

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
« ____ » _____ 2018 р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА

ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Мобільний додаток для роботи з заявками інтернет-компанії

Освітньо-кваліфікаційний рівень “бакалавр”
Напрямок підготовки 6.050102 – “Комп’ютерна інженерія”

Керівник проекту:

(підпис)

Щербакова М.Є.

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я.О.

(ініціали, прізвище)

Здобувач вищої освіти:

(підпис)

Плотніков Є.Є.

(ініціали, прізвище)

Група:

КІ-14з

Севєродонецьк 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 6.050102 Комп'ютерна інженерія

(шифр і назва)

Спеціальність _____

(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____

І.С. Скарга-Бандурова

« _____ » _____ 20 ____ р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Плотнікова Євгена Євгеновича

(прізвище, ім'я, по батькові)

1. Тема роботи Мобільний додаток для роботи з заявками
інтернет-компанії

керівник проекту (роботи) Щербакова Марина Євгенівна доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 14 05 2018 р. № 117/48

2. Строк подання студентом роботи 11.06.2018

3. Вихідні дані до роботи Матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області та постановка задачі; вибір засобів для розробки; розробка клієнтського додатку; охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	ст. викл.Критська Я.О.		

7. Дата видачі завдання _____

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Ознайомлення з предметною галуззю	30.04.18 – 05.05.18	
2	Аналіз існуючих аналогів	06.05.18 – 10.05.18	
3	Вибір засобів для розробки	11.05.18 – 19.05.18	
4	Розробка клієнтського додатку	20.05.18 – 05.06.18	
5	Розробка розділу «Охорона праці та безпека в надзвичайних ситуаціях»	06.06.18 – 07.06.18	
6	Оформлення пояснювальної записки	08.06.18 – 11.06.18	

Студент _____

(підпис)

Плотніков Є.Є.

(прізвище та ініціали)

Науковий керівник _____

(підпис)

Щербакова М.Є.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра складається з трьох розділів, містить 55 сторінок, 17 рисунків, 3 таблиці, 10 джерел.

Об'єкт розробки: клієнтський додаток для роботи з заявками інтернет - компанії.

Мета роботи: створення мобільного додатку для операційної системи (ОС) Android, що отримує заявки на обліковий запис кожного робітника, виконання їх і звіт в кінці дня.

В дипломному проекті виконано:

1. Проаналізовані переваги і недоліки існуючих програм для виконання заявок співробітниками інтернет-компаній.

2. Розроблений мобільний клієнтський додаток для роботи з інтернет-заявками.

Отримані наступні результати: мобільний додаток з формою авторизації, звітом в кінці дня та формою роботи з заявками.

Практичне значення, галузь застосування роботи: додаток може бути використаний для отримання та виконання заявок інтернет-компаніями з використанням пристроїв на ОС Android.

Ключові слова: ANDROID, ANDROID STUDIO, MYSQL, ОПЕРАЦІЙНА СИСТЕМА, ЗАЯВКИ, ІНТЕРНЕТ, ЗВІТ.

Умови одержання дипломного проекту: СНУ ім. В.Даля, пр. Центральний 59-А, м. Северодонецьк, 93400.

ЗМІСТ

ВСТУП.....	7
1.1 Загальні відомості про Android	8
1.2 Розробка програмного забезпечення для Android	8
1.3 Створення інтерфейсу користувача	10
1.4 Основні типи елементів інтерфейсу	11
1.5 Використання адаптерів.....	14
1.6 Технічне завдання на розробку	17
1.6.1 Постановка завдання на розробку мобільного додатку.....	17
2 ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ІНТЕРНЕТ-КОМПАНІЇ.....	18
2.1 Розробка структури сайту	18
2.1 Форма привітання та розробка меню мобільного додатка	20
2.2 Елементи меню.....	21
2.3 Реалізація переходу між екранами за допомогою методу intent.....	23
2.4 Основні процеси програм для Android	25
3 ОХОРОНА ПРАЦІ	27
3.1 Вступ	27
3.1 Аналіз стану умов праці.....	27
3.1.1 Вимоги до приміщення	27
3.1.2 Вимоги до організації місця праці	28
3.2 Виробнича санітарія	28
3.3 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу	29
3.4 Пожежна безпека	30
3.5 Електробезпека.....	32
3.6 Гігієнічні вимоги до параметрів виробничого середовища	33
3.6.1 Мікроклімат.....	33
3.6.2 Освітлення	33
3.7 Вентилювання	35

3.8 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій	35
3.9 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі)..	36
ВИСНОВКИ.....	40
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	41
ДОДАТОК А. Лістинг програми	42
ДОДАТОК Б. Комп'ютерна презентація	50

ВСТУП

У сучасному світі мобільні електронні пристрої поширюються дуже швидко. Потужності збільшуються в геометричній прогресії. З'являються нові технології, які дозволяють розмістити ще більші потужності на меншій площі. Компанії зі сфери розробки програмного забезпечення пропонують безліч засобів для розв'язання тих чи інших проблем. Проте часто виникає потреба розв'язати нетривіальну задачу або просто створити новий продукт. У такому разі єдине, що залишається – це розробка унікального продукту. Для того, щоб він виявився успішним, він повинен відповідати найвищим стандартам своєї галузі. Тому кожний створений продукт повинен бути потрібний користувачам, добре продуманий, спроектований та реалізований. Якщо потрібно, щоб продукт став поширений серед користувачів, він повинен бути також професійно представлений.

Взагалі, розробка будь-якого програмного продукту потребує великих затрат: як часу, так і ресурсів (людських та матеріальних). Тому під час розробки чогось нового, потрібно врахувати усі ризики.

Була обрана тема «Розробка програми для операційної системи Android», оскільки зараз це найпоширеніша операційна система для мобільних пристроїв. Це відносно нова система, тому кількість якісних програм для неї досить не велика. Багато сайтів та компанії потребують реалізацію їх продуктів для платформи Android.

Метою даної роботи було створити програму для виконання заявок інтернет-компанії її співробітниками. У програмі повинна бути реєстрація користувача та можливість доступу до інформації про заявки на виконання робіт.

1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальні відомості про Android

Android – операційна система від Google для мобільних пристроїв, таких як планшетні комп'ютери, комунікатори, мобільні телефони, камери та багато інших. Вона створена на базі ядра Linux. Android використовує для роботи віртуальну машину *Dalvik*.

Dalvik – віртуальна машина Java, створена під час розробки Android для мінімального використання ресурсів. Вона, як і звичайна віртуальна машина Java (JVM), інтерпретує байт-код в машинний код. Проте *Dalvik* використовує свій байт-код, який відрізняється від звичайного байт-коду Java. Причиною цього є те, що в Android використовується своя реалізація Java, відмінна від Java SE чи Java ME. Вихідний код конвертується у байт-код (файли з розширенням *.class*, які можуть бути об'єднані в архіви *.jar* – *Java Archive*). Після цього утиліта *dx* конвертує *.class* файли у файли з розширенням *.dex*, які використовує віртуальна машина на Android. При конвертації коду в байт-код всі ідентичні ресурси конвертуються тільки один раз для економії пам'яті. Так як саму віртуальну машину можна встановити майже на будь-якому пристрої, Android можна запускати навіть на електронних годинниках.

1.2 Розробка програмного забезпечення для Android

Для розробки програмного забезпечення для Android використовується *Android Software Development Kit (Android SDK)*. Код пишеться на Java. Для тестування програм можна використовувати як пристрій, підключений до комп'ютера, так і емулятор Android, який іде у пакеті з *Android SDK*. Також можна використовувати *Android Native Development Kit (Android NDK)*. *NDK* дозволяє писати програмне забезпечення для Android на C, C++ та інших

поширених мовах програмування. Проте у цьому разі при розробці не буде доступне відлагодження. Програми для Android можна розповсюджувати через *Android Market (Google Play)*.

ПЗ (програмне забезпечення) для Android зазвичай розробляється у програмі *Eclipse*. У пакеті *Android SDK* є версія даної програми зі всіма потрібними для розробки бібліотеками.

Щоб створити порожній проект, достатньо вибрати відповідний пункт меню. Після цього будуть створені базові папки та файли, необхідні для роботи з проектом. Створюються папки *src*, *gen*, *assets*, *bin*, *libs*, *res*. У папці *src* (source) знаходяться вихідні коди програми у файлах з розширенням *.java*. У папці *gen* (generated) знаходяться автоматично згенеровані файли. Вони присвоюють кожному логічному елементу програми унікальний ідентифікатор, зберігають дані про ресурси та інше. У папці *assets* знаходяться додаткові ресурси, якщо вони потрібні. У папці *bin* розташовані файли компільованого проекту. Також там знаходиться *.apk* файл, який використовується для встановлення програми на пристрої з Android. У папці *libs* (libraries) розміщені додаткові бібліотеки, якщо такі використовуються. Щоб додати додаткову бібліотеку, достатньо її завантажити та додати, використовуючи відповідний пункт меню. У останній згенерованій папці *res* (resources) знаходяться ресурси, потрібні для роботи програми. У цій папці розташовані дочірні папки *drawable*, *layout*, *menu*, *values* та інші. У папці *drawable* розміщують графічні ресурси, у папці *layout* – файли-макети програми, у папці *menu* – макети меню, у папці *values* – ресурси, такі як стрічки (*Strings*), кольори (*Colors*), розміри (*Dimens*), стилі макетів (*Styles*) та інші. Якщо для різних розширень потрібно використати різні ресурси, їх розміщують у відповідних папках з суфіксами. У папках з суфіксом – *hdpi* – ресурси та макети для великих екранів, – *ldpi* – не великих екранів, – *mdpi* – середніх за щільністю екранів. Щільність екрана вимірюється у *DPI* (dots per inch) – крапках на дюйм. Відповідні розміри: ~120 dpi – невеликий екран (*-ldpi*), ~160 dpi – середній екран (*-mdpi*), ~240 dpi – великий екран (*-hdpi*). З появою екранів з великим розширенням (*Retina*) були введені ще два розміри екранів:

~260 dpi – дуже великий екран (*-xhdpi*) та >260 dpi – надвеликий екран (*-xxhdpi*). Також використовують суфікси, які залежать не від щільності, а від розміру екрану:

– *small*, – *medium*, – *large*, – *xlarge*.

Якщо програма використовує різні макети для різних положень екранів, файли відповідних макетів розміщують у різних папках. За вертикальне положення відповідає суфікс – *port*, за горизонтальне – *land*. При повороті екрану програма буде використовувати ресурси з відповідних директорій. Для локалізації програми при розробці достатньо продублювати папку *values* з відповідними суфіксами. Суфікс папки *values* повинен відповідати міжнародному позначенню країни (ресурси для української версії розміщують у папці *values-ua*, англійської – *values-en*). Якщо для країни не вказана відповідна папка, програма буде використовувати ресурси з папки *values* (без суфікса).

Отже, структура програм для Android добре продумана. Можна легко додати локалізації чи адаптувати програму для різних розмірів екранів. Це надзвичайно важливо, так як існують сотні різних пристроїв на Android з різною конфігурацією та різними розмірами екранів.

1.3 Створення інтерфейсу користувача

Інтерфейс користувача створюється за допомогою виглядів (*View*), груп виглядів (*ViewGroup*), та макетів (*Layouts*) (рис. 1.1). Об'єкти вигляду зазвичай є елементами інтерфейсу користувача, такими як кнопки, поля для введення інформації, текстові поля та інші. Групи виглядів – невидимі об'єкти. Вони є контейнерами, які визначають вигляд дочірніх виглядів, наприклад, сіткою (*grid*) чи вертикальним списком (*vertical list*). Весь інтерфейс базується на ієрархії виглядів. Самі вигляди зберігаються у файлах формату *.xml*.

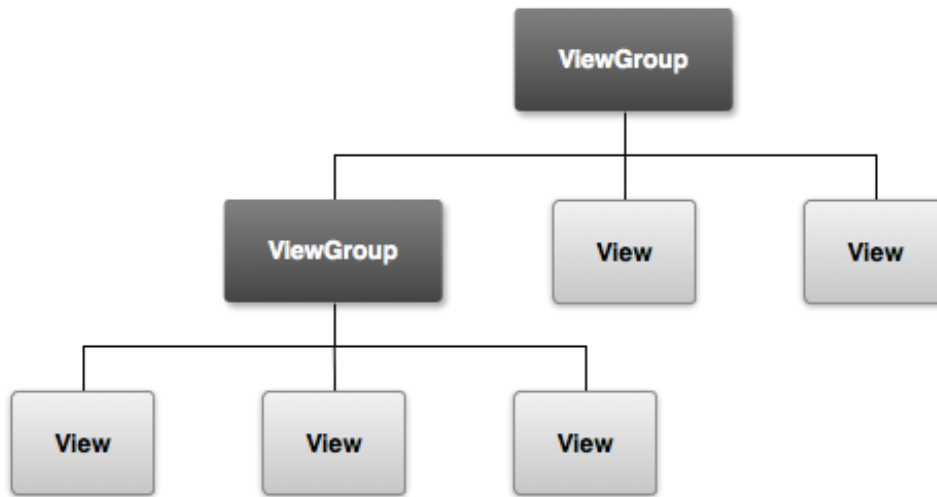


Рисунок 1.1 – Ієрархія виглядів деякого макету

Вигляди та макети можуть бути задані як у файлах з розширенням *.xml*, так і програмним шляхом. Але розробники Android пропонують визначати елементи інтерфейсу у файлах макетів, оскільки це забезпечує більш гнучке пристосування до різних екранів: достатньо створити альтернативний макет для іншого розширення та розмістити його у відповідній папці, після чого програму можна буде використовувати на іншому екрані. Визначення інтерфейсу програмним шляхом вимагає дублювання частини коду. Також у програмі *Eclipse* є візуальний редактор, який дозволяє візуально редагувати файли макетів. Якщо їх визначати програмним шляхом, візуальний редактор буде недоступний.

1.4 Основні типи елементів інтерфейсу

Linear Layout – лінійне розташування. У ньому об'єкти інтерфейсу розміщуються горизонтально або вертикально у тій послідовності, у якій вони задані. Розробник сам визначає, як відображати об'єкти у блоці *LinearLayout* за допомогою атрибуту *android:orientation*, який може приймати усього два значення: *horizontal*(горизонтально) або *vertical* (вертикально).

При використанні *LinearLayout* доступний ще один важливий атрибут – *android:layout_weight*. Він дозволяє задати розміри дочірніх елементів

пропорційно батьківському (рис. 1.2).

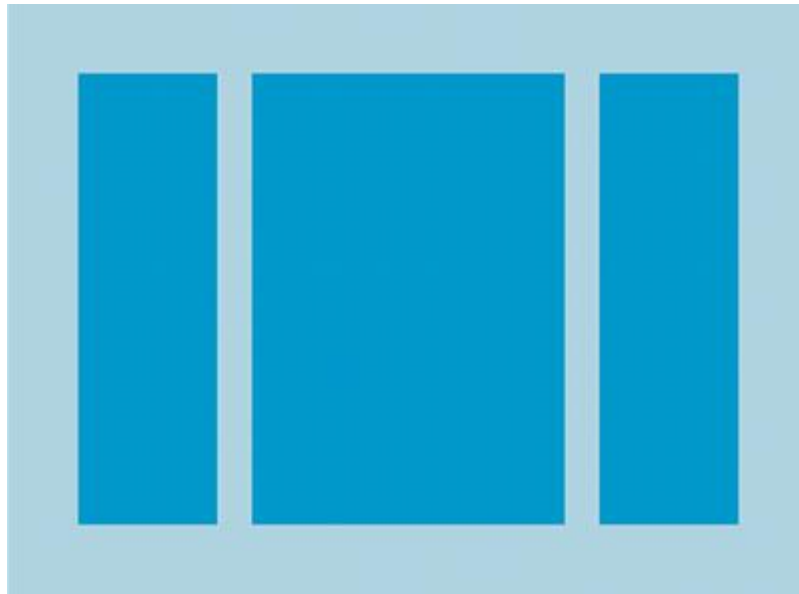


Рисунок 1.2 – Приклад лінійного розташування

Relative Layout – відносне розташування. У ньому об'єкти інтерфейсу розташовуються відносно один одного (справа, зліва, під та інші). Для визначення розміщення використовуються такі атрибути: *android:layout_alignParentTop* (приймає значення *true* або *false*, якщо *true*, то верхня межа елемента рівняється з верхньою межею батьківського елемента), *android:layout_centerVertical* (якщо *true*, то дочірній елемент розміщується вертикально по середині), *android:layout_below* (аргументом виступає ідентифікатор елемента, під яким буде розташований даний елемент) та багато інших. Всі атрибути такого типу приймають значення *boolean* (*true* або *false*) або ідентифікатор елемента, відносно якого вони мають розташовуватись (рис. 1.3).



Рисунок 1.3 – Відносне розташування

ListView – список, група виглядів (*View Group*), у якій всі елементи розташовуються у списку з можливістю прокрутки по вертикалі. Для додання елементів у список використовується клас адаптерів (*Adapter*), який отримує інформацію з деякого джерела (масив, список і т. д.), та конвертує її у елементи списку, які додаються до елементу *ListView* (рис. 1.4).



Рисунок 1.4 – Приклад списку елементів

Grid View – сітка, група виглядів, яка відображає елементи у вигляді двовимірної сітки, елементи якої можуть прокручуватися вертикально. Для додання елементів до елементу *GridView* також використовують клас адаптерів (рис. 1.5).

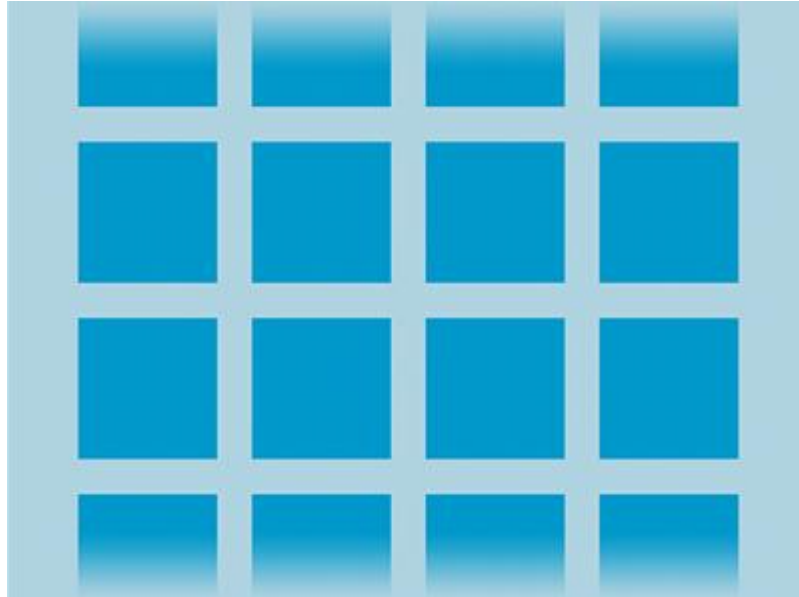


Рисунок 1.5 – Приклад сітки

1.5 Використання адаптерів

Коли інформація для деякого елементу інтерфейсу динамічна, використовують спеціалізований шаблонний клас адаптерів `Adapter`, та його дочірні класи: `Simple Adapter`, `SimpleCursorAdapter` та `ArrayAdapter<T>`, де `T` – тип елементів, які будуть використовуватись у списку. Адаптер виступає у ролі буфера між списком та джерелом даних.

```
ArrayAdapter adapter = new ArrayAdapter<String> (this,  
android.R.layout.simple_list_item_1, myStringArray);
```

Даний адаптер використовує елементи типу *String*. Макет для кожного елемента має ідентифікатор *android.R.layout.simple_list_item_1*. Даний макет стандартний та розташований у стандартній бібліотеці, на що вказує префікс *android.R.layout.simple_list_item_1*. Якщо потрібно використати власний макет, досить його створити та вказати у якості ідентифікатора *R.layout.my_own_item_view*. Файл *R.java* у директорії *gen* містить автоматично згенеровані ідентифікатори для усіх ресурсів та макетів проекту, тому досить звернутися до нього, щоб отримати адресу потрібного нам елемента. Першим аргументом конструктору є елемент типу *Context* (контекст активного вікна програми), останнім – масив елементів типу *String*. Також масив стрічок можна передати у адаптер використовуючи метод *createFromResource()*, у випадку, якщо він (масив) заданий у файлі ресурсів.

Розглянемо ще один варіант адаптеру – *SimpleCursorAdapter*. При використанні цього адаптеру, ми повинні вказати ідентифікатор для елемента нашого списку, який, на відміну від попереднього прикладу, може містити декілька елементів (минулий містив тільки один елемент, якому присвоювалося значення елемента *String* з масиву). Після того, як макет елемента створений, при зберіганні файлу йому автоматично буде наданий ідентифікатор, який, буде збережений у файлі *R.java*.

Припустимо, що у макеті є два текстових поля для імені та телефону абонентів зі списку контактів. Потрібно сформувати масив даних, та відповідний йому масив ідентифікаторів, куди будуть записані дані:

```
String[] fromColumns = {ContactsContract. Data.DISPLAY_NAME,
ContactsContract. CommonDataKinds. Phone.NUMBER};
int[] toViews = {R.id.display_name, R.id.phone_number};
```

Щоб отримати контакти, потрібно використати об'єкт *Cursor*, що поверне список пар *ім'я – телефон*, які будуть використані для запису до макету та формування списку. Далі потрібно створити сам адаптер, вибрати список, та передати йому об'єкт адаптера.

```
SimpleCursorAdapter adapter = new SimpleCursorAdapter (this,
R.layout.person_name_and_number, cursor, fromColumns, toViews, 0);
ListView listView = getListView();
listView.setAdapter(adapter);
```

Spinner – елемент інтерфейсу, який створює меню, що випадає. За замовчуванням обраний перший елемент. Для заповнення цього елемента також використовуються адаптери: *ArrayAdapter*, *CursorAdapter* та *SpinnerAdapter*.

Важливо, що діяльність перезапускається при повороті екрану. Тому якщо це може вплинути на коректну роботу програми, потрібно обробити такі ситуації. Якщо потрібно, можна заборонити перезапуск діяльності. Для цього досить вказати у файлі маніфесту атрибут для даної діяльності.

Файл маніфесту (*AndroidManifest.xml*) обов'язково повинен бути у кожній програмі. Він містить важливу інформацію для системи Android про програму, її вимоги, які діяльності вона включає, назву програми, які можливості пристрою вона буде використовувати, як вона буде взаємодіяти з іншими програмами, а також мінімальну та максимельну версію *API*, потрібну для коректної роботи системи.

1.6 Технічне завдання на розробку

1.6.1 Постановка завдання на розробку мобільного додатку

Головним завданням проектування було створення мобільного додатку для управління заявками Інтернет-компанії. Панель керування додатка має використовуватися для редагування, управління заявками, перевірки кількості заявок та додавання виконаних заявок. Повинна дозволяти співробітникам компанії оперативно виконувати, отримувати нові заявки та звітувати відносно виконаних заявок.

Мобільний додаток для інтернет-компанії повинен мати такі елементи:

- форма входу під обліковий запис;
- зручне меню для вибору заявки;
- форма звіту в кінці дня;
- кнопка виходу з облікового запису.

2 ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ІНТЕРНЕТ-КОМПАНІЇ

2.1 Розробка структури сайту

Розробка структури мобільного додатку інтернет-компанії виконувалася з урахуванням наступних вимог.

Перше завдання: надати кожному працівнику свій логін і пароль.

Друге завдання: зробити форму входу; після вводу даних додаток буде переходити на наступну сторінку.

Третє завдання: вітання своїх робітників у програмі.

Четверте завдання: зробити меню, в якому буде добавлено заявки, звіт і вихід з програми.

Проаналізувавши і узагальнивши ці дані, створимо структуру мобільного додатку Інтернет-компанії (рис. 2.1).

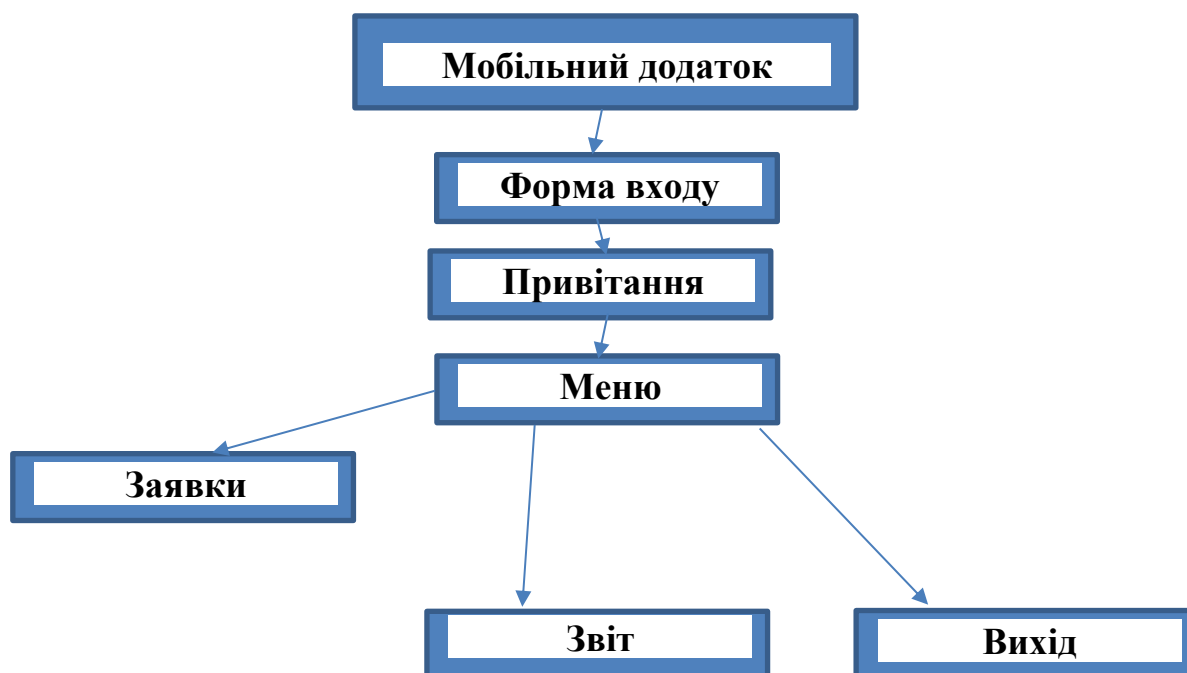


Рисунок 2.1 - Структурна схема мобільного додатку

При відкритті програми відображається форма авторизації. Форма зроблена для того, щоб кожен працівник мав свій обліковий запис.

Відповідно до структури була спроектована форма входу (рис. 2.2).

Вона містить логін і пароль для входу в облікову запис.

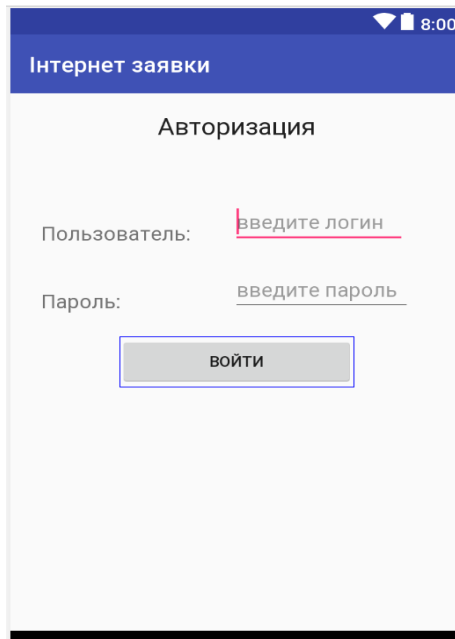


Рисунок 2.2 – Форма входу

Якщо в поле логіну і паролю введенні невірні дані, то зменшується кількість спроб увійти в свій обліковий запис. Якщо кількість спроб закінчилась, то вхід на деякий час блокується (рис. 2.3).

```
// В другом случае выдаем сообщение с ошибкой:
else {
    Toast.makeText(getApplicationContext(), text: "Неправильные данные!", Toast.LENGTH_SHORT).show();
    numberOfRemainingLoginAttempts--;

    // Делаем видимыми текстовые поля, указывающие на количество оставшихся попыток:
    attempts.setVisibility(View.VISIBLE);
    numberOfAttempts.setVisibility(View.VISIBLE);
    numberOfAttempts.setText(Integer.toString(numberOfRemainingLoginAttempts));

    // Когда выполнено 3 безуспешных попытки залогиниться,
    // делаем видимым текстовое поле с надписью, что все пропало и выставляем
    // кнопку настройку невозможности нажатия setEnabled(false):
    if (numberOfRemainingLoginAttempts == 3) {
        login.setEnabled(false);
        loginLocked.setVisibility(View.VISIBLE);
        loginLocked.setBackgroundColor(Color.RED);
        loginLocked.setText("Введите правильный логин или пароль");
    }
}
```

Рисунок 2.3 – Блокування форми входу

Якщо будуть введенні коректний логін і пароль, то після натискання на кнопку «Вхід» будуть проаналізовані поле логіну і паролю, буде зроблений вхід у свій обліковий запис, та буде перехід на наступний екран.

2.1 Форма привітання та розробка меню мобільного додатка

При переході на другу сторінку мобільного додатку ми бачимо форму привітання, яка зображена на рис 2.4.

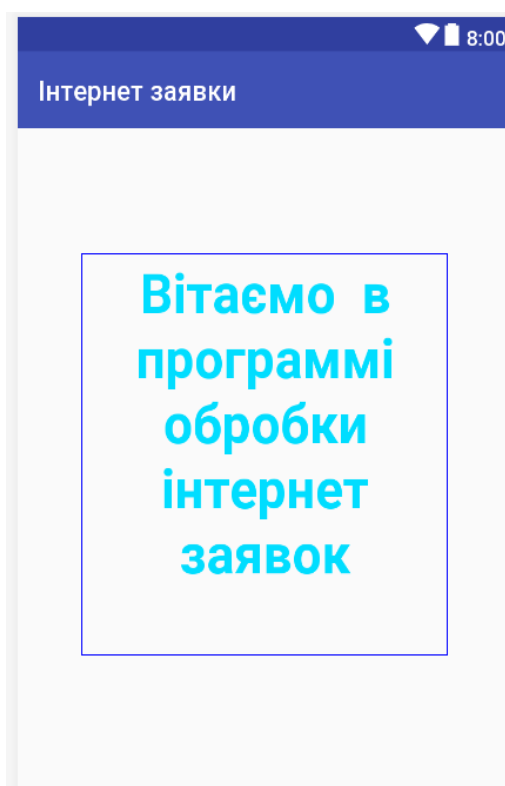


Рисунок 2.4 – Форма привітання

Також у правому верхньому кутку розташована кнопка з меню, в якій реалізовано заявки, звіт та вихід з облікового запису. При натисканні на кожний елемент меню виконується перехід на відповідну категорію.

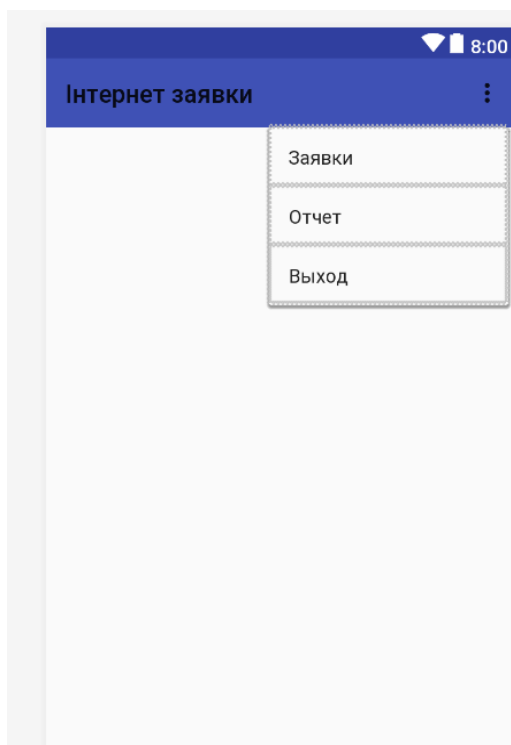


Рисунок 2.5 – Меню мобильного додатку

2.2 Элементы меню

При переходе на первый элемент меню открываются заявки, які повинна виконати людина, яка зайшла в свій обліковий запис.

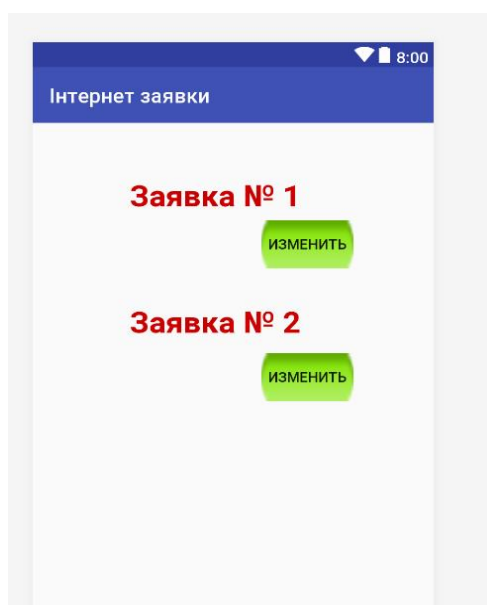


Рисунок 2.6 – Заявки

Для того щоб виконати заявку, необхідно натиснути на кнопку «Змінити» та подивитися після цього, що потрібно тобі виконати. Після того, як ти виконав заявку, натиснути на кнопку «Виконано» та переходити до виконання наступної заявки.

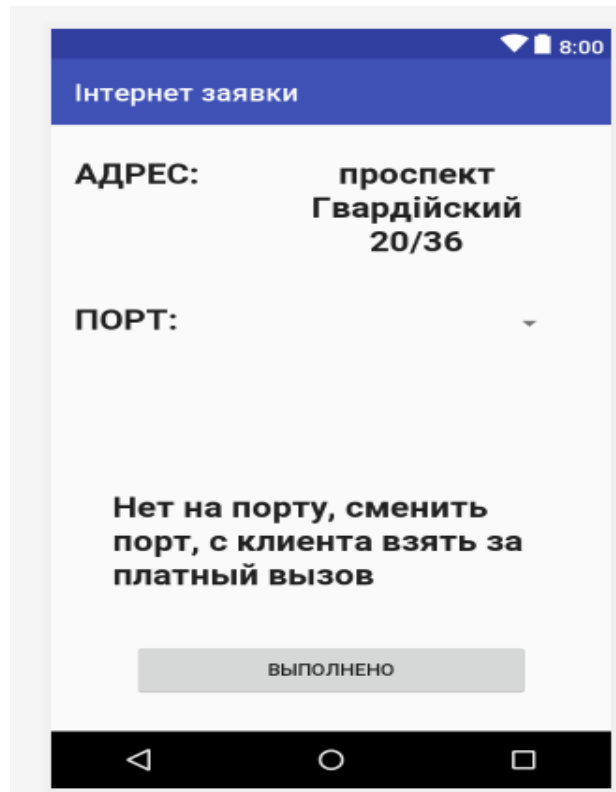


Рисунок 2.6 – Зміна заявки

При переході на другий елемент меню відкривається звіт, в який записуються всі заявки, які співробітник виконав за весь день, та після натискання на кнопку йде збереження звіту.

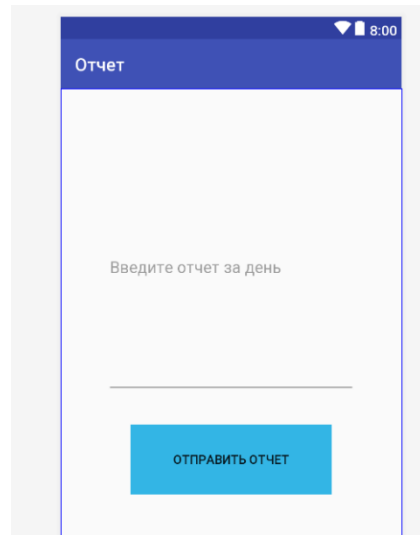


Рисунок 2.7 - Форма звітності

2.3 Реалізація переходу між екранами за допомогою методу `intent`

`Intent` є об'єктом обміну повідомленнями, за допомогою якого можна запросити виконання дії у компонента іншої програми. Незважаючи на те, що об'єкти `Intent` спрощують обмін даними між компонентами у декількох аспектах, в основному вони використовуються в трьох ситуаціях: Для запуску операції: Компонент `Activity` є один екран в додатку. Для запуску нового екземпляра компонента `Activity` необхідно передати об'єкт `Intent` методу `startActivity ()`. Об'єкт `Intent` описує, для чого потрібно запустити, а також містить всі інші необхідні дані.

Якщо після завершення операції від неї вимагається отримати результат, викличте метод `startActivityForResult ()`. Ваша операція отримає результат у вигляді окремого об'єкта `Intent` в зворотному виклику методу `onActivityResult ()` операції.

Є два типи об'єктів `Intent`:

- Явні об'єкти `Intent` вказують компонент, який потрібно запустити, по імені (повне ім'я класу). Явні об'єкти `Intent` зазвичай використовуються для

запуску компонента з вашого власного додатка, оскільки вам відомо ім'я класу операції або служби, яку необхідно запустити. Наприклад, можна запустити нову операцію у відповідь на дію користувача або запустити службу, щоб завантажити файл у фоновому режимі.

- Неявні об'єкти Intent не містять імені конкретного компонента. Замість цього вони в цілому оголошують дію, яке потрібно виконати, що дає можливість компоненту з іншої програми обробити цей запит. Наприклад, якщо потрібно показати користувачеві місце на карті, то за допомогою неявного об'єкту Intent можна запросити, щоб це зробило інше додаток, в якому така можливість передбачена.

Коли створено неявний об'єкт Intent, система Android знаходить відповідний компонент шляхом порівняння вмісту об'єкта Intent з фільтрами Intent, оголошеними в файлах маніфесту інших додатків, наявних на пристрої. Якщо об'єкт Intent збігається з фільтром Intent, система запускає цей компонент і передає йому об'єкт Intent. Якщо відповідними виявляються кілька фільтрів Intent, система виводить діалогове вікно, де користувач може вибрати програму для додавання коментарів.

Фільтр Intent є виразом у файлі маніфесту додатки, яке вказує типи об'єктів Intent, які міг би приймати компонент. Наприклад, оголосивши фільтр Intent для операції, ви даєте іншим додаткам можливість безпосередньо запускати вашу операцію за допомогою деякого об'єкту Intent. Точно так же, якщо ви не оголосите будь-які фільтри Intent для операції, то її можна буде запустити тільки за допомогою явного об'єкта Intent.

Об'єкт Intent містить інформацію, на підставі якої система Android визначає, який компонент потрібно запустити (наприклад, точне ім'я компонента або категорію компонентів, які повинні отримати цей об'єкт Intent), а також відомості, які необхідні компоненту-одержувачу, щоб належним чином виконати дію (а саме - виконувану дію і дані, з якими його потрібно виконати).

Основні відомості, що містяться в об'єкті Intent: ім'я компонента Ім'я компонента, який потрібно запустити. Ця інформація є необов'язковою, але

саме вона і робить об'єкт Intent явним. Її наявність означає, що об'єкт Intent слід доставити тільки компоненту програми, визначеним на ім'я. При відсутності імені компонента об'єкт Intent є неявним, а система визначає, який компонент отримає цей об'єкт Intent за іншими відомостями, які в ньому містяться (наприклад, за дією, даними і категорії - см. Опис далі). Тому, якщо вам потрібно запустити певний компонент зі свого додатка, слід вказати його ім'я.

2.4 Основні процеси програм для Android

Під час користування програмою, об'єкти типу *Activity* («діяльність») змінюють свої стани, зупиняються, продовжуються та завершуються. Під час написання програми, можна вказати, як повинна поводитись діяльність впродовж своєї роботи. Наприклад, якщо ви створюєте онлайн відео програвач, можна переривати з'єднання з Інтернетом та призупиняти мовлення коли користувач згортає програму та переключається на іншу. Це забезпечить економію інтернет-трафіку користувача. Коли користувач знову переключиться на вашу програму, діяльність може відновити підключення та продовжити відео-мовлення з моменту, на якому воно було зупинене. Принцип роботи Activity продемонстровано на рис. 2.8.

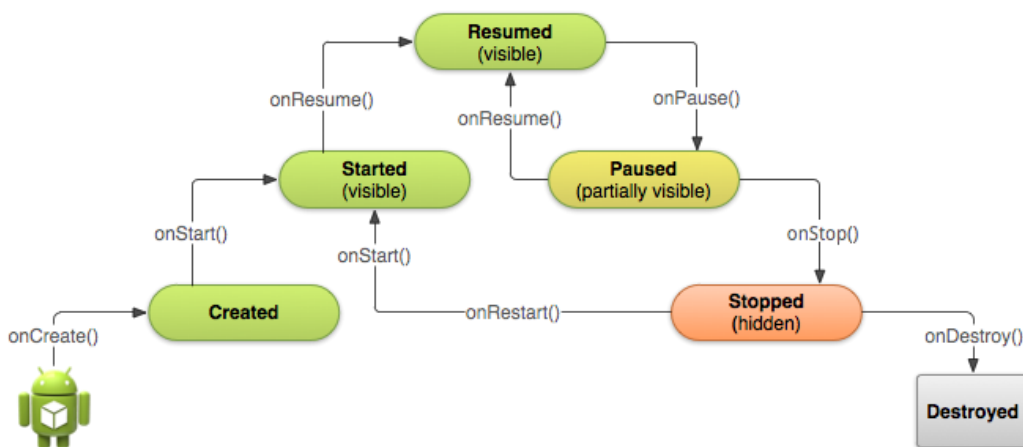


Рисунок 2.8 – Зміна станів активності

Кожна активність починається з методу *void onCreate (Bundle instance)*. Для того, щоб створити свою активність, потрібно створити клас, який буде розширювати клас *Activity*, та перевизначити метод *onCreate()*. У ньому потрібно викликати батьківський конструктор, та передати йому об'єкт типу *Bundle*, який буде передано в перевизначений метод *onCreate()* автоматично при запуску діяльності. Далі, якщо дана діяльність використовує макет, потрібно його вказати, передавши методу *setContentView()* ідентифікатор макету. Тільки після цього з ним можна буде працювати. Простий приклад діяльності:

```
public class Start extends Activity {
    @Override
    public void onCreate (Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView (R.layout.activity_start);
    }
}
```

Важливо, що діяльність перезапускається при повороті екрану. Тому якщо це може вплинути на коректну роботу програми, потрібно обробити такі ситуації. Якщо потрібно, можна заборонити перезапуск діяльності. Для цього досить вказати у файлі маніфесту атрибут для даної діяльності.

Файл маніфесту (*AndroidManifest.xml*) обов'язково повинен бути у кожній програмі. Він містить важливу інформацію для системи Android про програму, її вимоги, які діяльності вона включає, назву програми, які можливості пристрою вона буде використовувати, як вона буде взаємодіяти з іншими програмами, а також мінімальну та максимальну версію *API*, потрібну для коректної роботи системи.

3 ОХОРОНА ПРАЦІ

3.1 Вступ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної роботи бакалавра було створення системи он-лайн замовлення їжі, і як результат було створено онлайн - систему, яка виконує всі зазначені вимоги. Так як в процесі проектування використовувався персональний комп'ютер, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде розроблятися та використовуватися розроблена система.

3.1 Аналіз стану умов праці

3.1.1 Вимоги до приміщення

Таблиця 3.1 - Розміри приміщення

Найменування	Значення
Довжина, м	4,4
Ширина, м	2,8
Висота, м	2,5
Площа, м ²	12,32
Об'єм, м ³	30,8

Згідно з ДСН 3.3.6.042-99 [18] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для забезпечення потрібного рівного освітленості кімната має вікно та

систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

3.1.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за ДСанПіН 3.3.2.007-98 [19] (табл. 3.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 3.2 - Характеристики робочого місця

Найменування параметра	Фактичне Значення	Нормативне Значення
Висота робочої поверхні, мм	700	680 ÷ 800
Висота простору для ніг, мм	650	не менше 600
Ширина простору для ніг, мм	540	не менше 500
Глибина простору для ніг, мм	660	не менше 650
Висота поверхні сидіння, мм	420	400 ÷ 500
Ширина сидіння, мм	410	не менше 400
Глибина сидіння, мм	420	не менше 400
Висота поверхні спинки, мм	500	не менше 300
Ширина опорної поверхні спинки, мм	400	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	750	700 ÷ 800

3.2 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання

необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

3.3 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 3.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-1.28-10 [22], яке встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга $U=+220\text{В} \pm 5\%$;
- робочий струм $I=2\text{А}$;
- споживана потужність $P=350\text{ Вт}$.

Таблиця 3.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна Оцінка	Нормативні Документи
1	2	3	4
Фізичні:			
підвищена або знижена вологість повітря	-//-	3	[18]
підвищений рівень напруги електричної	-//-	3	[20] [21]

мережі, замикання якої може відбутися через тіло людини			
Психофізіологічні:			
1	2	3	4
нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- формулювання теми; - пошук інформацію про предметну область; - пошук інформації про наявні аналоги; - проектування структур та алгоритмів; - виконання роботи; - оформлення записки.	4	[21] [22]
фізичні (статичне - сидіння)	порушення умов організації робочого часу (безперервна робота)	2	[19] [22]

Робочі місця в обов'язковому порядку повинні відповідати вимогам до санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, що затверджені постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [19].

3.4 Пожежна безпека

Висока щільність елементів в електронних схемах призводить до значного підвищення температури окремих вузлів (80...100°C). При проходженні електричного струму по провідниках і деталей виділяється тепло, що в умовах їх високої щільності може привести до перегріву, і може служити причиною запалювання ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів дії температури може викликати порушення ізоляції і привести до короткого замикання між струмоведучими частинами обладнання (шини,

електроди).

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), надійно захищені діелектричними щитками та/або сітками з метою недопущення потрапляння працівника під напругу.

В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

1) поліамід - матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420 °С;

2) полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С;

3) склотекстоліт ДЦ - матеріал друкарських плат, важкогорючий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

4) пластикат кабельний №489 - матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1;

5) деревина - будівельний і обробний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255 °С, температура самозаймання 399 °С.

Простори усередині приміщень в межах, яких можуть утворюватися або знаходиться пожежонебезпечні речовини і матеріали відповідно до НАПБ Б.03.002-2007 [23] відносяться до пожежонебезпечної зони класу П-ІІа. Це обумовлено тим, що в приміщенні знаходяться тверді горючі та важкозаймисті речовини та матеріали. Приміщенню, у якому розташоване робоче місце, присвоюється ІІ ступень вогнестійкості.

Причинами можливого загоряння і пожежі можуть бути:

1) несправність електроустановки;

2) конструктивні недоліки устаткування;

3) коротке замикання в електричних мережах;

4) запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол [24].

3.5 Електробезпека

Виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

3.6 Гігієнічні вимоги до параметрів виробничого середовища

3.6.1 Мікроклімат

Мікроклімат робочих приміщень - це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт 1а. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають ДСН 3.3.6.042-99 [18] і наведені в табл. 3.4:

Таблиця 3.4 – Норми мікроклімату робочої зони об'єкту

Період Року	Категорія Робіт	Температура С ⁰	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	Легка-1а	22-24	40-60	0,1
Тепла	Легка-1а	23-25	40-60	0,1

3.6.2 Освітлення

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше -1/8, в побутових - 1/10:

$$S_b = \left(\frac{1}{5} / \frac{1}{10} \right) * S_n \quad (4.1)$$

де S_b – площа віконних прорізів, м²;

S_n – площа підлоги, м².

$$S_n = a \cdot b = 4,4 \cdot 2,8 = 12,32 \text{ м}^2 ,$$

$$S = 1/10 \cdot 25 = 1,232 \text{ м}^2 .$$

Приймаємо 1 вікно площею $S=1,6 \text{ м}^2$.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 4,4 м, ширина 2,8 м, світильниками ЛПО2П, оснащеними лампою типа ЛБ (одна - 80 Вт) з світловим потоком 5400 лм. Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E * S * Z * K}{F * U * M} \quad (4.2)$$

де E - нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S - освітлювана площа, м^2 ; $S = 12,32 \text{ м}^2$;

Z - поправочний коефіцієнт світильника ($Z = 1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп) приймаємо рівним 1,1;

K - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U - коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п.

- 0,575 M - число люмінесцентних ламп в світильнику - 1;

F - світловий потік лампи - 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 * 12,32 * 1,15 * 1,5}{5400 * 0,575 * 1} \approx 2,0$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, оснащених лампами типа ЛБ (одна - 80 Вт) зі світловим потоком 5400 лм.

3.7 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при V приміщення $> 40 \text{ м}^3$ на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП. Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

3.8 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала $2/3$ нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря:

а) якщо об'єм приміщення 20 м^3 , то потрібно подати не менш як 30 м^3 /год повітря;

б) якщо об'єм приміщення у межах від 20 до 40 м^3 , то потрібно подати не

менш як 20 м³/год повітря;

в) якщо об'єм приміщення становить понад 40 м³, допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі;
- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;
- не тягнути за мережевий кабель, щоб витягти вилку з розетки;
- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;
- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;
- не залишати включені електроприлади без нагляду;

3.9 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [25], приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контуру заземлення повинен мати не більше 4 Ом.

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = \frac{R_0 * R_{пр.з.}}{R_{пр.з.} - R_0} \quad (4.3)$$

де $R_{пр.з.}$ - опір природних заземлювачів;

R_0 - допустимий опір заземлення.

Якщо природні заземлювачі відсутні, то $R_{шт.з.} = R_0$.

Підставивши числові значення у формулу (А.3), отримуємо:

$$R_{шт.з.} = \frac{4 * 40}{40 - 4} \approx 40 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом·м. Приблизне значення питомого опору глини приймаємо $\rho = 40$ Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту, $R_{розр.}$, Ом·м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в.}$, і горизонтальних $R_{розр.г.}$, Ом·м за формулою:

$$R_{розр.} = \Psi * \rho \quad (4.4)$$

де Ψ - коефіцієнт сезонності для вертикальних заземлювачів І кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $R_{розр.в.} = 1,7$ і горизонтальних $R_{розр.г.} = 5,5$ Ом·м

$$R_{розр.в.} = 1,7 * 40 = 68 \text{ Ом/м}$$

$$R_{розр.г.} = 5,5 * 40 = 220 \text{ Ом/м}$$

4) Розраховується опір розтікання струму вертикального заземлювача R_B , Ом, за (4.5).

$$R_B = \frac{R_{розр.в.}}{2 * \pi * l_B} * \left(\ln \frac{2 * l_B}{d_{ст}} + \frac{1}{2} * \ln \frac{4 * t + l_B}{4 * t - l_B} \right) \quad (4.5)$$

де l_B - довжина вертикального заземлювача (для труб - 2-3 м; $l_B = 3$ м);

$d_{ст}$ - діаметр стержня (для труб - 0,03-0,05 м; $d_{ст} = 0,05$ м);

t - відстань від поверхні землі до середини заземлювача, яка визначається за ф. (4.6):

$$t = h_B + \frac{l_B}{2} \quad (4.6)$$

де h_B - глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м}$$

$$R_B = \frac{68}{2 * \pi * 3} * \left(\ln \frac{2 * 3}{0,05} + \frac{1}{2} * \ln \frac{4 * 2,3 + 3}{4 * 2,3 - 3} \right) = 18,5 \text{ Ом}$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання η_B :

$$n = \frac{2 * R_B}{R_0} = \frac{2 * 18,5}{4} = 9,25 \quad (4.7)$$

6) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання n_B , шт:

$$n_B = \frac{2 * R_B}{R_0 * \eta_B} = \frac{2 * 18,5}{4 * 0,57} = 16,2 \approx 16 \quad (4.8)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача l_c , м:

$$l_c = 1,05 * L_B * (n_B - 1) \quad (4.9)$$

де L_B - відстань між вертикальними заземлювачами, (прийняти за $L_B = 3\text{м}$);
 n_B - необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 * 3 * (16 - 1) \approx 48\text{м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) R_G , Ом:

$$R_G = \frac{P_{розр.г}}{2 * \pi * l_c} * \ln \frac{2 * l_c^2}{d_{cm} * h_G} \quad (4.10)$$

де d_{cm} - еквівалентний діаметр смуги шириною b , $d_{cm} = 0,95b$, $b = 0,15$ м;

h_G - глибина закладання горизонтальних заземлювачів (0,5 м);

l_c - довжина з'єднувальної стрічки горизонтального заземлювача l_c , м

$$R_G = \frac{220}{2 * \pi * 48} * \ln \frac{2 * 48^2}{0,95 * 0,15 * 0,5} = 8,1 \text{ Ом}$$

9) Визначається коефіцієнт використання горизонтального заземлювача η_c . відповідно до необхідної кількості вертикальних заземлювачів n_B . Коефіцієнт використання з'єднувальної смуги $\eta_c = 0,3$ (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{заг}} = \frac{R_B * R_{\Gamma}}{R_B * \eta_c + R_{\Gamma} * n_B * \eta_B} \quad (4.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{\text{заг}} \leq R_0$, а саме:

$$R_{\text{заг}} = \frac{18,5 * 8,1}{18,5 * 0,3 + 8,1 * 16 * 0,57} = 1,9 \leq R_0$$

ВИСНОВКИ

Тільки додатки можуть зробити будь-яку операційну систему придатною для роботи, розваг, виходу в Інтернет, перегляду веб-сторінок і багато чого іншого, що перетворює звичайний телефон в маленький кишеньковий комп'ютер з повним набором функціональних можливостей. В дипломному проєкті був розроблений клієнтський додаток, що дозволяє робітникам інтернет-компанії виконувати заявки та здавати звіт в кінці дня за допомогою пристроїв з ОС Android.

Основні можливості розробленої програми:

- обліковий запис для кожного робітника;
- форма звітності в кінці дня;
- форма для роботи з заявками інтернет-компанії.

Також ця програма може бути модернізована за допомогою бази даних Інтернет компанії, в якій буде інформація про кожного клієнта, і на сервер буде відправлятися звіт в кінці дня.

Для розробки додатку була використана мова програмування Java і середовище програмування Android Studio 3.1. Зовнішній вигляд активностей (форм) додатку зберігається у файлах xml-формату.

Були розроблені заходи з охорони праці та безпеки в надзвичайних ситуаціях.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Dave MacLean Pro Android 5 / Dave MacLean, Satya Komatineni, Grant Allen. - Springer Science+Business Media, New York, 2015. – 813 pp.
2. Flanagan D. JavaScript: The Definitive Guide, Sixth Edition / David Flanagan. - Sebastopol : O'Reilly Media, Inc., 2011. – 1098 p.
3. Revill L. jQuery 2.0 Development Cookbook / Leon Revill. - Birmingham : Packt Publishing Ltd, 2014. – 410 p.
4. Neuburg M. Programming iOS 7. Fourth Edition / Matt Neuburg . - Sebastopol : O'Reilly Media, Inc., 2014. – 929 p.
5. Gargenta M. Learning Android, Second Edition / Marko Gargenta, Masumi Nakamura. – Sebastopol : O'Reilly Media, Inc., 2014. – 288 p.
6. Программирование под Android. Уроки разработки, примеры кода [Электронный ресурс]. - Режим доступа: <http://androiddocs.ru/pattern-viewholder-v-realizacii-spiska-listview/>
7. Жидецький В.Ц., Джигирей В.С ., Мельников О.В. Основи охорони праці , 2000 – 350 с.
8. Закон України «Про пожежну безпеку»
9. Android design helper [Електронний ресурс]. - Режим доступу: <http://angrytools.com/>
10. Введение в программирование для Android [Электронный ресурс]. - Режим доступа: <https://startandroid.ru/ru/>

ДОДАТОК А

Лістинг програми

Файл «Main activity.java»

```

package com.example.evgeny.startdiplom;

import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

// меню для программы

public class MainActivity extends AppCompatActivity {

    // Объявляем об использовании следующих объектов:
    private EditText username;
    private EditText password;
    private Button login;
    private TextView loginLocked;
    private TextView attempts;
    private TextView numberOfAttempts;

    // Число для подсчета попыток залогиниться:
    int numberOfRemainingLoginAttempts = 3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Связываемся с элементами нашего интерфейса:
        username = (EditText) findViewById(R.id.edit_user);
        password = (EditText) findViewById(R.id.edit_password);
        login = (Button) findViewById(R.id.button_login);
        loginLocked = (TextView) findViewById(R.id.login_locked);
        attempts = (TextView) findViewById(R.id.attempts);
        numberOfAttempts = (TextView) findViewById(R.id.number_of_attempts);

        numberOfAttempts.setText(Integer.toString(numberOfRemainingLoginAttempts));
    }

    // Обрабатываем нажатие кнопки "Войти":
    public void Login(View view) {

        // Если введенные логин и пароль будут словом "admin",
        // показываем Toast сообщение об успешном входе:
        if (username.getText().toString().equals("admin") &&
            password.getText().toString().equals("admin")) {
            Toast.makeText(getApplicationContext(), "Вход

```

```

выполнен!", Toast.LENGTH_SHORT).show();

    // Выполняем переход на другой экран:
    Intent intent = new Intent(MainActivity.this, Second.class);
    startActivity(intent);
}

// В другом случае выдаем сообщение с ошибкой:
else {
    Toast.makeText(getApplicationContext(), "Неправильные
данные!", Toast.LENGTH_SHORT).show();
    numberOfRemainingLoginAttempts--;

    // Делаем видимыми текстовые поля, указывающие на количество
оставшихся попыток:
    attempts.setVisibility(View.VISIBLE);
    numberOfAttempts.setVisibility(View.VISIBLE);

numberOfAttempts.setText(Integer.toString(numberOfRemainingLoginAttempts));

    // Когда выполнено 3 безуспешных попытки залогиниться,
// делаем видимым текстовое поле с надписью, что все пропало и
выставляем
    // кнопке настройку невозможности нажатия setEnabled(false):
if (numberOfRemainingLoginAttempts == 3) {
    login.setEnabled(false);
    loginLocked.setVisibility(View.VISIBLE);
    loginLocked.setBackgroundColor(Color.RED);
    loginLocked.setText("Введите правильный логин или пароль");
    }
}
}
}

```

Файл «Activity_main.xml»

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin">

    <TextView
        android:id="@+id/Login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="Авторизация"
        android:textAppearance="?android:attr/textAppearanceLarge"
        tools:layout_editor_absoluteX="115dp"
        tools:layout_editor_absoluteY="49dp" />

    <TextView
        android:id="@+id/user_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```

        android:layout_alignParentLeft="true"
        android:layout_below="@+id/Login"
        android:layout_marginTop="75dp"
        android:layout_marginLeft="10dp"
        android:text="Пользователь:"
        android:textAppearance="?android:attr/textAppearanceMedium" />

<EditText
    android:id="@+id/edit_user"
    android:layout_width="148dp"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/user_text"
    android:layout_marginLeft="35dp"
    android:layout_toRightOf="@+id/user_text"
    android:hint="введите логин">

    <requestFocus />
</EditText>

<TextView
    android:id="@+id/password_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/user_text"
    android:layout_below="@+id/user_text"
    android:layout_marginTop="40dp"
    android:text="Пароль:"
    android:textAppearance="?android:attr/textAppearanceMedium" />

<EditText
    android:id="@+id/edit_password"
    android:layout_width="127dp"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/password_text"
    android:layout_alignLeft="@+id/edit_user"
    android:layout_alignRight="@+id/edit_user"
    android:layout_marginRight="-5dp"
    android:hint="введите пароль"
    android:inputType="textPassword" />

<TextView
    android:id="@+id/attempts"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="140dp"
    android:layout_toStartOf="@+id/Login"
    android:text="Попыток:"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:visibility="invisible" />

<Button
    android:id="@+id/button_login"
    android:layout_width="199dp"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="216dp"
    android:onClick="Login"
    android:text="Войти"
    tools:layout_editor_absoluteX="89dp"
    tools:layout_editor_absoluteY="215dp" />

```

```

<TextView
    android:id="@+id/number_of_attempts"
    android:layout_width="26dp"
    android:layout_height="30dp"
    android:layout_alignRight="@+id/user_text"
    android:layout_alignTop="@+id/attempts"
    android:layout_marginRight="-26dp"
    android:visibility="invisible" />

<TextView
    android:id="@+id/login_locked"
    android:layout_width="wrap_content"
    android:layout_height="49dp"
    android:layout_alignEnd="@+id/edit_password"
    android:layout_alignLeft="@+id/attempts"
    android:layout_alignParentBottom="true"
    android:layout_alignRight="@+id/edit_password"
    android:layout_alignStart="@+id/attempts"
    android:layout_marginBottom="48dp"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:visibility="invisible" />

</RelativeLayout>

```

Файл «Secondactivity.java»

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView"
        android:layout_width="284dp"
        android:layout_height="293dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="92dp"
        android:text="Вітаємо в программі обробки інтернет заявок"
        android:textAlignment="center"
        android:textAppearance="@style/TextAppearance.AppCompat.Display2"
        android:textColor="@android:color/holo_blue_bright"
        android:textSize="40sp"
        android:textStyle="bold"
        android:typeface="normal"
        android:visibility="visible" />

</RelativeLayout>

```

Файл «SecondActivity.java»

```

package com.example.evgeny.startdiplom;

import android.os.Bundle;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;

```

```

public class Second extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.second_activity);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

        getMenuInflater().inflate(R.menu.main_menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.action_zayavki:

                Intent intent = new Intent(Second.this, Three.class);
                startActivity(intent);

                return true;

            case R.id.action_zvit:

                Intent Internet = new Intent(Second.this, Four.class);
                startActivity(Internet);

                return true;

            case R.id.action_quit:
                finish();
                System.exit(0);

                return true;
        }

        return super.onOptionsItemSelected(item);
    }

    public void onBackPressed() {
        // do nothing
    }
}

```

Файл «Three.java»

```

package com.example.evgeny.startdiplom;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Three extends AppCompatActivity {

```

```

private Button change;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_three);
    change = (Button) findViewById(R.id.button_change);
}

public void Change (View view) {

    Intent intent = new Intent(Three.this, Zayavka.class);
    startActivity(intent);
}
}

```

Файл «Zayavka.java»

```

package com.example.evgeny.startdiplom;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Three extends AppCompatActivity {

    private Button change;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_three);
        change = (Button) findViewById(R.id.button_change);
    }

    public void Change (View view) {

        Intent intent = new Intent(Three.this, Zayavka.class);
        startActivity(intent);
    }
}

```

Файл «Activity_three.xml»

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/changer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/textView2"
        android:layout_centerVertical="true"
        android:background="@android:drawable/button_onoff_indicator_on"
        android:text="Изменить" />

```

```

<Button
    android:id="@+id/button_change"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignEnd="@+id/textView2"
    android:layout_below="@+id/textView2"
    android:background="@android:drawable/button_onoff_indicator_on"
    android:onClick="Change"
    android:text="ИЗМЕНИТЬ" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="265dp"
    android:layout_height="46dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="42dp"
    android:layout_marginTop="179dp"
    android:text="Заявка № 2"
    android:textAlignment="center"
    android:textAppearance="@style/TextAppearance.AppCompat"
    android:textColor="@android:color/holo_red_dark"
    android:textSize="30sp"
    android:textStyle="bold"
    tools:text="Заявка № 2" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="265dp"
    android:layout_height="46dp"
    android:layout_alignParentTop="true"
    android:layout_alignStart="@+id/textView4"
    android:layout_marginTop="52dp"
    android:text="Заявка № 1"
    android:textAlignment="center"
    android:textAppearance="@style/TextAppearance.AppCompat"
    android:textColor="@android:color/holo_red_dark"
    android:textSize="30sp"
    android:textStyle="bold"
    tools:text="Заявка № 1" />

```

```
</RelativeLayout>
```

Файл «Activityfour.xml»

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Four">

    <EditText
        android:id="@+id/editText"
        android:layout_width="281dp"
        android:layout_height="287dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="57dp"
        android:ems="10"

```



```
    android:hint="@string/name"
    android:inputType="text"
    android:selectAllOnFocus="false"
    android:singleLine="false"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.504"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/button"
    android:layout_width="227dp"
    android:layout_height="79dp"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="55dp"
    android:background="@android:color/holo_blue_light"
    android:text="@string/name2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.503"
    app:layout_constraintStart_toStartOf="parent"
    tools:layout_editor_absoluteY="392dp" />
</RelativeLayout>
```

ДОДАТОК Б

Комп'ютерна презентація

Східноукраїнський національний університет імені Володимира Даля
Факультет інформаційних технологій та електроніки
Кафедра комп'ютерних наук та інженерії
Напрямок підготовки 6.050102 – “Комп'ютерна інженерія”

Дипломний проект бакалавра на тему

Мобільний додаток для роботи з заявками інтернет-компанії

Студента групи КІ-143 Плотнікова Євгена Євгеновича

Севєродонецьк 2018

Рисунок Б.1 – Титульний слайд

Структура програми

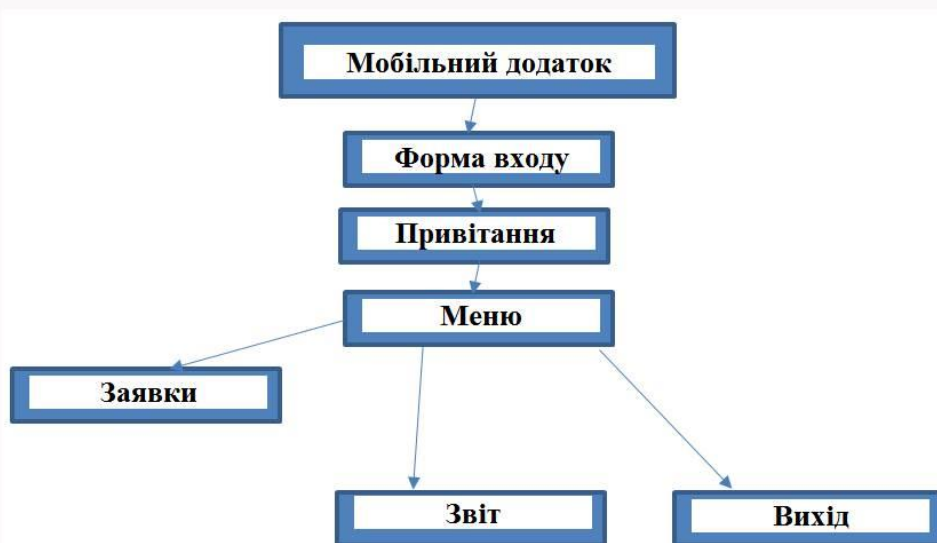


Рисунок Б.2 – Структура розробленого додатку

Форма входу

The screenshot shows a mobile application interface for login. At the top, there is a blue header with the text 'Интернет заявки'. Below it, the title 'Авторизация' is centered. There are two input fields: the first is labeled 'Пользователь:' and contains the placeholder text 'введите логин'; the second is labeled 'Пароль:' and contains the placeholder text 'введите пароль'. Below these fields is a button with the text 'ВОЙТИ'.

Рисунок Б.3 – Форма входу додатку

Блокування форми входу

```
// В другом случае выведем сообщение с ошибкой:
else {
    Toast.makeText(getApplicationContext(), text: "Неправильные данные!", Toast.LENGTH_SHORT).show();
    numberOfRemainingLoginAttempts--;

    // Делаем видимыми текстовые поля, указывающие на количество оставшихся попыток:
    attempts.setVisibility(View.VISIBLE);
    numberOfAttempts.setVisibility(View.VISIBLE);
    numberOfAttempts.setText(Integer.toString(numberOfRemainingLoginAttempts));

    // Когда выполнено 3 безуспешных попытки залогиниться,
    // делаем видимым текстовое поле с надписью, что все пропало и выставляем
    // кнопке настройку невозможности нажатия setEnabled(false):
    if (numberOfRemainingLoginAttempts == 3) {
        login.setEnabled(false);
        loginLocked.setVisibility(View.VISIBLE);
        loginLocked.setBackgroundColor(Color.RED);
        loginLocked.setText("Введите правильный логин или пароль");
    }
}
```

Рисунок Б.4 – Фрагмент програми блокування входу

ФОРМА ПРИВІТАННЯ

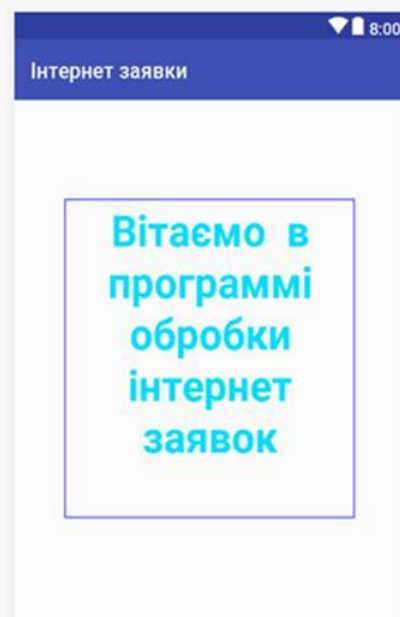


Рисунок Б.5 – Екран привітання користувача

ЕЛЕМЕНТИ МЕНЮ

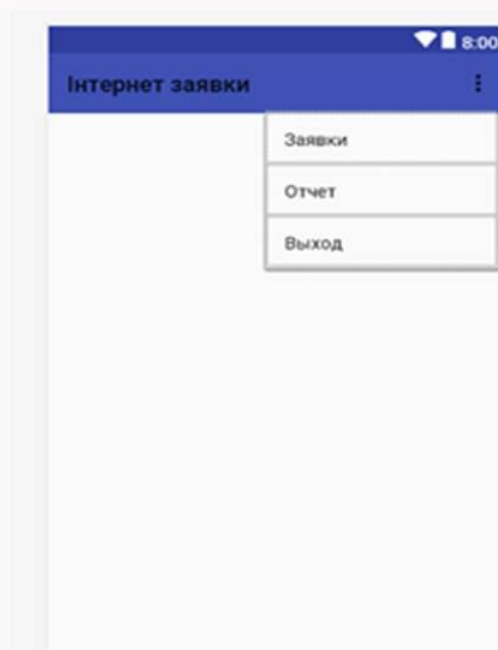


Рисунок Б.6 – Меню програми

ЗАЯВКИ



Рисунок Б.7 – Перелік заявок

ЗАЯВКИ

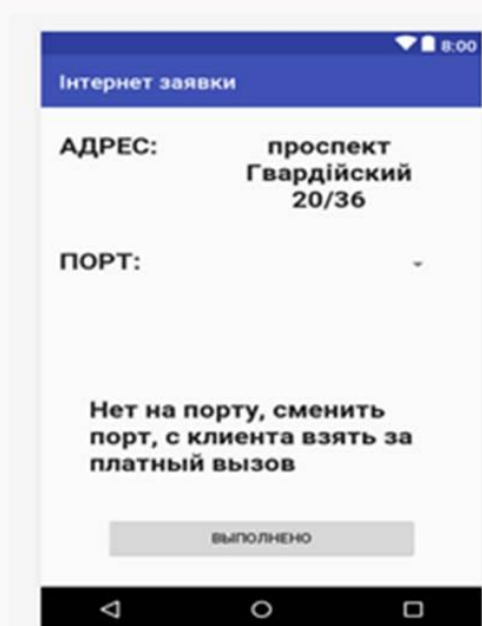


Рисунок Б.8 – Заявку виконано

Форма звітності

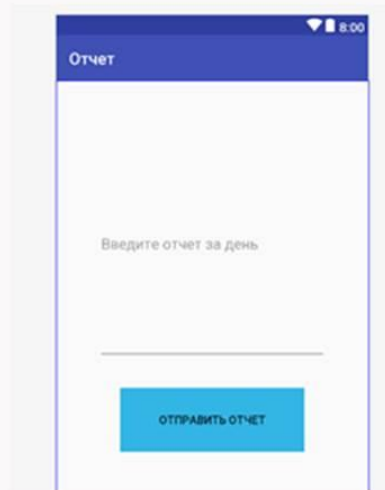


Рисунок Б.9 – Звіт за виконанні заявки

Висновки

В дипломному проекті був розроблений клієнтський додаток, що дозволяє робітниками інтернет-компанії виконувати заявки та здавати звіт в кінці дня з ОС Android

Основні можливості розробленої програми:

- облікова запис для кожного робітника;
- форма звітності в кінці дня
- форма для роботи з заявками інтернет компанії

Також ця програма може бути модернізована за допомогою бази даних Інтернет компанії в якій буде інформація про кожного клієнта і на сервер буде відправлятися звіт в кінці дня

Рисунок Б.10 - Висновки