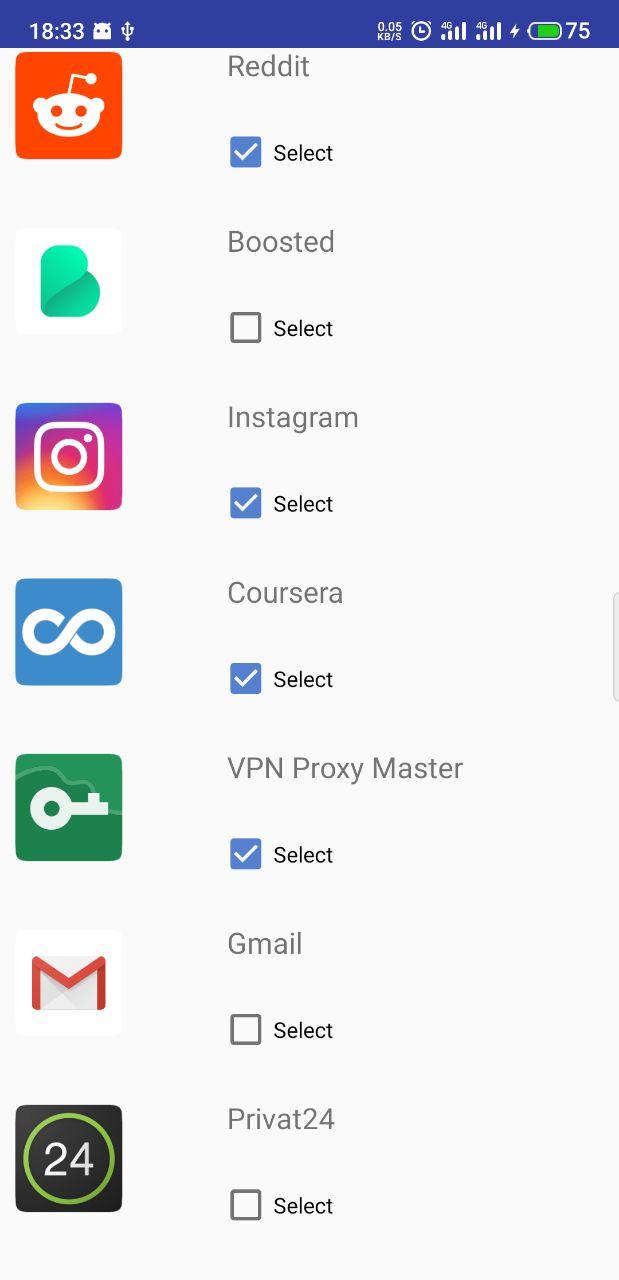
ADT у звичному розумінні цього поняття, наш проект не вимагає. Проте ми використовуємо базу даних, яка має певний інтерфейс, і її можна вважати, як певний абстрактний тип даних, бо користувач може взаємодіяти з нею через наданий інтерфейс.

Програмна база даних складається з двох менших баз даних. Перша - база даних, яка містить інформацію про користувачів. В ній будуть такі колонки: ID, Ім’я користувача, резервний пароль (який в усіх буде однаковий), інформація про його обличчя, статус користувача. Статус - true/false, якщо true, то обличчям цього користувача можна буде розблокувати програми. Пароль буде хешуватися, а інформація про обличчя зберігається у вигляді матриці (двовимірний масив розмірами 200x200 (фотографія стискається для кращої продуктивності)) векторів градієнту, оскільки використовується HOG метод розпізнавання обличчя (див. ДЗ№2). Для кращої точності, можна додавати декілька фотографій (наприклад під різними ракурсами та з різним освітленням). Вони зберігаються у черзі розміром 25 фотографій. Якщо юзер додасть більше як 25 фотографій, ті, які перші в черзі видаляються.

Друга - це база даних, в якій є список всіх заблокованих програм для різних профілів. Вона містить багато профілів, в кожного з яких є параметр який приймає true\false значення. true означає, що профіль вибраний. Кожен профіль містить такі колонки: ID, назва програм, назва пакунку, статус програми (locked\unlocked).

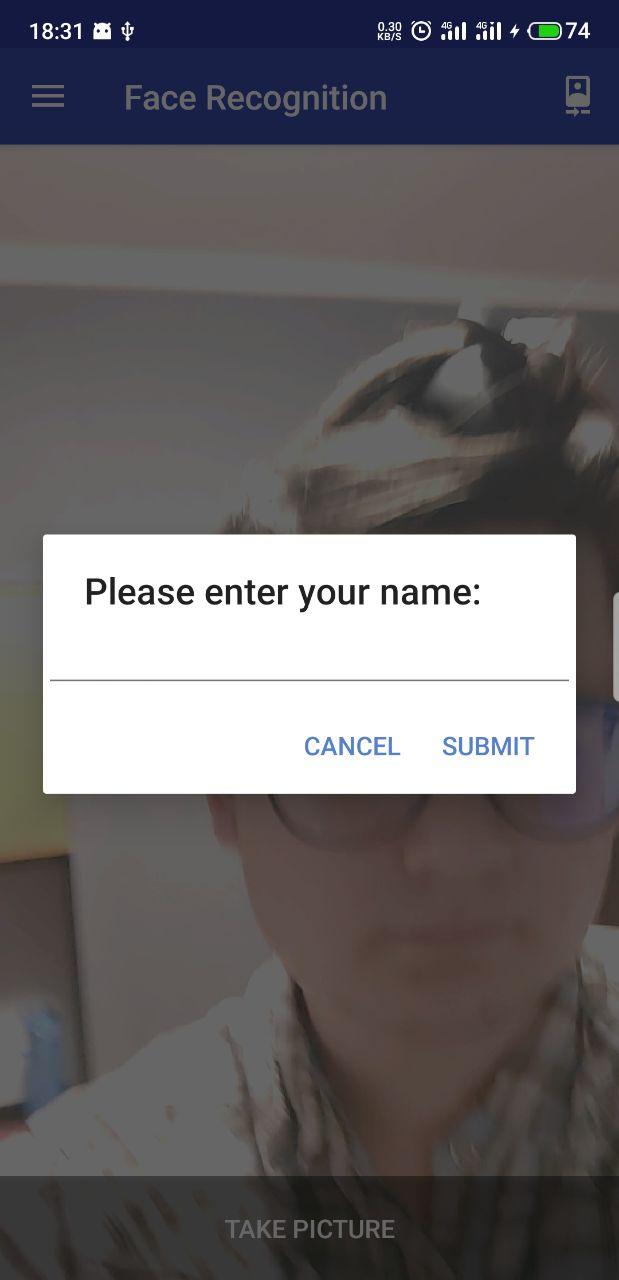
Інтерфейс виглядає так: при першому запуску створюється обліковий запис. При створенні першого користувача юзер вводить ім’я, резервний пароль (на випадок якщо розблокування по обличчю з якихось причин не доступне), та робить кілька фотографій себе для кращої точності. Після цього він може створити нового користувача нажавши на відповідну кнопку. Тоді йому треба буде ввести ім’я користувача та також зробити фото для бази даних. Юзер буде мати можливість обирати користувачів, себто по якому обличчю розблоковувати, але перед тим йому треба буде ввести резервний пароль, щоби підтвердити, що він адміністратор цього пристрою. Можна обрати декілька облич. Також користувач має змогу обирати профілі. Вони містяться з окремій базі даних.

Інтерфейс вибору програм для блокування:



Користувач може скролити і вибирати програми для блокування. В майбутньому можливо з’явиться пошук за іменем.

Додавання нового обличчя:



Юзер натискає Take Picture щоб зробити фотографію. Далі він вводить ім’я користувача

На момент написання ДЗ, програма зреалізована частково. Поки що є окрема програма для розпізнавання обличчя, і окрема для блокування програм. [База даних з профілями]

(https://github.com/Pavlik1400/FaceIt/blob/master/examples/android\_app\_locker/AppLocker4/app/src/main/java/com/example/applocker4/DataBase.java) має лише один профіль. Використовується SQLite, зокрема апі, яке надає Android Studio - клас SQLiteOpenHelper.

Цей клас надає можливість створювати таблички і стовпці. Потім можна взаємодіяти з базою даних через звичайні запити (виглядає це приблизно так: Cursor allApps = db.rawQuery("select \* from " + DataBase.FeedEntry.TABLE\_NAME,null); )

Для збереження даних про обличчя використовується клас PreferenceManager та його метод getDefaultSharedPreferences. Грубо кажучи це такий собі спільний для всіх програм словник, в який можна за ключем зберігати будь-які дані. В майбутньому заміниться на SQLite базу даних.

Для кращої читаємості коду, планується створити інтерфейс для бази даних, щоби не робити прямих запитів, а робити через зручні методи.

Кінцевий вигляд діаграми баз даних:

