getCover() {  
 **return cover**;  
 }  
  
 **public void** setCover(String cover) {  
 **this**.**cover** = cover;  
 }  
}

UserProfileActivity.java:

**package** com.roichomsky.socialmid;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.ActionBar;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.recyclerview.widget.LinearLayoutManager;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**import** android.annotation.SuppressLint;  
**import** android.os.Bundle;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.tasks.OnSuccessListener;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
**import** com.squareup.picasso.Picasso;  
  
**import** java.util.ArrayList;  
**import** java.util.Collections;  
  
**public class** UserProfileActivity **extends** AppCompatActivity {  
  
 FirebaseDatabase **firebaseDatabase**;  
 DatabaseReference **databaseReference**;  
  
 ActionBar **actionBar**;  
  
 ImageView **avatarIv**, **coverIv**;  
 TextView **nameTv**, **emailTv**;  
 Button **addBtn**;  
  
 FirebaseUser **firebaseUser**;  
  
 ArrayList<Post> **postList**;  
 PostAdapter **postAdapter**;  
 RecyclerView **recyclerView**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_user\_profile***);  
  
 *//init views* **avatarIv** = findViewById(R.id.***avatarIv***);  
 **coverIv** = findViewById(R.id.***coverIv***);  
 **nameTv** = findViewById(R.id.***nameTv***);  
 **emailTv** = findViewById(R.id.***emailTv***);  
 **addBtn** = findViewById(R.id.***addBtn***);  
  
 **final** String uid = getIntent().getStringExtra(**"uid"**);  
  
 **firebaseUser** = FirebaseAuth.*getInstance*().getCurrentUser();  
  
 **firebaseDatabase** = FirebaseDatabase.*getInstance*();  
 **databaseReference** = **firebaseDatabase**.getReference(**"Users"**).child(uid);  
  
 *//support bar* **actionBar** = getSupportActionBar();  
  
 *//set back button enabled* **actionBar**.setDisplayShowHomeEnabled(**true**);  
 **actionBar**.setDisplayHomeAsUpEnabled(**true**);  
 *//init recyclerView* **recyclerView** = findViewById(R.id.***posts\_recyclerView***);  
 *//set it's properties* **recyclerView**.setLayoutManager(**new** LinearLayoutManager(getApplicationContext()));  
 **recyclerView**.setHasFixedSize(**true**);  
  
 **postList** = **new** ArrayList<>();  
  
 **databaseReference**.addListenerForSingleValueEvent(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 User userProfile = dataSnapshot.getValue(User.**class**);  
  
 **actionBar**.setTitle(userProfile.getName());  
  
 **nameTv**.setText(userProfile.getName());  
 **emailTv**.setText(userProfile.getEmail());  
 **if** (userProfile.getImage()!= **null**){  
 Picasso.*get*().load(userProfile.getImage()).into(**avatarIv**);  
 }  
 **if** (userProfile.getCover() != **null**){  
 Picasso.*get*().load(userProfile.getCover()).into(**coverIv**);  
 }  
  
 getAllPosts(userProfile);  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
 finish();  
 }  
 });  
  
 alreadyFriend(uid, **addBtn**);  
  
 **addBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @SuppressLint(**"ResourceAsColor"**)  
 @Override  
 **public void** onClick(View v) {  
 DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**).child(**firebaseUser**.getUid())  
 .child(**"friendsList"**).child(uid);  
 **if** (!**addBtn**.getText().equals(**"Friends"**)){  
 reference.setValue(uid).addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @SuppressLint(**"ResourceAsColor"**)  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 Toast.*makeText*(getApplicationContext(), **"Added..."**, Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 }  
 **else**{  
 reference.removeValue();  
 **addBtn**.setText(**"Add Friend"**);  
 Toast.*makeText*(getApplicationContext(), **"Removed..."**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 });  
 }  
  
 **private void** getAllPosts(**final** User user) {  
 *//get path of database named "Posts" containing posts info* DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Posts"**);  
 *//get all data from path* reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **postList**.clear();  
 **for** (DataSnapshot ds : dataSnapshot.getChildren()) {  
 Post post = ds.getValue(Post.**class**);  
 **if** (post.getPublisherID().equals(user.getUid())){  
 **postList**.add(post);  
 }  
  
 Collections.*reverse*(**postList**);  
  
 *//adapter* **postAdapter** = **new** PostAdapter(UserProfileActivity.**this**, **postList**, user.getUid(), **"UserProfileActivity"**);  
 **recyclerView**.setAdapter(**postAdapter**);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == android.R.id.***home***){  
 finish();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **private void** alreadyFriend(**final** String uid, **final** Button addBtn){  
 **final boolean**[] has = {**false**};  
 **final** DatabaseReference reference = FirebaseDatabase.*getInstance*().getReference(**"Users"**).child(**firebaseUser**.getUid()).child(**"friendsList"**).child(uid);  
 reference.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 has[0] = dataSnapshot.exists();  
 **if** (has[0]){  
 addBtn.setText(**"Friends"**);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 }  
}

UserService.java:

**package** com.roichomsky.socialmid;  
  
**import** android.app.Service;  
**import** android.content.Intent;  
**import** android.os.IBinder;  
  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
  
**public class** UserService **extends** Service {  
  
 FirebaseAuth **firebaseAuth**;  
 Boolean **loggedIn** = **false**;  
  
 @Override  
 **public void** onCreate() {  
 **firebaseAuth** = FirebaseAuth.*getInstance*();  
 }  
  
 @Override  
 **public int** onStartCommand(Intent intent, **int** flags, **int** startId) {  
 checkUserStatus();  
 **if** (!**loggedIn**){  
 Intent mainIntent = **new** Intent(**this**, MainActivity.**class**);  
 mainIntent.addFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 startActivity(mainIntent);  
 }  
 **return super**.onStartCommand(intent, flags, startId);  
 }  
  
 @Override  
 **public void** onDestroy() {  
 }  
  
 @Override  
 **public** IBinder onBind(Intent intent) {  
 **return null**;  
 }  
  
 **private void** checkUserStatus(){  
 *// Get current user* FirebaseUser user = **firebaseAuth**.getCurrentUser();  
 **if** (user != **null**){  
 *//user is signed in* **loggedIn** = **true**;  
 }  
 **else** {  
 *//user is'nt signed* **loggedIn** = **false**;  
 }  
 }  
  
}

menu\_main.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"**>  
  
 <**item android:id="@+id/action\_upload"  
 android:title="Upload Post"  
 android:icon="@drawable/ic\_upload\_post"  
 app:showAsAction="always"** />  
  
 <**item android:id="@+id/action\_search"  
 android:title="Search"  
 android:icon="@drawable/ic\_menu\_search"  
 app:showAsAction="always"  
 app:actionViewClass="android.widget.SearchView"** />  
  
 <**item android:id="@+id/action\_logout"  
 android:icon="@drawable/ic\_logout"  
 app:showAsAction="always"  
 android:title="Logout"**/>  
  
</**menu**>

menu\_nav.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**menu xmlns:android="http://schemas.android.com/apk/res/android"**>  
  
 <**item android:id="@+id/nav\_home"  
 android:title="Home"  
 android:icon="@drawable/ic\_home\_black"**/>  
 <**item android:id="@+id/nav\_profile"  
 android:title="Profile"  
 android:icon="@drawable/ic\_profile\_black"**/>  
 <**item android:id="@+id/nav\_friends"  
 android:title="Friends"  
 android:icon="@drawable/ic\_users\_black"**/>  
  
</**menu**>

colors.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**resources**>  
 <**color name="colorPrimary"**>#008CBB</**color**>  
 <**color name="colorPrimaryDark"**>#007CA6</**color**>  
 <**color name="colorAccent"**>#0099cc</**color**>  
 <**color name="colorBlack"**>#000</**color**>  
 <**color name="colorWhite"**>#fff</**color**>  
 <**color name="colorGrey"**>#F1EDED</**color**>  
 <**color name="colorDarkGrey"**>#808080</**color**>  
 <**color name="greyPrimary"**>#3C7C92</**color**>  
</**resources**>

strings.xml:

<**resources**>  
 <**string name="app\_name"**>SocialMid</**string**>  
 <**string name="hello\_blank\_fragment"**>Hello blank fragment</**string**>  
</**resources**>

styles.xml:

<**resources**>  
  
 *<!-- Base application theme. -->* <**style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar"**>  
 <**item name="colorPrimary"**>@color/colorPrimary</**item**>  
 <**item name="colorPrimaryDark"**>@color/colorPrimaryDark</**item**>  
 <**item name="colorAccent"**>@color/colorAccent</**item**>  
 </**style**>  
  
 <**style name="AppThemeChat" parent="Theme.AppCompat.Light.NoActionBar"**>  
 <**item name="colorPrimary"**>@color/colorPrimary</**item**>  
 <**item name="colorPrimaryDark"**>@color/colorPrimaryDark</**item**>  
 <**item name="colorAccent"**>@color/colorAccent</**item**>  
 </**style**>  
</**resources**>

activit\_chat.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 tools:context=".ChatActivity"**>  
  
 <**androidx.appcompat.widget.Toolbar  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@id/chat\_toolbar"  
 app:contentInsetLeft="0dp"  
 app:contentInsetStartWithNavigation="0dp"  
 app:contentInsetStart="0dp"  
 android:background="@color/colorPrimary"**>  
  
 <**LinearLayout  
 android:orientation="horizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 <**com.blogspot.atifsoftwares.circularimageview.CircularImageView  
 app:c\_border="false"  
 android:layout\_marginTop="1dp"  
 android:id="@+id/avatarIv"  
 android:layout\_width="38dp"  
 android:layout\_height="38dp"  
 android:src="@drawable/ic\_default\_img"** />  
  
 <**TextView  
 android:layout\_marginTop="8dp"  
 android:layout\_marginStart="7dp"  
 android:textColor="@color/colorWhite"  
 android:textSize="18sp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/toolbar\_name"  
 android:text="NAME"** />  
 </**LinearLayout**>  
 </**androidx.appcompat.widget.Toolbar**>  
  
 <**androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerview\_message\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/chat\_toolbar"  
 android:layout\_above="@+id/layout\_chatbox"  
 android:descendantFocusability="beforeDescendants"  
 android:focusableInTouchMode="true"  
 android:focusable="true"** />  
  
 <**LinearLayout  
 android:id="@+id/layout\_chatbox"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:minHeight="48dp"  
 android:layout\_margin="10dp"  
 android:background="@drawable/chatbox\_rect"  
 android:layout\_alignParentBottom="true"**>  
  
 <**EditText  
 android:id="@+id/chatboxEt"  
 android:hint="Enter message..."  
 android:background="@android:color/transparent"  
 android:layout\_gravity="center"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:layout\_width="0dp"  
 android:layout\_weight="1"  
 android:layout\_height="wrap\_content"  
 android:maxLines="6"**/>  
  
 <**Button  
 android:id="@+id/chatbox\_sendBtn"  
 android:text="SEND"  
 android:textSize="14dp"  
 android:background="?attr/selectableItemBackground"  
 android:clickable="true"  
 android:layout\_width="64dp"  
 android:layout\_height="48dp"  
 android:gravity="center"  
 android:layout\_gravity="bottom"** />  
  
 </**LinearLayout**>  
  
</**RelativeLayout**>

activity\_comments.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".CommentsActivity"  
 android:orientation="vertical"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
  
 <**com.blogspot.atifsoftwares.circularimageview.CircularImageView  
 android:layout\_marginStart="10dp"  
 android:id="@+id/avatarIv"  
 android:src="@drawable/ic\_default\_img"  
 android:layout\_marginTop="7dp"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/usernameTv"  
 android:layout\_marginTop="7dp"  
 android:layout\_marginStart="5dp"  
 android:textColor="@color/colorBlack"  
 android:text="USERNAME"  
 android:textStyle="bold"  
 android:textSize="12sp"** />  
  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/descriptionTv"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginStart="5dp"  
 android:textColor="@color/colorBlack"  
 android:textSize="14sp"  
 android:text="THIS IS A DESCRIPTION TEXT VIEW"**/>  
  
 </**LinearLayout**>  
  
 <**androidx.recyclerview.widget.RecyclerView  
 android:background="@drawable/topline\_grey"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="5dp"  
 android:id="@+id/comments\_recyclerView"**/>  
  
</**LinearLayout**>

activity\_dashboard.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 tools:context=".DashboardActivity"**>  
  
 *<!--FrameLayout: show fragment-->* <**FrameLayout  
 android:id="@+id/content"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"**>  
 </**FrameLayout**>  
  
 *<!--Bottom Navigation: show menu-->* <**com.google.android.material.bottomnavigation.BottomNavigationView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/navigation"  
 android:layout\_gravity="bottom"  
 android:background="?android:attr/windowBackground"  
 app:menu="@menu/menu\_nav"**>  
 </**com.google.android.material.bottomnavigation.BottomNavigationView**>  
  
</**LinearLayout**>

activity\_login.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".LoginActivity"**>  
  
 *<!--TextView: Login Label-->* <**TextView  
 android:layout\_marginTop="40dp"  
 android:textAlignment="center"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Login"  
 android:textSize="25sp"  
 android:textColor="@color/colorAccent"**/>  
  
 *<!--EditText: Input Email-->* <**com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/emailTIL"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"**>  
 <**com.google.android.material.textfield.TextInputEditText  
 android:hint="Email"  
 android:id="@+id/emailEt"  
 android:inputType="textEmailAddress"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**/>  
 </**com.google.android.material.textfield.TextInputLayout**>  
  
 *<!--EditText: Input Password-->* <**com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/passwordTIL"  
 app:passwordToggleEnabled="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/emailTIL"  
 android:layout\_centerVertical="true"**>  
 <**EditText  
 android:hint="Password"  
 android:id="@+id/passwordEt"  
 android:inputType="textPassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**/>  
 </**com.google.android.material.textfield.TextInputLayout**>  
  
 *<!--Button: Login User-->* <**Button  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/passwordTIL"  
 android:text="LOGIN"  
 android:drawablePadding="5dp"  
 android:drawableStart="@drawable/ic\_login\_white"  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:id="@+id/loginBtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"**/>  
  
 *<!--TextView: Show dialog to recover password-->* <**TextView  
 android:textAlignment="center"  
 android:layout\_below="@id/loginBtn"  
 android:id="@+id/forgotPassTv"  
 android:text="Forgot Password?"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**/>  
  
 *<!--TextView: Click to go to RegisterActivity-->* <**TextView  
 android:id="@+id/noAccountTv"  
 android:textAlignment="center"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginBottom="30dp"  
 android:text="Don't have an account? Sign up."  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**/>  
  
</**RelativeLayout**>

activity\_main.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="85dp"  
 tools:context=".MainActivity"**>  
  
 <**ImageView  
 android:layout\_width="250dp"  
 android:layout\_height="250dp"  
 android:src="@drawable/icon"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="20dp"**/>  
  
 *<!--Button: Register-->* <**Button  
 android:textSize="17dp"  
 android:drawablePadding="5dp"  
 android:drawableLeft="@drawable/ic\_register\_white"  
 android:layout\_above="@id/loginBtn"  
 android:text="REGISTER"  
 android:id="@+id/registerBtn"  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"** />  
  
 *<!--Button: Login-->* <**Button  
 android:textSize="17dp"  
 android:drawablePadding="5dp"  
 android:drawableLeft="@drawable/ic\_login\_white"  
 android:text="LOGIN"  
 android:id="@+id/loginBtn"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginBottom="100dp"  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"** />  
  
</**RelativeLayout**>

activity\_upload\_post.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".UploadPostActivity"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="22sp"  
 android:layout\_marginTop="10dp"  
 android:textStyle="bold"  
 android:text="Upload Post"  
 android:layout\_gravity="center"  
 android:textColor="@color/colorBlack"**/>  
  
 <**ImageView  
 android:layout\_width="720px"  
 android:layout\_height="720px"  
 android:id="@+id/previewIv"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="10dp"  
 android:src="@color/colorDarkGrey"**/>  
  
 <**LinearLayout  
 android:layout\_marginTop="10dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_gravity="center"**>  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Gallery"  
 android:background="@drawable/underline"  
 android:id="@+id/galleryBtn"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Camera"  
 android:background="@drawable/underline"  
 android:id="@+id/cameraBtn"** />  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_marginTop="10dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Description:"  
 android:textStyle="bold"  
 android:layout\_marginStart="8dp"  
 android:textColor="@color/colorBlack"  
 android:textSize="16sp"**/>  
  
 <**EditText  
 android:layout\_marginStart="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="top"  
 android:scrollbars="vertical"  
 android:inputType="textMultiLine"  
 android:overScrollMode="always"  
 android:scrollbarStyle="insideInset"  
 android:maxLength="256"  
 android:scrollHorizontally="false"  
 android:id="@+id/descriptionEt"  
 android:hint="Write a caption..."  
 android:background="@drawable/rectangle"  
 android:textSize="16sp"**/>  
  
 </**LinearLayout**>  
  
 <**Button  
 android:id="@+id/uploadBtn"  
 android:layout\_gravity="center"  
 android:background="@drawable/rectangle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Upload"**/>  
  
</**LinearLayout**>

activity\_user\_profile.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".UserProfileActivity"  
 android:background="@color/colorGrey"**>  
  
 <**ScrollView  
 android:id="@+id/bio\_scrollView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="180dp"**>  
  
 *<!--Cover Photo-->* <**ImageView  
 android:id="@+id/coverIv"  
 android:layout\_width="match\_parent"  
 android:layout\_height="180dp"  
 android:scaleType="fitXY"  
 android:background="@color/colorPrimaryDark"**>  
  
 </**ImageView**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginTop="60dp"**>  
  
 <**ImageView  
 android:id="@+id/avatarIv"  
 android:padding="5dp"  
 android:background="@color/colorPrimary"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 android:scaleType="fitXY"  
 android:src="@drawable/ic\_default\_img\_dark"**/>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="81dp"  
 android:orientation="vertical"  
 android:background="#666F7475"**>  
 <**TextView  
 android:id="@+id/nameTv"  
 android:alpha="3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:textColor="@color/colorWhite"  
 android:layout\_marginTop="5dp"  
 android:textSize="25sp"**/>  
  
 <**TextView  
 android:id="@+id/emailTv"  
 android:alpha="3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:textColor="@color/colorWhite"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Add Friend"  
 android:background="#00000000"  
 android:id="@+id/addBtn"  
 android:textSize="16sp"** />  
  
  
 </**LinearLayout**>  
  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
 </**ScrollView**>  
  
 <**androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/posts\_recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_below="@id/bio\_scrollView"  
 android:layout\_height="wrap\_content"**/>  
  
</**RelativeLayout**>

comment\_item.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="horizontal" android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 <**com.blogspot.atifsoftwares.circularimageview.CircularImageView  
 android:layout\_marginStart="10dp"  
 android:id="@+id/avatarIv"  
 android:src="@drawable/ic\_default\_img"  
 android:layout\_marginTop="7dp"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"** />  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="7dp"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/usernameTv"  
 android:layout\_marginStart="5dp"  
 android:textColor="@color/colorBlack"  
 android:text="USERNAME"  
 android:textStyle="bold"  
 android:textSize="12sp"** />  
  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/commentTv"  
 android:layout\_marginTop="1dp"  
 android:layout\_marginStart="5dp"  
 android:textColor="@color/colorBlack"  
 android:textSize="14sp"  
 android:text="THIS IS A COMMENT TEXT VIEW"** />  
  
 </**LinearLayout**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Delete Comment"  
 android:textSize="11sp"  
 android:visibility="invisible"  
 android:id="@+id/dltTv"  
 android:textColor="@color/colorDarkGrey"  
 android:layout\_marginStart="8dp"**/>  
  
 </**LinearLayout**>  
  
</**LinearLayout**>

fragment\_friends.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".FriendsFragment"**>  
  
 <**androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/users\_recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
  
</**FrameLayout**>

fragment\_home.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".HomeFragment"**>  
  
 <**androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/posts\_recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
  
</**FrameLayout**>

fragment\_profile.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".ProfileFragment"  
 android:background="@color/colorGrey"**>  
   
 <**ScrollView  
 android:id="@+id/bio\_scrollView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="180dp"**>  
  
 *<!--Cover Photo-->* <**ImageView  
 android:id="@+id/coverIv"  
 android:layout\_width="match\_parent"  
 android:layout\_height="180dp"  
 android:scaleType="fitXY"  
 android:background="@color/colorPrimaryDark"**>  
  
 </**ImageView**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginTop="60dp"**>  
  
 <**ImageView  
 android:id="@+id/avatarIv"  
 android:padding="5dp"  
 android:background="@color/colorPrimary"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 android:scaleType="fitXY"  
 android:src="@drawable/ic\_add\_image"**/>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="81dp"  
 android:orientation="vertical"  
 android:background="#666F7475"**>  
 <**TextView  
 android:id="@+id/nameTv"  
 android:alpha="3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:textColor="@color/colorWhite"  
 android:layout\_marginTop="5dp"  
 android:textSize="25sp"**/>  
  
 <**TextView  
 android:id="@+id/emailTv"  
 android:alpha="3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:textColor="@color/colorWhite"** />  
  
  
 </**LinearLayout**>  
  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
 </**ScrollView**>  
  
 <**androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/posts\_recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_below="@id/bio\_scrollView"  
 android:layout\_height="wrap\_content"**/>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:id="@+id/noPostsLl"  
 android:visibility="invisible"  
 android:layout\_centerVertical="true"  
 android:layout\_centerHorizontal="true"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="When you share photos, they'll appear on your profile."  
 android:layout\_gravity="center"  
 android:textStyle="bold"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/noPostsBtn"  
 android:text="Share new post"  
 android:textStyle="bold"  
 android:textColor="@color/colorPrimaryDark"  
 android:layout\_gravity="center"** />  
  
 </**LinearLayout**>  
  
 *<!--Floating Action Button-->* <**com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/fab"  
 android:src="@drawable/ic\_edit\_white"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_margin="10dp"** />  
  
</**RelativeLayout**>

item\_message\_received.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="horizontal"  
 android:paddingTop="5dp"**>  
  
 <**TextView  
 android:id="@+id/text\_message\_body"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:layout\_marginTop="3dp"  
 android:background="@drawable/rectangle\_darkblue\_rounded"  
 android:maxWidth="240dp"  
 android:padding="8dp"  
 android:text="hi man, how are you?"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 android:textColor="#ffffff"** />  
  
 <**TextView  
 android:id="@+id/text\_message\_time"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="3dp"  
 android:text="11:40"  
 android:textSize="10sp"  
 app:layout\_constraintLeft\_toRightOf="@id/text\_message\_body"  
 app:layout\_constraintBottom\_toBottomOf="@id/text\_message\_body"**/>  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

item\_message\_sent.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingTop="5dp"  
 xmlns:app="http://schemas.android.com/apk/res-auto"**>  
  
 <**TextView  
 android:id="@+id/text\_message\_body"  
 android:text="hello, hello!"  
 android:background="@drawable/rectangle\_lightblue\_rounded"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:maxWidth="240dp"  
 android:padding="8dp"  
 android:textColor="#ffffff"  
 android:layout\_marginTop="3dp"  
 android:layout\_marginEnd="5dp"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"**/>  
  
 <**TextView  
 android:id="@+id/text\_message\_seen"  
 android:text="sent"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="11sp"  
 android:layout\_marginEnd="3dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/text\_message\_time"  
 app:layout\_constraintRight\_toLeftOf="@+id/text\_message\_body"** />  
  
 <**TextView  
 android:id="@+id/text\_message\_time"  
 android:text="11:40"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="10sp"  
 android:layout\_marginEnd="3dp"  
 app:layout\_constraintBottom\_toBottomOf="@+id/text\_message\_body"  
 app:layout\_constraintRight\_toLeftOf="@+id/text\_message\_body"** />  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

post\_item.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="vertical"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:paddingStart="15dp"  
 android:orientation="horizontal"**>  
  
 <**com.blogspot.atifsoftwares.circularimageview.CircularImageView  
 android:id="@+id/avatarIv"  
 android:src="@drawable/ic\_default\_img"  
 android:layout\_marginTop="7dp"  
 android:layout\_width="35dp"  
 android:layout\_height="35dp"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/usernameTv"  
 android:layout\_marginTop="16dp"  
 android:paddingStart="3dp"  
 android:paddingEnd="3dp"  
 android:textColor="@color/colorBlack"  
 android:text="USERNAME"  
 android:textStyle="bold"  
 android:textSize="12sp"** />  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"  
 android:gravity="end"**>  
  
 <**ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:id="@+id/optionIv"  
 android:layout\_marginEnd="7dp"  
 android:visibility="invisible"  
 android:src="@drawable/ic\_post\_option"**/>  
  
 </**LinearLayout**>  
  
  
  
 </**LinearLayout**>  
  
 <**ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="1080px"  
 android:id="@+id/postIv"  
 android:scaleType="fitXY"  
 android:src="@color/colorDarkGrey"** />  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:orientation="horizontal"**>  
  
 <**com.like.LikeButton  
 android:id="@+id/likeBtn"  
 android:layout\_width="35dp"  
 android:layout\_height="35dp"  
 android:layout\_marginStart="15dp"  
 android:layout\_marginTop="7dp"  
 app:icon\_size="10dp"  
 app:anim\_scale\_factor="2.5"  
 app:is\_enabled="true"  
 app:icon\_type="heart"** />  
  
 <**ImageView  
 android:layout\_width="35dp"  
 android:layout\_height="35dp"  
 android:layout\_marginStart="9dp"  
 android:id="@+id/commentIv"  
 android:layout\_marginTop="7dp"  
 android:src="@drawable/ic\_comments"** />  
  
 </**LinearLayout**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="13sp"  
 android:layout\_marginStart="15dp"  
 android:id="@+id/likesTv"  
 android:textColor="@color/colorBlack"  
 android:textStyle="bold"  
 android:text="0 likes"**/>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="3dp"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/username2Tv"  
 android:layout\_marginStart="15dp"  
 android:textColor="@color/colorBlack"  
 android:text="USERNAME"  
 android:textStyle="bold"  
 android:textSize="12sp"** />  
  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/descriptionTv"  
 android:layout\_marginStart="5dp"  
 android:textColor="@color/colorBlack"  
 android:textSize="14sp"  
 android:text="THIS IS A DESCRIPTION TEXT VIEW"**/>  
  
 </**LinearLayout**>  
  
 <**RelativeLayout  
 android:layout\_marginStart="15dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="3dp"  
 android:id="@+id/commentsTv"  
 android:textColor="#808080"  
 android:text="View all 0 comments"**/>  
  
 <**TextView  
 android:layout\_below="@+id/commentsTv"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="3dp"  
 android:textSize="11sp"  
 android:id="@+id/timeTv"  
 android:textColor="#808080"  
 android:text="00/00/00 00:00 AM"**/>  
  
 </**RelativeLayout**>  
  
</**LinearLayout**>

register\_dialog.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"**>  
  
 *<!--TextView: Label Registration-->* <**TextView  
 android:textAlignment="center"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Register"  
 android:textSize="25sp"  
 android:textColor="@color/colorAccent"**/>  
  
 *<!--EditText: Input Email-->* <**com.google.android.material.textfield.TextInputLayout  
 android:layout\_marginTop="20dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/emailTIL"  
 android:layout\_centerHorizontal="true"**>  
 <**com.google.android.material.textfield.TextInputEditText  
 android:hint="Email"  
 android:id="@+id/emailEt"  
 android:inputType="textEmailAddress"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**/>  
 </**com.google.android.material.textfield.TextInputLayout**>  
  
 *<!--EditText: Input Password-->* <**com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/passwordTIL"  
 app:passwordToggleEnabled="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/emailTIL"**>  
 <**EditText  
 android:hint="Password"  
 android:id="@+id/passwordEt"  
 android:inputType="textPassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**/>  
 </**com.google.android.material.textfield.TextInputLayout**>  
  
 *<!--EditText: Input Nickname-->* <**com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/nameTIL"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/passwordTIL"**>  
 <**com.google.android.material.textfield.TextInputEditText  
 android:hint="Nickname"  
 android:id="@+id/nameEt"  
 android:inputType="text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**/>  
 </**com.google.android.material.textfield.TextInputLayout**>  
  
 *<!--Button: Register User-->* <**Button  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/nameTIL"  
 android:text="REGISTER"  
 android:drawableStart="@drawable/ic\_register\_white"  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:id="@+id/registerBtn"  
 android:drawablePadding="5dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"**/>  
  
 *<!--TextView: Click to go to LoginActivity-->* <**TextView  
 android:layout\_marginTop="3dp"  
 android:layout\_below="@id/registerBtn"  
 android:textAlignment="center"  
 android:id="@+id/haveAccountTv"  
 android:text="Already have an account?"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**/>  
  
</**RelativeLayout**>

row\_users.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.cardview.widget.CardView  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 app:cardUseCompatPadding="true"  
 app:cardCornerRadius="2dp"  
 app:cardElevation="2dp"  
 app:cardBackgroundColor="@color/colorWhite"  
 app:contentPadding="3dp"**>  
  
 <**LinearLayout  
 android:id="@+id/ll"  
 android:orientation="horizontal"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"**>  
  
 <**com.blogspot.atifsoftwares.circularimageview.CircularImageView  
 android:id="@+id/avatarIv"  
 android:src="@drawable/ic\_default\_img"  
 android:layout\_width="70dp"  
 android:layout\_height="70dp"**/>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"**>  
  
 <**TextView  
 android:id="@+id/nameTv"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Person Name"  
 android:textColor="@color/colorBlack"  
 android:textSize="18sp"** />  
  
 <**TextView  
 android:id="@+id/emailTv"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Person Email"  
 android:textColor="@color/colorBlack"** />  
 <**TextView  
 android:id="@+id/unfriendTv"  
 android:layout\_width="110dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="3dp"  
 android:background="@drawable/rectangle"  
 android:text="Unfriend"  
 android:textAlignment="center"  
 android:textColor="@color/colorBlack"** />  
 </**LinearLayout**>  
 </**LinearLayout**>

</**androidx.cardview.widget.CardView**>