**ספר פרויקט**

**שם האפליקציה :** **FindYourSherulLeumi**

אפליקציה לחיפוש תקני שירות-לאומי למועמדים לשירות-הלאומי



**שם הסטודנט: שיר שגב**

**ת"ז, 207385592**

**מנחה: נתן דילברי**

הפרויקט מוגש כפרויקט גמר לקבלת דיפלומת הנדסאי

במגמת : תוכנה

המרכז לעיצוב וטכנולוגיה באריאל

**תוכן עניינים**

[1. מבוא 4](#_Toc115385394)

[1.1 מטרת הפרויקט 4](#_Toc115385395)

[1.2 רקע 4](#_Toc115385396)

[1.3 סקירת מצב קיים בשוק 4](#_Toc115385397)

[1.4 תיאור 4](#_Toc115385398)

[1.5 מה הפרויקט אמור לשפר או לחדש 4](#_Toc115385399)

[2. תיאור התוכנה 5](#_Toc115385400)

[2.1 תכנון מערכת 5](#_Toc115385401)

[2.2 דרישות ופונקציונאליות 5](#_Toc115385402)

[3. נתונים טכניים 5](#_Toc115385403)

[3.1 ממשקים מוצעים 5](#_Toc115385404)

[3.2 סוגי שירותים 5](#_Toc115385405)

[3.3 ציוד חומרה ותוכנה נדרשים לביצוע הפרויקט 5](#_Toc115385406)

[3.4 ציוד חומרה ותוכנה נדרשים לביצוע הפרויקט 6](#_Toc115385407)

[3.5 בעיות צפויות במהלך פיתוח ופתרונות אפשריים 6](#_Toc115385408)

[3.6 דרישות המערכת 6](#_Toc115385409)

[3.7 גבולות המערכת 6](#_Toc115385410)

[4. ארכיטקטורה ופונקציונאליות 7](#_Toc115385411)

[4.1 ארכיטקטורת המערכת 7](#_Toc115385412)

[4.2 טופולוגיית ארכיטקטורת המערכת 7](#_Toc115385413)

[5. שימוש במבני נתונים וארגון קבצים 8](#_Toc115385414)

[5.1 בסיס הנתונים 8](#_Toc115385415)

[5.2 שיטת אחסון נתונים 8](#_Toc115385416)

[5.3 מנגנוני התאוששות 8](#_Toc115385417)

[6. טכנולוגיות בשימוש 9](#_Toc115385418)

[6.1 שפות הפיתוח 9](#_Toc115385419)

[7. תרשימי מערכת מרכזיים 10](#_Toc115385420)

[USE CASE 10](#_Toc115385421)

[8. אבטחת מידע 18](#_Toc115385422)

[8.1 יעדים עיקריים 18](#_Toc115385423)

[8.2 נכסים מרכזיים 18](#_Toc115385424)

[8.3 גורמי סיכון 18](#_Toc115385425)

[8.4 אסטרטגיית הגנה 18](#_Toc115385426)

[8.5 אמצעים 18](#_Toc115385427)

[8.6 בעיות אפשרית בתהליך הזדהות 19](#_Toc115385428)

[9. בדיקות המערכת 19](#_Toc115385429)

[9.1 בדיקות תהליכיות ברמת המשתמש 19](#_Toc115385430)

[10. קוד האפליקציה 20](#_Toc115385431)

[11. סיכום תהליך הפרויקט 138](#_Toc115385432)

[11.1 תהליך הפיתוח 138](#_Toc115385433)

[11.2 משאבים הנדרשים לפרויקט 138](#_Toc115385434)

[11.3 תרשים גאנט 138](#_Toc115385435)

[12. ביבליוגרפיה 139](#_Toc115385436)

1. מבוא
   1. מטרת הפרויקט

מטרת הפרויקט היא להקל על מועמדים לשירות לאומי את תהליך חיפוש התקן המתאים להם בתוך מגוון רחב של תקני שירות לאומי ברחבי הארץ מעמותות שונות, בעזרת ריכוז כל התקנים באפליקציה אחת, בצורה נוחה, פשוטה ונגישה.

* 1. רקע

בכל שנה ישנם נערים ונערות אשר קיבלו פטור מהשירות הצבאי מסיבות שונות, שבכל זאת רוצים לתרום למדינה ע"י התנדבות בשירות הלאומי .  
קיימים כיום 8 גופים שונים המוכרים כעמותות שירות לאומי. כל עמותה אחראית על איתור, מיון, ושיבוץ המתנדבים בתקנים השונים. לאור ריבוי התקנים ומגוון אגודות השירות הלאומי, חיפוש תקן מתאים נהפך לתהליך מבלבל ומסורבל, אשר מהווה אתגר לנער/נערה, בייחוד שחיפוש התקן המתאים בד"כ מתבצע בכיתה י"ב ועליהם גם ללמוד לבגרויות.

* 1. סקירת מצב קיים בשוק

בכל שנה נפתחים עוד ועוד תקני שירות לאומי במגוון תחומים ומועמדי השירות הלאומי לא מודעים לכל התחומים ולכל התקנים האופציונליים היכולים לעניין אותם ולהתאים להם, וכן לפני שהמועמד מתקבל תפקיד ישנם מיונים לתפקידים השונים ועליו להשקיע זמן ומחשבה בתהליך.

כיום עוד לא קיימת מערכת המשלבת מאגר של תקני שירות לאומי ממספר עמותות (אלא, כל עמותה מציגה את התקנים שלה בלבד) ואין ריכוז של המידע במערכת אחת, מה שיכול להקל משמעותית למועמד את תהליך החיפוש.

* 1. תיאור

הפרויקט מאפשר לרכז את כל התקנים השונים באפליקציה אחת ולהקל למתנדב/ת את תהליך חיפוש התקן המתאים במקום אחד ממגוון תחומים שונים ומגוון עמותות.

* 1. מה הפרויקט אמור לשפר או לחדש

לאחר הבנת הצורך להקל בתהליך חיפוש תקן שירות לאומי עבור מועמדים ולהיחשף בקלות למגוון רחב יותר של תפקידים מעמותות שירות לאומי שונות, הוחלט לייצר פתרון שיאפשר לרכז את התקנים במאגר מידע אחד, התקנים יופיעו באפליקציה אחת עם הפרטים החשובים על התקן כגון: עמותה, תחום וכו' וכן, הרכז/ת שאחראי/ת על תקן יוכל להוסיף תקנים למערכת, וע"י כך תהליך חיפוש התקן המתאים יקל משמעותית על המועמד.

1. תיאור התוכנה
   1. תכנון מערכת

מבחינת סביבת הטמעה ושימוש, האפליקציה תיכתב בשפת Java בסביבת אנדרואיד, באמצעות ANDROID STUDIO.   
האפליקציה תוכל לתמוך במספר משתמשים במקביל ובכך לאפשר עבודה רציפה ומקבילית של המשתמשים.

* 1. דרישות ופונקציונאליות

האפליקציה תצטרך לתמוך בחיבור לרשת, על מנת לבצע חיבור לבסיס הנתונים MySQL לצורך אימות ושליפת נתוני הנתונים.

בנוסף, האפליקציה תצטרך לתמוך במסך מגע לצורך ניווט ושימוש בה.

1. נתונים טכניים
   1. ממשקים מוצעים

אפליקציה למכשירים בעלי מערכת הפעלה אנדרואיד.

* 1. סוגי שירותים

**שירות למשתמש מסוג מתנדב/ת** – השירות יתאפשר למשתמש שנרשם למערכת, השירות ידרוש אימות זהות משתמש ע"י login, לאחר שיאושר שהמשתמש אכן רשום למערכת והמערכת תזהה שהמשתמש הוא מתנדב, הוא יקבל בהתאם גישה לDatabase לצורך שליפת מידע וקבלת נתונים.

**שירות למשתמש מסוג רכז/ת** – השירות יתאפשר למשתמש שנרשם למערכת, השירות ידרוש אימות זהות משתמש ע"י login, לאחר שיאושר שהמשתמש אכן רשום למערכת והמערכת תזהה שהמשתמש הוא רכז/ת, הוא יקבל בהתאם גישה לDatabase לצורך שליפת מידע, קבלת נתונים והוספת מידע חדש. שירות זה מאפשר לרכז/ת להוסיף תקנים חדשים למערכת וכן, למחוק אותם בהתאם לרלוונטיות של התקן.

* 1. ציוד חומרה ותוכנה נדרשים לביצוע הפרויקט
* תוכנת Android Studio
* תוכנת Visual Studio Code
* תוכנת MySQL Workbench
  1. ציוד חומרה ותוכנה נדרשים לביצוע הפרויקט
* מכשיר בעל מערכת הפעלה אנדרואיד / Emulator להרצת מערכת Android
* מחשב PC
* גישה לאינטרנט
  1. בעיות צפויות במהלך פיתוח ופתרונות אפשריים

|  |  |
| --- | --- |
| **בעיה** | **פתרון** |
| בעיות חיבור בין בסיס הנתונים בצד הלקוח | אימות הנתונים מול השרת |
| בעיות אימות נתוני המשתמש של הלקוח אל מול השרת | ווידוא כי מערכת שחזור הסיסמאות עובדת תקין |
| בעיות בזמן בניית ותכנון בסיס הנתונים מבחינת מבנה , חיבור בין בסיס הנתונים שמכיל את נתוני התקנים ושליפתם לכל רכז/ת ספציפי | בדיקת הדרך הנכונה ביותר לבניית בסיס הנתונים, כולל ארכיטקטורה, בדיקה שהמידע מסודר כצורך ולפי הקריטריונים הרצויים והתייעצות עם המנחה |
| בעיות של ביצועים של התוכנה כלומר איטיות, תקיעות, קריסות וכדומה | בדיקת הקוד הכולל של האפליקציה לוודא סדר בקוד וכתיבה נכונה ושימוש בבקשות HTTP המחזירות קוד שגיאה במקרה של תקלה כלשהי |
| בעיית העברת מידע בפורמט אחיד בין מערכות שונות המתבססות על שפות תכנות שונות | העברת נתונים בין השרתים כאובייקטי JSON על מנת ליצור פורמט אחיד |

* 1. דרישות המערכת

**צד הלקוח:** מכשיר סמארטפון בעל מערכת הפעלה אנדרואיד.

**צד שרת:** NodeJS וMySQL לDatabase

* 1. גבולות המערכת

לאפליקציה נדרש קישור לרשת האינטרנטית על מנת לגשת לשרת ולכן, ללא קישור לאינטרנט לא ניתן יהיה לגשת לנתונים.

1. ארכיטקטורה ופונקציונאליות
   1. ארכיטקטורת המערכת

מכיוון שהמערכת כוללת בתוכה אובייקטים רבים, לוגיקות ומניפולציות על אותם אובייקטים, אחסון של אובייקטים שונים במסד הנתונים ,שליפתם בעת הצורך, הצגתם למשתמש ועוד חלקי קוד רבים, לכן ישנו צורך לבנות את המערכת מעוצבת על ידי ארכיטקטורה, שתקל על מהלך הפיתוח המסועף.

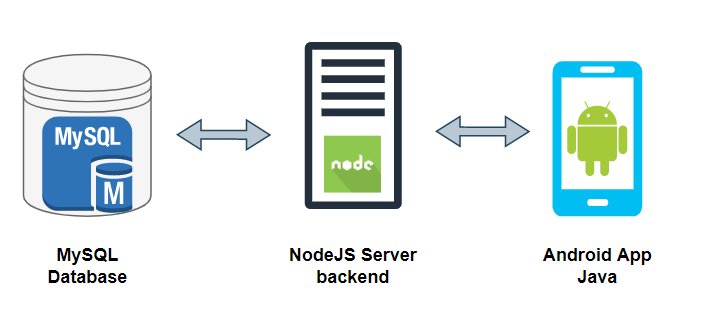
מודל שרת-לקוח הוא ארכיטקטורת תוכנה המפרידה בין שכבה ושכבה, כך שלכל שכבה יש גישה רק לשכבה שמעליה או מתחתיה. כך יותר קל לנהל את עבודת הפרויקט בצורה יעילה ומסודרת. בנוסף לכך במידה ויהיה צורך לבצע שינויים באחת מן השכבות לא יהיה צורך לשנות את הפרויקט כולו וכן ארכיטקטורת זו תאפשר החלפה או שדרוג בעתיד של כל אחת מהשכבות באופן בלתי תלוי בשכבות האחרות, בהתאם לדרישות או לשינויים בטכנולוגיה.

מבנה המערכת:

צד שרת + מסד נתונים

צד לקוח

* 1. טופולוגיית ארכיטקטורת המערכת

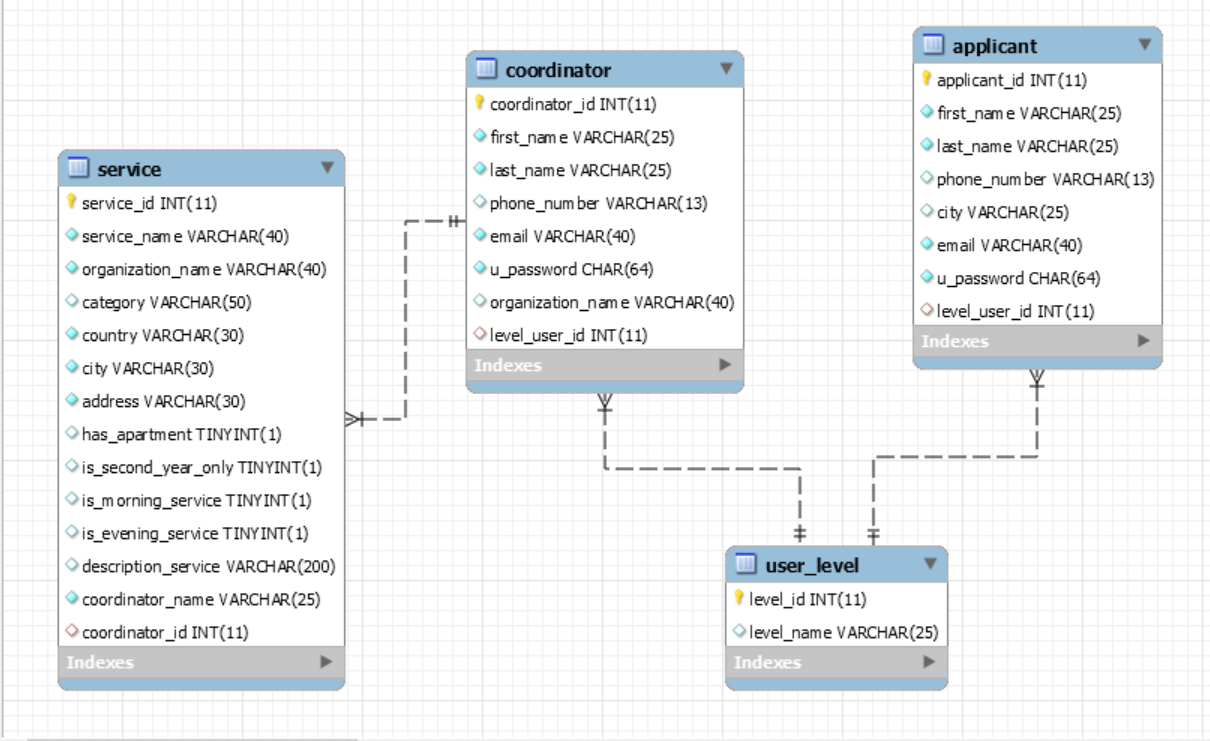
****

1. שימוש במבני נתונים וארגון קבצים
   1. בסיס הנתונים

ימומש באמצעות MySQL בצד השרת

בסיס הנתונים של המערכת יכיל 4 טבלאות:

1. טבלת תקני שירות לאומי.
2. טבלה עבור מועמדים.
3. טבלת עבור רכזים.
4. טבלת רמות משתמשים.



* 1. שיטת אחסון נתונים

הנתונים ישמרו בסיס הנתונים MySQL

* 1. מנגנוני התאוששות

המידע יגובה על בסיס יומי

1. טכנולוגיות בשימוש

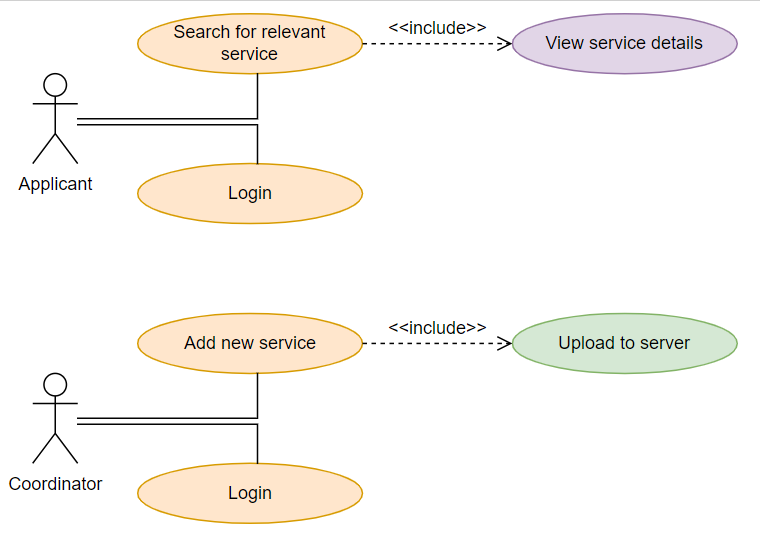
* **Android Studio -** הינה סביבת עבודה (IDE – Integrated Development Environment) של Google לפיתוח אפליקציות למערכת ההפעלה Android שנבנתה על בסיס IntelliJ IDEA ותוכננה לפיתוח במערכת Android.

פיתוח האפליקציות ניתן לביצוע בשפות Java או Kotlin (בפרויקט הנ"ל הפיתוח התבצע בשפת Java).

* **שרת Node.js -** Node.js הוא מפרש לשפת JavaScript שנכתב ב-C++ ומבוסס על מנוע V8. הוא תוכנן לכתיבת אפליקציות רשת כגון שרתי HTTP. Node.js בניגוד לJavaScript אינו מורץ בצד הלקוח, אלא בצד השרת.
* **MySQL –** הוא מסד נתונים טבלאי, מבוסס שפת SQL‏ (Structured Query Language). התוכנה פותחה במקור ע"י חברת MySQL AB אך כיום בבעלות חברת Oracle. בMySQL משתמשות מערכות ניהול תוכן רבות כבסיס נתונים.
  1. שפות הפיתוח
* **Java -** שפת תכנות מונחית עצמים. שפת Java משמשת לפיתוח תוכנות ואפליקציות הן למחשב והן למכשירי סלולר.
* **XML (eXtensible Markup Language) -** תקן לקידוד וייצוג נתונים ומידע באופן טקסטואלי, תקן ה-XML מגדיר כיצד לייצג מידע, בפורמט שניתן לקריאה ע"י אדם וע"י מכונה. ייצוג המידע מתבצע תוך שימוש בתגיות שמגדירות רכיבי מידע.

1. תרשימי מערכת מרכזיים

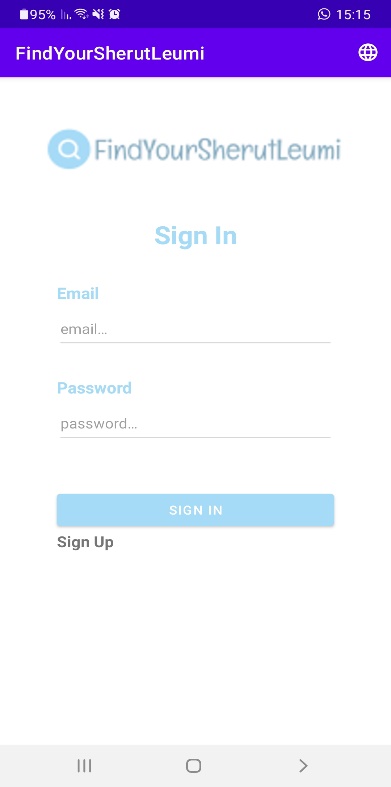
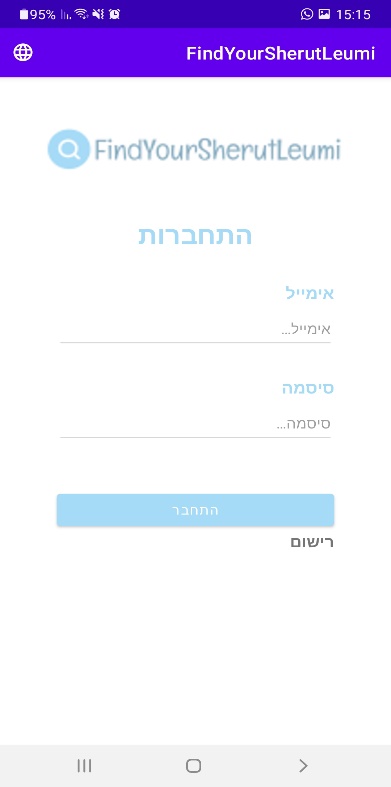
USE CASE **6.1**



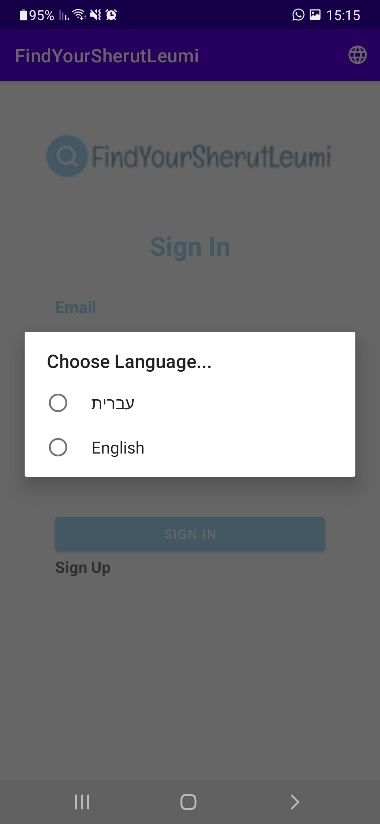
**6.2** ממשקי מערכת

**דף כניסה – עברית/אנגלית**

לצורך כניסה למערכת יש להקיש כתובת אימייל וסיסמה

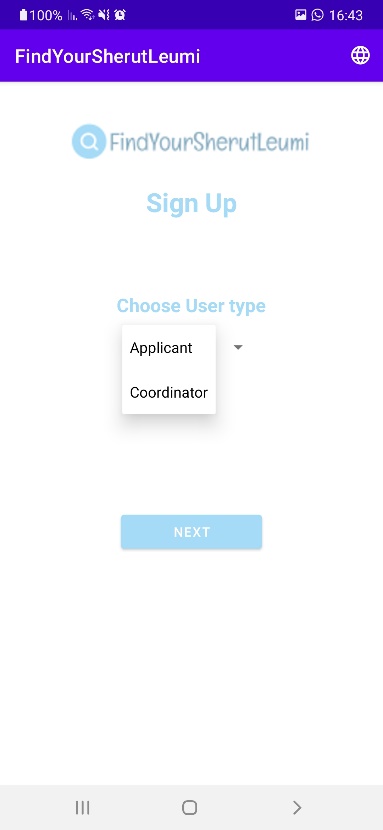


לאחר לחיצה על מקש החלפת שפה – ניתן לבחור עברית/אנגלית

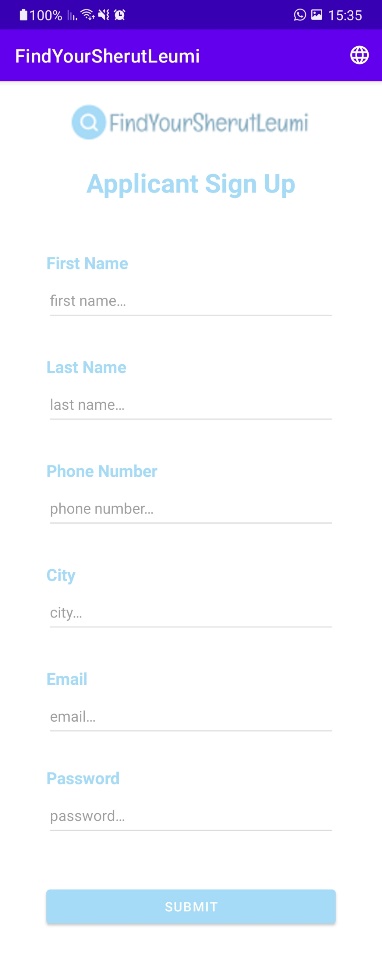
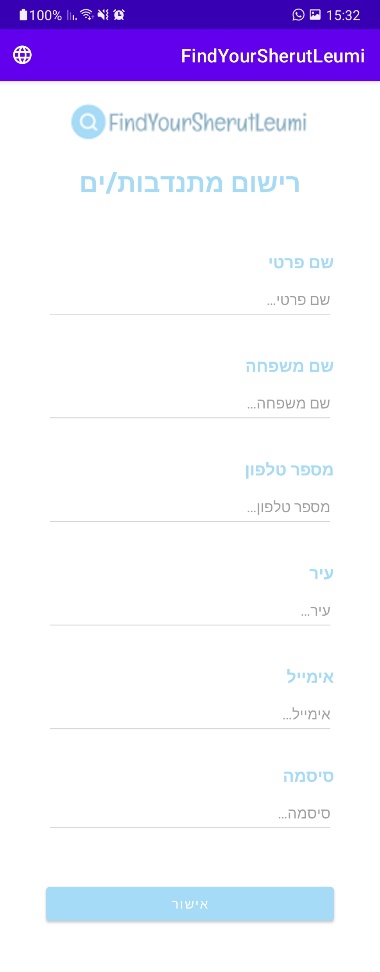
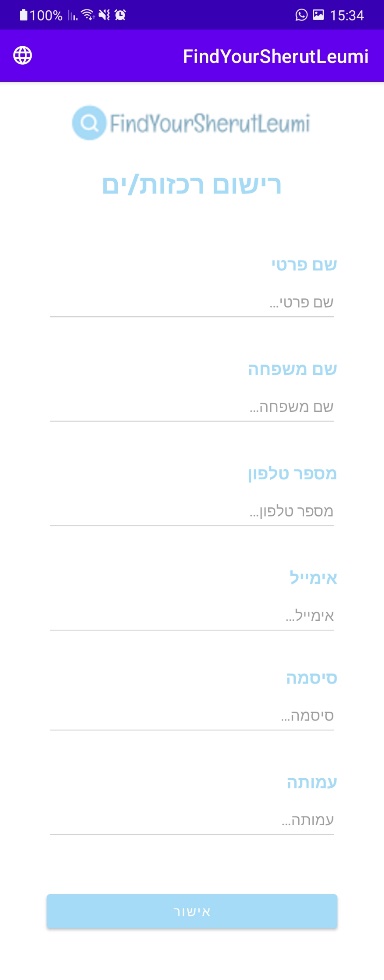
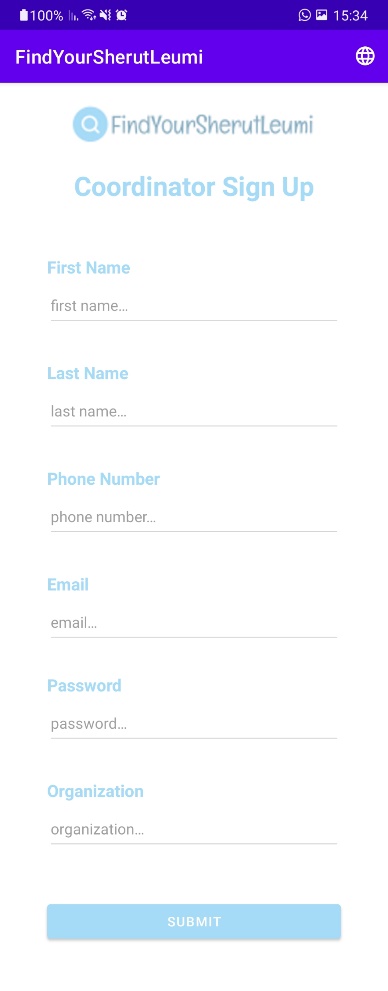


**דף רישום למערכת**

ניתן להירשם למערכת לאחר בחירת סוג הרשמה

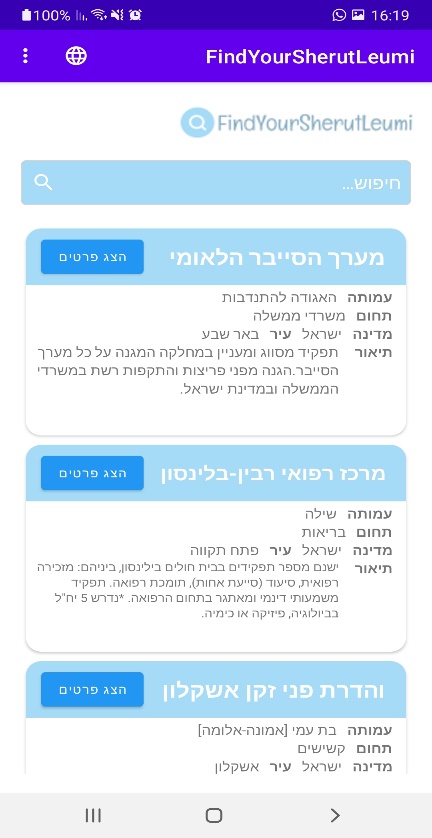
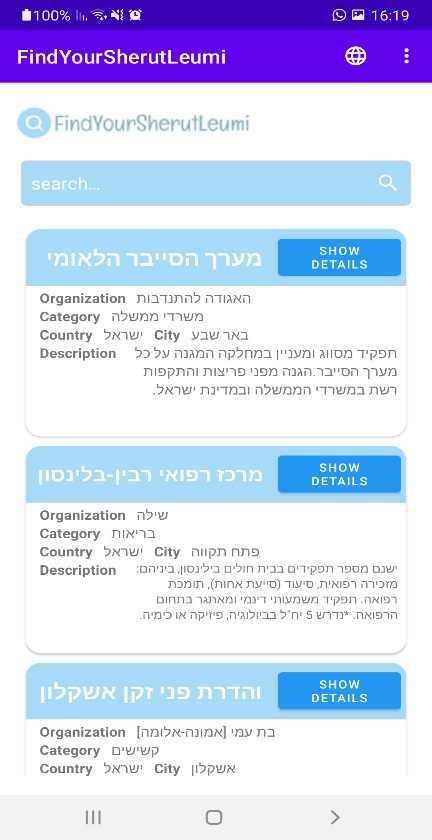


עמוד מילוי פרטים לאחר בחירת סוג משתמש



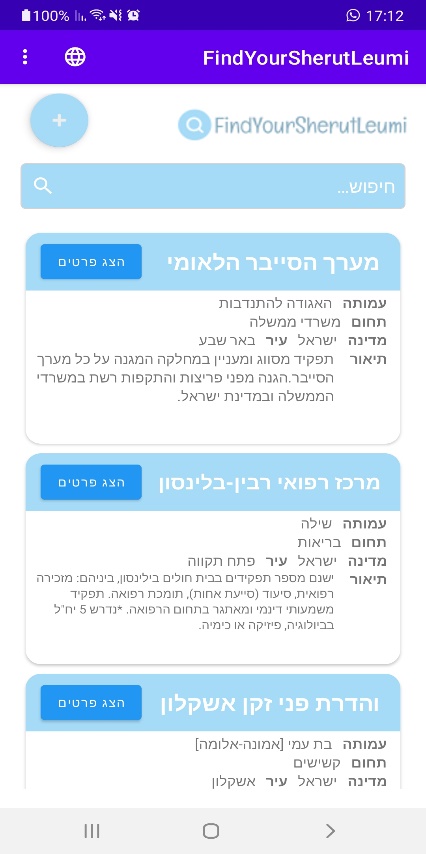
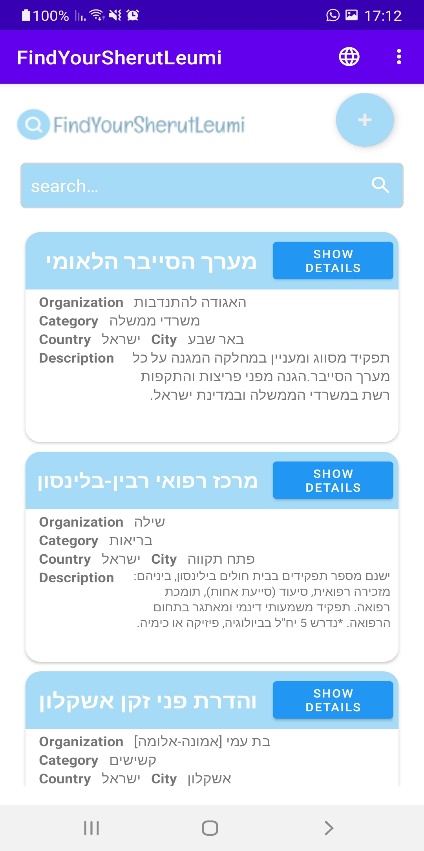
**העמוד הראשי – מתנדב/ת**

התקנים הזמינים באפליקציה



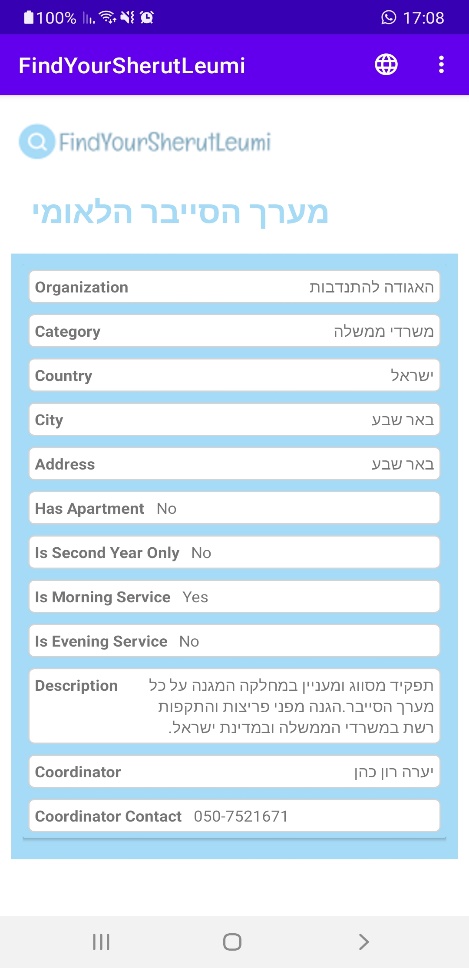
**העמוד הראשי – רכז/ת**

התקנים הזמינים באפליקציה

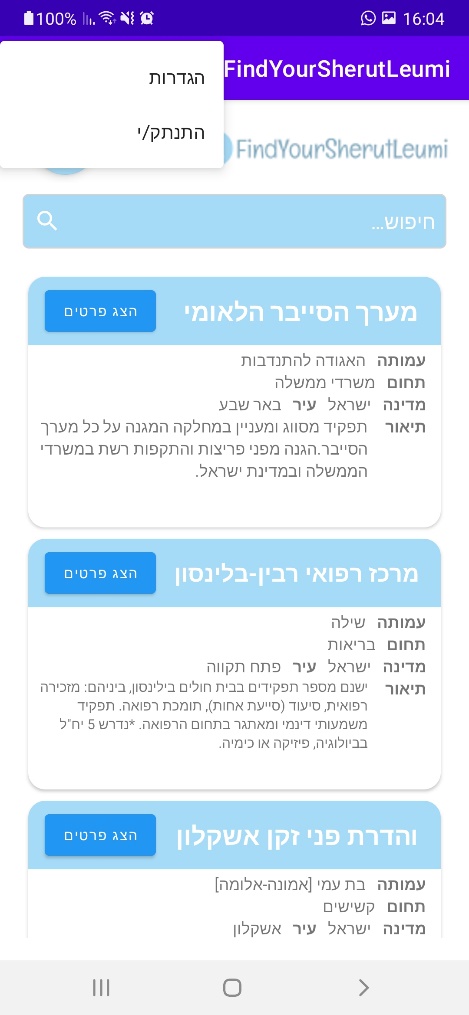


**עמוד פרטי תקן**

הפרטים המלאים של אותו תקן

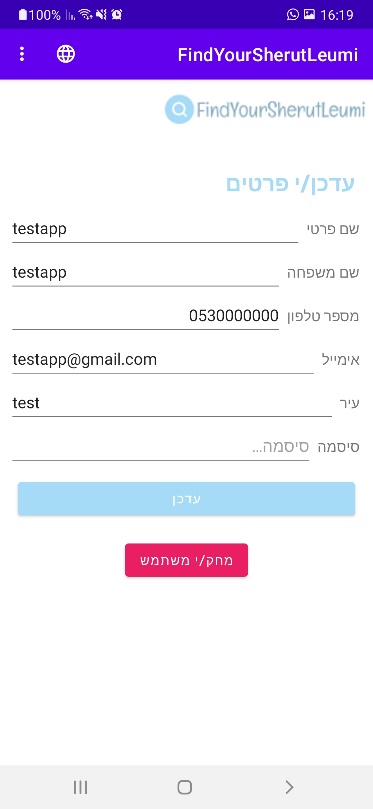
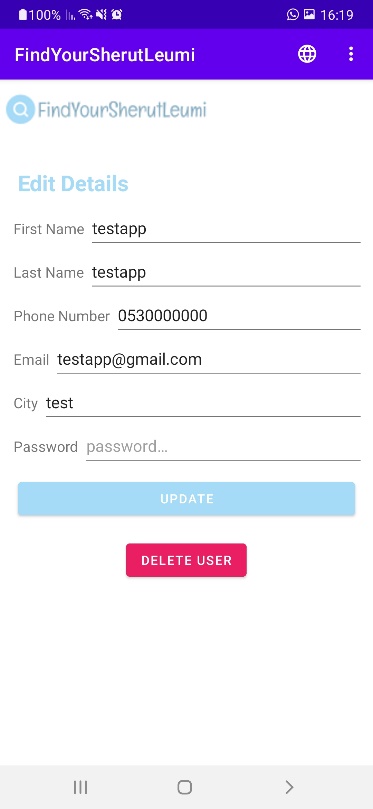


לאחר לחיצה על התפריט בראש האפליקציה – ניתן לגשת להגדרות או להתנתק מהמערכת



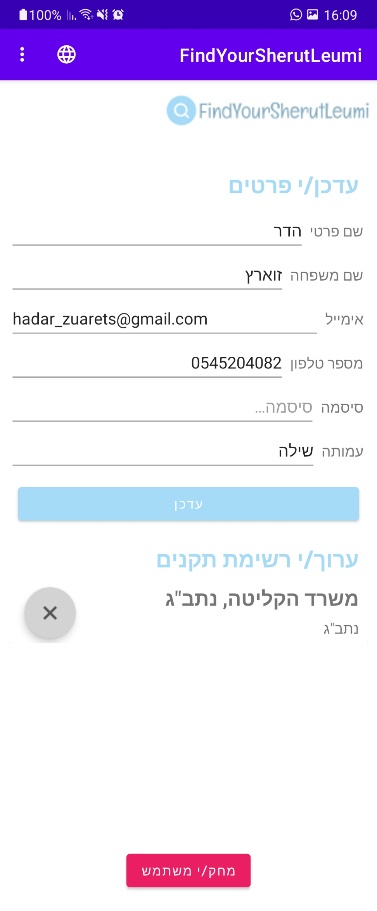
**עמוד הגדרות – מתנדב/ת**

ניתן לערוך פרטים אישיים ולמחוק את החשבון



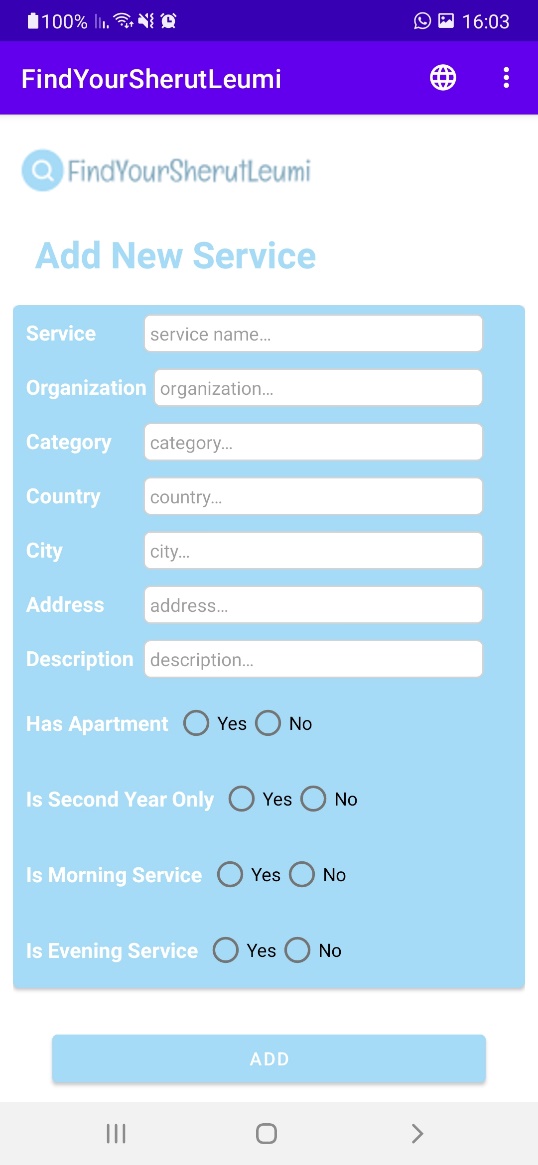
**עמוד הגדרות – רכז/ת**

ניתן לערוך פרטים אישיים, למחוק תקן או למחוק את החשבון



**עמוד הוספת תקן**

הרכז/ת ממלא/ת את כל השדות הנדרשים על מנת להוסיף תקן חדש למערכת



1. אבטחת מידע

החלקים בהם נדרשת אבטחה, סיסמאות המשתמשים שיוכנסו בעת הרשמה יוצפנו בקוד השרת בשיטת BYCRYPT, לאחר מילוי פרטי ההרשמה ואישורם, הסיסמא תעבור הצפנה וכך תישמר במסד הנתונים. כאשר המשתמש יתחבר למערכת, הוא יכניס את סיסמתו המקורית אשר תעבור שוב הצפנה, וסיסמא זו תיבדק מול הסיסמא המוצפנת ששמורה במסד הנתונים.

* 1. יעדים עיקריים

הגבלת גישה לאלו הלא רשאים.

שמירה על יעילות וזמינות הגישה אל המידע.

* 1. נכסים מרכזיים

חיסיון המידע - המידע יהיה נגיש לגורם מורשה בלבד וכן, מערכת המידע יהיו זמינים בהתאם לרמת המשתמש שהוגדרה בעת ההרשמה למערכת.

* 1. גורמי סיכון

**טכנולוגיות** – מערכות החומרה, התוכנה והתקשורת המהוות "מערכות".

**אנשים** – המשתמשים בטכנולוגיות מהווים חוליה חלשה במיוחד, מכיוון שהם מפעילים או משתמשים בטכנולוגיות, ונוטים לעשות שגיאות המייצרות פרצות במערך ההגנה אשר מאפשרות לתוקף לנצלן.

**תהליכים** – מידת הנזק משגיאת אנוש או נקודת תורפה טכנולוגית עשויה לִהצטמצם מאוד אם מבוצע תכנון נכון של התהליכים בארגון. הגדרת תהליכים בצורה נכונה מאפשר זיהוי מוקדם של פגיעה או ניסיון פגיעה במערכות.

* 1. אסטרטגיית הגנה

על מנת לאבטח את פרטיות המשתמשים, על כל משתמש ליצור פרופיל המכיל שם משתמש וסיסמא. הסיסמה נשמרת מוצפנת ובעת התחברות למערכת, מתבצע אימות בין הקלט לבין המידע השמור במערכת, רק לאחר אימות הפרטים, המשתמש יקבל גישה לאפליקציה.

* 1. אמצעים

**הרשאות** - מתן אפשרות למשתמש מסוים לבצע פעולה נתונה במערכת מידע. הפעולות עשויות להיות יצירה, כתיבה, מחיקה או שינוי של קובץ או כל פעולה אחרת שהמערכת תוכננה להכיל. מערכת הרשאות מתבססת על הזדהות כתנאי מקדים, שכן המערכת נדרשת לקשר בין משתמש לפעולה.

**הזדהות** - תהליך בו משתמש (אדם או שירות ממוחשב) מספק למערכת מידע פריט מידע המזהה אותו ואמצעי לווידוא הזיהוי. האמצעי הנפוץ ביותר הוא השימוש בשם משתמש וסיסמה.

* 1. בעיות אפשרית בתהליך הזדהות

אובדן יכולת התחברות (לרוב איבוד סיסמה או שם משתמש).

אובדן תקשורת עם השרת.

שינוי הסיסמה על ידי פריצה.

1. בדיקות המערכת
   1. בדיקות תהליכיות ברמת המשתמש

|  |  |
| --- | --- |
| **פעולה לביצוע** | **תוצאה** |
| פתיחת האפליקציה | הצגת עמוד התחברות |
| בחירת שפה של האפליקציה | הצגת חלון שפות ושינוי שפה בהתאם |
| רישום לאפליקציה | הצגת עמוד התחברות |
| התחברות לאפליקציה | הצגת עמוד ראשי |
| צפייה בפרטים מלאים של תקן מסוים | הצגת פרטים מלאים של התקן הנבחר |
| הוספת תקן חדש ממשתמש מסוג רכז/ת | הצגת עמוד הוספת תקן חדש |
| מחיקת תקן ממשתמש מסוג רכז/ת | תקן לא יופיע יותר ברשימת התקנים של הרכז/ת |
| עדכון פרטים אישיים | שינוי הפרטים האישיים והצגתם בהתאם |

* **דרישה:** על המערכת להיות נגישה אך ורק למשתמשים הרשומים למערכת דרך התחברות מאובטחת.
* **בדיקה:** לחיצה על כפתור התחברות כאשר שדות הקלט ריקים / מלאים בפרטים שלא רשומים במערכת, ההתחברות תיכשל והמשתמש לא יעבור לדף הראשי.
* **דרישה:** על המערכת לזהות תוכן ריק בשדות קלט שחובה להזין בהם קלט.
* **בדיקה:** לחיצה על כפתור הוספת תקן כאשר שדות הקלט ריקים, ההוספה תיכשל והתקן לא יתווסף למאגר.

1. קוד האפליקציה

**10.1** צד שרת

**10.1.1** עץ תיקיות

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

**10.1.2** קבצי JS

**config**

config.js

const mysql = require("mysql2");

// creating a connection with the db

const pool = mysql.createPool({

  host: process.env.DB\_HOST,

  user: process.env.DB\_USER,

  database: process.env.DB\_NAME,

  password: process.env.DB\_PASSWORD,

});

module.exports = pool.promise();

**controllers**

applicantController.js

const pool = require("../config/db");

const bcrypt = require("bcrypt");

// GET - gets all applicants from db

exports.getAllApplicants = async (req, res, next) => {

  try {

    const [applicants] = await pool.execute("CALL get\_all\_applicants()");

    res.status(200).json(applicants[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// gets specific applicant from db

exports.getApplicantById = async (req, res, next) => {

  const id = req.params.id;

  try {

    const [result, \_] = await pool.execute("CALL get\_applicant\_by\_id(?)", [

      id,

    ]);

    const applicant = result[0][0];

    res.status(200).json(applicant);

    console.log(applicant[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// gets specific applicant from db by email

exports.getApplicantByEmail = async (req, res, next) => {

  const email = req.params.email;

  try {

    const [result, \_] = await pool.execute("CALL get\_applicant\_by\_email(?)", [

      email,

    ]);

    const applicant = result[0][0];

    res.status(200).json(applicant);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// POST - post new applicant to the db

exports.createNewApplicant = async (req, res, next) => {

  const {

    first\_name,

    last\_name,

    phone\_number,

    city,

    email,

    u\_password,

  } = req.body;

  try {

      const hash\_password = await bcrypt.hash(u\_password, 12);

      const applicant = await pool.execute(

      "CALL add\_new\_applicant(?,?,?,?,?,?)",

      [first\_name, last\_name, phone\_number, city, email, hash\_password]

    );

    res.status(201).json({ applicant });

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// PUT - put new update to a applicant from the db

exports.updateApplicantById = async (req, res, next) => {

  const id = req.params.id;

  const {

    first\_name,

    last\_name,

    phone\_number,

    city,

    email,

    u\_password,

  } = req.body;

  try {

    let hash\_password = u\_password;

    console.log("hello: " + u\_password.length);

    if(u\_password.length != 60){

      hash\_password = await bcrypt.hash(u\_password, 12);

    }

    const applicant = await pool.execute(

      "CALL update\_applicant(?,?,?,?,?,?,?)",

      [id, first\_name, last\_name, phone\_number, city, email, hash\_password]

    );

    res.status(201).json(applicant[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// DELETE - delete applicant from db by id

exports.deleteApplicantById = async (req, res, next) => {

  const id = req.params.id;

  try {

    const deleted = await pool.execute("CALL delete\_applicant(?)", [id]);

    res.sendStatus(200);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

coordinatorController.js

const pool = require("../config/db");

const bcrypt = require("bcrypt");

// GET - gets all coordinators from db

exports.getAllCoordinators = async (req, res, next) => {

  try {

    const [coordinators] = await pool.execute("CALL get\_all\_coordinators()");

    res.status(200).json(coordinators[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// gets specific coordinator from db

exports.getCoordinatorById = async (req, res, next) => {

  const id = req.params.id;

  try {

    const [result, \_] = await pool.execute(

      "CALL get\_coordinator\_by\_id(?)",

      [id]

    );

    const coordinator = result[0][0];

    res.status(200).json(coordinator);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

exports.getCoordinatorByEmail = async (req, res, next) => {

  const email = req.params.email;

  try {

    const [result, \_] = await pool.execute(

      "CALL get\_coordinator\_by\_email(?)",

      [email]

    );

    const coordinator = result[0][0];

    console.log(coordinator);

    res.status(200).json(coordinator);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// POST - post new coordinator to the db

exports.createNewCoordinator = async (req, res, next) => {

  const {

    first\_name,

    last\_name,

    phone\_number,

    email,

    u\_password,

    organization\_name,

  } = req.body;

  try {

    const hash\_password = await bcrypt.hash(u\_password, 12);

    const coordinator = await pool.execute(

      "CALL add\_new\_coordinator(?,?,?,?,?,?)",

      [

        first\_name,

        last\_name,

        phone\_number,

        email,

        hash\_password,

        organization\_name,

      ]

    );

    res.status(201).json({ coordinator });

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// PUT - put new update to a coordinator from the db

exports.updateCoordinatorById = async (req, res, next) => {

  const id = req.params.id;

  const {

    first\_name,

    last\_name,

    phone\_number,

    email,

    u\_password,

    organization\_name,

  } = req.body;

  try {

    let hash\_password = u\_password;

    console.log("hello: " + u\_password.length);

    if(u\_password.length != 60){

      hash\_password = await bcrypt.hash(u\_password, 12);

    }

    const coordinator = await pool.execute(

      "CALL update\_coordinator(?,?,?,?,?,?,?)",

      [

        id,

        first\_name,

        last\_name,

        phone\_number,

        email,

        hash\_password,

        organization\_name,

      ]

    );

    res.status(201).json(coordinator[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// DELETE - delete coordinator from db by id

exports.deleteCoordinatorById = async (req, res, next) => {

  const id = req.params.id;

  try {

    const deleted = await pool.execute("CALL delete\_coordinator(?)", [id]);

    res.sendStatus(200);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

generalController.js

const pool = require("../config/db");

const bcrypt = require("bcrypt");

// gets user from db by email

exports.getUserByEmail = async (req, res, next) => {

    const email = req.body.email;

    try {

      const [user, \_] = await pool.execute(

        "CALL get\_user\_by\_email(?)",

        [email]

      );

      res.status(200).json(user[0]);

    } catch (err) {

      console.log(err);

      next(err);

    }

  };

exports.authenticateUser = async (req, res, next) => {

  const email = req.body.email;

  const password = req.body.u\_password;

  try {

    const [result, \_] = await pool.execute("CALL get\_user\_by\_email(?)", [email]);

    const user = result[0][0];

    console.log(user);

    if(user != undefined){

        const isvalid = await bcrypt.compare(password, user.u\_password).then((valid) => {

          if(valid){

              res.status(200).json(user);

          } else {

              res.status(401).json("Email and password does not match");

              console.log("code 401");

          }

        })

      .catch((err) => next(err));

    }

    else res.status(400).json("Email does not exist");

  } catch (err) {

    console.log(err);

    next(err);

  }

};

serviceController.js

const pool = require("../config/db");

// GET - gets all services from db

exports.getAllServices = async (req, res, next) => {

  try {

    const [services] = await pool.execute("CALL get\_all\_services()");

    res.status(200).json(services[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// gets all services partialy from db

exports.getAllServicesPartially = async (req, res, next) => {

  try {

    const [services] = await pool.execute("CALL get\_all\_services\_partially()");

    res.status(200).json(services[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// gets specific service from db

exports.getServiceById = async (req, res, next) => {

  const id = req.params.id;

  try {

    const [result, \_] = await pool.execute("CALL get\_service\_by\_id(?)", [id]);

    const service = result[0][0];

    res.status(200).json(service);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// gets specific service from db

exports.getServicesByCoordinatorId = async (req, res, next) => {

  const id = req.params.id;

  try {

    const [service, \_] = await pool.execute("CALL get\_service\_by\_coordinator\_id(?)", [id]);

    res.status(200).json(service[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// POST - post new service to the db

exports.createNewService = async (req, res, next) => {

  const {

    service\_name,

    organization\_name,

    category,

    country,

    city,

    address,

    has\_apartment,

    is\_second\_year\_only,

    is\_morning\_service,

    is\_evening\_service,

    description\_service,

    coordinator\_name,

    coordinator\_id

  } = req.body;

  try {

    const service = await pool.execute(

      "CALL add\_new\_service(?,?,?,?,?,?,?,?,?,?,?,?,?)",

      [

        service\_name,

        organization\_name,

        category,

        country,

        city,

        address,

        has\_apartment,

        is\_second\_year\_only,

        is\_morning\_service,

        is\_evening\_service,

        description\_service,

        coordinator\_name,

        coordinator\_id

      ]

    );

    res.status(201).json({ service });

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// PUT - put new update to a service from the db

exports.updateServiceById = async (req, res, next) => {

  const id = req.params.id;

  const {

    service\_name,

    organization\_name,

    category,

    country,

    city,

    address,

    has\_apartment,

    is\_second\_year\_only,

    is\_morning\_service,

    is\_evening\_service,

    description\_service,

    coordinator\_name,

  } = req.body;

  try {

    const service = await pool.execute(

      "CALL update\_service(?,?,?,?,?,?,?,?,?,?,?,?,?)",

      [

        id,

        service\_name,

        organization\_name,

        category,

        country,

        city,

        address,

        has\_apartment,

        is\_second\_year\_only,

        is\_morning\_service,

        is\_evening\_service,

        description\_service,

        coordinator\_name,

      ]

    );

    res.status(201).json(service[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// DELETE - delete service from db by id

exports.deleteServiceById = async (req, res, next) => {

  const id = req.params.id;

  try {

    const deleted = await pool.execute("CALL delete\_service(?)", [id]);

    res.sendStatus(200);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// delete all services from db by coordinator id

exports.deleteServicesByCoordinatorId = async (req, res, next) => {

  const id = req.params.id;

  try {

    const deleted = await pool.execute("CALL delete\_services\_by\_coordinator\_id(?)", [id]);

    res.sendStatus(200);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

userLevelController.js

const pool = require("../config/db");

// GET - gets all levels from db

exports.getAllLevels = async (req, res, next) => {

  try {

    const [levels] = await pool.execute("CALL get\_all\_levels()");

    res.status(200).json(levels[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// gets specific level from db

exports.getLevelById = async (req, res, next) => {

  const id = req.params.id;

  try {

    const [result, \_] = await pool.execute("CALL get\_level\_by\_id(?)", [id]);

    const level = result[0][0];

    res.status(200).json(level);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// POST - post new level user to the db

exports.createNewLevelUser = async (req, res, next) => {

  const { level\_name } = req.body;

  try {

    const level = await pool.execute("CALL add\_new\_level(?)", [level\_name]);

    res.status(201).json({ level });

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// PUT - put new update to a level user from the db

exports.updateLevelById = async (req, res, next) => {

  const id = req.params.id;

  const { level\_name } = req.body;

  try {

    const level = await pool.execute("CALL update\_level(?,?)", [

      id,

      level\_name,

    ]);

    res.status(201).json(level[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

// DELETE - delete level from db by id

exports.deleteLevelById = async (req, res, next) => {

  const id = req.params.id;

  try {

    const deleted = await pool.execute("CALL delete\_level(?)", [id]);

    res.status(200).json(deleted[0]);

  } catch (err) {

    console.log(err);

    next(err);

  }

};

**routes**

applicantRoute.js

const express = require("express");

const applicantController = require("../controllers/applicantController");

const router = express.Router();

// GET

router.route("/").get(applicantController.getAllApplicants);

router.route("/email/:email").get(applicantController.getApplicantByEmail); // GET by email

router

  .route("/:id")

  .get(applicantController.getApplicantById) // GET by id

  .put(applicantController.updateApplicantById) // PUT by id

  .delete(applicantController.deleteApplicantById); // DELETE by id

// POST

router.post("/add", applicantController.createNewApplicant);

module.exports = router;

coordinatorRoute.js

const express = require("express");

const coordinatorController = require("../controllers/coordinatorController");

const router = express.Router();

// GET

router.route("/").get(coordinatorController.getAllCoordinators);

router.route("/email/:email").get(coordinatorController.getCoordinatorByEmail);

router

  .route("/:id")

  .get(coordinatorController.getCoordinatorById) // GET by id

  .put(coordinatorController.updateCoordinatorById) // PUT by id

  .delete(coordinatorController.deleteCoordinatorById); // DELETE by id

// POST

router.post("/add", coordinatorController.createNewCoordinator);

module.exports = router;

generalRoute.js

const express = require("express");

const generalController = require("../controllers/generalController");

const router = express.Router();

router.route("/").get(generalController.getUserByEmail);

router.route("/").post(generalController.authenticateUser);

module.exports = router;

serviceRoute.js

const express = require("express");

const serviceController = require("../controllers/serviceController");

const router = express.Router();

// GET

router.route("/").get(serviceController.getAllServices);

router.route("/coordinator/:id").get(serviceController.getServicesByCoordinatorId);

// POST

router.post("/add", serviceController.createNewService);

// DELETE

router.route("/coordinator/delete/:id").delete(serviceController.deleteServicesByCoordinatorId);

router

  .route("/:id")

  .get(serviceController.getServiceById) // GET by id

  .put(serviceController.updateServiceById) // PUT by id

  .delete(serviceController.deleteServiceById); // DELETE by id

module.exports = router;

userLevelRoute.js

const express = require("express");

const userLevelController = require("../controllers/userLevelController");

const router = express.Router();

// GET

router.route("/").get(userLevelController.getAllLevels);

router

  .route("/:id")

  .get(userLevelController.getLevelById) // GET by id

  .put(userLevelController.updateLevelById) // PUT by id

  .delete(userLevelController.deleteLevelById); // DELETE by id

// POST

router.post("/add", userLevelController.createNewLevelUser);

module.exports = router;

server.js

require("dotenv").config();

const express = require("express");

const app = express();

app.use(express.urlencoded({ extended: true }));

// const PORT = process.env.PORT || 3000;

app.use(express.json());

app.use("/services", require("./routes/serviceRoute"));

app.use("/userLevels", require("./routes/userLevelRoute"));

app.use("/applicants", require("./routes/applicantRoute"));

app.use("/coordinators", require("./routes/coordinatorRoute"));

app.use("/login", require("./routes/generalRoute"));

app.use((err, req, res, next) => {

  console.log(err.stack);

  console.log(err.name);

  console.log(err.code);

  res.status(500).json({

    massage: "Somthing went wrong...",

  });

});

app.all("/\*", (req, res) => {

  res.status(404)

  .json({massage : `${req.method} on ${req.originalUrl} not found`});

})

// app.listen(PORT, () => console.log(`Listening on port ${PORT}...`));

module.exports = app;

connection.js

require("dotenv").config();

const PORT = process.env.PORT || 3000;

const server = require("./server");

const connect = require("http").createServer(server);

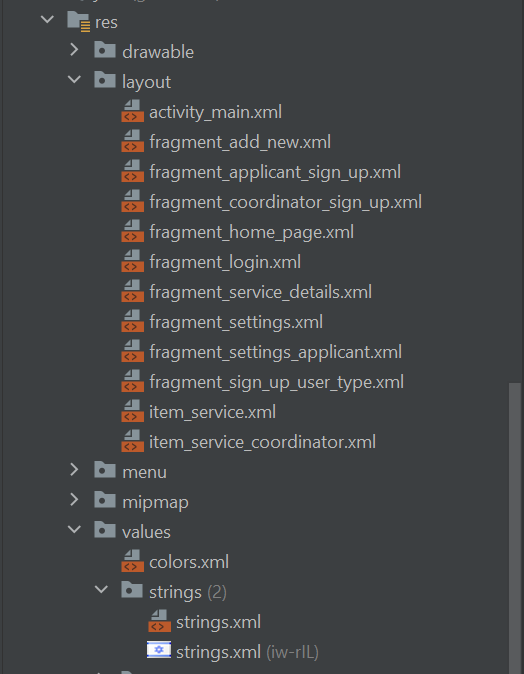
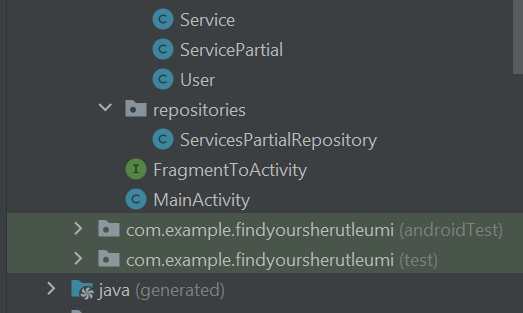
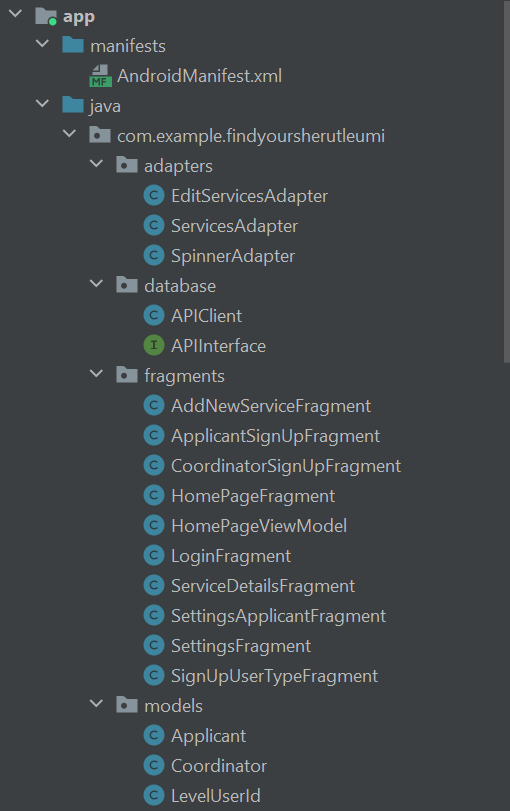
module.exports = { connect };

connect.listen(PORT, () => console.log(`Listening on port ${PORT}...`))

**10.2** צד לקוח

**10.2.1** עץ תיקיות

**XML Java**



**10.2.2** קבצי Java

**manifests**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="com.example.findyoursherutleumi">  
 <uses-permission android:name="android.permission.INTERNET"/>  
 <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@drawable/rounded\_logo"  
 android:label="@string/app\_name"  
 android:roundIcon="@drawable/rounded\_logo"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.FindYourSherutLeumi"  
 android:usesCleartextTraffic="true"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

**database**

APIClient.java

package com.example.findyoursherutleumi.database;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.example.findyoursherutleumi.models.User;  
import com.google.gson.Gson;  
import com.google.gson.GsonBuilder;  
  
import retrofit2.Retrofit;  
import retrofit2.converter.gson.GsonConverterFactory;  
  
*/\*\*  
 \* This class is the connection between the server to the client  
 \*/*public class APIClient extends AppCompatActivity {  
  
 private static Retrofit *instance*;  
// private static final String BASE\_URL = "http://10.0.2.2:3000/";  
 // *TODO: switch BASE\_URL if needed* private static final String *BASE\_URL* = "http://192.168.1.108:3000/";  
  
 public static Retrofit getInstance(){  
 if (*instance* == null){  
 Gson gson = new GsonBuilder().setLenient().create();  
 *instance* = new Retrofit.Builder()  
 .baseUrl(*BASE\_URL*)  
 .addConverterFactory(GsonConverterFactory.*create*(gson))  
 .build();  
 }  
 return *instance*;  
 }  
  
 public static User getUserService(){  
 User userService = *getInstance*().create(User.class);  
  
 return userService;  
 }  
  
  
  
}

APIInterfacet.java

package com.example.findyoursherutleumi.database;  
  
import com.example.findyoursherutleumi.models.Applicant;  
import com.example.findyoursherutleumi.models.Coordinator;  
import com.example.findyoursherutleumi.models.Service;  
import com.example.findyoursherutleumi.models.ServicePartial;  
import com.example.findyoursherutleumi.models.User;  
  
import java.util.List;  
  
import okhttp3.ResponseBody;  
import retrofit2.Call;  
import retrofit2.http.DELETE;  
import retrofit2.http.Field;  
import retrofit2.http.FormUrlEncoded;  
import retrofit2.http.GET;  
import retrofit2.http.POST;  
import retrofit2.http.PUT;  
import retrofit2.http.Path;  
  
*/\*\*  
 \* This interface has the http requests from the server,  
 \* that the user needs while using the app.  
 \*/*public interface APIInterface {  
  
 // GET requests  
 @GET("applicants")  
 Call<List<Applicant>> getAllApplicants();  
  
 @GET("applicants/{id}")  
 Call<Applicant> getApplicantById(@Path("id") int id);  
  
 @GET("services/{id}")  
 Call<Service> getServiceById(@Path("id") int id);  
  
 @GET("services")  
 Call<List<ServicePartial>> getAllServicesPartially();  
  
 @GET("services/coordinator/{id}")  
 Call<List<Service>> getServicesByCoordinatorId(@Path("id") int id);  
  
 @GET("coordinators/{id}")  
 Call<Coordinator> getCoordinatorById(@Path("id") int id);  
  
 @GET("coordinators/email/{email}")  
 Call<Coordinator> getCoordinatorByEmail(@Path("email") String email);  
  
 @GET("applicants/email/{email}")  
 Call<Applicant> getApplicantByEmail(@Path("email") String email);  
  
  
 // POST requests  
 @POST("login")  
 @FormUrlEncoded  
 Call<User> authenticateUser(@Field("email") String email,  
 @Field("u\_password") String password);  
  
 @POST("applicants/add")  
 @FormUrlEncoded  
 Call<Applicant> createNewApplicant(@Field("first\_name") String firstName,  
 @Field("last\_name") String lastName,  
 @Field("phone\_number") String phoneNumber,  
 @Field("city") String city,  
 @Field("email") String email,  
 @Field("u\_password") String password);  
  
 @POST("coordinators/add")  
 @FormUrlEncoded  
 Call<Coordinator> createNewCoordinator(@Field("first\_name") String firstName,  
 @Field("last\_name") String lastName,  
 @Field("phone\_number") String phoneNumber,  
 @Field("email") String email,  
 @Field("u\_password") String password,  
 @Field("organization\_name") String organizationName);  
  
 @POST("services/add")  
 @FormUrlEncoded  
 Call<Service> createNewService(@Field("service\_name") String serviceName,  
 @Field("organization\_name") String organizationName,  
 @Field("category") String category,  
 @Field("country") String country,  
 @Field("city") String city,  
 @Field("address") String address,  
 @Field("has\_apartment") Integer hasApartment,  
 @Field("is\_second\_year\_only") Integer secondYear,  
 @Field("is\_morning\_service") Integer isMorning,  
 @Field("is\_evening\_service") Integer isEvening,  
 @Field("description\_service") String description,  
 @Field("coordinator\_name") String coordinatorName,  
 @Field("coordinator\_id") int coordinatorId);  
  
  
 // PUT requests  
 @PUT("coordinators/{id}")  
 @FormUrlEncoded  
 Call<Coordinator> updateCoordinatorById(@Path("id") int coordinator\_id,  
 @Field("first\_name") String first\_name,  
 @Field("last\_name") String last\_name,  
 @Field("phone\_number") String phone\_number,  
 @Field("email") String email,  
 @Field("u\_password") String u\_password,  
 @Field("organization\_name") String organization\_name);  
  
 @PUT("applicants/{id}")  
 @FormUrlEncoded  
 Call<Applicant> updateApplicantById(@Path("id") int applicant\_id,  
 @Field("first\_name") String first\_name,  
 @Field("last\_name") String last\_name,  
 @Field("phone\_number") String phone\_number,  
 @Field("city") String city,  
 @Field("email") String email,  
 @Field("u\_password") String u\_password);  
  
  
 // DELETE requests  
 @DELETE("services/{id}")  
 Call<ResponseBody> deleteServiceById(@Path("id") int service\_id);  
  
 @DELETE("applicants/{id}")  
 Call<ResponseBody> deleteApplicantById(@Path("id") int applicant\_id);  
  
 @DELETE("coordinators/{id}")  
 Call<ResponseBody> deleteCoordinatorById(@Path("id") int coordinator\_id);  
  
 @DELETE("services/coordinator/delete/{id}")  
 Call<ResponseBody> deleteServicesByCoordinatorId (@Path("id") int coordinator\_id);  
  
  
}

**models**

Applicant.java

package com.example.findyoursherutleumi.models;  
  
import com.google.gson.annotations.Expose;  
import com.google.gson.annotations.SerializedName;  
  
*/\*\*  
 \* This class is applicant model  
 \*/*public class Applicant {  
  
 @SerializedName("applicant\_id")  
 @Expose  
 private Integer applicantId;  
 @SerializedName("first\_name")  
 @Expose  
 private String firstName;  
 @SerializedName("last\_name")  
 @Expose  
 private String lastName;  
 @SerializedName("phone\_number")  
 @Expose  
 private String phoneNumber;  
 @SerializedName("city")  
 @Expose  
 private String city;  
 @SerializedName("email")  
 @Expose  
 private String email;  
 @SerializedName("u\_password")  
 @Expose  
 private String uPassword;  
 @SerializedName("level\_user\_id")  
 @Expose  
 private Integer levelUserId;  
  
 public Integer getApplicantId() {  
 return applicantId;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public String getPhoneNumber() {  
 return phoneNumber;  
 }  
  
 public void setPhoneNumber(String phoneNumber) {  
 this.phoneNumber = phoneNumber;  
 }  
  
 public String getCity() {  
 return city;  
 }  
  
 public void setCity(String city) {  
 this.city = city;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getUPassword() {  
 return uPassword;  
 }  
  
 public void setUPassword(String uPassword) {  
 this.uPassword = uPassword;  
 }  
  
 public Integer getLevelUserId() {  
 return levelUserId;  
 }  
  
}

Coordinator.java

package com.example.findyoursherutleumi.models;  
  
import com.google.gson.annotations.Expose;  
import com.google.gson.annotations.SerializedName;  
  
*/\*\*  
 \* This class is coordinator model  
 \*/*public class Coordinator {  
  
 @SerializedName("coordinator\_id")  
 @Expose  
 private Integer coordinatorId;  
 @SerializedName("first\_name")  
 @Expose  
 private String firstName;  
 @SerializedName("last\_name")  
 @Expose  
 private String lastName;  
 @SerializedName("phone\_number")  
 @Expose  
 private String phoneNumber;  
 @SerializedName("email")  
 @Expose  
 private String email;  
 @SerializedName("u\_password")  
 @Expose  
 private String uPassword;  
 @SerializedName("organization\_name")  
 @Expose  
 private String organizationName;  
 @SerializedName("level\_user\_id")  
 @Expose  
 private Integer levelUserId;  
  
 public Integer getCoordinatorId() {  
 return coordinatorId;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public String getPhoneNumber() {  
 return phoneNumber;  
 }  
  
 public void setPhoneNumber(String phoneNumber) {  
 this.phoneNumber = phoneNumber;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getUPassword() {  
 return uPassword;  
 }  
  
 public void setUPassword(String uPassword) {  
 this.uPassword = uPassword;  
 }  
  
 public String getOrganizationName() {  
 return organizationName;  
 }  
  
 public void setOrganizationName(String organizationName) {  
 this.organizationName = organizationName;  
 }  
  
 public Integer getLevelUserId() {  
 return levelUserId;  
 }  
  
}

LevelUserId.java

package com.example.findyoursherutleumi.models;  
  
import com.google.gson.annotations.Expose;  
import com.google.gson.annotations.SerializedName;  
  
*/\*\*  
 \* This class is level user id model  
 \*/*public class LevelUserId {  
  
 @SerializedName("level\_id")  
 @Expose  
 private Integer levelId;  
 @SerializedName("level\_name")  
 @Expose  
 private String levelName;  
  
 public Integer getLevelId() {  
 return levelId;  
 }  
  
 public void setLevelId(Integer levelId) {  
 this.levelId = levelId;  
 }  
  
 public String getLevelName() {  
 return levelName;  
 }  
  
 public void setLevelName(String levelName) {  
 this.levelName = levelName;  
 }  
  
 @Override  
 public String toString() {  
 return "LevelUserId{" +  
 "levelId=" + levelId +  
 ", levelName='" + levelName + '\'' +  
 '}';  
 }  
}

Service.java

package com.example.findyoursherutleumi.models;  
  
import com.google.gson.annotations.Expose;  
import com.google.gson.annotations.SerializedName;  
  
*/\*\*  
 \* This class is service model  
 \*/*public class Service {  
  
 @SerializedName("service\_id")  
 @Expose  
 private Integer serviceId;  
 @SerializedName("service\_name")  
 @Expose  
 private String serviceName;  
 @SerializedName("organization\_name")  
 @Expose  
 private String organizationName;  
 @SerializedName("category")  
 @Expose  
 private String category;  
 @SerializedName("country")  
 @Expose  
 private String country;  
 @SerializedName("city")  
 @Expose  
 private String city;  
 @SerializedName("address")  
 @Expose  
 private String address;  
 @SerializedName("has\_apartment")  
 @Expose  
 private Integer hasApartment;  
 @SerializedName("is\_second\_year\_only")  
 @Expose  
 private Integer isSecondYearOnly;  
 @SerializedName("is\_morning\_service")  
 @Expose  
 private Integer isMorningService;  
 @SerializedName("is\_evening\_service")  
 @Expose  
 private Integer isEveningService;  
 @SerializedName("description\_service")  
 @Expose  
 private String descriptionService;  
 @SerializedName("coordinator\_name")  
 @Expose  
 private String coordinatorName;  
 @SerializedName("coordinator\_id")  
 @Expose  
 private Integer coordinatorId;  
  
 public Integer getServiceId() {  
 return serviceId;  
 }  
  
 public String getServiceName() {  
 return serviceName;  
 }  
  
 public void setServiceName(String serviceName) {  
 this.serviceName = serviceName;  
 }  
  
 public String getOrganizationName() {  
 return organizationName;  
 }  
  
 public void setOrganizationName(String organizationName) {  
 this.organizationName = organizationName;  
 }  
  
 public String getCategory() {  
 return category;  
 }  
  
 public void setCategory(String category) {  
 this.category = category;  
 }  
  
 public String getCountry() {  
 return country;  
 }  
  
 public void setCountry(String country) {  
 this.country = country;  
 }  
  
 public String getCity() {  
 return city;  
 }  
  
 public void setCity(String city) {  
 this.city = city;  
 }  
  
 public String getAddress() {  
 return address;  
 }  
  
 public void setAddress(String address) {  
 this.address = address;  
 }  
  
 public boolean getHasApartment() {  
 if (hasApartment == 0)  
 return false;  
 return true;  
 }  
  
 public void setHasApartment(Integer hasApartment) {  
 this.hasApartment = hasApartment;  
 }  
  
 public boolean getIsSecondYearOnly() {  
 if (isSecondYearOnly == 0)  
 return false;  
 return true;  
  
 }  
  
 public void setIsSecondYearOnly(Integer isSecondYearOnly) {  
 this.isSecondYearOnly = isSecondYearOnly;  
  
 }  
  
 public boolean getIsMorningService() {  
 if (isMorningService == 0)  
 return false;  
 return true;  
 }  
  
 public void setIsMorningService(Integer isMorningService) {  
 this.isMorningService = isMorningService;  
 }  
  
 public boolean getIsEveningService() {  
 if (isEveningService == 0)  
 return false;  
 return true;  
 }  
  
 public void setIsEveningService(Integer isEveningService) {  
 this.isEveningService = isEveningService;  
 }  
  
 public String getDescriptionService() {  
 return descriptionService;  
 }  
  
 public void setDescriptionService(String descriptionService) {  
 this.descriptionService = descriptionService;  
 }  
  
 public String getCoordinatorName() {  
 return coordinatorName;  
 }  
  
 public void setCoordinatorName(String coordinatorName) {  
 this.coordinatorName = coordinatorName;  
 }  
  
 public Integer getCoordinatorId() {  
 return coordinatorId;  
 }  
  
 public void setCoordinatorId(Integer coordinatorId) {  
 this.coordinatorId = coordinatorId;  
 }  
  
 @Override  
 public String toString() {  
 return "Service{" +  
 "serviceId=" + serviceId +  
 ", serviceName='" + serviceName + '\'' +  
 ", organizationName='" + organizationName + '\'' +  
 ", category='" + category + '\'' +  
 ", country='" + country + '\'' +  
 ", city='" + city + '\'' +  
 ", address='" + address + '\'' +  
 ", hasApartment=" + hasApartment +  
 ", isSecondYearOnly=" + isSecondYearOnly +  
 ", isMorningService=" + isMorningService +  
 ", isEveningService=" + isEveningService +  
 ", descriptionService='" + descriptionService + '\'' +  
 ", coordinatorName='" + coordinatorName + '\'' +  
 ", coordinatorId=" + coordinatorId +  
 '}';  
 }  
}

ServicePartial.java

package com.example.findyoursherutleumi.models;  
  
import com.google.gson.annotations.Expose;  
import com.google.gson.annotations.SerializedName;  
  
*/\*\*  
 \* This class is partial service details model  
 \*/*public class ServicePartial {  
  
 @SerializedName("service\_id")  
 @Expose  
 private Integer serviceId;  
 @SerializedName("service\_name")  
 @Expose  
 private String serviceName;  
 @SerializedName("organization\_name")  
 @Expose  
 private String organizationName;  
 @SerializedName("category")  
 @Expose  
 private String category;  
 @SerializedName("country")  
 @Expose  
 private String country;  
 @SerializedName("city")  
 @Expose  
 private String city;  
 @SerializedName("description\_service")  
 @Expose  
 private String descriptionService;  
  
 public Integer getServiceId() {  
 return serviceId;  
 }  
  
 public void setServiceId(Integer serviceId) {  
 this.serviceId = serviceId;  
 }  
  
 public String getServiceName() {  
 return serviceName;  
 }  
  
 public void setServiceName(String serviceName) {  
 this.serviceName = serviceName;  
 }  
  
 public String getOrganizationName() {  
 return organizationName;  
 }  
  
 public void setOrganizationName(String organizationName) {  
 this.organizationName = organizationName;  
 }  
  
 public String getCategory() {  
 return category;  
 }  
  
 public void setCategory(String category) {  
 this.category = category;  
 }  
  
 public String getCountry() {  
 return country;  
 }  
  
 public void setCountry(String country) {  
 this.country = country;  
 }  
  
 public String getCity() {  
 return city;  
 }  
  
 public void setCity(String city) {  
 this.city = city;  
 }  
  
 public String getDescriptionService() {  
 return descriptionService;  
 }  
  
 public void setDescriptionService(String descriptionService) {  
 this.descriptionService = descriptionService;  
 }  
  
 @Override  
 public String toString() {  
 return "Organization: '" + organizationName + '\n' +  
 "Category: '" + category + '\n' +  
 "Country: '" + country + '\n' +  
 "City: '" + city + '\n' +  
 "Description: '" + descriptionService;  
 }  
}

User.java

package com.example.findyoursherutleumi.models;  
  
import com.google.gson.annotations.Expose;  
import com.google.gson.annotations.SerializedName;  
  
*/\*\*  
 \* This class is user model  
 \*/*public class User {  
  
 @SerializedName("email")  
 @Expose  
 private String email;  
 @SerializedName("u\_password")  
 @Expose  
 private String uPassword;  
 @SerializedName("level\_user\_id")  
 @Expose  
 private Integer userTypeId;  
  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getUPassword() {  
 return uPassword;  
 }  
  
 public void setUPassword(String uPassword) {  
 this.uPassword = uPassword;  
 }  
  
 public Integer getUserTypeId() {  
 return userTypeId;  
 }  
  
 public void setUserTypeId(Integer userTypeId) {  
 this.userTypeId = userTypeId;  
 }  
  
  
 @Override  
 public String toString() {  
 return "User{" +  
 "email='" + email + '\'' +  
 ", uPassword='" + uPassword + '\'' +  
 '}';  
 }  
}

**fragments**

AddNewServiceFragment.java

package com.example.findyoursherutleumi.fragments;  
  
import androidx.fragment.app.FragmentTransaction;  
  
import android.annotation.SuppressLint;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.Toast;  
  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.database.APIClient;  
import com.example.findyoursherutleumi.database.APIInterface;  
import com.example.findyoursherutleumi.models.Coordinator;  
import com.example.findyoursherutleumi.models.Service;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
*/\*\*  
 \* This class is the AddNewService fragment,  
 \* which the authorized user can add service to the app  
 \*/*public class AddNewServiceFragment extends Fragment {  
  
 APIInterface apiInterface;  
  
 Button addServiceBtn;  
 EditText serviceNameInput;  
 EditText organizationInput;  
 EditText categoryInput;  
 EditText countryInput;  
 EditText cityInput;  
 EditText addressInput;  
 EditText descriptionInput;  
 RadioGroup apartmentRadio;  
 RadioButton apartmentBtn;  
 RadioGroup secondYearRadio;  
 RadioButton secondYearBtn;  
 RadioGroup morningRadio;  
 RadioButton morningBtn;  
 RadioGroup eveningRadio;  
 RadioButton eveningBtn;  
  
 private String userEmail;  
  
 private int hasApartment = 0;  
 private int isSecondYear = 0;  
 private int isMorning = 0;  
 private int isEvening = 0;  
  
  
 public static AddNewServiceFragment newInstance() {  
 return new AddNewServiceFragment();  
 }  
  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,  
 @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_add\_new*, container, false);  
 apiInterface = APIClient.*getInstance*().create(APIInterface.class);  
  
 serviceNameInput = view.findViewById(R.id.*service\_input\_add*);  
 organizationInput = view.findViewById(R.id.*organization\_input\_add*);  
 categoryInput = view.findViewById(R.id.*category\_input\_add*);  
 countryInput = view.findViewById(R.id.*country\_input\_add*);  
 cityInput = view.findViewById(R.id.*city\_input\_add*);  
 addressInput = view.findViewById(R.id.*address\_input\_add*);  
 descriptionInput = view.findViewById(R.id.*description\_input\_add*);  
  
 apartmentRadio = view.findViewById(R.id.*apartment\_radio*);  
 apartmentRadio.setOnCheckedChangeListener((radioGroup, i) -> onRadioButtonClicked1(view));  
 secondYearRadio = view.findViewById(R.id.*second\_year\_radio*);  
 secondYearRadio.setOnCheckedChangeListener((radioGroup, i) -> onRadioButtonClicked2(view));  
 morningRadio = view.findViewById(R.id.*is\_morning\_radio*);  
 morningRadio.setOnCheckedChangeListener((radioGroup, i) -> onRadioButtonClicked3(view));  
 eveningRadio = view.findViewById(R.id.*is\_evening\_radio*);  
 eveningRadio.setOnCheckedChangeListener((radioGroup, i) -> onRadioButtonClicked4(view));  
  
 assert getArguments() != null;  
 userEmail = getArguments().getString("email");  
  
 addServiceBtn = view.findViewById(R.id.*add\_new\_service\_btn*);  
 addServiceBtn.setOnClickListener(view1 -> {  
 if (view1.getId() == R.id.*add\_new\_service\_btn*) {  
 // checks if the user filled all the fields  
 boolean isComplete = isEmpty(serviceNameInput.getText().toString(),  
 organizationInput.getText().toString(),  
 categoryInput.getText().toString(),  
 countryInput.getText().toString(),  
 cityInput.getText().toString(),  
 addressInput.getText().toString(),  
 descriptionInput.getText().toString()  
 );  
 if (isComplete) {  
 Call<Coordinator> call = apiInterface.getCoordinatorByEmail(userEmail);  
 call.enqueue(new Callback<Coordinator>() {  
 @Override  
 public void onResponse(@NonNull Call<Coordinator> call, @NonNull Response<Coordinator> response) {  
 if (!response.isSuccessful()) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 assert response.body() != null;  
 Call<Service> call2 = apiInterface.createNewService(  
 serviceNameInput.getText().toString(),  
 organizationInput.getText().toString(),  
 categoryInput.getText().toString(),  
 countryInput.getText().toString(),  
 cityInput.getText().toString(),  
 addressInput.getText().toString(),  
 hasApartment, isSecondYear, isMorning, isEvening,  
 descriptionInput.getText().toString(),  
 response.body().getFirstName() + " " + response.body().getLastName(),  
 response.body().getCoordinatorId());  
  
 call2.enqueue(new Callback<Service>() {  
 @SuppressLint("NotifyDataSetChanged")  
 @Override  
 public void onResponse(@NonNull Call<Service> call, @NonNull Response<Service> response) {  
 if (!response.isSuccessful()) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 // after adding, all the fields cleared  
 serviceNameInput.getText().clear();  
 organizationInput.getText().clear();  
 categoryInput.getText().clear();  
 countryInput.getText().clear();  
 cityInput.getText().clear();  
 addressInput.getText().clear();  
 descriptionInput.getText().clear();  
  
 assert getFragmentManager() != null;  
 Fragment newFragment = new HomePageFragment();  
 FragmentTransaction transaction = getFragmentManager().beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment)  
 .addToBackStack(null)  
 .commit();  
  
 Toast.*makeText*(getContext(), R.string.*new\_service\_added*, Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<Service> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 }  
  
  
 @Override  
 public void onFailure(@NonNull Call<Coordinator> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 }  
 });  
  
 return view;  
 }  
  
 /\*  
 those radio methods for update the right value to the new item  
 \*/  
 public void onRadioButtonClicked1(View view) {  
 int radioId = apartmentRadio.getCheckedRadioButtonId();  
 apartmentBtn = view.findViewById(radioId);  
 if (apartmentBtn.getId() == R.id.*has\_apartment\_yes\_radio*)  
 hasApartment = 1;  
 }  
  
 public void onRadioButtonClicked2(View view) {  
 int radioId = secondYearRadio.getCheckedRadioButtonId();  
 secondYearBtn = view.findViewById(radioId);  
 if (secondYearBtn.getId() == R.id.*second\_year\_yes\_radio*)  
 isSecondYear = 1;  
 }  
  
 public void onRadioButtonClicked3(View view) {  
 int radioId = morningRadio.getCheckedRadioButtonId();  
 morningBtn = view.findViewById(radioId);  
 if (morningBtn.getId() == R.id.*morning\_service\_yes\_radio*)  
 isMorning = 1;  
 }  
  
 public void onRadioButtonClicked4(View view) {  
 int radioId = eveningRadio.getCheckedRadioButtonId();  
 eveningBtn = view.findViewById(radioId);  
 if (eveningBtn.getId() == R.id.*evening\_service\_yes\_radio*)  
 isEvening = 1;  
 }  
  
 /\*  
 this method checks if one of the fields is empty,  
 before adding a new item.  
 \*/  
 private boolean isEmpty(String input1, String input2, String input3,  
 String input4, String input5, String input6,  
 String input7) {  
 if (input1.isEmpty() || input2.isEmpty() || input3.isEmpty()  
 || input4.isEmpty() || input5.isEmpty() || input6.isEmpty() || input7.isEmpty()) {  
 Toast.*makeText*(getContext(), R.string.*fields\_not\_filled*, Toast.*LENGTH\_SHORT*).show();  
 return false;  
 }  
 return true;  
 }  
  
  
}

ApplicantSignUpFragment.java

package com.example.findyoursherutleumi.fragments;  
  
import androidx.fragment.app.FragmentTransaction;  
  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.database.APIClient;  
import com.example.findyoursherutleumi.database.APIInterface;  
import com.example.findyoursherutleumi.models.Applicant;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
*/\*\*  
 \* This class is the ApplicantSignUp fragment,  
 \* which user from type - applicant can sign up to the app.  
 \*/*public class ApplicantSignUpFragment extends Fragment {  
  
 APIInterface apiInterface;  
  
 EditText firstNameInput;  
 EditText lastNameInput;  
 EditText phoneNumberInput;  
 EditText cityInput;  
 EditText emailInput;  
 EditText passwordInput;  
  
 Button submitBtn;  
  
 public static ApplicantSignUpFragment newInstance() {  
 return new ApplicantSignUpFragment();  
 }  
  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,  
 @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_applicant\_sign\_up*, container, false);  
 apiInterface = APIClient.*getInstance*().create(APIInterface.class);  
  
 setHasOptionsMenu(true);  
  
 firstNameInput = view.findViewById(R.id.*first\_name\_input*);  
 lastNameInput = view.findViewById(R.id.*last\_name\_input*);  
 phoneNumberInput = view.findViewById(R.id.*phone\_number\_input*);  
 cityInput = view.findViewById(R.id.*city\_input*);  
 emailInput = view.findViewById(R.id.*email\_address\_input*);  
 passwordInput = view.findViewById(R.id.*password\_input*);  
  
 submitBtn = view.findViewById(R.id.*sign\_in\_btn*);  
  
 submitBtn.setOnClickListener(view1 -> {  
 if (view1.getId() == R.id.*sign\_in\_btn*) {  
 // checks if the user filled all the fields  
 boolean isComplete = isEmpty(firstNameInput.getText().toString(),  
 lastNameInput.getText().toString(),  
 phoneNumberInput.getText().toString(),  
 cityInput.getText().toString(),  
 emailInput.getText().toString(),  
 passwordInput.getText().toString());  
 if (isComplete) {  
 Call<Applicant> call = apiInterface.createNewApplicant(firstNameInput.getText().toString(),  
 lastNameInput.getText().toString(), phoneNumberInput.getText().toString(),  
 cityInput.getText().toString(), emailInput.getText().toString(),  
 passwordInput.getText().toString());  
 call.enqueue(new Callback<Applicant>() {  
 @Override  
 public void onResponse(@NonNull Call<Applicant> call, @NonNull Response<Applicant> response) {  
 if(!response.isSuccessful()) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 } else{  
 // all the fields is cleared after submission  
 firstNameInput.getText().clear();  
 lastNameInput.getText().clear();  
 phoneNumberInput.getText().clear();  
 cityInput.getText().clear();  
 emailInput.getText().clear();  
 passwordInput.getText().clear();  
  
 // transact back to the login screen  
 Fragment newFragment = new LoginFragment();  
 assert getFragmentManager() != null;  
 FragmentTransaction transaction = getFragmentManager().beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment);  
 transaction.addToBackStack(null);  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<Applicant> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 }  
 });  
  
 return view;  
 }  
  
 /\*  
 this method checks if one of the fields is empty,  
 or if some fields not filled correctly  
 before sign up the user.  
 \*/  
 private boolean isEmpty(String input1, String input2, String input3,  
 String input4, String email, String pass) {  
 if (input1.isEmpty() || input2.isEmpty() || input3.isEmpty()  
 || input4.isEmpty() || email.isEmpty() || pass.isEmpty()) {  
 Toast.*makeText*(getContext(), R.string.*fields\_not\_filled*, Toast.*LENGTH\_SHORT*).show();  
 return false;  
 }  
 else if (pass.length() < 6){  
 Toast.*makeText*(getContext(), R.string.*pass\_length\_short*, Toast.*LENGTH\_SHORT*).show();  
 return false;  
 }  
 else if (!android.util.Patterns.*EMAIL\_ADDRESS*.matcher(email).matches()){  
 Toast.*makeText*(getContext(), R.string.*email\_address\_not\_valid*, Toast.*LENGTH\_SHORT*).show();  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public void onPrepareOptionsMenu(@NonNull Menu menu) {  
 MenuItem item = menu.findItem(R.id.*more\_item*);  
 if(item != null)  
 item.setVisible(false);  
 }  
  
  
}

CoordinatorSignUpFragment.java

package com.example.findyoursherutleumi.fragments;  
  
import androidx.fragment.app.FragmentTransaction;  
  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.database.APIClient;  
import com.example.findyoursherutleumi.database.APIInterface;  
import com.example.findyoursherutleumi.models.Coordinator;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
*/\*\*  
 \* This class is the CoordinatorSignUp fragment,  
 \* which user from type - coordinator can sign up to the app.  
 \*/*public class CoordinatorSignUpFragment extends Fragment {  
  
 APIInterface apiInterface;  
  
 EditText firstNameInput;  
 EditText lastNameInput;  
 EditText phoneNumberInput;  
 EditText emailInput;  
 EditText passwordInput;  
 EditText organizationInput;  
  
 Button submitBtn;  
  
 public static CoordinatorSignUpFragment newInstance() {  
 return new CoordinatorSignUpFragment();  
 }  
  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,  
 @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_coordinator\_sign\_up*, container, false);  
 apiInterface = APIClient.*getInstance*().create(APIInterface.class);  
  
 setHasOptionsMenu(true);  
  
 firstNameInput = view.findViewById(R.id.*first\_name\_input*);  
 lastNameInput = view.findViewById(R.id.*last\_name\_input*);  
 phoneNumberInput = view.findViewById(R.id.*phone\_number\_input*);  
 emailInput = view.findViewById(R.id.*email\_address\_input*);  
 passwordInput = view.findViewById(R.id.*password\_input*);  
 organizationInput = view.findViewById(R.id.*organization\_input*);  
  
 submitBtn = view.findViewById(R.id.*sign\_in\_btn*);  
  
 submitBtn.setOnClickListener(view1 -> {  
 if (view1.getId() == R.id.*sign\_in\_btn*) {  
 // checks if the user filled all the fields  
 boolean isComplete = isEmpty(firstNameInput.getText().toString(),  
 lastNameInput.getText().toString(),  
 phoneNumberInput.getText().toString(),  
 emailInput.getText().toString(),  
 passwordInput.getText().toString(),  
 organizationInput.getText().toString());  
 if (isComplete) {  
 Call<Coordinator> call = apiInterface.createNewCoordinator(firstNameInput.getText().toString(),  
 lastNameInput.getText().toString(), phoneNumberInput.getText().toString(),  
 emailInput.getText().toString(), passwordInput.getText().toString(),  
 organizationInput.getText().toString());  
 call.enqueue(new Callback<Coordinator>() {  
 @Override  
 public void onResponse(@NonNull Call<Coordinator> call, @NonNull Response<Coordinator> response) {  
 if(!response.isSuccessful()) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 } else{  
 // all the fields is cleared after submission  
 firstNameInput.getText().clear();  
 lastNameInput.getText().clear();  
 phoneNumberInput.getText().clear();  
 emailInput.getText().clear();  
 passwordInput.getText().clear();  
 organizationInput.getText().clear();  
  
 // transact back to the login screen  
 Fragment newFragment = new LoginFragment();  
 assert getFragmentManager() != null;  
 FragmentTransaction transaction = getFragmentManager().beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment);  
 transaction.addToBackStack(null);  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<Coordinator> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 }  
 });  
 return view;  
 }  
  
 /\*  
 this method checks if one of the fields is empty,  
 or if some fields not filled correctly  
 before sign up the user.  
 \*/  
 private boolean isEmpty(String input1, String input2, String input3,  
 String email, String pass, String input5) {  
 if (input1.isEmpty() || input2.isEmpty() || input3.isEmpty()  
 || email.isEmpty() || input5.isEmpty() || pass.isEmpty()) {  
 Toast.*makeText*(getContext(), R.string.*fields\_not\_filled*, Toast.*LENGTH\_SHORT*).show();  
 return false;  
 }  
 else if (pass.length() < 6){  
 Toast.*makeText*(getContext(), R.string.*pass\_length\_short*, Toast.*LENGTH\_SHORT*).show();  
 return false;  
 }  
 else if (!android.util.Patterns.*EMAIL\_ADDRESS*.matcher(email).matches()){  
 Toast.*makeText*(getContext(), R.string.*email\_address\_not\_valid*, Toast.*LENGTH\_SHORT*).show();  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public void onPrepareOptionsMenu(@NonNull Menu menu) {  
 MenuItem item = menu.findItem(R.id.*more\_item*);  
 if(item != null)  
 item.setVisible(false);  
 }  
  
  
}

HomePageFragment.java

package com.example.findyoursherutleumi.fragments;  
  
import androidx.fragment.app.FragmentTransaction;  
import androidx.lifecycle.ViewModelProvider;  
  
import android.annotation.SuppressLint;  
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.text.Editable;  
import android.text.TextWatcher;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.EditText;  
import android.widget.TextView;  
  
import com.example.findyoursherutleumi.MainActivity;  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.adapters.ServicesAdapter;  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
  
*/\*\*  
 \* This class is the HomePage fragment,  
 \* the main screen of the app, contains the list of all services.  
 \*/*public class HomePageFragment extends Fragment {  
  
 // arguments from login  
 private int userTypeId;  
 private String userEmail;  
  
 HomePageViewModel mViewModel;  
 RecyclerView recyclerView;  
 ServicesAdapter servicesAdapter;  
 FloatingActionButton addServiceBtn;  
 private EditText searchBar;  
 TextView searchNoResult;  
  
 public static HomePageFragment newInstance() {  
 return new HomePageFragment();  
 }  
  
 @SuppressLint("NotifyDataSetChanged")  
 @Override  
 public void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 mViewModel = new ViewModelProvider(this).get(HomePageViewModel.class);  
 mViewModel.init();  
 }  
  
 @SuppressLint("NotifyDataSetChanged")  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,  
 @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_home\_page*, container, false);  
 setHasOptionsMenu(true);  
 addServiceBtn = view.findViewById(R.id.*add\_service\_btn*);  
 searchNoResult = view.findViewById(R.id.*no\_result\_text*);  
  
 if (getArguments() != null) {  
 userEmail = getArguments().getString("email");  
 userTypeId = getArguments().getInt("typeId");  
 }else{  
 userEmail = MainActivity.*getUser*().getEmail();  
 userTypeId = MainActivity.*getUser*().getUserTypeId();  
 }  
 if (userTypeId == 1) { // if the user type id is 1, the user cannot add new service to the app  
 addServiceBtn.setVisibility(View.*GONE*);  
 }  
  
 recyclerView = view.findViewById(R.id.*services\_lst*);  
 mViewModel.getServices().observe(getViewLifecycleOwner(), servicePartials -> {  
 servicesAdapter.updateServicesList(servicePartials);  
 servicesAdapter.notifyDataSetChanged();  
 });  
  
 initRecyclerView();  
  
 addServiceBtn.setOnClickListener(view1 -> {  
 Fragment newFragment = new AddNewServiceFragment();  
 Bundle bundle = new Bundle();  
 bundle.putString("email", userEmail);  
 newFragment.setArguments(bundle);  
  
 assert getFragmentManager() != null;  
 FragmentTransaction transaction = getFragmentManager().beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment);  
 transaction.addToBackStack(null);  
 transaction.commit();  
 });  
  
 searchBar = view.findViewById(R.id.*search\_bar*);  
 // filters items from recyclerview while user is searching by search bar  
 searchBar.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
  
 }  
  
 @SuppressLint("NotifyDataSetChanged")  
 @Override  
 public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
 servicesAdapter.filter(charSequence);  
 if (servicesAdapter.getItemCount() == 0)  
 searchNoResult.setVisibility(View.*VISIBLE*);  
 else searchNoResult.setVisibility(View.*GONE*);  
 }  
  
 @Override  
 public void afterTextChanged(Editable editable) {  
  
 }  
 });  
  
 servicesAdapter.notifyDataSetChanged();  
 return view;  
 }  
  
 // initializing the recyclerview  
 private void initRecyclerView() {  
 servicesAdapter = new ServicesAdapter(getLayoutInflater(), mViewModel.getServices().getValue());  
 RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(this.getContext());  
 recyclerView.setLayoutManager(layoutManager);  
 recyclerView.setAdapter(servicesAdapter);  
 }  
  
 @SuppressLint("NotifyDataSetChanged")  
 public void updateAdapter() {  
 servicesAdapter.notifyDataSetChanged();  
 }  
  
 /\* when the fragment is restarting, the viewmodel gets all services from db,  
 and updating the range of the adapter.  
 \*/  
 @Override  
 public void onStart() {  
 super.onStart();  
 mViewModel.loadServices();  
 updateAdapter();  
 servicesAdapter.notifyItemRangeChanged(0,recyclerView.getItemDecorationCount());  
 }  
  
 @SuppressLint("NotifyDataSetChanged")  
 @Override  
 public void onResume() {  
 super.onResume();  
 mViewModel.loadServices();  
 updateAdapter();  
 }  
  
 @Override  
 public void onPause() {  
 super.onPause();  
 updateAdapter();  
 }  
  
 @Override  
 public void onStop() {  
 super.onStop();  
 updateAdapter();  
 }  
  
 @Override  
 public void onDetach() {  
 super.onDetach();  
 updateAdapter();  
 }  
  
}

HomePageViewModel.java

package com.example.findyoursherutleumi.fragments;  
  
import androidx.lifecycle.LiveData;  
import androidx.lifecycle.MutableLiveData;  
import androidx.lifecycle.ViewModel;  
  
import com.example.findyoursherutleumi.models.ServicePartial;  
import com.example.findyoursherutleumi.repositories.ServicesPartialRepository;  
  
import java.util.List;  
  
*/\*\*  
 \* This class is the HomePage viewmodel of the main screen,  
 \* to set up the services list from db.  
 \*/*public class HomePageViewModel extends ViewModel {  
  
 private MutableLiveData<List<ServicePartial>> servicesPartialLst;  
 private ServicesPartialRepository repository;  
  
 public void init(){  
 if (servicesPartialLst != null)  
 return;  
 repository = ServicesPartialRepository.*getInstance*();  
 servicesPartialLst = repository.getServices();  
 }  
 public LiveData<List<ServicePartial>> getServices(){  
 return servicesPartialLst;  
 }  
  
 public void loadServices() {  
 servicesPartialLst.postValue(repository.getServices().getValue());  
 }  
  
}

LoginFragment.java

package com.example.findyoursherutleumi.fragments;  
  
import androidx.fragment.app.FragmentTransaction;  
  
import android.content.Context;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.findyoursherutleumi.FragmentToActivity;  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.database.APIClient;  
import com.example.findyoursherutleumi.database.APIInterface;  
import com.example.findyoursherutleumi.models.User;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
*/\*\*  
 \* This class is the Login fragment,  
 \* which the user can login to the app by email and password.  
 \*/*public class LoginFragment extends Fragment {  
  
 Button signInBtn;  
 EditText inputEmail;  
 EditText inputPassword;  
 TextView signUpClickable;  
  
 APIInterface apiInterface;  
  
 private FragmentToActivity mCallback;  
  
 // load context from main activity, before the fragment is created  
 @Override  
 public void onAttach(@NonNull Context context) {  
 super.onAttach(context);  
 try {  
 mCallback = (FragmentToActivity) context;  
 } catch (ClassCastException e) {  
 throw new ClassCastException(context  
 + " must implement FragmentToActivity");  
 }  
 }  
  
 public static LoginFragment newInstance() {  
 return new LoginFragment();  
 }  
  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,  
 @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_login*, container, false);  
 apiInterface = APIClient.*getInstance*().create(APIInterface.class);  
  
 setHasOptionsMenu(true);  
  
 inputEmail = view.findViewById(R.id.*email\_address\_input*);  
 inputPassword = view.findViewById(R.id.*password\_input*);  
  
 signInBtn = view.findViewById(R.id.*sign\_in\_btn*);  
 signUpClickable = view.findViewById(R.id.*sign\_up\_clickable*);  
  
 signInBtn.setOnClickListener(view1 -> {  
 if (view1.getId() == R.id.*sign\_in\_btn*){  
 // checks if the user filled all the fields  
 boolean isComplete = isEmpty(inputEmail.getText().toString(), inputPassword.getText().toString());  
 if (isComplete) {  
 Call<User> call = apiInterface.authenticateUser(inputEmail.getText().toString(), inputPassword.getText().toString());  
 call.enqueue(new Callback<User>() {  
 @Override  
 public void onResponse(@NonNull Call<User> call, @NonNull Response<User> response) {  
 if (!response.isSuccessful()) {  
 if (response.code() == 400) Toast.*makeText*(getContext(), R.string.*email\_does\_not\_exist*, Toast.*LENGTH\_SHORT*).show();  
 else if (response.code() == 401) Toast.*makeText*(getContext(), R.string.*email\_and\_password\_does\_not\_match*, Toast.*LENGTH\_SHORT*).show();  
 else Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 mCallback.communicate(response.body()); // sends user to the main activity  
  
 // all the fields is cleared after login  
 inputEmail.getText().clear();  
 inputPassword.getText().clear();  
  
 Fragment newFragment = new HomePageFragment();  
  
 Bundle bundle = new Bundle();  
 assert response.body() != null;  
  
 bundle.putInt("typeId", response.body().getUserTypeId());  
 bundle.putString("email", response.body().getEmail());  
 newFragment.setArguments(bundle);  
  
 assert getFragmentManager() != null;  
 FragmentTransaction transaction = getFragmentManager().beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment, "home\_page\_tag");  
 transaction.addToBackStack(null);  
 transaction.commit();  
 }  
 }  
 @Override  
 public void onFailure(@NonNull Call<User> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 }  
 });  
  
 signUpClickable.setOnClickListener(view12 -> {  
 if (view12.getId() == R.id.*sign\_up\_clickable*){  
 Fragment newFragment = new SignUpUserTypeFragment();  
 assert getFragmentManager() != null;  
 FragmentTransaction transaction = getFragmentManager().beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment);  
 transaction.addToBackStack(null);  
 transaction.commit();  
 }  
 });  
  
 return view;  
 }  
  
 /\*  
 this method checks if one of the fields is empty  
 \*/  
 private boolean isEmpty(String input1, String input2) {  
 if (input1.isEmpty() || input2.isEmpty()) {  
 Toast.*makeText*(getContext(), R.string.*fields\_not\_filled*, Toast.*LENGTH\_SHORT*).show();  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public void onPrepareOptionsMenu(@NonNull Menu menu) {  
 MenuItem item = menu.findItem(R.id.*more\_item*);  
 if(item != null)  
 item.setVisible(false);  
 }  
  
}

ServiceDetailsFragment.java

package com.example.findyoursherutleumi.fragments;  
  
import android.annotation.SuppressLint;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.database.APIClient;  
import com.example.findyoursherutleumi.database.APIInterface;  
import com.example.findyoursherutleumi.models.Coordinator;  
import com.example.findyoursherutleumi.models.Service;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
*/\*\*  
 \* This class is the ServiceDetails fragment,  
 \* where all service details is shown.  
 \*/*public class ServiceDetailsFragment extends Fragment {  
  
 private int serviceId;  
  
 TextView serviceTxt;  
 TextView organizationTxt;  
 TextView categoryTxt;  
 TextView countryTxt;  
 TextView cityTxt;  
 TextView addressTxt;  
 TextView hasApartmentTxt;  
 TextView isSecondYearOnlyTxt;  
 TextView isMorningServiceTxt;  
 TextView isEveningServiceTxt;  
 TextView descriptionTxt;  
 TextView coordinatorNameTxt;  
 TextView coordinatorContactTxt;  
  
 APIInterface apiInterface;  
  
 public static ServiceDetailsFragment newInstance() {  
 return new ServiceDetailsFragment();  
 }  
  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,  
 @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_service\_details*, container, false);  
 apiInterface = APIClient.*getInstance*().create(APIInterface.class);  
  
 serviceTxt = view.findViewById(R.id.*service\_details\_title*);  
 organizationTxt = view.findViewById(R.id.*organization\_txt\_details*);  
 categoryTxt = view.findViewById(R.id.*category\_txt*);  
 countryTxt = view.findViewById(R.id.*country\_txt*);  
 cityTxt = view.findViewById(R.id.*city\_txt*);  
 addressTxt = view.findViewById(R.id.*address\_txt*);  
 hasApartmentTxt = view.findViewById(R.id.*has\_apartment\_txt*);  
 isSecondYearOnlyTxt = view.findViewById(R.id.*is\_second\_year\_only\_txt*);  
 isMorningServiceTxt = view.findViewById(R.id.*is\_morning\_service\_txt*);  
 isEveningServiceTxt = view.findViewById(R.id.*is\_evening\_service\_txt*);  
 descriptionTxt = view.findViewById(R.id.*description\_service\_txt*);  
 coordinatorNameTxt = view.findViewById(R.id.*coordinator\_name\_txt*);  
 coordinatorContactTxt = view.findViewById(R.id.*coordinator\_contact\_txt*);  
  
 // arguments from ServicesAdapter  
 assert getArguments() != null;  
 serviceId = getArguments().getInt("id");  
  
 Call<Service> call = apiInterface.getServiceById(serviceId);  
 call.enqueue(new Callback<Service>() {  
 @Override  
 public void onResponse(@NonNull Call<Service> call, @NonNull Response<Service> response) {  
 if (!response.isSuccessful())  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 else {  
 // sets service details from db  
 assert response.body() != null;  
 serviceTxt.setText(response.body().getServiceName());  
 organizationTxt.setText(response.body().getOrganizationName());  
 categoryTxt.setText(response.body().getCategory());  
 countryTxt.setText(response.body().getCountry());  
 cityTxt.setText(response.body().getCity());  
 addressTxt.setText(response.body().getAddress());  
 hasApartmentTxt.setText(response.body().getHasApartment() ? R.string.*yes* : R.string.*no*);  
 isSecondYearOnlyTxt.setText(response.body().getIsSecondYearOnly() ? R.string.*yes* : R.string.*no*);  
 isMorningServiceTxt.setText(response.body().getIsMorningService() ? R.string.*yes* : R.string.*no*);  
 isEveningServiceTxt.setText(response.body().getIsEveningService() ? R.string.*yes* : R.string.*no*);  
 descriptionTxt.setText(response.body().getDescriptionService());  
  
 Call<Coordinator> call1 = apiInterface.getCoordinatorById(response.body().getCoordinatorId());  
 call1.enqueue(new Callback<Coordinator>() {  
 @SuppressLint("SetTextI18n")  
 @Override  
 public void onResponse(@NonNull Call<Coordinator> call, @NonNull Response<Coordinator> response) {  
 if (!response.isSuccessful()){  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 assert response.body() != null;  
 coordinatorNameTxt.setText(response.body().getFirstName() + " " + response.body().getLastName());  
 coordinatorContactTxt.setText(response.body().getPhoneNumber());  
 }  
 }  
 @Override  
 public void onFailure(@NonNull Call<Coordinator> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 System.*out*.println(t.getMessage());  
 }  
 });  
  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<Service> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 System.*out*.println(t.getMessage());  
 }  
 });  
  
 return view;  
 }  
  
  
}

SettingsApplicantFragment.java

package com.example.findyoursherutleumi.fragments;  
  
import androidx.fragment.app.FragmentManager;  
import androidx.fragment.app.FragmentTransaction;  
  
import android.app.AlertDialog;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.database.APIClient;  
import com.example.findyoursherutleumi.database.APIInterface;  
import com.example.findyoursherutleumi.models.Applicant;  
  
  
import okhttp3.ResponseBody;  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
*/\*\*  
 \* This class is the SettingsApplicant fragment,  
 \* which applicant user type can edit it's own details  
 \*/*public class SettingsApplicantFragment extends Fragment {  
  
 EditText firstNameInput;  
 EditText lastNameInput;  
 EditText phoneInput;  
 EditText emailInput;  
 EditText cityInput;  
 EditText passwordInput;  
 Button updateDetailsBtn;  
 Button deleteApplicantBtn;  
 APIInterface apiInterface;  
 private int applicantId;  
 private String password;  
  
 public static SettingsApplicantFragment newInstance() {  
 return new SettingsApplicantFragment();  
 }  
  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,  
 @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_settings\_applicant*, container, false);  
 apiInterface = APIClient.*getInstance*().create(APIInterface.class);  
  
 firstNameInput = view.findViewById(R.id.*a\_first\_name\_input*);  
 lastNameInput = view.findViewById(R.id.*a\_last\_name\_input*);  
 phoneInput = view.findViewById(R.id.*a\_phone\_input*);  
 emailInput = view.findViewById(R.id.*a\_email\_input*);  
 cityInput = view.findViewById(R.id.*a\_city\_input*);  
 passwordInput = view.findViewById(R.id.*a\_password\_input*);  
  
 deleteApplicantBtn = view.findViewById(R.id.*a\_delete\_user\_edit\_details\_btn*);  
  
 fetchData(); // gets user details from db  
  
 updateDetailsBtn = view.findViewById(R.id.*update\_applicant\_edit\_details\_btn*);  
 updateDetailsBtn.setOnClickListener(view1 -> { // update the new details from fields  
 if (passwordInput.getText().toString().length() < 6 && passwordInput.getText().toString().length() > 0)  
 Toast.*makeText*(inflater.getContext(), R.string.*pass\_length\_short*, Toast.*LENGTH\_SHORT*).show();  
 else {  
 if (passwordInput.getText().toString().length() > 0)  
 password = passwordInput.getText().toString();  
 Call<Applicant> call = apiInterface.updateApplicantById(applicantId,  
 firstNameInput.getText().toString(),  
 lastNameInput.getText().toString(),  
 phoneInput.getText().toString(),  
 cityInput.getText().toString(),  
 emailInput.getText().toString(),  
 password);  
 call.enqueue(new Callback<Applicant>() {  
 @Override  
 public void onResponse(@NonNull Call<Applicant> call, @NonNull Response<Applicant> response) {  
 if (!response.isSuccessful()) {  
 Toast.*makeText*(inflater.getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 System.*out*.println("code1: " + response.code());  
 } else {  
 Toast.*makeText*(getContext(), R.string.*details\_updated\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<Applicant> call, @NonNull Throwable t) {  
 Toast.*makeText*(inflater.getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 });  
  
 deleteApplicantBtn.setOnClickListener(view12 -> {  
 // alert dialog to make sure the user actually wants to delete the account.  
 final AlertDialog.Builder builder = new AlertDialog.Builder(getContext());  
 builder.setMessage(R.string.*delete\_user\_dialog*);  
 builder.setCancelable(true);  
  
 builder.setPositiveButton(R.string.*yes*, (dialogInterface, i) -> { // if the user click 'yes'  
  
 Fragment newFragment = new LoginFragment();  
 assert getFragmentManager() != null;  
 FragmentManager fragmentManager = getFragmentManager();  
 FragmentTransaction transaction = fragmentManager.beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment).disallowAddToBackStack();  
 transaction.commit();  
  
 // clears fragment's back stack to prevent user from going back after logging out.  
 int count = fragmentManager.getBackStackEntryCount();  
 for (int index = 0; index < count; ++index) {  
 fragmentManager.popBackStackImmediate();  
 }  
  
 Call<ResponseBody> call1 = apiInterface.deleteApplicantById(applicantId);  
 call1.enqueue(new Callback<ResponseBody>() {  
 @Override  
 public void onResponse(@NonNull Call<ResponseBody> call, @NonNull Response<ResponseBody> response) {  
 if (!response.isSuccessful()) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 System.*out*.println("code1: " + response.code());  
 } else {  
 Toast.*makeText*(newFragment.getContext(), R.string.*user\_deleted\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<ResponseBody> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 });  
  
 builder.setNegativeButton(R.string.*no*, (dialogInterface, i) -> dialogInterface.cancel()); // if the user click 'no'  
 AlertDialog alertDialog = builder.create();  
 alertDialog.show();  
 });  
  
 return view;  
 }  
  
 /\*  
 this method gets the current user data from db  
 \*/  
 public void fetchData() {  
 assert getArguments() != null;  
 Call<Applicant> call1 = apiInterface.getApplicantByEmail(getArguments().getString("email"));  
 call1.enqueue(new Callback<Applicant>() {  
 @Override  
 public void onResponse(@NonNull Call<Applicant> call, @NonNull Response<Applicant> response) {  
 if (!response.isSuccessful()) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 System.*out*.println("code1: " + response.code());  
 } else {  
 assert response.body() != null;  
 password = response.body().getUPassword();  
 applicantId = response.body().getApplicantId();  
 firstNameInput.setText(response.body().getFirstName());  
 lastNameInput.setText(response.body().getLastName());  
 phoneInput.setText(response.body().getPhoneNumber());  
 cityInput.setText(response.body().getCity());  
 emailInput.setText(response.body().getEmail());  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<Applicant> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
  
 }  
  
  
}

SettingsFragment.java

package com.example.findyoursherutleumi.fragments;  
  
import androidx.fragment.app.FragmentManager;  
import androidx.fragment.app.FragmentTransaction;  
  
import android.annotation.SuppressLint;  
import android.app.AlertDialog;  
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.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.adapters.EditServicesAdapter;  
import com.example.findyoursherutleumi.database.APIClient;  
import com.example.findyoursherutleumi.database.APIInterface;  
import com.example.findyoursherutleumi.models.Coordinator;  
import com.example.findyoursherutleumi.models.Service;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import okhttp3.ResponseBody;  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
*/\*\*  
 \* This class is the SettingsApplicant fragment,  
 \* which coordinator user type can edit it's own details  
 \*/*public class SettingsFragment extends Fragment {  
  
 EditServicesAdapter editServicesAdapter;  
 List<Service> cServicesLst = new ArrayList<>();  
 RecyclerView recyclerView;  
 TextView noItems;  
 EditText firstNameInput;  
 EditText lastNameInput;  
 EditText emailInput;  
 EditText phoneInput;  
 EditText passwordInput;  
 EditText organizationInput;  
 Button updateDetailsBtn;  
 Button deleteCoordinatorBtn;  
 APIInterface apiInterface;  
 private int coordinatorId;  
 private String password;  
  
 public static SettingsFragment newInstance() {  
 return new SettingsFragment();  
 }  
  
  
 @SuppressLint({"SetTextI18n", "NotifyDataSetChanged"})  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,  
 @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_settings*, container, false);  
 apiInterface = APIClient.*getInstance*().create(APIInterface.class);  
 noItems = view.findViewById(R.id.*no\_items\_text*);  
  
 recyclerView = view.findViewById(R.id.*coordinator\_services\_list\_edit*);  
 RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(this.getContext());  
 recyclerView.setLayoutManager(layoutManager);  
 editServicesAdapter = new EditServicesAdapter(inflater, cServicesLst);  
 recyclerView.setAdapter(editServicesAdapter);  
  
 firstNameInput = view.findViewById(R.id.*c\_first\_name\_input*);  
 lastNameInput = view.findViewById(R.id.*c\_last\_name\_input*);  
 emailInput = view.findViewById(R.id.*c\_email\_input*);  
 phoneInput = view.findViewById(R.id.*c\_phone\_input*);  
 passwordInput = view.findViewById(R.id.*c\_password\_input*);  
 organizationInput = view.findViewById(R.id.*c\_organization\_input*);  
  
 deleteCoordinatorBtn = view.findViewById(R.id.*c\_delete\_user\_edit\_details\_btn*);  
  
 fetchData(); // gets user details from db  
  
 updateDetailsBtn = view.findViewById(R.id.*update\_edit\_details\_btn*);  
 updateDetailsBtn.setOnClickListener(view12 -> {  
 if (passwordInput.getText().toString().length() < 6 && passwordInput.getText().toString().length() > 0)  
 Toast.*makeText*(inflater.getContext(), R.string.*pass\_length\_short*, Toast.*LENGTH\_SHORT*).show();  
 else {  
 if (passwordInput.getText().toString().length() > 0)  
 password = passwordInput.getText().toString();  
 Call<Coordinator> call1 = apiInterface.updateCoordinatorById(coordinatorId,  
 firstNameInput.getText().toString(),  
 lastNameInput.getText().toString(),  
 phoneInput.getText().toString(),  
 emailInput.getText().toString(),  
 password,  
 organizationInput.getText().toString());  
 call1.enqueue(new Callback<Coordinator>() {  
 @Override  
 public void onResponse(@NonNull Call<Coordinator> call, @NonNull Response<Coordinator> response) {  
 if (!response.isSuccessful()) {  
 Toast.*makeText*(inflater.getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 System.*out*.println("code1: " + response.code());  
 } else {  
 assert response.body() != null;  
 Toast.*makeText*(getContext(), R.string.*details\_updated\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<Coordinator> call, @NonNull Throwable t) {  
 Toast.*makeText*(inflater.getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 });  
  
 deleteCoordinatorBtn.setOnClickListener(view1 -> {  
 // alert dialog to make sure the user actually wants to delete the account.  
 final AlertDialog.Builder builder = new AlertDialog.Builder(getContext());  
 builder.setMessage(R.string.*delete\_user\_dialog*);  
 builder.setCancelable(true);  
 builder.setPositiveButton(R.string.*yes*, (dialogInterface, i) -> { // if the user click 'yes'  
 Fragment newFragment = new LoginFragment();  
 assert getFragmentManager() != null;  
 FragmentManager fragmentManager = getFragmentManager();  
 FragmentTransaction transaction = fragmentManager.beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment).disallowAddToBackStack();  
 transaction.commit();  
  
 // clears fragment's back stack to prevent user from going back after logging out.  
 int count = fragmentManager.getBackStackEntryCount();  
 for (int index = 0; index < count; ++index) {  
 fragmentManager.popBackStackImmediate();  
 }  
  
 // *TODO: delete successfully but by login with another account it gets an error* Call<ResponseBody> call1 = apiInterface.deleteServicesByCoordinatorId(coordinatorId);  
 call1.enqueue(new Callback<ResponseBody>() {  
 @Override  
 public void onResponse(@NonNull Call<ResponseBody> call, @NonNull Response<ResponseBody> response) {  
 if (!response.isSuccessful()) {  
 Toast.*makeText*(newFragment.getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 System.*out*.println("code1: " + response.code());  
 } else {  
 Call<ResponseBody> call2 = apiInterface.deleteCoordinatorById(coordinatorId);  
 call2.enqueue(new Callback<ResponseBody>() {  
 @Override  
 public void onResponse(@NonNull Call<ResponseBody> call, @NonNull Response<ResponseBody> response) {  
 if (!response.isSuccessful()) {  
 Toast.*makeText*(newFragment.getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 System.*out*.println("code2: " + response.code());  
 } else {  
 Toast.*makeText*(newFragment.getContext(), R.string.*user\_deleted\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<ResponseBody> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<ResponseBody> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 });  
  
 builder.setNegativeButton(R.string.*no*, (dialogInterface, i) -> dialogInterface.cancel()); // if the user click 'no'  
 AlertDialog alertDialog = builder.create();  
 alertDialog.show();  
 });  
  
 editServicesAdapter.notifyDataSetChanged();  
 return view;  
 }  
  
 /\*  
 this method gets the current user data from db  
 \*/  
 public void fetchData() {  
 assert getArguments() != null;  
 Call<Coordinator> call = apiInterface.getCoordinatorByEmail(getArguments().getString("email"));  
 call.enqueue(new Callback<Coordinator>() {  
 @Override  
 public void onResponse(@NonNull Call<Coordinator> call, @NonNull Response<Coordinator> response) {  
 if (!response.isSuccessful()) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 assert response.body() != null;  
 password = response.body().getUPassword();  
 coordinatorId = response.body().getCoordinatorId();  
 firstNameInput.setText(response.body().getFirstName());  
 lastNameInput.setText(response.body().getLastName());  
 emailInput.setText(response.body().getEmail());  
 phoneInput.setText(response.body().getPhoneNumber());  
 organizationInput.setText(response.body().getOrganizationName());  
  
 Call<List<Service>> call1 = apiInterface.getServicesByCoordinatorId(coordinatorId);  
 call1.enqueue(new Callback<List<Service>>() {  
 @SuppressLint("NotifyDataSetChanged")  
 @Override  
 public void onResponse(@NonNull Call<List<Service>> call, @NonNull Response<List<Service>> response) {  
 if (!response.isSuccessful())  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 else {  
 assert response.body() != null;  
 cServicesLst.addAll(response.body());  
 editServicesAdapter.notifyDataSetChanged();  
  
 if (editServicesAdapter.getItemCount() == 0)  
 noItems.setVisibility(View.*VISIBLE*);  
 else noItems.setVisibility(View.*GONE*);  
  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<List<Service>> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call<Coordinator> call, @NonNull Throwable t) {  
 Toast.*makeText*(getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 }  
  
  
}

SignUpUserTypeFragment.java

package com.example.findyoursherutleumi.fragments;  
  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentTransaction;  
  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.Spinner;  
  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.adapters.SpinnerAdapter;  
  
*/\*\*  
 \* This is the SignUpUserType fragment,  
 \* which the user chooses applicant/coordinator signing up.  
 \*/*public class SignUpUserTypeFragment extends Fragment {  
  
 Button nextBtn;  
 public static int *chose*;  
  
 public static SignUpUserTypeFragment newInstance() {  
 return new SignUpUserTypeFragment();  
 }  
  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,  
 @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_sign\_up\_user\_type*, container, false);  
  
 setHasOptionsMenu(true);  
  
 Spinner spinner = view.findViewById(R.id.*user\_type\_spinner*);  
 spinner.setOnItemSelectedListener(new SpinnerAdapter());  
 ArrayAdapter<CharSequence> adapter = ArrayAdapter.*createFromResource*(getContext(),  
 R.array.*user\_types\_array*, android.R.layout.*simple\_spinner\_item*);  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 spinner.setAdapter(adapter);  
  
 nextBtn = view.findViewById(R.id.*sign\_up\_next\_btn*);  
 nextBtn.setOnClickListener(view1 -> {  
 if (view1.getId() == R.id.*sign\_up\_next\_btn*){  
 Fragment newFragment = null;  
 if (*chose* == 0) newFragment = new ApplicantSignUpFragment();  
 else if (*chose* == 1) newFragment = new CoordinatorSignUpFragment();  
 assert getFragmentManager() != null;  
 FragmentTransaction transaction = getFragmentManager().beginTransaction();  
 assert newFragment != null;  
 transaction.replace(R.id.*fragmentContainerView*, newFragment);  
 transaction.addToBackStack(null);  
 transaction.commit();  
 }  
 });  
  
 return view;  
 }  
  
 @Override  
 public void onPrepareOptionsMenu(@NonNull Menu menu) {  
 MenuItem item = menu.findItem(R.id.*more\_item*);  
 if(item!=null)  
 item.setVisible(false);  
 }  
  
  
  
}

**adapters**

EditServicesAdapter.java

package com.example.findyoursherutleumi.adapters;  
  
import android.annotation.SuppressLint;  
import android.app.AlertDialog;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.database.APIClient;  
import com.example.findyoursherutleumi.database.APIInterface;  
import com.example.findyoursherutleumi.models.Service;  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
  
import java.util.List;  
  
import okhttp3.ResponseBody;  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
*/\*\*  
 \* This class is the adapter for the service's list in the 'SettingsFragment'.  
 \*/*public class EditServicesAdapter extends RecyclerView.Adapter<EditServicesAdapter.ViewHolder> {  
  
 private final LayoutInflater inflater;  
 int id;  
 static int *deleted* = 0;  
 APIInterface apiInterface;  
 private List<Service> servicesLst;  
  
 public EditServicesAdapter(LayoutInflater inflater, List<Service> servicesLst) {  
 this.inflater = inflater;  
 this.servicesLst = servicesLst;  
 }  
  
 @NonNull  
 @Override  
 public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View rootView = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*item\_service\_coordinator*, parent, false);  
 apiInterface = APIClient.*getInstance*().create(APIInterface.class);  
 return new ViewHolder(rootView);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ViewHolder holder, int position) {  
 holder.serviceNameTxt.setText(servicesLst.get(position).getServiceName());  
 holder.cityTxt.setText(servicesLst.get(position).getCity());  
  
 holder.deleteBtn.setOnClickListener(view -> {  
 if (view.getId() == holder.deleteBtn.findViewById(R.id.*delete\_service\_edit\_btn*).getId()) {  
 id = servicesLst.get(holder.getAdapterPosition()).getServiceId();  
  
 // alert dialog to make sure the user actually wants to delete the current item  
 final AlertDialog.Builder builder = new AlertDialog.Builder(inflater.getContext());  
 builder.setMessage(R.string.*delete\_service\_dialog*);  
 builder.setCancelable(true);  
 builder.setPositiveButton(R.string.*yes*, (dialogInterface, i) -> { // if the user click 'yes'  
 Call<ResponseBody> call = apiInterface.deleteServiceById(id);  
 call.enqueue(new Callback<ResponseBody>() {  
 @SuppressLint("NotifyDataSetChanged")  
 @Override  
 public void onResponse(@NonNull Call<ResponseBody> call, @NonNull Response<ResponseBody> response) {  
 if (!response.isSuccessful()) {  
 Toast.*makeText*(inflater.getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 servicesLst.remove(holder.getAdapterPosition());  
 notifyItemRemoved(holder.getAdapterPosition());  
 notifyItemRangeChanged(holder.getAdapterPosition(),servicesLst.size()-1);  
 Toast.*makeText*(inflater.getContext(), R.string.*item\_deleted\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 @Override  
 public void onFailure(@NonNull Call<ResponseBody> call, @NonNull Throwable t) {  
 Toast.*makeText*(inflater.getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 });  
 builder.setNegativeButton(R.string.*no*, (dialogInterface, i) -> dialogInterface.cancel()); // if the user click 'no'  
 AlertDialog alertDialog = builder.create();  
 alertDialog.show();  
 }  
 });  
 }  
  
  
 @Override  
 public int getItemCount() {  
 return servicesLst.size();  
 }  
  
 @SuppressLint("NotifyDataSetChanged")  
 public void updateServicesList(final List<Service> mServicesLst) {  
 this.servicesLst = mServicesLst;  
 notifyDataSetChanged();  
 }  
  
 public class ViewHolder extends RecyclerView.ViewHolder {  
  
 FloatingActionButton deleteBtn;  
 TextView serviceNameTxt;  
 TextView cityTxt;  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
 deleteBtn = itemView.findViewById(R.id.*delete\_service\_edit\_btn*);  
 serviceNameTxt = itemView.findViewById(R.id.*service\_title\_lst\_edit*);  
 cityTxt = itemView.findViewById(R.id.*service\_city\_lst\_edit*);  
 }  
  
 }  
}

ServicesAdapter.java

package com.example.findyoursherutleumi.adapters;  
  
import android.annotation.SuppressLint;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.fragments.ServiceDetailsFragment;  
import com.example.findyoursherutleumi.models.ServicePartial;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.stream.Collectors;  
  
*/\*\*  
 \* This class is the adapter for the service's list in the 'HomePageFragment'.  
 \*/*public class ServicesAdapter extends RecyclerView.Adapter<ServicesAdapter.ViewHolder> {  
  
 private final LayoutInflater inflater;  
 private List<ServicePartial> servicesLst;  
 private List<ServicePartial> filteredList;  
  
 public ServicesAdapter(LayoutInflater inflater, List<ServicePartial> servicesLst){  
 this.inflater = inflater;  
 this.servicesLst = servicesLst;  
 this.filteredList = new ArrayList<>(servicesLst); // copy of the list for filtering  
 }  
  
 @NonNull  
 @Override  
 public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View rootView = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*item\_service*,parent,false);  
 return new ViewHolder(rootView);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ViewHolder holder, int position) {  
 holder.title.setText(filteredList.get(position).getServiceName());  
 holder.detailsBtn.setText(R.string.*show\_details*);  
  
 holder.organizationTxt.setText(filteredList.get(position).getOrganizationName());  
 holder.categoryTxt.setText(filteredList.get(position).getCategory());  
 holder.countryTxt.setText(filteredList.get(position).getCountry());  
 holder.cityTxt.setText(filteredList.get(position).getCity());  
 holder.descriptionTxt.setText(filteredList.get(position).getDescriptionService());  
  
 int id = filteredList.get(position).getServiceId();  
 holder.detailsBtn.setOnClickListener(view -> {  
 if (view.getId() == holder.detailsBtn.findViewById(R.id.*show\_details\_btn*).getId()) {  
  
 // get to the specific 'ServiceDetailsFragment' by the service id  
 AppCompatActivity activity = (AppCompatActivity) view.getContext();  
 Fragment newFragment = new ServiceDetailsFragment();  
 Bundle bundle = new Bundle();  
 bundle.putInt("id", id);  
 newFragment.setArguments(bundle);  
  
 activity.getSupportFragmentManager()  
 .beginTransaction()  
 .replace(R.id.*fragmentContainerView*, newFragment)  
 .addToBackStack(null)  
 .commit();  
 }  
 });  
 }  
  
 @Override  
 public int getItemCount() {  
 return filteredList.size();  
 }  
  
 @SuppressLint("NotifyDataSetChanged")  
 public void updateServicesList(final List<ServicePartial> mServicesLst) {  
 filteredList = mServicesLst;  
 notifyDataSetChanged();  
 }  
  
 public class ViewHolder extends RecyclerView.ViewHolder{  
  
 TextView title;  
 Button detailsBtn;  
 TextView organizationTxt;  
 TextView categoryTxt;  
 TextView countryTxt;  
 TextView cityTxt;  
 TextView descriptionTxt;  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
 title = itemView.findViewById(R.id.*service\_title\_lst*);  
 detailsBtn = itemView.findViewById(R.id.*show\_details\_btn*);  
  
 organizationTxt = itemView.findViewById(R.id.*organization\_details\_txt*);  
 categoryTxt = itemView.findViewById(R.id.*category\_details\_txt*);  
 countryTxt = itemView.findViewById(R.id.*country\_details\_txt*);  
 cityTxt = itemView.findViewById(R.id.*city\_details\_txt*);  
 descriptionTxt = itemView.findViewById(R.id.*description\_details\_txt*);  
 }  
  
 }  
  
 /\*  
 this method is for filtering items from 'servicesLst',  
 by the text that the user write in the search box at 'HomePageFragment'.  
 \*/  
 @SuppressLint("NotifyDataSetChanged")  
 public void filter(CharSequence charSequence) {  
 if (TextUtils.*isEmpty*(charSequence))  
 this.filteredList = new ArrayList<>(servicesLst);  
 this.filteredList = this.filteredList.stream()  
 .filter(data ->  
 data.getServiceName().toLowerCase().contains(charSequence) ||  
 data.getCategory().toLowerCase().contains(charSequence) ||  
 data.getCity().toLowerCase().contains(charSequence))  
 .collect(Collectors.*toList*());  
 updateServicesList(filteredList);  
 notifyDataSetChanged();  
 }  
}

SpinnerAdapter.java

package com.example.findyoursherutleumi.adapters;  
  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.Toast;  
  
import com.example.findyoursherutleumi.fragments.SignUpUserTypeFragment;  
  
*/\*\*  
 \* This class is for the spinner at 'SignUpUserTypeFragment',  
 \* where the user can choose the user type for signing up.  
 \*/*public class SpinnerAdapter implements AdapterView.OnItemSelectedListener {  
  
 public void onItemSelected(AdapterView<?> parent, View view, int pos, long id) {  
 SignUpUserTypeFragment.*chose* = pos;  
 Toast.*makeText*(parent.getContext(),  
 "You chose : " + parent.getItemAtPosition(pos).toString() + " sign up",  
 Toast.*LENGTH\_SHORT*).show();  
 }  
  
 public void onNothingSelected(AdapterView<?> parent) {  
 // Another interface callback  
 }  
  
  
}

**repositories**

ServicesPartialRepository.java

package com.example.findyoursherutleumi.repositories;  
  
import android.widget.Button;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.lifecycle.MutableLiveData;  
  
import com.example.findyoursherutleumi.R;  
import com.example.findyoursherutleumi.database.APIClient;  
import com.example.findyoursherutleumi.database.APIInterface;  
import com.example.findyoursherutleumi.models.ServicePartial;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
// Singleton pattern - can have only one object  
public class ServicesPartialRepository {  
  
 private static ServicesPartialRepository *instance*;  
 private final ArrayList<ServicePartial> dataSet = new ArrayList<>();  
 private Button detailsBtn;  
  
 APIInterface apiInterface;  
  
 public static ServicesPartialRepository getInstance(){  
 if (*instance* == null)  
 *instance* = new ServicesPartialRepository();  
 return *instance*;  
 }  
  
 public MutableLiveData<List<ServicePartial>> getServices(){  
 setServices();  
 MutableLiveData<List<ServicePartial>> data = new MutableLiveData<>();  
 data.setValue(dataSet);  
 return data;  
 }  
  
 public void setServices(){  
 apiInterface = APIClient.*getInstance*().create(APIInterface.class);  
 Call<List<ServicePartial>> call = apiInterface.getAllServicesPartially();  
 call.enqueue(new Callback<List<ServicePartial>>() {  
 @Override  
 public void onResponse(@NonNull Call<List<ServicePartial>> call, @NonNull Response<List<ServicePartial>> response) {  
 if (!response.isSuccessful())  
 Toast.*makeText*(detailsBtn.getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 assert response.body() != null;  
 dataSet.clear();  
 dataSet.addAll(response.body());  
 }  
 @Override  
 public void onFailure(@NonNull Call<List<ServicePartial>> call, @NonNull Throwable t) {  
 Toast.*makeText*(detailsBtn.getContext(), R.string.*connection\_failed\_toast*, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
  
  
  
  
}

FragmentToActivity.java

package com.example.findyoursherutleumi;  
  
import com.example.findyoursherutleumi.models.User;  
  
// send data from fragment to activity  
public interface FragmentToActivity {  
 void communicate(User user);  
  
}

MainActivity.java

package com.example.findyoursherutleumi;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentManager;  
import androidx.fragment.app.FragmentTransaction;  
  
import android.app.AlertDialog;  
import android.content.SharedPreferences;  
import android.content.res.Configuration;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
  
import com.example.findyoursherutleumi.fragments.HomePageFragment;  
import com.example.findyoursherutleumi.fragments.LoginFragment;  
import com.example.findyoursherutleumi.fragments.SettingsApplicantFragment;  
import com.example.findyoursherutleumi.fragments.SettingsFragment;  
import com.example.findyoursherutleumi.models.User;  
  
import java.util.Locale;  
  
*/\*\*  
 \* This class is the main activity,  
 \* which all the fragments is shown  
 \*/*public class MainActivity extends AppCompatActivity implements FragmentToActivity {  
  
 private static User *user*;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 loadLocale(); // load default language  
 setContentView(R.layout.*activity\_main*);  
  
 ActionBar actionBar = getSupportActionBar();  
 assert actionBar != null;  
 actionBar.setTitle(getResources().getString(R.string.*app\_name*));  
 if (*user* == null && savedInstanceState != null) { // sets user details  
 *user* = new User();  
 *user*.setEmail(savedInstanceState.getString("email"));  
 *user*.setUserTypeId(savedInstanceState.getInt("typeId"));  
 *user*.setUPassword("");  
 }  
 }  
  
 @Override  
 protected void onSaveInstanceState(@NonNull Bundle outState) {  
 // saves bundles after configuration changes  
 super.onSaveInstanceState(outState);  
 if(*user* != null) {  
 outState.putString("email", *user*.getEmail());  
 outState.putInt("typeId", *user*.getUserTypeId());  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 // create menu  
 getMenuInflater().inflate(R.menu.*nav\_menu*, menu);  
 MenuItem languagesItem = menu.findItem(R.id.*languages\_btn*);  
 MenuItem logoutItem = menu.findItem(R.id.*logout\_item*);  
 MenuItem settingsItem = menu.findItem(R.id.*settings\_item*);  
  
 languagesItem.setOnMenuItemClickListener(menuItem -> {  
 showChangeLanguageDialog();  
 return true;  
 });  
  
 logoutItem.setOnMenuItemClickListener(menuItem -> { // logout dialog  
 final AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);  
 builder.setMessage(R.string.*logout\_dialog*);  
 builder.setCancelable(true);  
  
 builder.setPositiveButton(R.string.*yes*, (dialogInterface, i) -> {  
 assert getFragmentManager() != null;  
 FragmentManager fragmentManager = getSupportFragmentManager();  
  
 // clears fragment's back stack to prevent user from going back after logging out.  
 int count = fragmentManager.getBackStackEntryCount();  
 for(int index = 0; index < count; ++index) {  
 fragmentManager.popBackStackImmediate();  
 }  
 Fragment newFragment = new LoginFragment();  
 FragmentTransaction transaction = fragmentManager.beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment).disallowAddToBackStack();  
 transaction.commit();  
  
 });  
 builder.setNegativeButton(R.string.*no*, (dialogInterface, i) -> dialogInterface.cancel());  
 AlertDialog alertDialog = builder.create();  
 alertDialog.show();  
 return true;  
 });  
  
 settingsItem.setOnMenuItemClickListener(menuItem -> { // user settings dialog  
 Fragment newFragment;  
 if (*user*.getUserTypeId() == 2){  
 newFragment = new SettingsFragment();  
 }  
 else {  
 newFragment = new SettingsApplicantFragment();  
 }  
 FragmentManager fragmentManager = getSupportFragmentManager();  
  
 Bundle bundle = new Bundle();  
 bundle.putString("email", *user*.getEmail());  
 bundle.putInt("typeId", *user*.getUserTypeId());  
 newFragment.setArguments(bundle);  
  
 FragmentTransaction transaction = fragmentManager.beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment);  
 transaction.addToBackStack(null);  
 transaction.commit();  
 return true;  
 });  
 return true;  
 }  
  
 // change language method  
 private void showChangeLanguageDialog() {  
 final String [] languagesLst = {"עברית", "English"};  
 AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);  
 builder.setTitle("Choose Language...");  
 builder.setSingleChoiceItems(languagesLst, -1, (dialogInterface, i) -> {  
 if (i == 0) {  
 setLocale("iw");  
 recreate();  
 }  
 else if (i == 1) {  
 setLocale("en");  
 recreate();  
 }  
 dialogInterface.dismiss();  
 });  
 AlertDialog alertDialog = builder.create();  
 alertDialog.show();  
 }  
  
 // sets the chosen language  
 private void setLocale(String lang) {  
 Locale locale = new Locale(lang);  
 Locale.*setDefault*(locale);  
 Configuration config = new Configuration();  
 config.setLocale(locale);  
 getBaseContext().getResources().updateConfiguration(config, getBaseContext().getResources().getDisplayMetrics());  
 SharedPreferences.Editor editor = getSharedPreferences("Settings", *MODE\_PRIVATE*).edit();  
 editor.putString("My\_Lang",lang);  
 editor.apply();  
 }  
  
 // loads the language  
 public void loadLocale(){  
 SharedPreferences preferences = getSharedPreferences("Settings", *MODE\_PRIVATE*);  
 String language = preferences.getString("My\_Lang","");  
 setLocale(language);  
 }  
  
 @Override  
 public void communicate(User user) {  
 // communicate with fragment to get user details from login fragment  
 this.*user* = user;  
 }  
  
 // get user from activity to homepage fragment  
 public static User getUser(){  
 return *user*;  
 }  
  
 /\* when back button pressed on device and the user is already logged in,  
 homepage fragment is restarting to load recyclerview  
 \*/  
 @Override  
 public void onBackPressed() {  
 super.onBackPressed();  
 assert getFragmentManager() != null;  
 if (getSupportFragmentManager().getBackStackEntryCount() > 1 && *user* != null) { // checks if user logged in  
 Fragment newFragment = new HomePageFragment();  
 FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();  
 transaction.replace(R.id.*fragmentContainerView*, newFragment)  
 .addToBackStack(null)  
 .commit();  
 }  
 }  
  
  
}

**10.2.3** קבצי XML

activity\_main.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="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity"  
 >  
  
 <androidx.fragment.app.FragmentContainerView  
 android:id="@+id/fragmentContainerView"  
 android:name="com.example.findyoursherutleumi.fragments.LoginFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
</LinearLayout>

fragment\_add\_new.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"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 tools:context=".fragments.AddNewServiceFragment">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/logo\_service\_details\_view"  
 android:layout\_width="236dp"  
 android:layout\_height="63dp"  
 android:layout\_marginBottom="15dp"  
 android:contentDescription="@string/logo"  
 android:src="@drawable/logo" />  
  
 <TextView  
 android:id="@+id/service\_details\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_marginBottom="20dp"  
 android:paddingStart="7dp"  
 android:paddingEnd="7dp"  
 android:text="@string/add\_new\_service"  
 android:textAlignment="textStart"  
 android:textColor="@color/light\_blue"  
 android:textSize="28sp"  
 android:textStyle="bold" />  
  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fillViewport="true"  
 tools:ignore="SpeakableTextPresentCheck">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <androidx.cardview.widget.CardView  
 android:id="@+id/service\_details\_card"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 app:cardBackgroundColor="@color/light\_blue"  
 app:cardCornerRadius="5dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/service\_title\_add"  
 android:layout\_width="95dp"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/service"  
 android:textColor="@color/white"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/service\_input\_add"  
 android:layout\_width="260dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/rounded\_corners"  
 android:hint="@string/service\_name\_hint"  
 android:inputType="text"  
 android:padding="5dp"  
 android:textSize="14sp"  
 tools:ignore="TouchTargetSizeCheck" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/organization\_title\_add"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/organization"  
 android:textColor="@color/white"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/organization\_input\_add"  
 android:layout\_width="252dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/rounded\_corners"  
 android:hint="@string/organization\_hint"  
 android:inputType="text"  
 android:padding="5dp"  
 android:textSize="14sp"  
 tools:ignore="TouchTargetSizeCheck" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/category\_title"  
 android:layout\_width="95dp"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:textSize="16sp"  
 android:text="@string/category"  
 android:textColor="@color/white"  
 android:textStyle="bold" />  
  
  
 <EditText  
 android:id="@+id/category\_input\_add"  
 android:layout\_width="260dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/rounded\_corners"  
 android:hint="@string/category\_hint"  
 android:inputType="text"  
 android:padding="5dp"  
 android:textSize="14sp"  
 tools:ignore="TouchTargetSizeCheck" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/country\_title"  
 android:layout\_width="95dp"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:textSize="16sp"  
 android:text="@string/country"  
 android:textColor="@color/white"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/country\_input\_add"  
 android:layout\_width="260dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/rounded\_corners"  
 android:hint="@string/country\_hint"  
 android:inputType="text"  
 android:padding="5dp"  
 android:textSize="14sp"  
 tools:ignore="TouchTargetSizeCheck" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/city\_title"  
 android:layout\_width="95dp"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:textSize="16sp"  
 android:text="@string/city"  
 android:textColor="@color/white"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/city\_input\_add"  
 android:layout\_width="260dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/rounded\_corners"  
 android:hint="@string/city\_hint"  
 android:inputType="text"  
 android:padding="5dp"  
 android:textSize="14sp"  
 tools:ignore="TouchTargetSizeCheck" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/address\_title"  
 android:layout\_width="95dp"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:textSize="16sp"  
 android:text="@string/address"  
 android:textColor="@color/white"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/address\_input\_add"  
 android:layout\_width="260dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/rounded\_corners"  
 android:hint="@string/address\_hint"  
 android:inputType="text"  
 android:padding="5dp"  
 android:textSize="14sp"  
 tools:ignore="TouchTargetSizeCheck" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/description\_service\_title"  
 android:layout\_width="95dp"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/description"  
 android:textColor="@color/white"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:counterEnabled="true"  
 app:counterMaxLength="200"  
 app:boxStrokeColor="@color/light\_gray"  
 app:counterTextAppearance="@style/counterText"  
 app:counterOverflowTextAppearance="@style/counterOverride">  
  
 <EditText  
 android:id="@+id/description\_input\_add"  
 android:layout\_width="260dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/rounded\_corners"  
 android:inputType="text"  
 android:padding="5dp"  
 android:textSize="14sp"  
 tools:ignore="TouchTargetSizeCheck,SpeakableTextPresentCheck" />  
 </com.google.android.material.textfield.TextInputLayout>  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="5dp"  
 android:gravity="center\_vertical"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/has\_apartment\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/has\_apartment"  
 android:textColor="@color/white"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
  
 <RadioGroup  
 android:id="@+id/apartment\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <RadioButton  
 android:id="@+id/has\_apartment\_yes\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/yes" />  
  
 <RadioButton  
 android:id="@+id/has\_apartment\_no\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/no" />  
 </RadioGroup>  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="5dp"  
 android:gravity="center\_vertical"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/is\_second\_year\_only\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/is\_second\_year\_only"  
 android:textColor="@color/white"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
  
 <RadioGroup  
 android:id="@+id/second\_year\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <RadioButton  
 android:id="@+id/second\_year\_yes\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/yes" />  
  
 <RadioButton  
 android:id="@+id/second\_year\_no\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/no" />  
 </RadioGroup>  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="5dp"  
 android:gravity="center\_vertical"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/is\_morning\_service\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/is\_morning\_service"  
 android:textColor="@color/white"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
  
 <RadioGroup  
 android:id="@+id/is\_morning\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <RadioButton  
 android:id="@+id/morning\_service\_yes\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/yes" />  
  
 <RadioButton  
 android:id="@+id/morning\_service\_no\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/no" />  
 </RadioGroup>  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="5dp"  
 android:gravity="center\_vertical"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/is\_evening\_service\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/is\_evening\_service"  
 android:textColor="@color/white"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
  
 <RadioGroup  
 android:id="@+id/is\_evening\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <RadioButton  
 android:id="@+id/evening\_service\_yes\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/yes" />  
  
 <RadioButton  
 android:id="@+id/evening\_service\_no\_radio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/no" />  
 </RadioGroup>  
 </LinearLayout>  
  
  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
  
 <Button  
 android:id="@+id/add\_new\_service\_btn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="30dp"  
 android:backgroundTint="@color/light\_blue"  
 android:text="@string/add\_btn" />  
  
 </LinearLayout>  
 </ScrollView>  
 </LinearLayout>  
  
</FrameLayout>

fragment\_applicant\_sign\_up.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=".fragments.ApplicantSignUpFragment">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/logo\_sign\_up\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="65dp"  
 android:layout\_marginLeft="30dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginRight="30dp"  
 android:layout\_marginBottom="15dp"  
 android:contentDescription="@string/logo"  
 android:src="@drawable/logo" />  
  
 <TextView  
 android:id="@+id/service\_details\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="20dp"  
 android:text="@string/applicant\_sign\_up"  
 android:textAlignment="center"  
 android:textColor="@color/light\_blue"  
 android:textSize="28sp"  
 android:textStyle="bold" />  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:contentDescription="@string/sign\_up"  
 android:fillViewport="true">  
  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/first\_name\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/first\_name"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/first\_name\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/first\_name\_hint"  
 android:inputType="text"  
 android:minHeight="48dp"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/last\_name\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/last\_name"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/last\_name\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/last\_name\_hint"  
 android:inputType="text"  
 android:minHeight="48dp"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/phone\_number\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/phone\_number"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/phone\_number\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/phone\_number\_hint"  
 android:inputType="phone"  
 android:minHeight="48dp"  
 android:textSize="16sp"  
 tools:ignore="TextFields" />  
  
 <TextView  
 android:id="@+id/city\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/city"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/city\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/city\_hint"  
 android:inputType="text"  
 android:minHeight="48dp"  
 android:textSize="16sp" />  
  
  
 <TextView  
 android:id="@+id/email\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/email"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/email\_address\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/email\_hint"  
 android:inputType="textEmailAddress"  
 android:minHeight="48dp"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/password\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/password"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/password\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:inputType="textPassword"  
 android:minHeight="48dp"  
 android:textSize="16sp"  
 android:hint="@string/password\_hint"  
 />  
  
 <Button  
 android:id="@+id/sign\_in\_btn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="50dp"  
 android:backgroundTint="@color/light\_blue"  
 android:text="@string/submit\_btn" />  
  
  
 </LinearLayout>  
 </ScrollView>  
  
 </LinearLayout>  
  
</FrameLayout>

fragments\_coordinator\_sign\_up.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=".fragments.CoordinatorSignUpFragment">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/logo\_sign\_up\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="65dp"  
 android:layout\_marginLeft="30dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginRight="30dp"  
 android:layout\_marginBottom="15dp"  
 android:contentDescription="@string/logo"  
 android:src="@drawable/logo" />  
  
 <TextView  
 android:id="@+id/coordinator\_sign\_up\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="20dp"  
 android:text="@string/coordinator\_sign\_up"  
 android:textAlignment="center"  
 android:textColor="@color/light\_blue"  
 android:textSize="28sp"  
 android:textStyle="bold" />  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:contentDescription="@string/sign\_up"  
 android:fillViewport="true">  
  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/first\_name\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/first\_name"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/first\_name\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/first\_name\_hint"  
 android:inputType="text"  
 android:minHeight="48dp"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/last\_name\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/last\_name"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/last\_name\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/last\_name\_hint"  
 android:inputType="text"  
 android:minHeight="48dp"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/phone\_number\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/phone\_number"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/phone\_number\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/phone\_number\_hint"  
 android:inputType="phone"  
 android:minHeight="48dp"  
 android:textSize="16sp"  
 tools:ignore="TextFields" />  
  
  
 <TextView  
 android:id="@+id/email\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/email"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/email\_address\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/email\_hint"  
 android:inputType="textEmailAddress"  
 android:minHeight="48dp"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/password\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/password"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/password\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:inputType="textPassword"  
 android:minHeight="48dp"  
 android:textSize="16sp"  
 android:hint="@string/password\_hint"  
 />  
  
 <TextView  
 android:id="@+id/organization\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/organization"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/organization\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/organization\_hint"  
 android:inputType="text"  
 android:minHeight="48dp"  
 android:textSize="16sp" />  
  
 <Button  
 android:id="@+id/sign\_in\_btn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="50dp"  
 android:backgroundTint="@color/light\_blue"  
 android:text="@string/submit\_btn" />  
  
  
 </LinearLayout>  
 </ScrollView>  
  
  
 </LinearLayout>  
</FrameLayout>

fragments\_home\_page.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:id="@+id/home\_page\_fragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:orientation="vertical"  
 tools:context=".fragments.HomePageFragment">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="63dp"  
 android:orientation="horizontal">  
  
 <ImageView  
 android:id="@+id/logo\_service\_details\_view"  
 android:layout\_width="236dp"  
 android:layout\_height="63dp"  
 android:layout\_gravity="start|top"  
 android:layout\_marginEnd="80dp"  
 android:layout\_marginBottom="15dp"  
 android:contentDescription="@string/logo"  
 android:src="@drawable/logo"  
 tools:ignore="ImageContrastCheck" />  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/add\_service\_btn"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="end|top"  
 android:layout\_weight="1"  
 android:clickable="true"  
 android:contentDescription="todo"  
 android:src="@android:drawable/ic\_input\_add"  
 app:backgroundTint="@color/light\_blue"  
 app:tint="#EEE"  
 tools:ignore="SpeakableTextPresentCheck,ImageContrastCheck" />  
  
 </LinearLayout>  
  
 <EditText  
 android:id="@+id/search\_bar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:autofillHints=""  
 android:background="@drawable/rounded\_corners\_light\_blue"  
 android:drawableEnd="@drawable/ic\_search"  
 android:hint="@string/search"  
 android:inputType="text|textNoSuggestions"  
 android:minHeight="48dp"  
 android:paddingStart="10dp"  
 android:paddingEnd="10dp"  
 android:textColor="@color/white"  
 android:textColorHint="@color/white"  
 tools:ignore="TextContrastCheck" />  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:orientation="vertical">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/services\_lst"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <TextView  
 android:id="@+id/no\_result\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_centerInParent="true"  
 android:gravity="center"  
 android:text="@string/no\_result"  
 android:textSize="20sp"  
 android:visibility="gone" />  
  
  
 </RelativeLayout>  
</LinearLayout>

fragment\_login.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"  
 android:layout\_margin="10dp"  
 android:id="@+id/login\_fragment"  
 tools:context=".fragments.LoginFragment">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/logo\_login\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="79dp"  
 android:layout\_margin="30dp"  
 android:contentDescription="@string/logo"  
 android:src="@drawable/logo" />  
  
 <TextView  
 android:id="@+id/sign\_in\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:text="@string/sign\_in"  
 android:textAlignment="center"  
 android:textColor="@color/light\_blue"  
 android:textSize="28sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/email\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/email"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/email\_address\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:hint="@string/email\_hint"  
 android:inputType="textEmailAddress"  
 android:minHeight="48dp"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/password\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginRight="50dp"  
 android:text="@string/password"  
 android:textColor="@color/light\_blue"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/password\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_gray"  
 android:ems="10"  
 android:inputType="textPassword"  
 android:minHeight="48dp"  
 android:textSize="16sp"  
 android:hint="@string/password\_hint"  
 />  
  
 <Button  
 android:id="@+id/sign\_in\_btn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="50dp"  
 android:layout\_marginRight="50dp"  
 android:backgroundTint="@color/light\_blue"  
 android:text="@string/login\_btn" />  
  
 <TextView  
 android:id="@+id/sign\_up\_clickable"  
 android:layout\_width="81dp"  
 android:layout\_height="48dp"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginRight="50dp"  
 android:layout\_marginBottom="40dp"  
 android:clickable="true"  
 android:minHeight="48dp"  
 android:text="@string/sign\_up"  
 android:textSize="17sp"  
 android:textStyle="bold" />  
  
 </LinearLayout>  
</FrameLayout>

fragment\_service\_details.xml

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout 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=".fragments.ServiceDetailsFragment">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/logo\_service\_details\_view"  
 android:layout\_width="236dp"  
 android:layout\_height="63dp"  
 android:layout\_marginBottom="15dp"  
 android:contentDescription="@string/logo"  
 android:src="@drawable/logo" />  
  
 <TextView  
 android:id="@+id/service\_details\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_marginBottom="20dp"  
 android:paddingStart="7dp"  
 android:paddingEnd="7dp"  
 android:text="Service Title"  
 android:textAlignment="textStart"  
 android:textColor="@color/light\_blue"  
 android:textSize="28sp"  
 android:textStyle="bold" />  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/light\_blue"  
 android:fillViewport="true"  
 android:padding="10dp"  
 tools:ignore="SpeakableTextPresentCheck">  
  
 <androidx.cardview.widget.CardView  
 android:id="@+id/service\_details\_card"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 app:cardBackgroundColor="@color/light\_blue">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/organization\_title\_details"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/organization"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/organization\_txt\_details"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/category\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/category"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/category\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/country\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/country"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/country\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/city\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/city"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/city\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/address\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/address"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/address\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/has\_apartment\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/has\_apartment"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/has\_apartment\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/is\_second\_year\_only\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/is\_second\_year\_only"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/is\_second\_year\_only\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/is\_morning\_service\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/is\_morning\_service"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/is\_morning\_service\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/is\_evening\_service\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/is\_evening\_service"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/is\_evening\_service\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/description\_service\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/description"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/description\_service\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/coordinator\_name\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/coordinator"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/coordinator\_name\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="5dp"  
 android:background="@drawable/rounded\_corners"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/coordinator\_contact\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/coordinator\_contact"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/coordinator\_contact\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="5dp"  
 android:text="TextView" />  
 </LinearLayout>  
  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
 </ScrollView>  
  
 </LinearLayout>  
  
</FrameLayout>

fragments\_settings.xml

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout 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:id="@+id/corrdinator\_settings\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".fragments.SettingsFragment">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/logo\_service\_details\_view"  
 android:layout\_width="236dp"  
 android:layout\_height="63dp"  
 android:layout\_gravity="start|top"  
 android:layout\_marginEnd="80dp"  
 android:layout\_marginBottom="15dp"  
 android:contentDescription="@string/logo"  
 android:src="@drawable/logo"  
 tools:ignore="ImageContrastCheck" />  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fillViewport="true"  
 tools:ignore="SpeakableTextPresentCheck">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/user\_edit\_details\_settings\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:text="@string/edit\_details"  
 android:textColor="@color/light\_blue"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/user\_first\_name\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/first\_name"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/c\_first\_name\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/first\_name\_hint"  
 android:inputType="text"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/user\_last\_name\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/last\_name"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/c\_last\_name\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/last\_name\_hint"  
 android:inputType="text"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/user\_email\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/email"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/c\_email\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/email\_hint"  
 android:inputType="textEmailAddress"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/user\_phone\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/phone\_number"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/c\_phone\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/phone\_number\_hint"  
 android:inputType="text"  
 android:minHeight="48dp" />  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/user\_password\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/password"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/c\_password\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/password\_hint"  
 android:inputType="textPassword"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/user\_organization\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/organization"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/c\_organization\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/organization\_hint"  
 android:inputType="text"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/update\_edit\_details\_btn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:backgroundTint="@color/light\_blue"  
 android:text="@string/update\_btn" />  
  
 <TextView  
 android:id="@+id/user\_edit\_services\_list\_settings\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:text="@string/edit\_services\_list"  
 android:textColor="@color/light\_blue"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:orientation="vertical">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/coordinator\_services\_list\_edit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="250dp"  
 android:layout\_marginBottom="30dp"/>  
  
 <TextView  
 android:id="@+id/no\_items\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_centerInParent="true"  
 android:gravity="center"  
 android:text="@string/no\_items"  
 android:textSize="20sp"  
 android:visibility="gone" />  
 </RelativeLayout>  
  
  
 <Button  
 android:id="@+id/c\_delete\_user\_edit\_details\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_marginBottom="30dp"  
 android:backgroundTint="@color/red"  
 android:text="@string/delete\_user\_btn" />  
  
 </LinearLayout>  
 </ScrollView>  
 </LinearLayout>  
  
</FrameLayout>

fragments\_settings\_applicant.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=".fragments.SettingsApplicantFragment">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/logo\_service\_details\_view"  
 android:layout\_width="236dp"  
 android:layout\_height="63dp"  
 android:layout\_gravity="start|top"  
 android:layout\_marginEnd="80dp"  
 android:layout\_marginBottom="15dp"  
 android:contentDescription="@string/logo"  
 android:src="@drawable/logo"  
 tools:ignore="ImageContrastCheck" />  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fillViewport="true"  
 tools:ignore="SpeakableTextPresentCheck">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/applicant\_edit\_details\_settings\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:text="@string/edit\_details"  
 android:textColor="@color/light\_blue"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/a\_first\_name\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/first\_name"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/a\_first\_name\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/first\_name\_hint"  
 android:inputType="text"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/a\_last\_name\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/last\_name"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/a\_last\_name\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/last\_name\_hint"  
 android:inputType="text"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/a\_phone\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/phone\_number"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/a\_phone\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/phone\_number\_hint"  
 android:inputType="phone"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/a\_email\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/email"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/a\_email\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/email\_hint"  
 android:inputType="textEmailAddress"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/a\_city\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/city"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/a\_city\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/city\_hint"  
 android:inputType="text"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/a\_password\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:text="@string/password"  
 android:textSize="16sp" />  
  
 <EditText  
 android:id="@+id/a\_password\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/password\_hint"  
 android:inputType="textPassword"  
 android:minHeight="48dp" />  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/update\_applicant\_edit\_details\_btn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:backgroundTint="@color/light\_blue"  
 android:text="@string/update\_btn" />  
  
 <Button  
 android:id="@+id/a\_delete\_user\_edit\_details\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_marginBottom="30dp"  
 android:backgroundTint="@color/red"  
 android:text="@string/delete\_user\_btn" />  
  
 </LinearLayout>  
 </ScrollView>  
 </LinearLayout>  
  
</FrameLayout>

fragment\_sign\_up\_user\_type.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=".fragments.SignUpUserTypeFragment">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/logo\_sign\_up\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="65dp"  
 android:layout\_marginLeft="30dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginRight="30dp"  
 android:layout\_marginBottom="15dp"  
 android:contentDescription="@string/logo"  
 android:src="@drawable/logo" />  
  
 <TextView  
 android:id="@+id/sign\_up\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="70dp"  
 android:text="@string/sign\_up"  
 android:textAlignment="center"  
 android:textColor="@color/light\_blue"  
 android:textSize="28sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/user\_type\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="43dp"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:text="@string/user\_type"  
 android:textColor="@color/light\_blue"  
 android:textSize="20sp"  
 android:textStyle="bold" />  
  
 <Spinner  
 android:id="@+id/user\_type\_spinner"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:minHeight="48dp"  
 android:spinnerMode="dropdown"  
 tools:ignore="SpeakableTextPresentCheck" />  
  
 <Button  
 android:id="@+id/sign\_up\_next\_btn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="130dp"  
 android:layout\_marginTop="150dp"  
 android:layout\_marginRight="130dp"  
 android:backgroundTint="@color/light\_blue"  
 android:text="@string/next\_btn" />  
  
  
 </LinearLayout>  
</FrameLayout>

item\_service.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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:id="@+id/service\_row\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:outlineSpotShadowColor="#D3D3D3">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="0dp"  
 android:layout\_height="220dp"  
 android:layout\_margin="5dp"  
 android:outlineAmbientShadowColor="#D3D3D3"  
 app:cardCornerRadius="15dp"  
 app:circularflow\_radiusInDP="30dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="60dp">  
  
 <LinearLayout  
 android:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/light\_blue"  
 android:gravity="center"  
 android:orientation="horizontal"  
 android:paddingStart="5dp"  
 android:paddingEnd="5dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <TextView  
 android:id="@+id/service\_title\_lst"  
 android:layout\_width="225dp"  
 android:layout\_height="44dp"  
 android:layout\_marginEnd="5dp"  
 android:gravity="center\_vertical"  
 android:padding="5dp"  
 android:text="Service Title"  
 android:textColor="@color/white"  
 android:textSize="16sp"  
 android:textStyle="bold"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform"  
 tools:ignore="TextContrastCheck" />  
  
 <Button  
 android:id="@+id/show\_details\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:backgroundTint="#2196F3"  
 android:gravity="center"  
 android:text="@string/show\_details"  
 android:textSize="12sp"  
 tools:ignore="TextContrastCheck" />  
 </LinearLayout>  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="60dp"  
 android:orientation="vertical"  
 android:padding="3dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/organization\_details\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="10dp"  
 android:paddingEnd="10dp"  
 android:text="@string/organization"  
 android:textStyle="bold"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform" />  
  
 <TextView  
 android:id="@+id/organization\_details\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/category\_details\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="10dp"  
 android:paddingEnd="10dp"  
 android:text="@string/category"  
 android:textStyle="bold"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform" />  
  
 <TextView  
 android:id="@+id/category\_details\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/country\_details\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="10dp"  
 android:paddingEnd="10dp"  
 android:text="@string/country"  
 android:textStyle="bold"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform" />  
  
 <TextView  
 android:id="@+id/country\_details\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform" />  
  
 <TextView  
 android:id="@+id/city\_details\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="10dp"  
 android:paddingEnd="10dp"  
 android:text="@string/city"  
 android:textStyle="bold"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform" />  
  
 <TextView  
 android:id="@+id/city\_details\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/description\_details\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="10dp"  
 android:paddingEnd="10dp"  
 android:text="@string/description"  
 android:textStyle="bold"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform" />  
  
 <TextView  
 android:id="@+id/description\_details\_txt"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:paddingStart="5dp"  
 android:paddingEnd="5dp"  
 android:text="TextView"  
 app:autoSizeMaxTextSize="30sp"  
 app:autoSizeMinTextSize="10sp"  
 app:autoSizeStepGranularity="2sp"  
 app:autoSizeTextType="uniform" />  
 </LinearLayout>  
  
  
 </LinearLayout>  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
 </androidx.cardview.widget.CardView>  
</androidx.constraintlayout.widget.ConstraintLayout>

item\_service\_coordinator.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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="wrap\_content">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <LinearLayout  
 android:layout\_width="258dp"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="5dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_weight="4"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/service\_title\_lst\_edit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="33dp"  
 android:gravity="center\_vertical"  
 android:paddingStart="5dp"  
 android:paddingEnd="5dp"  
 android:text="TextView"  
 android:textSize="22sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/service\_city\_lst\_edit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center\_vertical"  
 android:paddingStart="5dp"  
 android:paddingEnd="5dp"  
 android:text="TextView"  
 android:textSize="16sp" />  
 </LinearLayout>  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/delete\_service\_edit\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="5dp"  
 android:layout\_weight="1"  
 android:clickable="true"  
 app:backgroundTint="@color/light\_gray"  
 app:srcCompat="@android:drawable/ic\_menu\_close\_clear\_cancel"  
 tools:ignore="SpeakableTextPresentCheck" />  
  
 </LinearLayout>  
  
 </androidx.cardview.widget.CardView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

nav\_menu.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/languages\_btn"  
 android:icon="@drawable/ic\_baseline\_language\_24"  
 android:title="@string/languages\_bar"  
 app:showAsAction="ifRoom|withText" />  
  
 <item  
 android:id="@+id/more\_item"  
 android:icon="@drawable/ic\_baseline\_more\_vert\_24"  
 android:title="@string/more\_menu"  
 app:showAsAction="always">  
  
 <menu>  
 <item  
 android:id="@+id/settings\_item"  
 android:title="@string/settings" />  
  
 <item  
 android:id="@+id/logout\_item"  
 android:title="@string/logout" />  
 </menu>  
 </item>  
  
</menu>

**10.3** מסד הנתונים

**10.3.1** קוד יצירת טבלאות

CREATE DATABASE IF NOT EXISTS FindYourSherutLeumi;

USE FindYourSherutLeumi;

-- ................. tables ...................

CREATE TABLE user\_level(

level\_id INT PRIMARY KEY AUTO\_INCREMENT,

level\_name NVARCHAR(25) UNIQUE

);

CREATE TABLE applicant(

applicant\_id INT PRIMARY KEY AUTO\_INCREMENT,

first\_name NVARCHAR(25) NOT NULL,

last\_name NVARCHAR(25) NOT NULL,

phone\_number NVARCHAR(13) UNIQUE,

city NVARCHAR(25),

email NVARCHAR(40) NOT NULL UNIQUE,

u\_password CHAR(64) NOT NULL,

level\_user\_id INT DEFAULT 1,

FOREIGN KEY (level\_user\_id) REFERENCES user\_level(level\_id)

);

CREATE TABLE coordinator(

coordinator\_id INT PRIMARY KEY AUTO\_INCREMENT,

first\_name NVARCHAR(25) NOT NULL,

last\_name NVARCHAR(25) NOT NULL,

phone\_number NVARCHAR(13) UNIQUE,

email NVARCHAR(40) NOT NULL UNIQUE,

u\_password CHAR(64) NOT NULL,

organization\_name NVARCHAR(40),

level\_user\_id INT DEFAULT 2,

FOREIGN KEY (level\_user\_id) REFERENCES user\_level(level\_id)

);

CREATE TABLE service(

service\_id INT PRIMARY KEY AUTO\_INCREMENT,

service\_name NVARCHAR(40) NOT NULL,

organization\_name NVARCHAR(40) NOT NULL,

category NVARCHAR(50),

country NVARCHAR(30) NOT NULL,

city NVARCHAR(30) NOT NULL,

address NVARCHAR(30) NOT NULL,

has\_apartment BOOLEAN,

is\_second\_year\_only BOOLEAN,

is\_morning\_service BOOLEAN,

is\_evening\_service BOOLEAN,

description\_service NVARCHAR(200),

coordinator\_name NVARCHAR(25) NOT NULL,

coordinator\_id INT,

FOREIGN KEY (coordinator\_id) REFERENCES coordinator(coordinator\_id)

);

**10.3.2** קוד יצירת פרוצדורות

-- SERVICE

-- procedure that calls all the services from db

DELIMITER $$

CREATE PROCEDURE `get\_all\_services`()

BEGIN

SELECT \* FROM service;

END$$

DELIMITER ;

-- procedure that calls all the services partialy from db

DELIMITER $$

CREATE PROCEDURE `get\_all\_services\_partialy`()

BEGIN

SELECT service\_id,service\_name,organization\_name,category,country,city,description\_service FROM service;

END$$

DELIMITER ;

-- procedure that calls a specific service from db

DELIMITER $$

CREATE PROCEDURE `get\_service\_by\_id`(IN id INT)

BEGIN

SELECT \* FROM service WHERE service\_id = id;

END$$

DELIMITER ;

-- procedure that calls services by coordinator id from db

DELIMITER $$

CREATE PROCEDURE `get\_service\_by\_coordinator\_id`(IN id INT)

BEGIN

SELECT \* FROM service WHERE coordinator\_id = id;

END$$

DELIMITER ;

-- procedure that adds new service into the db

DELIMITER $$

CREATE PROCEDURE `add\_new\_service`(IN p\_service\_name NVARCHAR(40), IN p\_organization\_name NVARCHAR(40),

IN p\_category NVARCHAR(50), IN p\_country NVARCHAR(30), IN p\_city NVARCHAR(30), IN p\_address NVARCHAR(30),

IN p\_has\_apartment BOOLEAN, IN p\_is\_second\_year\_only BOOLEAN, IN p\_is\_morning\_service BOOLEAN,

IN p\_is\_evening\_service BOOLEAN, IN p\_description\_service NVARCHAR(200), IN p\_coordinator\_name NVARCHAR(25),IN p\_coordinator\_id INT)

BEGIN

INSERT INTO service(service\_name,organization\_name,category,country,city,address,has\_apartment,

is\_second\_year\_only,is\_morning\_service,is\_evening\_service,description\_service,coordinator\_name,coordinator\_id)

VALUES(p\_service\_name, p\_organization\_name, p\_category, p\_country,p\_city,p\_address,p\_has\_apartment,

p\_is\_second\_year\_only,p\_is\_morning\_service,p\_is\_evening\_service,p\_description\_service,p\_coordinator\_name,p\_coordinator\_id);

END$$

DELIMITER ;

-- procedure that updates service in db

DELIMITER $$

CREATE PROCEDURE `update\_service`(IN id INT, IN p\_service\_name NVARCHAR(40), IN p\_organization\_name NVARCHAR(40),

IN p\_category NVARCHAR(50), IN p\_country NVARCHAR(30), IN p\_city NVARCHAR(30), IN p\_address NVARCHAR(30),

IN p\_has\_apartment BOOLEAN, IN p\_is\_second\_year\_only BOOLEAN, IN p\_is\_morning\_service BOOLEAN,

IN p\_is\_evening\_service BOOLEAN, IN p\_description\_service NVARCHAR(200), IN p\_coordinator\_name NVARCHAR(25))

BEGIN

UPDATE service

SET service\_name = p\_service\_name,

organization\_name = p\_organization\_name,

category = p\_category,

country = p\_country,

city = p\_city,

address = p\_address,

has\_apartment = p\_has\_apartment,

is\_second\_year\_only = p\_is\_second\_year\_only,

is\_morning\_service = p\_is\_morning\_service,

is\_evening\_service = p\_is\_evening\_service,

description\_service = p\_description\_service,

coordinator\_name = p\_coordinator\_name

WHERE service\_id = id;

END$$

DELIMITER ;

-- procedure that deletes service by id from db

DELIMITER $$

CREATE PROCEDURE `delete\_service`(IN id INT)

BEGIN

DELETE FROM service WHERE service\_id = id;

END$$

DELIMITER ;

-- procedure that deletes services by coordinator id from db

DELIMITER $$

CREATE PROCEDURE `delete\_services\_by\_coordinator\_id`(IN id INT)

BEGIN

DELETE FROM service WHERE coordinator\_id = id;

END$$

DELIMITER ;

-- USER\_LEVEL

-- procedure that calls all the levels from db

DELIMITER $$

CREATE PROCEDURE `get\_all\_levels`()

BEGIN

SELECT \* FROM user\_level;

END$$

DELIMITER ;

-- procedure that calls a specific level from db

DELIMITER $$

CREATE PROCEDURE `get\_level\_by\_id`(IN id INT)

BEGIN

SELECT \* FROM user\_level WHERE level\_id = id;

END$$

DELIMITER ;

-- procedure that adds new level into the db

DELIMITER $$

CREATE PROCEDURE `add\_new\_level`(IN p\_name NVARCHAR(25))

BEGIN

INSERT INTO user\_level(level\_name)

VALUES(p\_name);

END$$

DELIMITER ;

-- procedure that updates level in db

DELIMITER $$

CREATE PROCEDURE `update\_level`(IN id INT, IN p\_name NVARCHAR(25))

BEGIN

UPDATE user\_level

SET level\_name = p\_name

WHERE level\_id = id;

END$$

DELIMITER ;

-- procedure that deletes level by id from db

DELIMITER $$

CREATE PROCEDURE `delete\_level`(IN id INT)

BEGIN

DELETE FROM user\_level WHERE level\_id = id;

END$$

DELIMITER ;

-- APPLICANT

-- procedure that calls all the applicants from db

DELIMITER $$

CREATE PROCEDURE `get\_all\_applicants`()

BEGIN

SELECT first\_name,last\_name,phone\_number,

city,email,u\_password

FROM applicant;

END$$

DELIMITER ;

-- procedure that calls a specific applicant from db

DELIMITER $$

CREATE PROCEDURE `get\_applicant\_by\_id`(IN id INT)

BEGIN

SELECT \* FROM applicant WHERE applicant\_id = id;

END$$

DELIMITER ;

-- procedure that calls a specific applicant from db by email

DELIMITER $$

CREATE PROCEDURE `get\_applicant\_by\_email`(IN applicant\_email NVARCHAR(40))

BEGIN

SELECT \* FROM applicant WHERE email = applicant\_email;

END$$

DELIMITER ;

-- procedure that adds new applicant into the db

DELIMITER $$

CREATE PROCEDURE `add\_new\_applicant`(IN p\_first\_name NVARCHAR(25),

IN p\_last\_name NVARCHAR(25),IN p\_phone\_number NVARCHAR(13),

IN p\_city NVARCHAR(25),IN p\_email NVARCHAR(40),IN p\_u\_password CHAR(64))

BEGIN

INSERT INTO applicant(first\_name,last\_name,phone\_number,city,email,u\_password)

VALUES(p\_first\_name,p\_last\_name,p\_phone\_number,p\_city,p\_email,p\_u\_password);

END$$

DELIMITER ;

-- procedure that updates applicant in db

DELIMITER $$

CREATE PROCEDURE `update\_applicant`(IN id INT,IN p\_first\_name NVARCHAR(25),

IN p\_last\_name NVARCHAR(25),IN p\_phone\_number NVARCHAR(13),

IN p\_city NVARCHAR(25),IN p\_email NVARCHAR(40),IN p\_u\_password CHAR(64))

BEGIN

UPDATE applicant

SET first\_name = p\_first\_name,

last\_name = p\_last\_name,

phone\_number = p\_phone\_number,

city = p\_city,

email = p\_email,

u\_password = p\_u\_password

WHERE applicant\_id = id;

END$$

DELIMITER ;

-- procedure that deletes applicant by id from db

DELIMITER $$

CREATE PROCEDURE `delete\_applicant`(IN id INT)

BEGIN

DELETE FROM applicant WHERE applicant\_id = id;

END$$

DELIMITER ;

-- COORDINATOR

-- procedure that calls all the coordinators from db

DELIMITER $$

CREATE PROCEDURE `get\_all\_coordinators`()

BEGIN

SELECT coordinator\_id,first\_name,last\_name,phone\_number,

email,u\_password,organization\_name

FROM coordinator;

END$$

DELIMITER ;

-- procedure that calls a specific coordinator from db

DELIMITER $$

CREATE PROCEDURE `get\_coordinator\_by\_id`(IN id INT)

BEGIN

SELECT \* FROM coordinator WHERE coordinator\_id = id;

END$$

DELIMITER ;

-- procedure that calls a specific coordinator from db by email

DELIMITER $$

CREATE PROCEDURE `get\_coordinator\_by\_email`(IN coordinator\_email NVARCHAR(40))

BEGIN

SELECT \* FROM coordinator WHERE email = coordinator\_email;

END$$

DELIMITER ;

-- procedure that adds new coordinator into the db

DELIMITER $$

CREATE PROCEDURE `add\_new\_coordinator`(IN p\_first\_name NVARCHAR(25),p\_last\_name NVARCHAR(25),

p\_phone\_number NVARCHAR(13),p\_email NVARCHAR(40),p\_u\_password CHAR(64),

p\_organization\_name NVARCHAR(40))

BEGIN

INSERT INTO coordinator(first\_name,last\_name,phone\_number,email,u\_password,organization\_name)

VALUES(p\_first\_name,p\_last\_name,p\_phone\_number,p\_email,p\_u\_password,p\_organization\_name);

END$$

DELIMITER ;

-- procedure that updates coordinator in db

DELIMITER $$

CREATE PROCEDURE `update\_coordinator`(IN id INT,IN p\_first\_name NVARCHAR(25),p\_last\_name NVARCHAR(25),

p\_phone\_number NVARCHAR(13),p\_email NVARCHAR(40),p\_u\_password CHAR(64),

p\_organization\_name NVARCHAR(40))

BEGIN

UPDATE coordinator

SET first\_name = p\_first\_name,

last\_name = p\_last\_name,

phone\_number = p\_phone\_number,

email = p\_email,

u\_password = p\_u\_password,

organization\_name = p\_organization\_name

WHERE coordinator\_id = id;

END$$

DELIMITER ;

-- procedure that deletes coordinator by id from db

DELIMITER $$

CREATE PROCEDURE `delete\_coordinator`(IN id INT)

BEGIN

DELETE FROM coordinator WHERE coordinator\_id = id;

END$$

DELIMITER ;

-------------------------------- general procedures ---------------------------------

-- this procedure finds user(applicant or coordinator) by email address

DELIMITER $$

CREATE PROCEDURE `get\_user\_by\_email`(IN p\_email NVARCHAR(40))

BEGIN

SELECT first\_name,last\_name,phone\_number,email,u\_password,level\_user\_id FROM coordinator

WHERE email = p\_email

UNION

SELECT first\_name,last\_name,phone\_number,email,u\_password,level\_user\_id FROM applicant

WHERE email = p\_email;

END$$

DELIMITER ;

1. סיכום תהליך הפרויקט
   1. תהליך הפיתוח

פיתוח תוכנה זו ארך כשלושה חודשים. התהליך היה בחלקו מתוכנן ובחלקו ספונטאני. החלקים המתוכננים היו בעיקר מסד הנתונים, הגדרת המחלקות ורוב המסכים.

התהליכים הספונטאניים נבעו מ- 3 סיבות עיקריות:

* תובנות שעלו תוך כדי פיתוח, וע"י חווית השימוש בזמן הבדיקות.
* התקלות בקשיים והצורך לחפש פתרון או לחלופין שינוי מבני.
* הרחבת ההבנה של שפת התכנות, נתנה יכולת גמישה יותר לפיתוח חכם יותר עם ממשק מתקדם יותר מאשר בתחילת התהליך.
  1. משאבים הנדרשים לפרויקט

1) מספר שעות מוקדש לפרויקט 500 שעות

2) ציוד נדרש: מחשב נייד.

3) תוכנות נדרשות:

* Android studio
* Visual studio code
* MySql Workbench

4) ידע חדש שנדרש ללמוד: חיבור בין שרת ללקוח

5) ספרות ומקורות מידע- אינטרנט, פורומים של מתכנתים, המנחה

* 1. תרשים גאנט

**תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי**

1. ביבליוגרפיה

* <https://geeksforgeeks.org/> - פלטפורמת מאמרים בנושא שפות תכנות.
* <https://developer.android.com/guide> - אתר הדוקומנטציה עבור פיתוח אנדרואיד.
* <https://stackoverflow.com/> - פורום שאלות/תשובות המוביל בעולם.

