

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

# КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

*на тему: "Android-додаток  
для користувачів послуг АТ "Сумиобленерго"  
по передачі показів лічильника електроенергії"*

Студент групи ІН.мз-91с

\_\_\_\_\_  
(підпис)

С.В. Федірка

Керівник роботи

\_\_\_\_\_  
(підпис)

І.В. Шелехов

Суми 2020

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ .....	3
ВСТУП .....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	6
1.1 Огляд основних відомостей.....	6
1.2 Огляд методологій і технологій розробки ANDROID-ДОДАТКІВ .....	10
1.3 Огляд існуючих мобільних додатків .....	25
1.4 Постановка задачі.....	29
1.5 Висновки .....	30
2 ВИБІР МЕТОДУ РІШЕННЯ .....	31
2.1 Основні концепції розробки мобільного додатку .....	31
2.2 Аналіз обраної мови та середовища програмування .....	34
2.3 Висновки .....	37
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ .....	38
3.1 Програмна реалізація .....	38
3.2 Тестування мобільного додатку .....	49
3.3 Приклади реалізації додатку .....	52
3.4 Висновки .....	57
ВИСНОВКИ.....	58
СПИСОК ЛІТЕРАТУРИ.....	59
ДОДАТОК А.....	61
Основний програмний код ANDROID-ДОДАТКУ .....	61

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

IDE – Integrated Development Environment – інтегроване середовище розробки

АТ – Акціонерне товариство

XML – Extensible Markup Language – розширювана мова розмітки

ОНА – Open Handset Alliance – альянс 84 компаній по розробці відкритих стандартів для мобільних пристроїв

OS(OC) – операційна система

JIT – Just-in-time compilation – технологія збільшення продуктивності програмних систем

JDK – Java Development Kit – комплект розробника додатків на мові Java

CSS - Cascading Style Sheets – каскадна таблиця стилів

ТЗ – технічне завдання

XP – eXtreme Programming – методологія розробки програмного забезпечення

ПрАТ – Приватне акціонерне товариство

SDK – Software Development Kit – набір із засобів розробки, утиліт і документації

ADT – Android Development Tools – плагін для Eclipse

NDK – Native Development Kit – набір інструментарію для розробки компонентів програмного забезпечення для платформи Android

JVD – Java Virtual Device – вбудований емулятор в Android Studio

MVC – Model-view-controller – архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення

GPS – Global Positioning System – система глобального позиціонування

## ВСТУП

Сучасний етап технологічного розвитку компанії характеризується зростаючою роллю інформаційних технологій. Завдяки розвитку мобільних пристроїв почався бурхливий розвиток ринку мобільного контенту, зокрема, мобільних додатків. Кількість і складність завдань, які виконуються мобільними додатками незмінно зростає, це допомагає у формуванні та розвитку сучасного інформаційного простору компанії, що в свою чергу дає працівникам змогу не тільки полегшити життя, а ще й економити їхній час. Саме тому розробка нового інформаційного додатку для користувачів послуг Акціонерного Товариства "Сумиобленерго" (далі – АТ "Сумиобленерго") є актуальною.

Кожна успішна компанія повинна слідувати сучасним трендам і йти в ногу з розвитком інформаційних технологій. Розробка мобільних додатків сьогодні є однією з найпопулярніших послуг на інформаційному ринку. А найпопулярнішою на сьогоднішній день платформою для якої розробляють мобільні додатки є операційна система Android, яка базується на ядрі Linux. На даний час її підтримкою та розробкою займається консорціум Open Handset Alliance, ініційований компанією Google після покупки компанії Android Inc. Операційна система Android забезпечує простоту та зручність використання і налаштування системи. Android додатки пишуться на мові програмування Java та виконуються на віртуальній машині – Dalvik Virtual Machine.

Розробка даного мобільного додатку має на меті створити за допомогою сучасної інтегрованої середи розробки (IDE) Android Studio, що містить усі необхідні інструменти та бібліотеки для роботи з XML-розміткою (дизайн, інтерфейс), Java-кодом (логіка взаємодії даних) та побудови проектів, зручне та інтуїтивно зрозуміле розширення, яке полегшить та заощадить час користувачу послуг.

Метою роботи є розробка інформаційного Android-додатку для передачі показників лічильника електроенергії в АТ "Сумиобленерго". Завданнями роботи є:

- вибір мови програмування;
- уточнення способів організації даних;
- запис алгоритму на вибраній мові програмування;
- розробка зручного інтерфейсу для користувача;
- тестування програми.

Додатки для Android в своїй роботі використовують вікна (аналогічно Windows), проте в даній системі вище вказані вікна носять іншу назву – Activity. Як і в Windows, кожне вікно має свій життєвий цикл і свої особливості. При створенні нового вікна викликається метод onCreate(), при розробці даний метод перевизначається і в ньому відбувається ініціалізація програми та його компонентів. Далі викликаються методи onStart() і onResume(). Обидва методи викликаються перш ніж можна буде вікна при його створенні, або відновленні (при перемиканні з іншої програми, при розгортанні згорнутого додатку і т.п). При згортанні викликаються методи onPause() і onStop(). При закритті програми і вікна викликається onDestroy(), в даному методі можна зберегти призначені для користувача дані і параметри. [1,2]

У майбутньому планується вдосконалювати створений програмний продукт шляхом покращення інтерфейсу і додання кількох функцій.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд основних відомостей

Android – це операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, що є відкритою, створена компанією Google на базі ядра Linux з мовою програмування Java. Підтримується альянсом Open Handset Alliance (ОНА) який представив концерн відкритий стандарт для мобільних пристроїв. Android підтримує 2D і 3D-графіку, використовуючи бібліотеки OpenGL, а також зберігання даних в базі даних SQLite. Кожний Android-додаток запускається в своєму власному процесі. Тому додаток ізольовано від інших запущених додатків і неправильно працюючий додаток не може безперешкодно зашкодити іншим додаткам. [3]

Мобільний додаток – програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях. Багато мобільних додатків встановлені на самому пристрої або можуть бути завантажені на нього з онлайн магазинів мобільних додатків, таких як App Store, Google Play, Windows Phone Store та інших, безкоштовно або за відповідну плату. [4]

Оскільки операційна система Android (далі – OS Android) є відкритою, користувачеві незважаючи на початкову заборону на установку програм з "неперевірених джерел" (наприклад, з карти пам'яті), надається можливість це обмеження відключати штатними засобами в налаштуваннях пристрою, що дозволяє встановлювати програми на телефони та планшети без інтернет-підключення (наприклад користувачам, які не мають Wi-Fi-точки доступу і не бажають витратити зайві кошти на мобільний інтернет), а також дозволяє будь-кому безкоштовно писати програми для Android і тестувати на своєму пристрої.

Про OS Android стало відомо 5 листопада 2007 року, під час проведення консорціуму ОНА з 84 компаній. Більша частина коду Android була випущена під ліцензією Apache Software Foundation. Android програми включають в себе бібліотеки та java-додатки, які запускаються віртуальною

машиною Dalvik Virtual Machine з JIT компілятором. Бібліотеки включають в себе систему управління, графіку OpenGL ES 2.0, движок WebKit, графічний движок Skia Graphics Engine, Secure Sockets Layer і бібліотеки Bionic. OS Android складається з 12 мільйонів рядків коду, в тому числі 3-х мільйонів рядків XML, 2.8 мільйонів рядків на C, 2.1 мільйона рядків на Java і 1.75 мільйона рядків на C++.

Почалося все з Енді Рубіна, Річі Майнера, Ніка Сірса і Кріса Уайта, які заснували компанію Android inc. в жовтні 2003 року в містечку Пало Альто штат Каліфорнія. У серпні 2005 року компанія Google придбала компанію Android inc., яка після цього стала дочірньою. Після покупки в компанії залишилися Енді Рубін, Річ Майнер і Кріс Уайт, а також в мережі почали з'являтися чутки про те, що Google хоче вийти на ринок мобільних телефонів. Заручившись підтримкою Google, команда на чолі з Енді Рубіном почала працювати над операційною системою, яка базується на ядрі Linux. Того самого року, а саме грудень 2006 року поповзли чутки про те, що Google планує випустити смартфон під своїм брендом, що мав назву "Гуглофон". Всі ці чутки спростував Ерік Шмідт, заявивши наступне: "Сьогоднішня заява носить більш амбітний характер, ніж випуск смартфона під нашим брендом, про що преса спекулює останні тижні. Наше бачення полягає в тому, що найпотужніша платформа повинна віддати свої сили тисячам різних моделей телефонів". Після цієї заяви 5 листопада 2007 і був представлений Open Headset Alliance – консорціум з безлічі компаній, до якого увійшли такі гіганти як: Broadcom Corporation, Google, HTC, Intel , LG , Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile, Texas Instruments та інші.

Всі версії OS Android представляють собою назву якого-небудь десерту, перші букви яких відповідають літерами латинського алфавіту. Android Market – інтернет-магазин додатків для смартфонів на базі Android, яку просував альянс ОНА на чолі з Google. 22 жовтня 2008 Google оголосила

про відкриття цього онлайн-магазину додатків для Android, що включав в себе ігри, клієнти соціальних мереж, офісні додатки, додатки для читання новин, управління фінансами та інші. [5]

В Україні деякий час був відсутній офіційний магазин додатків на смартфонах. Повідомлення від 12 січня 2010 компанією Samsung Україна змінила дану ситуацію, а пізніше Android Market з'явився і у інших вендорів: Motorola, HTC, LG і Sony Ericsson, що значно підігріло інтерес до Android-смартфонам в Україні. Офіційно першим смартфоном з Android Market в Україні став Samsung i5700 Galaxy Spica. Програми для Android є програмами в нестандартному байт-кодi для віртуальної машини Dalvik Virtual Machine. Google пропонує для вільного скачування інструментарій для розробки (Android SDK), який призначений для x86-машин під операційними системами Windows XP, Windows Vista, Mac OS X (10.4.8 або вище) і Linux. Для розробки додатків для Android потрібно JDK 5 або JDK 6 мова програмування Java (версія не нижче Java 1.5). Існують плагіни для Eclipse – "Android Development Tools" (ADT), призначений для Eclipse версій 3.3 – 3.5, для IntelliJ IDEA плагін, який полегшує розробку Android-додатків і також для середовища розробки NetBeans IDE розроблено експериментальний плагін.

До складу Android входить комплект базових додатків (рівень додатків Applications): клієнти електронної пошти та SMS, календар, різні карти, браузер, програма для управління контактами і багато іншого. Всі додатки, що запускаються на платформі Android написані на мові Java.

Архітектура OS Android має п'ять рівнів.

Каркас (рівень додатків Application Framework): Android дозволяє використовувати всю міць API, використовуюваного в додатках ядра. Архітектура побудована таким чином, що будь-який додаток може використовувати вже реалізовані можливості іншої програми за умови, що остання відкриє доступ на використання своєї функціональності. Таким



чином, архітектура реалізує принцип багаторазового використання компонентів ОС і додатків.

Основою всіх додатків є набір систем і служб:

1. Система перегляду (View System) – це багатий набір уявлень з розширюваною функціональністю, який служить для побудови зовнішнього вигляду додатків, що включає такі компоненти, як списки, таблиці, поля введення, кнопки і т.п.

2. Контент-провайдери (Content Providers) – це служби, які дозволяють додаткам отримувати доступ до даних інших додатків, а також надавати доступ до своїх даних.

3. Менеджер ресурсів (Resource Manager) призначений для доступу до строкових, графічних та інших типів ресурсів.

4. Менеджер сповіщень (Notification Manager) дозволяє будь-якому додатку відображати для користувача повідомлення в рядку статусу.

5. Менеджер дій (Activity Manager) управляє життєвим циклом додатків і надає систему навігації по історії роботи з діями.

Рівень бібліотек (Libraries): платформа Android включає набір C/C++ бібліотек, використовуваних різними компонентами ОС. Для розробників доступ до функцій цих бібліотек реалізований через використання Application Framework. Нижче представлені деякі з них:

1. System C library-BSD – реалізація стандартної системної бібліотеки C (libc) для вбудованих пристроїв, заснованих на Linux.

2. Media Libraries – бібліотеки, засновані на Packet Video's Open CORE, призначені для підтримки програвання і запису популярних аудіо- та відеоформатів (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG і т.п.).

3. Surface Manager – менеджер поверхонь управляє доступом до підсистеми відображення 2D- і 3D- графічних шарів.

4. Lib Web Core – сучасний движок web-браузера, який надає всю міць вбудованого Android-браузера.

5. SGL – движок для роботи з 2D-графікою.
6. 3D libraries – движок для роботи з 3D-графікою, заснований на OpenGL ES 1.0 API.
7. Free Type – бібліотека, призначена для роботи зі шрифтами.
8. SQLite – потужний движок для роботи з реляційними базами даних.

Рівень середовища виконання (Android Runtime): до складу Android входить набір бібліотек ядра, які надають велику частину функціональності бібліотек ядра мови Java. Платформа використовує оптимізовану, реєстр-орієнтовану віртуальну машину Dalvik Virtual Machine, на відміну від неї стандартна віртуальна машина Java – стек-орієнтована. Кожна програма запускається у своєму власному процесі, зі своїм власним примірником віртуальної машини. Dalvik використовує формат Dalvik Executable (\*.Dex), оптимізований для мінімального використання пам'яті додатком. Це забезпечується такими базовими функціями ядра Linux, як організація потокової обробки і низькорівневе управління пам'яттю. Байт-код Java, на якому написані програми, компілюються в dex-формат за допомогою утиліти dx, що входить до складу SDK.

Рівень ядра Linux (Linux Kernel): Android заснований на ОС Linux версії 2.6, тим самим платформі доступні системні служби ядра, такі як управління пам'яттю і процесами, забезпечення безпеки, робота з мережею і драйверами. Також ядро служить шаром абстракції між апаратним та програмним забезпеченням. [5,6]

## **1.2 Огляд методологій і технологій розробки Android-додатків**

Мобільним додатком називають спеціальну програму для смартфона, встановлена на тій чи іншій платформі, яка володіє певним функціоналом, що дозволяє виконувати певні дії в залежності від тих завдань, які необхідно вирішити. Це свого роду адаптер, що допомагає користувачеві взаємодіяти з різноманітною інформацією.

До початку розробки мобільного додатку необхідно визначитися:

- хто буде користувачем (цільова аудиторія);
- для яких пристроїв призначений цей додаток;
- в якому середовищі буде розроблятися додаток;
- платформа, на якій додаток буде функціонувати;
- методологія і методи, які доцільно застосувати при розробці додатка.

Мобільні додатки діляться на нативні (рідні), веб-додатки та гібридні.

Нативний додаток – це додаток, який розроблений на своїй (рідній) мові програмування для вибраної платформи (наприклад: objective-c для iOS, Java для Android, C# для Windows Phone).

Використовуються по таким причинам:

- працюють швидше і стабільніше ніж додатки іншого типу;
- дозволяють зняти функціональні обмеження браузерів з доступу до ресурсів пристрою.

Мобільна платформа надає інструментарій для розробників SDK, який дозволяє отримати доступ практично до всіх сервісів пристрою, а також призначений для спрощення процесу створення додатків.

Веб-додаток – це додаток, розроблений на HTML, JavaScript, CSS і вимагає для свого виконання встановленого і налаштованого браузера мобільного пристрою з виходом в Інтернет.

HTML застосовується для розмітки елементів інтерфейсу. CSS описує візуальну складову і взаємне розташування віджетів і елементів управління. Мова програмування JavaScript реалізує логіку програми.

Використовуються по таким причинам:

- можливість повного або часткового повторного використання коду на різних платформах;
- існування версії веб-сайту для настільного комп'ютера і є необхідність отримання доступу через мобільний пристрій;
- не пред'являють особливих вимог до графіки і використання апаратних засобів пристрою;

– є величезний вибір інструментів, фреймворків, які прискорюють і спрощують процес розробки.

Гібридний додаток – це додаток в якому частково використовується нативна функціональність, а частково – можливості веб-додатків. Від нативних додатків: можливість часткового доступу до ресурсів пристрою. Від веб-додатків: підтримка HTML і робота в браузері).

Використовуються по таким причинам:

- загальна продуктивність і відгук інтерфейсу не є вирішальними;
- можна поширювати відразу на безліч платформ;
- можливість поширення (публікації) як готового продукту або тимчасового замітника до виходу нативного додатку (запустити процес маркетингу). [7]

Порівняльна інформація по всім мобільним додаткам занесена до таблиці 1.1.

Таблиця 1.1 – Порівняння типів мобільних додатків

Тип додатку Критерій	Нативні	Веб-додатки	Гібридні
Доступ до ресурсів	повний	відсутній	обмежений
Доступ до Інтернету	не обов'язковий	обов'язковий	обов'язковий
Установка	потребує	не потребує	потребує
Розповсюдження (публікація)	магазини додатків	не потребує	магазини додатків
Оновлення	повторне розміщення	не потребує	повторне розміщення
Платформи	розробка під кожену	кросплатформенні	кросплатформенні

Перша методологія розробки мобільних додатків представлена на рисунку 1.1 і має назву Каскадна модель або модель "водоспад" (Waterfall Model). [7,8]

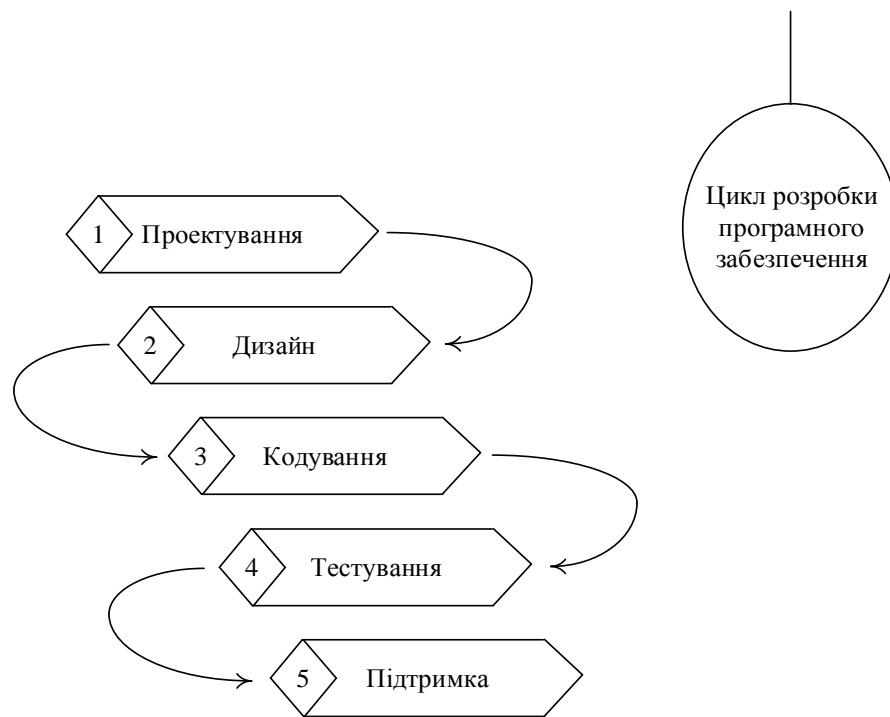


Рисунок 1.1 – Каскадна модель або модель "водоспад"

Дана модель одна з перших вона передбачає послідовне проходження стадій, кожна з яких має завершитися повністю до початку наступної. У моделі Waterfall легко керувати проектом. Завдяки її жорсткості, розробка проходить швидко, вартість і термін заздалегідь визначені. Проте каскадна модель буде давати відмінний результат тільки в проектах з чітко і заздалегідь визначеними вимогами і способами їх реалізації. Немає можливості зробити крок назад, тестування починається тільки після того, як розробка завершена або майже завершена. Продукти, розроблені за цією моделлю без обґрунтованого її вибору, можуть мати недоліки (список вимог не можна скорегувати в будь-який момент), про які стає відомо лише в кінці через суворі послідовності дій. Вартість внесення змін висока, так як для її ініціалізації доводиться чекати завершення всього проекту. Проте, фіксована вартість часто переважає мінуси підходу.

Каскадна методологія використовується:

– тільки тоді, коли вимоги відомі, зрозумілі і зафіксовані.  
Суперечливих вимог немає;

- немає проблем з доступністю програмістів потрібної кваліфікації;
- у відносно невеликих проектах.

Друга методологія розробки мобільних додатків представлена на рисунку 1.2 і має назву V-образна модель "V – Model". [7,8]

Структуру успадкувала від каскадної моделі тобто "крок за кроком". V-образна модель застосовується до систем, яким особливо важливе безперебійне функціонування. Особливістю моделі можна вважати те, що вона спрямована на ретельну перевірку і тестування продукту, що знаходиться вже на початкових стадіях проектування. Стадія тестування проводиться одночасно з відповідною стадією розробки, наприклад, під час кодування пишуться модульні тести. Прикладом такої моделі може бути мобільний додаток для європейського стільникового оператора, який економить витрати на роумінг під час подорожей. Проект виконується за чітким ТЗ, але в нього включений значний етап тестування: зручності інтерфейсу, функціонального, навантажувального та в тому числі інтеграційного, яке повинно підтверджувати, що кілька компонентів від різних виробників разом працюють стабільно, неможлива крадіжка грошей і кредитів.

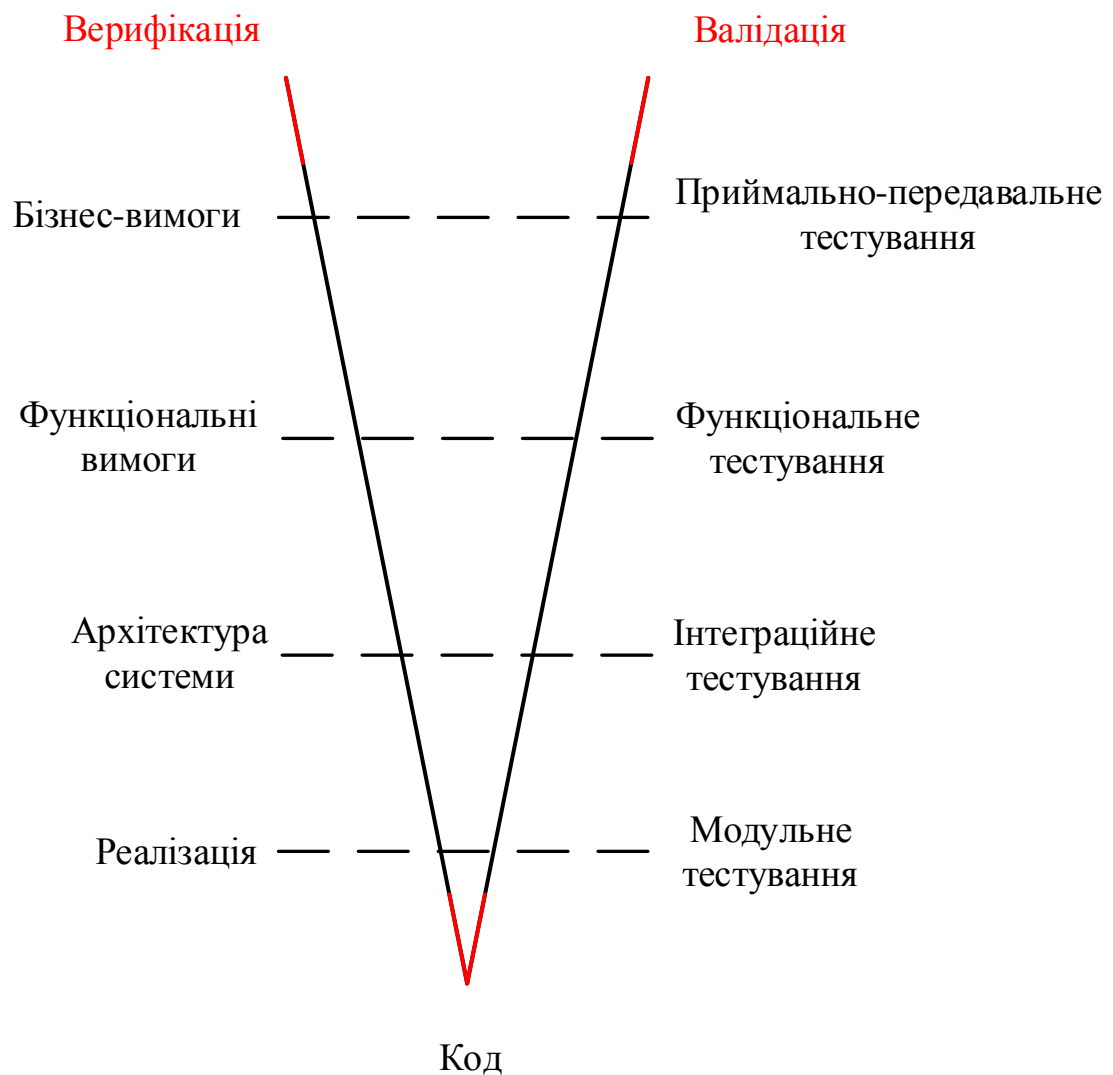


Рисунок 1.2 – V-образна модель

Методологія V-моделі використовується:

- якщо потрібне ретельне тестування продукту, то V-модель виправдає закладену в себе ідею: validation and verification;
- для малих і середніх проектів, де вимоги чітко визначені і фіксовані;
- в умовах доступності інженерів необхідної кваліфікації, особливо тестувальників.

Третя методологія розробки мобільних додатків представлена на рисунку 1.3 і має назву інкрементна модель (Incremental Model). [7,8]

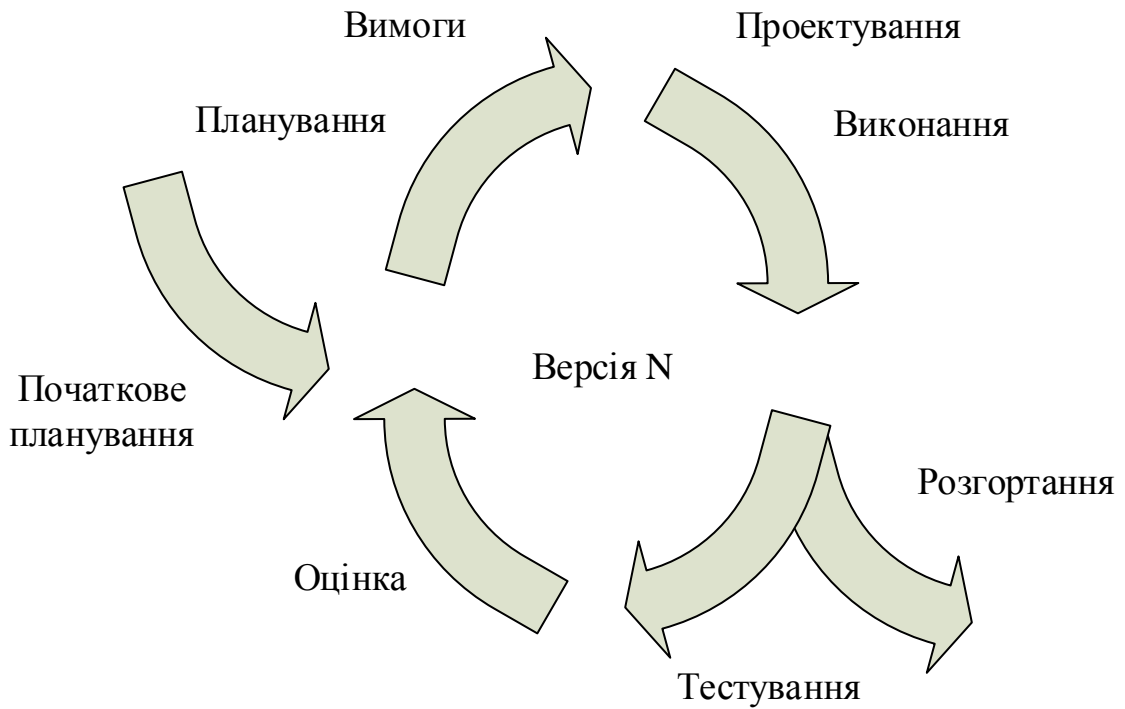


Рисунок 1.3 – Інкрементна модель

У інкрементній моделі повні вимоги до системи діляться на різні збірки. Термінологія часто використовується для опису поетапної збірки ПЗ. В моделі мають місце кілька циклів розробки і разом вони складають життєвий цикл "Мульти-водоспад". Цикл розділений на більш дрібні легко створювані модулі. Кожен модуль проходить через фази визначення вимог, проектування, кодування, впровадження та тестування. Процедура розробки за інкрементною моделлю передбачає випуск на першому великому етапі продукту в базовій функціональності, а потім вже послідовне додавання нових функцій, так званих "інкрементів". Процес триває до тих пір, поки не буде створена повна система.

Інкрементні моделі використовуються там, де окремі запити на зміну ясні, можуть бути легко формалізовані і реалізовані.

Ця модель використовується у наступних випадках:

- коли основні вимоги до системи чітко визначені і зрозумілі. У той же час деякі деталі можуть доопрацьовуватися з плином часу;
- потрібно ранній вивід продукту на ринок;



– є кілька ризикових функцій або цілей.

Четверта методологія розробки мобільних додатків представлена на рисунку 1.4 і має назву модель швидкої розробки додатків "RAD Model" (Rapid Application Development Model). [7,8]



Рисунок 1.4 – "RAD Model" або швидка розробка додатків

"RAD Model" або швидка розробка додатків – різновид інкрементної моделі. У RAD-моделі компоненти або функції розробляються декількома висококваліфікованими командами паралельно, ніби кілька міні-проектів. Тимчасові рамки одного циклу жорстко обмежені. Створені модулі потім інтегруються в один робочий прототип. Синергія дозволяє дуже швидко надати клієнту для огляду щось робоче з метою отримання зворотного зв'язку і внесення змін. Модель швидкої розробки додатків включає наступні фази:

- бізнес-моделювання: визначення переліку інформаційних потоків між різними підрозділами;
- моделювання даних: інформація, зібрана на попередньому етапі, використовується для визначення об'єктів і інших сутностей, необхідних для циркуляції інформації;
- моделювання процесу: інформаційні потоки пов'язують об'єкти для досягнення цілей розробки;
- збірка програми: використовуються засоби автоматичної збірки для перетворення моделей системи автоматичного проектування в код;
- тестування: тестуються нові компоненти і інтерфейси.

Модель може використовуватися тільки при наявності висококваліфікованих і вузькоспеціалізованих архітекторів. Бюджет проекту великий, щоб оплатити цих фахівців разом з вартістю готових інструментів автоматизованої збірки. RAD-модель може бути обрана при впевненому знанні цільового бізнесу і необхідності термінового виробництва системи протягом 2-3 місяців.

П'ята методологія розробки мобільних додатків представлена на рисунку 1.5 і має назву гнучка розробка моделі "Agile Model". [7,8]

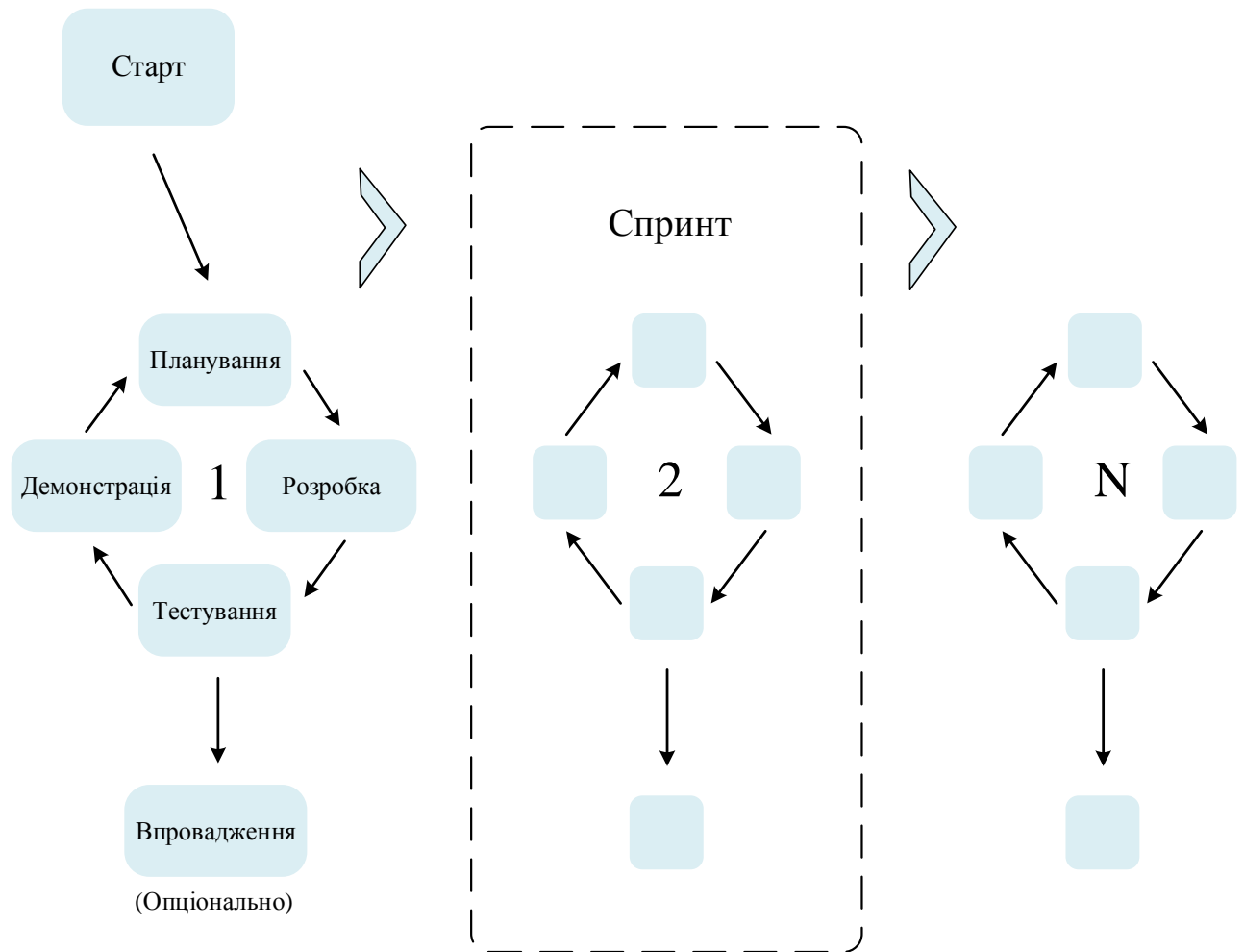


Рисунок 1.5 – Гнучка методологія розробки

У "гнучкої" методології розробки після кожної ітерації замовник може спостерігати результат і розуміти, задовольняє він його чи ні. Це одна з переваг гнучкої моделі. До її недоліків відносять те, що через відсутність конкретних формулювань результатів складно оцінити трудовитрати і вартість, необхідні на розробку.

Екстремальне програмування або XP (eXtreme Programming) є одним з найбільш відомих застосувань гнучкої моделі на практиці. В основі такого типу нетривалі щоденні зустрічі – "Scrum" і регулярно повторювані збори (раз в тиждень, раз на два тижні або раз на місяць), які називаються "Sprint". На щоденних нарадах учасники команди обговорюють:

- звіт про виконану роботу з моменту останнього Scrum'a;

- список завдань, які співробітник повинен виконати до наступних зборів;
- труднощі, що виникли в ході роботи.

Методологія підходить для великих або націлених на тривалий життєвий цикл проектів, що постійно адаптуються до умов ринку. Відповідно, в процесі реалізації вимоги змінюються. Гнучка розробка найкраще підходить для такого психотипу керівників.

Методологія використовується:

- коли потреби користувачів постійно змінюються в динамічному бізнесі;
- зміни на Agile реалізуються за меншу ціну через часті інкременти;
- на відміну від моделі водоспаду, в гнучкій моделі для старту проекту достатньо лише невеликого планування.

Scrum – це одна з Agile-методик, в якій створення продукту відбувається у вигляді серії ітерацій фіксованої тривалості. Методика робить акцент на якісному контролі процесу розробки, на управлінні завданнями в командному середовищі розробки. У загальному випадку, Scrum передбачає роботу над проектом короткими ітераціями (зазвичай 2-4 тижні), кожна з яких є зменшеним варіантом проекту і включає всі завдання, необхідні для видачі приросту по функціональності.

Кожен раз коли нова функція, модуль або оновлення завершено, здійснюється тестування. Завдяки тестуванню під час розробки, можливі помилки, які виправляються вчасно, що в підсумку призводить до набагато більш стабільному продукту. По закінченню ітерації замовник отримує нову версію продукту і якщо потрібно, проект або будь-яка його частина переглядається. Тут також враховується зворотний зв'язок від кінцевих користувачів. Потім цикл реалізації запускається знову. В результаті створюється рішення, яке максимально точно відповідає вимогам замовника та затребуване ринком.

Основні інструменти Scrum:

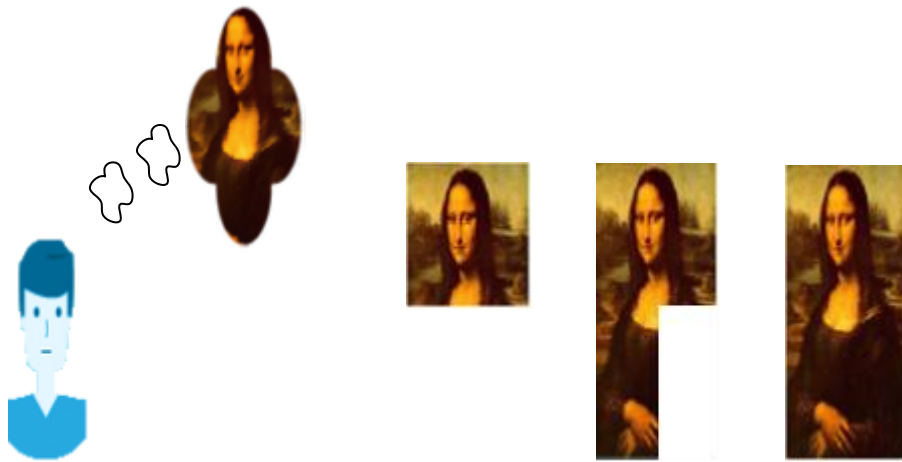
- резервування продукції (Product Backlog);
- історії користувачів (User Stories);
- планування спринту (ітерація в Scrum);
- резервування спринту (Sprint Backlog);
- дошка задач (у вигляді фізичної або цифрової, наприклад, всередині JIRA);
- діаграма згорання задач (Burndown Chart);
- швидкі наради (стендапи);
- огляд спринту (демо і ретроспектива).

Agile-методика XP сфокусована на інженерній стороні розвитку продукту. Це системний підхід до програмування. У той час як Scrum приділяє більше уваги управлінню і випуску, XP зосереджена на процесі виробництва. XP включає практики, що істотно поліпшують якість кінцевого продукту:

- безперервна інтеграція (Continuous Integration (CI));
- автоматизоване тестування;
- рефакторинг коду;
- рев'ю коду;
- парне програмування;
- стандарти кодування та інші.

Шоста методологія розробки мобільних додатків представлена на рисунку 1.6 і має назву ітеративна або ітераційна модель "Iterative Model".  
[7,8]

## Інкрементна модель



## Ітеративна модель

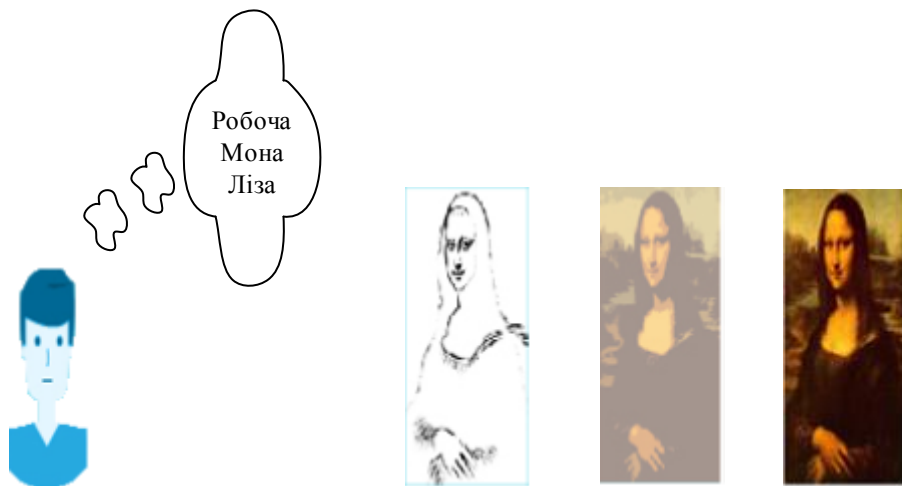


Рисунок 1.6 – Ітеративна або ітераційна модель

Ітераційна модель життєвого циклу на початку не вимагає повної специфікації вимог. Замість цього, створення починається з реалізації частини функціоналу, що стає базою для визначення подальших вимог. Цей процес повторюється. Версія може бути неідеальна, головне, щоб вона працювала. Розуміючи кінцеву мету, ми прагнемо до неї так, щоб кожен крок був результативним, а кожна версія – працездатна.

На рисунку 1.6 показана ітераційна "розробка" Мона Лізи. Як видно, в першій ітерації є лише начерк Джоконди, в другій – з'являються кольори, а третя ітерація додає деталей, насиченості і завершує процес. У інкрементній

моделі функціонал продукту нарощується по шматочках, продукт складається з частин. На відміну від ітераційної моделі, кожен шматочок являє собою цілісний елемент. Прикладом ітераційної розробки може служити розпізнавання голосу. Перші дослідження і підготовка наукового апарату почалися давно, на початку – в думках, потім – на папері. З кожною новою ітерацією якість розпізнавання поліпшувалося. Проте, ідеальне розпізнавання ще не досягнуто, отже, завдання ще не вирішена повністю.

Ітеративна модель використовується:

- вимоги до кінцевої системі заздалегідь чітко визначені і зрозумілі;
- проект великий або дуже великий;
- основне завдання повинно бути визначеним, але деталі реалізації можуть еволюціонувати з плином часу.

Сьома методологія розробки мобільних додатків представлена на рисунку 1.7 і має назву спіральна модель "Spiral Model". [7,8]

"Спіральна модель" схожа на інкрементну, але з акцентом на аналіз ризиків. Вона добре працює для вирішення критично важливих бізнес-задач, коли невдача несумісна з діяльністю компанії в умовах випуску нових продуктових лінійок, при необхідності наукових досліджень і практичної апробації. Спіральна модель передбачає 4 етапи для кожного витка:

1. Планування.
2. Аналіз ризиків.
3. Конструювання.
4. Оцінка результату і при задовільній якості перехід до нового витка.



Рисунок 1.7 – Спиральна модель

Ця модель не підійде для малих проектів, вона резонна для складних і дорогих (наприклад, таких, як розробка системи документообігу для банку, коли кожен наступний крок вимагає більшого аналізу для оцінки наслідків, ніж програмування). Державні проекти, зазвичай, починаються з підготовки експертним співтовариством дорогої концепції, яка аж ніяк не завжди марна, оскільки окупається в масштабах країни.



### 1.3 Огляд існуючих мобільних додатків

Для користувачів послуг по передачі показів лічильника електроенергії кожне з обленерго почало розробляти і впроваджувати свої мобільні додатки. На сьогоднішній день існує декілька областей, що запропонували і впровадили свій варіант мобільного додатку. Виберемо три області та проаналізуємо функціональні можливості кожного додатку.

Першим розглянемо додаток запропонований ПрАТ "Київобленерго", що має назву "Мережі онлайн". Інтерфейс додатка зображено на рисунку 1.8. По своїм набором функцій є основним орієнтиром.

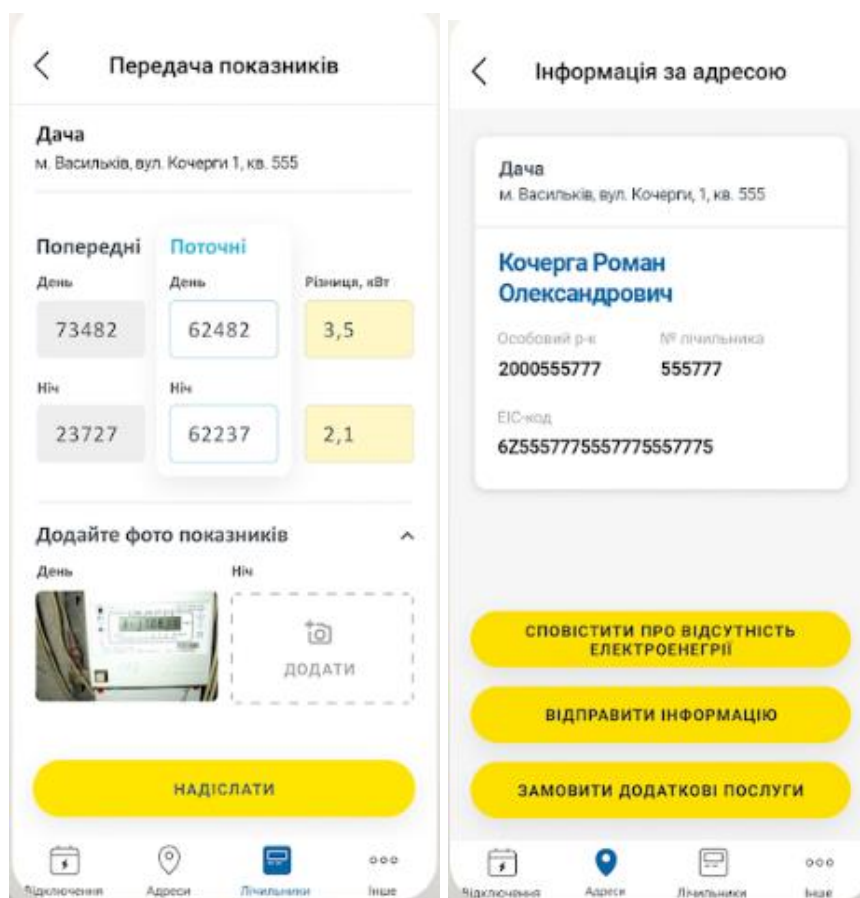


Рисунок 1.8 – Інтерфейс додатку "Мережі онлайн"

Щоб почати роботу в даному додатку спочатку потрібно ввести особливий рахунок, що слугує тригером для реєстрації користувача. Додаток має сучасне оформлення інтерфейсу і має такі можливості:

- оперативно отримувати інформацію про планові/аварійні відключення та час відсутності електропостачання;

- повідомляти про аварійні відключення електроенергії;
- зручно та швидко передавати покази лічильників;
- керувати декількома особовими рахунками з одного пристрою;
- дистанційно подавати заявки і звернення.

Проте за відгуками користувачів має рад недоліків:

- помилки при синхронізації з сервером;
- збої при реєстрації користувачів;
- неможливість передати показники лічильника без фотофіксації;
- не працює зворотній зв'язок.

Наступний додаток від АТ "Харківобленерго", що має назву "Харенерго". Інтерфейс додатка зображено на рисунку 1.9.

Додаток має інтуїтивно простий дизайн і зрозумілий користувачеві набір функцій:

- контроль необмеженої кількості особових рахунків;
- контроль переданих показань лічильників Харківобленерго;
- передача показань лічильників Харківобленерго онлайн;
- моніторинг нарахувань Харківенергозбут.

Особливістю програми варто відзначити оплату послуг електропостачання онлайн, без комісій та переплат.

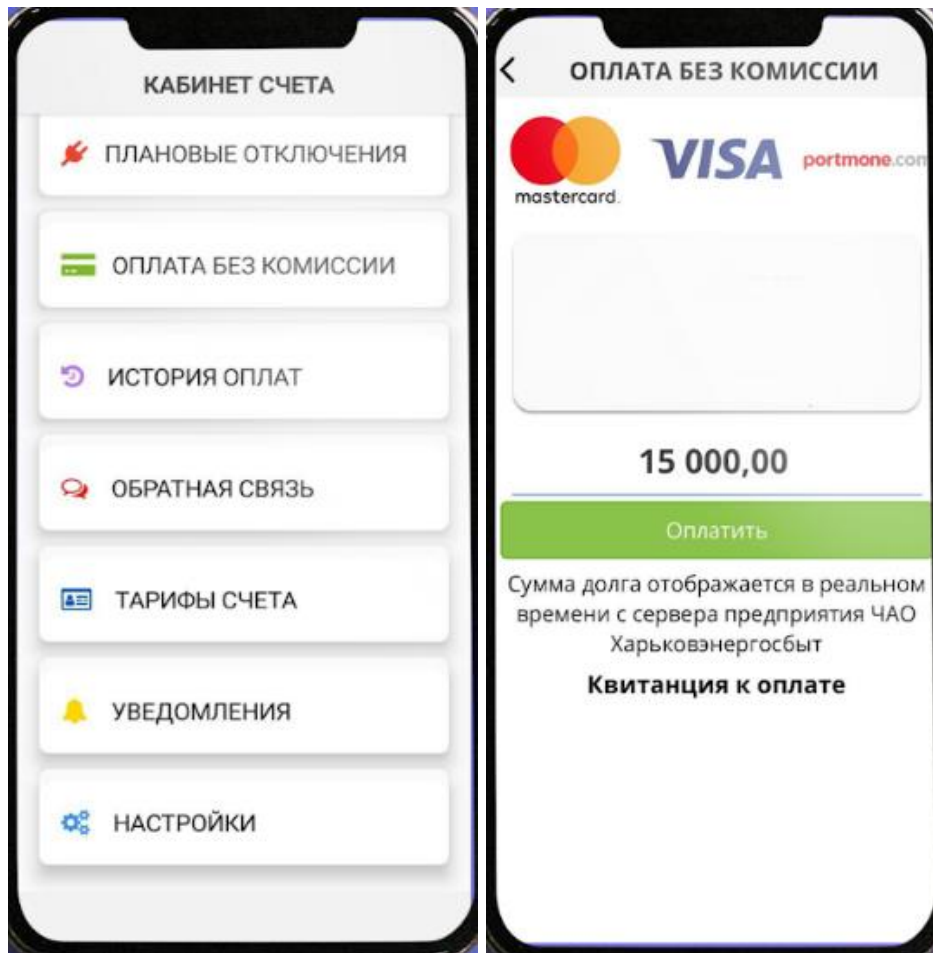


Рисунок 1.9 – Интерфейс додатку " Харенерго "

За словами розробників додаток має наступні переваги:

- при оплаті в нашому мобільному додатку, кошти зараховуються на рахунок ПрАТ "Харківенергозбут" в 4 рази швидше, ніж при оплаті в банківських касах і мобільних додатках;
- для оплати в мобільному додатку ви можете використовувати карту будь-якого Українського банку.

Недоліки за відгуками користувачів:

- відображення не актуальної інформації;
- часті збої та зависання в роботі додатку;
- занадто часте нагадування про сплату послуг;
- не працює зворотній зв'язок.

Останній додаток від АТ "Житомиробленого", що має назву "Ваш рахунок". Інтерфейс додатка зображено на рисунку 1.10.

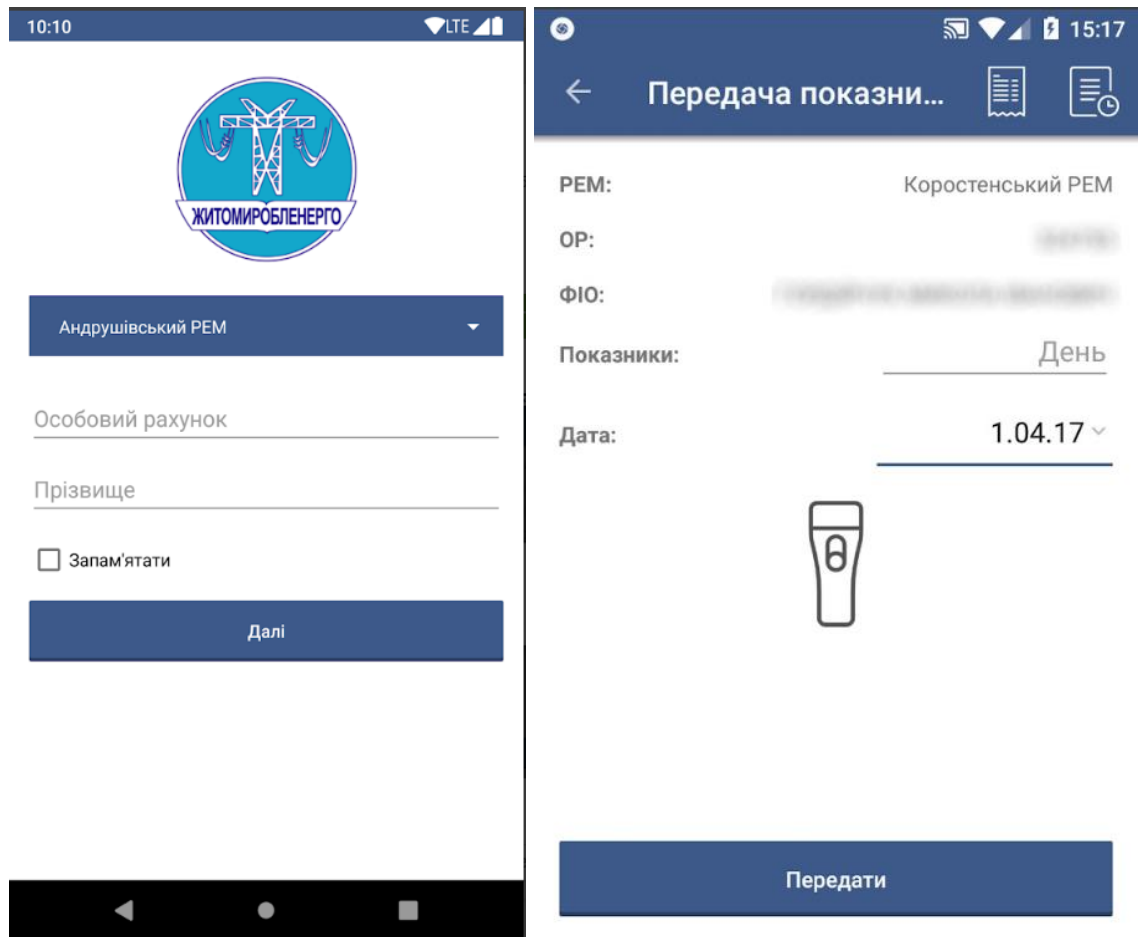


Рисунок 1.10 – Інтерфейс додатку " Ваш рахунок "

Для реєстрації потрібно ідентифікуватися по особовому рахунку та прізвищу особи з якою укладено договір на розподіл електричної енергії. Додаток простий у використанні і має такі можливості:

- передати покази лічильника;
- отримати рахунок;
- контролювати попередньо передані покази.

За відгуками користувачів має такі недоліки:

- обмежений функціонал;
- часті відмови працездатності при передачі даних;
- відсутній зворотній зв'язок.

Переваги в недоліки аналогів розробки мобільних додатків наведено в таблиці 1.2.

Таблиця 1.2 – Порівняння аналогів

Назва	Переваги	Недоліки
Мережі онлайн	<ul style="list-style-type: none"> <li>– отримання інформації про планові/аварійні відключення та час відсутності електропостачання;</li> <li>– керування декількома особовими рахунками з одного пристрою;</li> <li>– дистанційно подавати заявки і звернення.</li> </ul>	<ul style="list-style-type: none"> <li>– помилки при синхронізації з сервером;</li> <li>– збої при реєстрації користувачів;</li> <li>– неможливість передати показники лічильника без фотофіксації;</li> <li>– не працює зворотній зв'язок.</li> </ul>
Харенерго	<ul style="list-style-type: none"> <li>– контроль необмеженої кількості особових рахунків;</li> <li>– моніторинг нарахувань Харківенергозбут;</li> <li>– кошти зараховуються в 4 рази швидше, ніж при оплаті в банківських касах;</li> <li>– оплата в мобільному додатку будь-якою картою Українського банку</li> </ul>	<ul style="list-style-type: none"> <li>– відображення не актуальної інформації;</li> <li>– часті збої та зависання в роботі додатку;</li> <li>– занадто часте нагадування про сплату послуг;</li> <li>– не працює зворотній зв'язок.</li> </ul>
Ваш рахунок	<ul style="list-style-type: none"> <li>– отримання рахунок у форматі pdf;</li> <li>– контролювати попередньо передані покази</li> </ul>	<ul style="list-style-type: none"> <li>– обмежений функціонал;</li> <li>– часті відмови працездатності при передачі даних;</li> <li>– відсутній зворотній зв'язок.</li> </ul>

#### 1.4 Постановка задачі

В роботі потрібно розробити Android-додаток для користувачів послуг АТ "Сумиобленерго" по передачі показів лічильника електроенергії з такими можливостями:

- створити обліковий запис в особистому кабінеті;
- вести кілька особових рахунків;
- контролювати споживання;
- зручно і своєчасно передавати показання лічильника;
- переглянути планові відключення;
- переглянути графік споживання;
- переглянути історію внесених показників;
- замовити послугу і переглядати її статус.

Для досягнення поставленої мети роботи необхідно вирішити такі задачі:

- проаналізувати особливості підключення бази даних для мобільних додатків;
- розробити модель мобільного додатку;
- розробити структуру мобільного додатку;
- розробити алгоритм функціонування мобільного додатку;
- розробити інтерфейс автоматизованої системи;
- здійснити програмну реалізацію мобільного додатку;
- протестувати розроблений програмний продукт.

Моя частина роботи є саме відображення цих даних у зручному для користувача вигляді, зручному для сприйняття і аналізу. Результатом буде створення інтерфейсу з програмною реалізацією для приємної взаємодії з додатком без підключення до бази даних, що має особисту інформацію про користувачів. Реалізація потребує використання багатьох сучасних технологій які призначені покращити загальну якість додатку.

### **1.5 Висновки**

Розділ аналіз предметної області виступає фундаментом для з'ясування актуальності та необхідності розробки мобільного додатку.

Даний розділ включає огляд основних відомостей про теоретичні та історичні дані виникнення архітектури OS Android. Огляд методологій і технологій розробки Android-додатків при індустріальному розвитку мобільних додатків.

Проведено огляд існуючих мобільних додатків пов'язаних з основними аспектами розробки та надання послуг користувачам згідно обраної тематики роботи.

## 2 ВИБІР МЕТОДУ РІШЕННЯ

### 2.1 Основні концепції розробки мобільного додатку

Проаналізуємо особливості та засоби розробки мобільного додатку.

Мобільний додаток можна створити за допомогою таких мов програмування: Java, C++, C#, Python, Kotlin та інші. У процесі аналізу переваг і недоліків цих засобів та спираючись на власні уподобання для створення Android-додатку було обрано мову програмування Java. Оскільки цю мову вважають офіційною мовою Android та широко застосовують для створення мобільних додатків, а також web-сайтів [9]. Java – це об'єктно-орієнтована мова програмування, яка була розроблена Джеймсом Ослінгом і запущена в 1995 році в якості одного із ключових компонентів платформи Java Sun Microsystems [10].

Одним із середовищ для розробки під Android можна використовувати середовище Eclipse з встановленим плагіном ADT. Мова програмування для розробки Java. Підтримується можливість використання емулятора вбудованого в ADT або на мобільному пристрої з OS Android та майже повна зворотна сумісність версій. Існують декілька різних версій SDK, які використовуються для написання коду під різні версії Android. Крім мови Java підтримується можливість більш низькорівневої розробки з використанням Android NDK (Native Development Kit) на мові C/C++. Для такої платформи створення додатків відрізняється від розробки сайтів. В такому випадку обов'язково приділяється увага інтерфейсу, який повинен бути дружній та інтуїтивно зрозумілий користувачу. Це потрібно для стимуляції бажання використовувати додаток кожен день, важливо приділяти увагу при опрацюванні прототипу і дизайну програми. Розробка додатків вимагає від розробників глибоких знань законів дизайну і розуміння принципів ефективної взаємодії користувача з додатком.

Під платформу Android для розробників існує широкий спектр вибору OS, на яких можна розробляти додатки, а саме використовувати Windows,

Mac або Linux, на відміну від інших де єдиним варіантом є Mac і Windows відповідно.

Для програмування мобільних додатків існує спеціально розроблене середовище, що набирає великої популярності серед розробників таким вважається Android Studio – інтегроване середовище розробки (IDE) для платформи Android. Дане середовище є зручним як для написання програмного коду, так і для створення дизайну мобільного додатку, оскільки містить в собі весь необхідний для цього функціонал. Також в Android Studio є можливість перевірки програми на вбудованому емуляторі Java Virtual Device (JVD). Створення проекту програми розпочинається з вибору мінімальної версії SDK. SDK – це набір засобів розробки, за допомогою яких можна створити програми для певного пакету програм, програмного забезпечення базових засобів розробки, апаратної платформи, комп'ютерної системи, ігрових консолей та ін. Для розробки додатків під Android на мові Java потрібен набір Java Development Kit (JDK). Для тестування мобільного додатку було обрано мінімальну версію SDK: API 21: Android 5.0. Оскільки в такому випадку додаток можна буде використовувати на мобільних пристроях з версією ОС Android 5.0 та вище. [11] Наступним кроком було створення необхідних файлів згідно архітектури MVC. Тобто всі об'єкти програми діляться на три категорії:

1. Model (містять дані та так звану "бізнес-логіку" зазвичай їх використовують для моделювання змісту, з якою працює програма).

2. View (відображають себе на екрані та реагують на дії користувача, заповнюються на основі розмітки XML у файлі макету).

3. Controller (пов'язують об'єкти view і model, тобто містять "логіку додатку").

Перевагами даної архітектури є те, що розділяючи програмний код на класи можна підвищити його функціональність та спростити проектування й розуміння коду загалом. [12] Головними складовими мобільного додатку є



Activity та Layout. Activity – це клас з Android SDK, який відповідає за взаємодію користувача з інформацією на екрані. За допомогою класів реалізується функціонал, який необхідний додатку. Layout – це файли, які описуються на мові XML. На макеті розміщений певний набір об'єктів користувацького інтерфейсу. Layout формується з визначень, кожне визначення використовується для створення об'єктів, які виводяться на екран. [11] Файл LoginActivity.java є головним файлом, з якого починається виконання програми. В файлі містяться методи підключення layout та обробки подій. Даний файл пов'язаний з activity\_edit\_settings.xml, в якому містяться розмітки та кнопки, які відповідають за початок тестування та перегляд інструкції.

Android додаток включає в себе одну або кілька Activity (діяльностей). Діяльність можна представити у вигляді контейнера, що містить користувацький інтерфейс і код, який його запускає. Intents (наміри) складають систему повідомлень на Android. Намір складається з action (дії), яку необхідно виконати (подивитися, редагувати і ін.) і даних.

Дія – це те, що повинно бути зроблено при отриманні наміру і даних, з якими необхідно оперувати. Наміри використовуються при запуску діяльності і при комунікації між різними частинами Android системи. Додаток може отримувати або відправляти наміри. При передачі наміри фактично відправляється повідомлення системі зробити що-небудь, наприклад, запустити нову діяльність в поточному додатку або відкрити іншу програму. Якщо просто відправити намір, це не означає, що щось станеться автоматично. Для цього необхідно зареєструвати intent receiver (приймач намірів), який отримує намір і вказує Android системі, що потрібно зробити: виконати завдання в новій діяльності або запустити іншу програму. Якщо ж є кілька приймачів для отримання intent, можна створити інструмент, що дозволяє користувачеві самому вибрати необхідну дію. Одним наміром можуть бути знайдені кілька приймачів, тому користувач особисто визначає

action, яке потрібно виконати. Наприклад, при довгому натисненні на зображення в галереї, з'являється інструмент вибору, який пропонує відправити картинку через e-mail або соціальні мережі, редагувати або видалити і ін. Якщо система не може знайти підходящий намір, а інструмент вибору не було створено розробником, то додаток потерпить крах і видасть помилку виконання. Тому важливо стежити за тим, щоб вибори для комерційних, чи не націлених на інші діяльності в даному додатку, були створені.

## **2.2 Аналіз обраної мови та середовища програмування**

Для розробки мобільного додатку під платформу Android було вирішено використовувати середовище розробки Android Studio та мову програмування Java. Причини вибору зазначені нижче:

Java значно спрощує розробку адже встановлює жорсткі обмеження на способи передачі параметрів. Це особливо помітно при порівнянні з C++, де параметри можна передавати: за значенням та за посиланням, а враховуючи наявність покажчиків, код програми може стати надзвичайно заплутаним. Мова Java виграє у C++ у багатьох аспектах і насамперед спрощує написання коду за рахунок відсутності: а) покажчиків, б) заголовних файлів, с) множинного наслідування.

Властивості, характерні як для мови Java:

1) вихідний текст програми компілюється в проміжний код, що не залежить від мови і платформи. Цей код надалі виконується в спеціальному керованому середовищі;

2) автоматичний збір сміття і заборона на використання покажчиків. У інших мовах програмування таки як C# допускається обмежене використання покажчиків в блоках коду, позначених як "ненадійні";

3) об'єкти створюються за допомогою ключового слова new, виділення пам'яті здійснюється з "купи" (heap), що знаходиться в розпорядженні середовища виконання;

- 4) багатопотоковість підтримується шляхом блокування об'єктів;
- 5) інтерфейси, множинна реалізація інтерфейсів класом, одноразове наслідування базового класу похідним;
- 6) внутрішні класи;
- 7) відсутність глобальних функцій і констант, всі елементи повинні належати класам;
- 8) масиви і рядки з вбудованою довжиною і перевіркою кордонів;
- 9) не застосовуються оператори "->", "::". У всіх випадках використовується оператор ".";
- 10) null і boolean/bool є ключовими словами;
- 11) будь-яка величина повинна бути проініціалізована до того, як буде використана.

Java дозволяє створювати незалежні від платформи програми шляхом компіляції в проміжне представлення, яке називається байт-кодом.

Переваги мови Java:

1. Одна з основних переваг мови Java – незалежність від платформи, на якій виконуються програми: один і той же код можна запускати під управлінням операційних систем Windows, Linux, MacOS та ін. Це дійсно необхідно, коли програми завантажуються через Інтернет для подальшого виконання під управлінням різних операційних систем.

2. Java – повністю об'єктно-орієнтована мова програмування. Всі сутності в мові Java є об'єктами, за винятком не багатьох основних типів (primitive types), наприклад чисел.

3. Висока надійність. Розробники забезпечили мову Java засобами, що дозволяють виключити саму можливість створювати програми, в яких були б приховані найбільш поширені помилки. Для цього в мові Java зроблено наступне:

- виключена можливість явного виділення і звільнення пам'яті. Пам'ять у мові Java звільняється автоматично за допомогою механізму збірки

сміття. Програміст застрахований від помилок, пов'язаних з неправильним використанням пам'яті;

- введені істинні масиви і заборонена арифметика покажчиків. Тепер програмісти в принципі не можуть стерти дані з пам'яті внаслідок неправильного використання покажчиків;

- виключена можливість переплутати оператор присвоювання з оператором порівняння на рівність. Тепер не можна навіть скопіювати вираз `if (entries = 3)`;

- виключено множинне наслідування. Воно замінене новим поняттям – інтерфейсом. Інтерфейс дає програмісту майже все, що той може отримати від множинного наслідування, уникаючи при цьому складнощів, що виникають при управлінні ієрархіями класів.

Android Studio – це відносно нове середовище розробки андроїд додатків, що базується на платформі IntelliJ IDEA компанії JetBrains. Середовище розробки, зображено на рисунку 2.1, адаптоване для виконання типових завдань, що вирішуються в процесі розробки додатків для платформи Android. У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування додатків, що працюють на пристроях з екранами різної роздільності (планшети, смартфони, ноутбуки, годинники, окуляри тощо). Крім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій, таких як нова уніфікована підсистема складання, тестування і розгортання додатків, заснована на складальному інструментарії Gradle і підтримуюча використання засобів безперервної інтеграції. [13]

Для прискорення розробки додатків представлена колекція типових елементів інтерфейсу і візуальний редактор для їхнього компонування, що надає зручний попередній перегляд різних станів інтерфейсу додатку (наприклад, можна подивитися як інтерфейс буде виглядати для різних

версій Android і для різних розмірів екрану). Для створення нестандартних інтерфейсів присутній майстер створення власних елементів оформлення, що підтримує використання шаблонів.

Android Studio має деякі недоліки, а саме пов'язані зі швидкодією. Процеси виконуються невиправдано довго, оскільки постійно синхронізація з Gradle значно уповільнює компіляцію проектів, в порівнянні з Eclipse. Часті підвисання вікон програми, дуже повільно робота з Spell Checker.

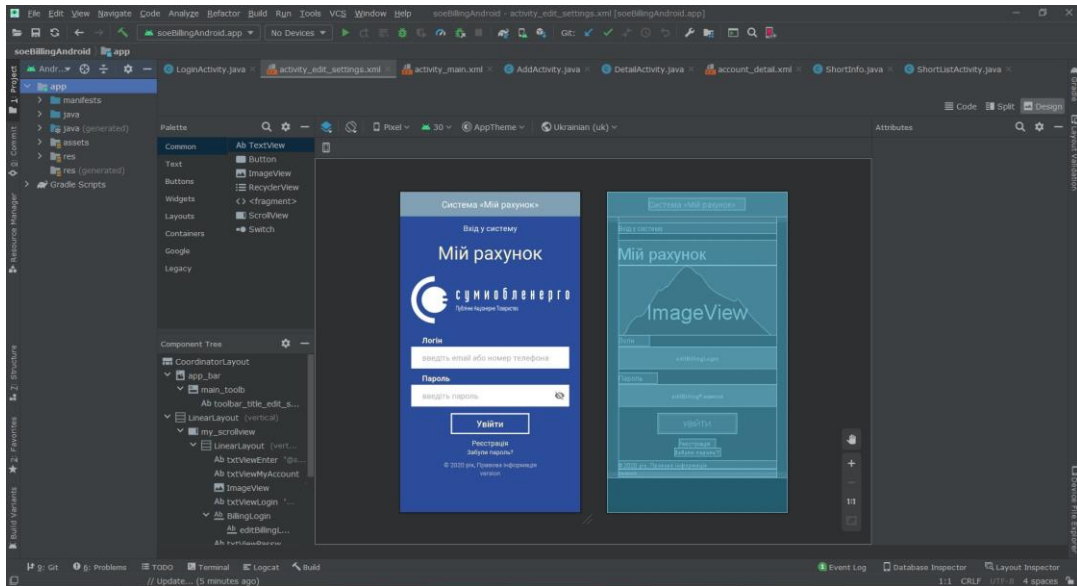


Рисунок 2.1 – Середовище розробки додатку

### 2.3 Висновки

У даному розділі проаналізовано особливості та засоби розробки мобільного додатку. Вирішено використовувати середовище розробки Android Studio та мову програмування Java, головними перевагами яких є простота, надійність та функціонал, що адаптоване до виконання поставленої задачі. Наведено середовище розробки та блок-схему інтерфейсу роботи реєстрації та авторизації користувача мобільного додатку.

## 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

### 3.1 Програмна реалізація

Програмна реалізація мого додатку складається з декількох основних частин:

#### I. Розробка інтерфейсу і дизайну мобільного додатку.

Розробка інтерфейсу і дизайну мобільного додатку – це один із найважливіших етапів створення програмного продукту. Від цієї роботи залежить те, як саме користувач буде сприймати додаток, чи сподобається йому створений дизайн, чи буде додаток зручним в експлуатації, і чи зможе він стати популярним.

Головною метою кожного додатка вважається можливість забезпечити максимальну ефективність і зручність роботи з інформацією, а саме: з графікою, зображенням, документами, базою даних.

Інтерфейс програмного продукту – це зовнішня оболонка будь-якого додатка разом із програмами керування доступом та іншими механізмами керування схованими від користувача, що забезпечує можливість працювати з різними документами, даними та іншою інформацією, що зберігається не тільки в комп'ютері, а й за його межами.

На відміну від моніторів для ПК, мобільні пристрої мають значно менші розміри екрану, тому при проектуванні додатків для них повинні керуватися іншими правилами. Дизайн додатку повинен бути зрозумілим та максимально простим. Розглянемо основні вимоги до дизайну мобільних додатків [14]:

– елементів по мінімуму. Не потрібно перенавантажувати великою кількістю об'єктів і так невеликий простір дисплея мобільного пристрою. Потрібно намагатися вмістити максимальну кількість функціоналу в дружній і лаконічний інтерфейс. Така вимога є головною для розробки не тільки мобільного, а будь-якого ПЗ;

– потрібно зважати на те, що сучасні телефони мають екрани з сенсорним управлінням реалізація якого відбувається за допомогою чуттєвого дотику. Оскільки площа дотику пальця значно більша за розміри покажчика комп'ютерної миші і стилуса, інтерфейс не повинен містити дрібних елементів. В першу чергу такі елементи погано помітні на невеликому екрані. А по-друге, якщо доторкнутися до деякої ділянки дисплея, користувач може натиснути не на той елемент, який він обирав, а це в свою чергу може призвести до зниження зручності використання додатка, або і того гірше;

– всі написи повинні мати розмір достатній для того, щоб користувач міг прочитати їх комфортно з відстані не менше 30 см;

– найбільш часті і важливі елементи інтерфейсу повинні знаходитися по центру екрану та мати достатній розмір для того, щоб виділятися серед інших;

– при перенесенні додатків на мобільну платформу з ПК можна обмежитися створенням зменшеної копії додатка. Для цього весь інтерфейс необхідно оптимізувати: прибрати всі зайві елементи, згрупувавши схожі по функціоналу. Якщо маємо велику кількість об'єктів то для них слід зробити додаткові "вікна", які будуть змінювати один одного на дисплеї. На мобільних платформах під "вікнами" розуміють елементи інтерфейсу, що займають весь простір екрану пристрою. За допомогою графічних елементів-навігаторів користувач здійснює переходи між такими "вікнами", або перетягуючи їх за допомогою пальця (це залежить від платформи і переваг розробників додатка).

У правильно спроектованого дизайну мобільного додатку повинні поєднуватися три основні властивості:

1. Зручність у використанні, тобто інтуїтивний дизайн, об'єднання та використання всіх можливостей мобільного пристрою.

2. Викликати цікавість для користувача. Сьогодні найбільшою популярністю користуються мобільні ігри. Адже кращий засіб змусити

користувача використовувати мобільний додаток, що має велику кількість функціоналу це внести елемент розваги.

3. Приносити користь користувачу. Ліпший рейтинг мають ті додатки, які здатні бути дійсно потрібними.

Розробляючи мобільний додаток мною було дотримано всі вище наведені правил та спроектовано оптимальний на вигляд і інтуїтивно зрозумілий інтерфейс для користування. На рисунку 3.1 наведено вікно додатку "Авторизація користувача".

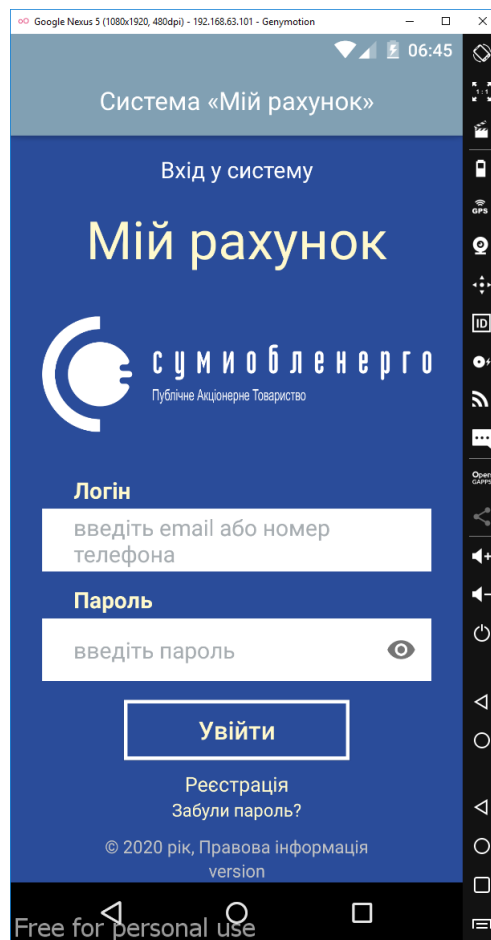


Рисунок 3.1 – Авторизація користувача

## II. Реалізація розмітки Android-додатку.

Під час розробки Android-додатку потрібно враховувати реалізацію візуальної частина вікон окремо від логіки програми. Кожна розмітки екрану має окремий код, що знаходиться у файлах з розширенням .xml у папці проекту, яка має назву "layout".



Розмітка для Android-додатку будується з використанням об'єктів ієрархії View і ViewGroup [15]. View об'єкти це віджети для інтерфейсу користувача, наприклад: кнопки або текстові поля. Об'єкти ViewGroup це невидимий вид контейнерів, які визначають розташування дочірніх представлень, наприклад: в сітці або вертикальному списку.

Розглянемо кілька стандартних типів розміток:

1. `FrameLayout` – це найпростіший тип розмітки. Зазвичай являє собою порожній простір на екрані, який можна заповнити тільки дочірніми об'єктами View або ViewGroup. Розміщення всіх дочірніх елементів `FrameLayout` відбувається шляхом прикріплення їх до верхнього лівого кута екрана. Дана розмітка для дочірнього об'єкта не може визначити різне розташування. Усі наступні дочірні об'єкти View будуть розташовуватися поверх попередніх компонентів, частково або повністю затінюючи їх. Якщо об'єкт, що знаходиться зверху, непрозорий, єдиний дочірній елемент для `FrameLayout` майже завжди розтягнутий до розмірів батьківського контейнера.

2. `LinearLayout` – це такий тип розмітки, що вирівнює всі дочірні об'єкти в одному напрямку тобто вертикально або горизонтально. Атрибут орієнтації `android: orientation` - задає напрямок. В стек один за одним поміщаються всі дочірні елементи, так щоб вертикальний список представлень матиме тільки один дочірній елемент в рядку незалежно від того, наскільки широким він є. Список розташований горизонтально буде розміщувати елементи в один рядок з висотою, що рівна висоті найвищого дочірнього елемента списку.

3. `TableLayout` – даний тип розмітки свої дочірні елементи позиціонує в рядки і стовпці, як це зазвичай звикли робити web-майстри в тезі `table`. `TableLayout` не показує лінії обрамлення для їх стовпців, рядків чи комірок. Ця розмітка може мати рядки з різною кількістю комірок. В розмітці таблиці при її формуванні, якщо це необхідно, можна залишати порожніми деякі

комірки. Для рядків коли створюють розмітки використовують об'єкти `TableRow`, які є дочірніми класами `TableLayout`. Кожен такий об'єкт `TableRow` визначає тільки один рядок в таблиці. Кожен рядок може взагалі не мати комірок або навпаки мати одну і більше комірок, що виступають контейнерами для інших об'єктів. Також в комірку дозволяється вкладати інший `TableLayout` або `LinearLayout`.

4. `RelativeLayout` – розмітка дозволяє дочірнім компонентам визначати свою позицію щодо батьківського компонента або відносно сусідніх дочірніх елементів, наприклад відносно ідентифікатора елемента. У `RelativeLayout` дочірні елементи розташовуються таким чином, щоб перший елемент який розташований по центру екрана та інші елементи, що вирівняні відносно першого елемента, будуть вирівняні відносно центру екрана. При такому розташуванні в оголошенні розмітки XML-файлі, елемент, на який буде спрямоване посилання для позиціонування інші об'єкти представлення, повинен бути оголошений раніше, ніж ті елементи, що звертаються до нього за його ідентифікатором.

5. `GridLayout` – спершу він має схожість з `TableLayout`, та це тільки здається на перший погляд. А насправді він має більший функціонал і набагато зручніший. Дана розмітка відноситься до класу `android.widget.GridLayout` при цьому має клітинки, рядки, колонки як в `TableLayout`, але його елементи здатні гнучко налаштовуватися. У `GridLayout` для будь-якого компонента можна вказати рядок і колонку відповідно в тому місці таблиці він буде розміщений. Не потрібно для кожної комірки вказувати елемент, це потрібно робити тільки для тих комірок, де дійсно повинні бути компоненти. Компоненти можуть розтягуватися на кілька елементів таблиці. Більш того, в одну комірку можна помістити кілька компонентів.

Отже вищеописані розмітки є підкласами `ViewGroup` і успадковують властивості, визначені в класі `View`. Щоб підключити створену розмітку

використовують код в методі onCreate. Для своєї роботи будемо застосовувати розмітку типу `LinearLayout` частина якої наведена нижче:

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">
    <ScrollView
        android:id="@+id/my_scrollview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="25dp"
        android:layout_marginRight="25dp"
        android:orientation="vertical">
        <TextView
            android:id="@+id/txtViewEnter"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp"
            android:gravity="center"
            android:text="@string/enter"
            android:textColor="@color/white"
            android:textSize="18sp"
        />
        <TextView
            android:id="@+id/txtViewMyAccount"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp"
            android:gravity="center"
            android:text="@string/my_account"
            android:textColor="@color/sumy_yellow"
            android:textSize="42sp"
        />
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="160dp"
            android:layout_gravity="center"
            android:focusable="true"
            android:focusableInTouchMode="true"
            android:src="@drawable/logo_soe_new_design" />
        <TextView
            android:id="@+id/txtViewLogin"
            android:layout_width="wrap_content"
            android:layout_height="fill_parent"
            android:layout_marginBottom="3dp"
            android:paddingLeft="25dp"

```

```

        android:text="@string/login"
        android:textColor="@color/sumy_yellow"
        android:textSize="18sp"
        android:textStyle="bold" />
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/BillingLogin"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    app:hintEnabled="false">
    <EditText
        android:id="@+id/editBillingLogin"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:background="@drawable/border"
        android:hint="@string/login_hint"
        android:imeOptions="actionNext|flagNoExtractUi"
        android:inputType="textEmailAddress|textMultiLine"
        android:paddingLeft="25dp"
        android:paddingRight="10dp"
        android:singleLine="true"
        android:maxLines="2"
        android:textColorHint="@color/placeholder" />
</com.google.android.material.textfield.TextInputLayout>
<TextView
    android:id="@+id/txtViewPassword"
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_marginBottom="3dp"
    android:layout_marginTop="10dp"
    android:paddingLeft="25dp"
    android:text="@string/password"
    android:textColor="@color/sumy_yellow"
    android:textSize="18sp"
    android:textStyle="bold" />
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/BillingPassword"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    app:hintEnabled="false"
    app:passwordToggleEnabled="true"
    app:errorTextAppearance="@drawable/border_error">
    <EditText
        android:id="@+id/editBillingPassword"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:background="@drawable/border"
        android:hint="@string/password_hint"
        android:imeOptions="actionDone"
        android:inputType="textPassword"
        android:paddingLeft="25dp"
        android:paddingRight="10dp"
        android:password="true"
        android:singleLine="true"

```

```

        android:textColorHint="@color/placeholder"
    />
</com.google.android.material.textfield.TextInputLayout>
<Button
    android:id="@+id/btnSave"
    android:layout_width="180dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="15dip"
    android:background="@drawable/border_btn_login"
    android:onClick="OnSettingSaveButtonClick"
    android:text="@string/sign_in"
    android:textAllCaps="false"
    android:textColor="@color/sumy_yellow"
    android:textSize="20sp" />
<TextView
    android:id="@+id/btnRegistration"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/transparent_color"
    android:onClick="OnSettingSaveButtonClick"
    android:text="@string/account_registration_btn_registration"
    android:textAllCaps="false"
    android:textColor="@color/sumy_yellow"
    android:textSize="16sp"
    />
<TextView
    android:id="@+id/btnForgot"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:layout_gravity="center"
    android:background="@drawable/transparent_color"
    android:onClick="OnSettingSaveButtonClick"
    android:text="@string/link_forgot"
    android:textAllCaps="false"
    android:textColor="@color/sumy_yellow"
    android:textSize="14sp"
    />
<TextView
    android:id="@+id/footer"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentBottom="true"
    android:gravity="bottom|center"
    android:text="@string/footer"
    android:textColor="@color/labelGray"
    android:textSize="14sp" />
<TextView
    android:id="@+id/version"
    android:layout_width="match_parent"

```

```
        android:layout_height="match_parent"  
        android:layout_alignParentBottom="true"  
        android:gravity="bottom|center"  
        android:text="version"  
        android:textColor="@color/labelGray"  
        android:textSize="14sp" />  
</LinearLayout>
```

### III. Реалізація логіки програмного продукту.

Головним класом при розробці Android-додатку є клас Activity. Представляє собою візуальну активність мобільного додатку та визначає дії, які може виконувати користувач. У процесі роботи такого класу спочатку створюється об'єкт класу Activity, після чого він запускається, відпрацьовує своє призначення та знищується, а користувач гаджета переходить до нового об'єкта.

Створюючи нову активність Android викликає метод onCreate. Даний метод проводить початкове налаштування Activity. Спочатку створюються об'єкти візуального інтерфейсу. Потім метод отримує об'єкт Bundle, який містить попередній стан Activity, якщо він у нього був. Якщо його не було Activity заново створюється, відповідно такий об'єкт приймає значення null. Якщо Activity вже була раніше створена, але перебувала в загальмованому стані, то Bundle містить зв'язану з Activity інформацію.

Після чого викликається метод onStart, а Activity змінює свій стан переходячи у "видимий". Щоб активність відображалася на екрані потрібно викликати метод onResume після чого користувач може з нею взаємодіяти. Якщо користувач вирішує перейти до іншої активності, то система викликає метод onPause. А якщо користувач вирішить повернутися до попередньої Activity, то система викличе знову метод onResume і така активність знову відобразиться на екрані гаджета. Отже, якщо Activity вже невидима, то викликається метод onStop, а після виклику цього методу користувач вирішить повернутися до попередньої активності, тоді система викличе метод onRestart. Завершення роботи активності відбувається викликом методу onDestroy, який виникає тоді коли система вирішить вбити Activity,

або при виклику методу `finish`. Схематичний взаємозв'язок між усіма цими зворотними викликами представлений на рисунку 3.2.

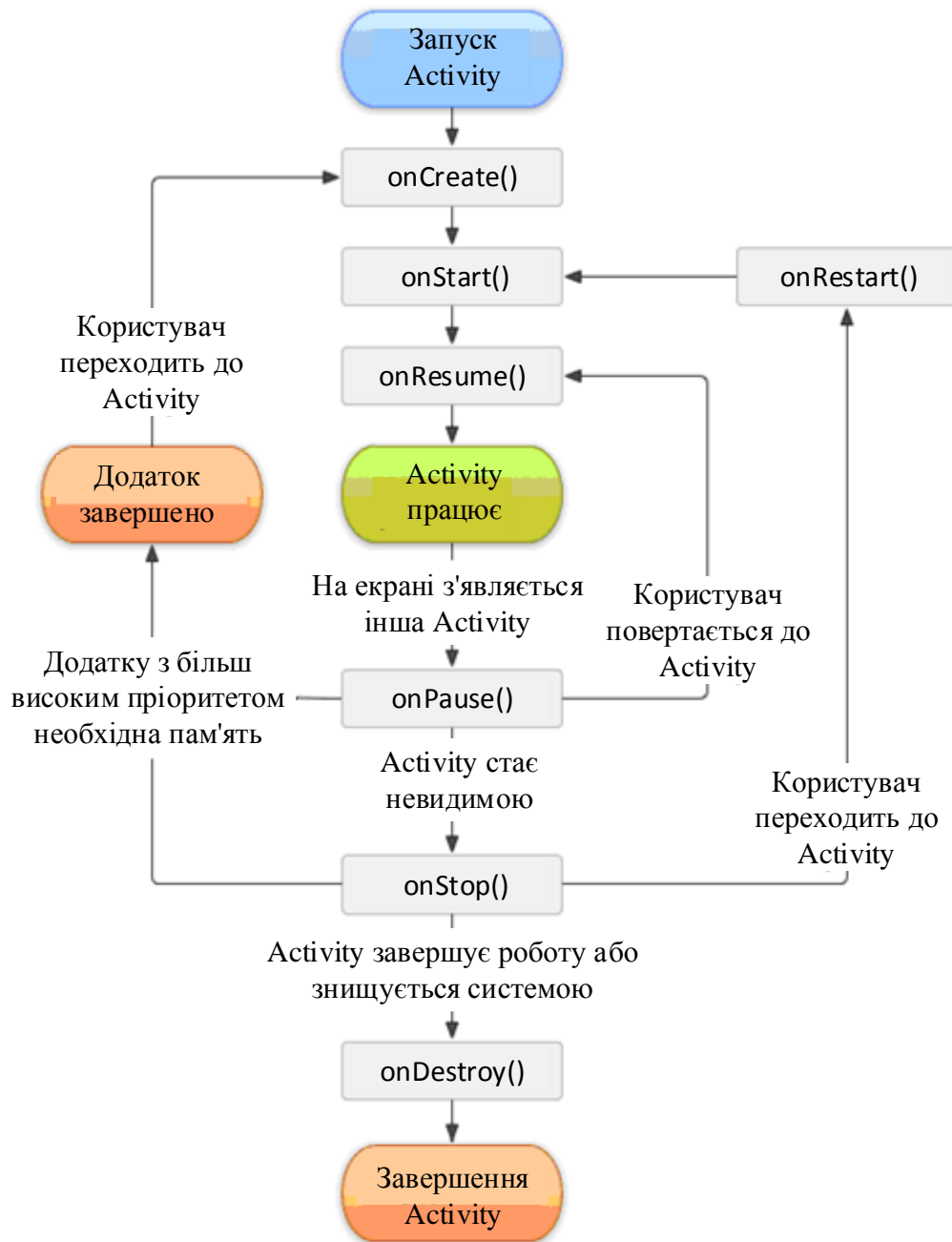


Рисунок 3.2 – Схематичне представлення процесу роботи класу `Activity`

Життєвий цикл `Activity` має три стани [16]:

- 1) активний та виконується – цей інтерфейс користувача знаходиться на передньому плані тобто на вершині стека `Activity`;
- 2) призупинений – даний інтерфейс користувача втратив фокус, та все ще видимий. При такому стані ніякий код не виконується;

3) завершений – інтерфейс користувача невидимий. При такому стані код не виконується.

Запустивши мобільний додаток, першим користувач бачить LoginActivity – вікно авторизації користувача, де йому необхідно ввести свій логін і пароль для подальшої роботи з ним.

Враховуючи вищезазначений метод наведемо частина коду реалізації LoginActivity.java:

```
public class LoginActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_settings);
        Button btnSave = findViewById(R.id.btnSave);
        btnSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String login = inputLogin.getText().toString().trim();
                String password = inputLogin.getText().toString().trim();
                if (!login.isEmpty() && !password.isEmpty()) {
                    checkLogin(login, password);
                } else {
                    Toast.makeText(getApplicationContext(), "Please enter the login
and password",
                        Toast.LENGTH_LONG).show();
                }
            }
        });
        private void checkLogin (final String login, final String password) {
            String req = "req_login";
            StringRequest strReq = new StringRequest(Method.POST,
AppConfig.URL_LOGIN, new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    Log.d(TAG, "Login Response: " + response.toString());
                    hideDialog();
                    try {
                        JSONObject jsonObj = new JSONObject(response);
                        boolean error = jsonObj.getBoolean("Error");
                        if (!error) {
                            session.setLogin(true);
                            String id = jsonObj.getString("id");
                            JSONObject user = jsonObj.getJSONObject("user");
                            String name = user.getString("name");
                            String login = user.getString("login");
                            String email = user.getString("email");
                            String phone = user.getString("phone");
                            String created = user.getString("created");
                            db.addUser(name, login, id, email, phone, created);
                            Intent intent = new Intent(getBaseContext(),
ShortListActivity.class);
                            startActivity(intent);
                        }
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                }
            });
        }
    }
}
```





б. Зовнішні переривання - вхідні дзвінки, повідомлення, відключення Wi-Fi і інші перешкоди не повинні зупинити або погіршити функціонування програми.

Також при тестуванні мобільних додатків враховується і дотримання конфіденційності. Це необхідно для збереження особистих даних користувача, його логінів, паролів та іншої інформації, яка міститься у пристрої.

Тестування мобільного додатку виконувалося в середовищі розробки Android Studio за допомогою альтернативного Android емулятора Genymotion, рисунок 3.3. Даний емулятор легко налаштовується та поєднує простоту використання з передовою продуктивністю 3D графіки, надаючи користувачу практично необмежені можливості роботи з мобільною ОС. Емулятор дає можливість в простий спосіб тестувати додаток на мобільному телефоні не використовуючи його фізично.

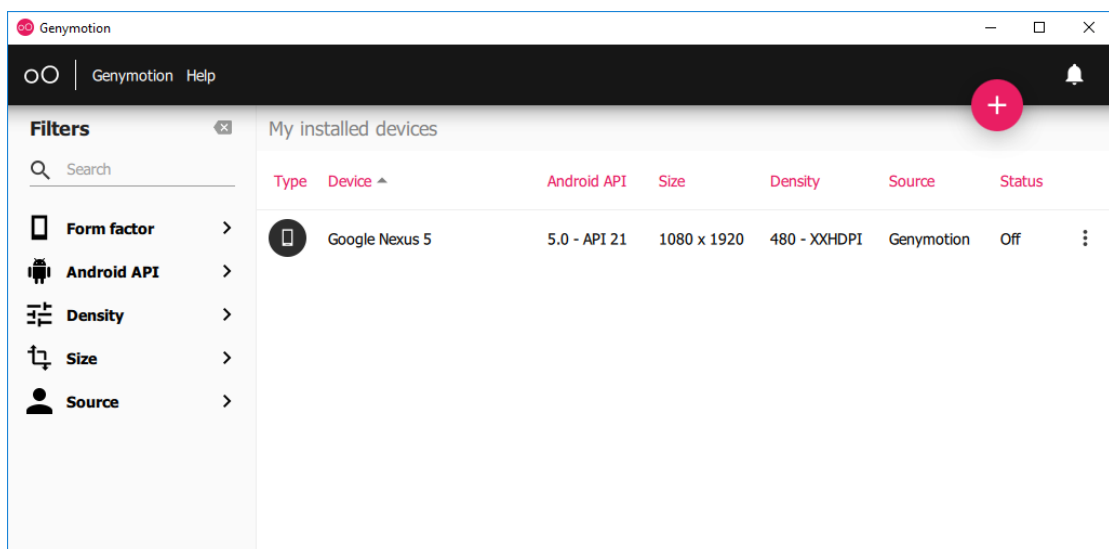


Рисунок 3.3 – Емулятор Genymotion

Genymotion має такі можливості:

- швидке завантаження і запуск попередньо налаштованих образів Android. Список пристроїв постійно розширюється;
- GPS з налаштованим координатами і батарея з налаштованим рівнем;

- відображення: апаратне прискорення з OpenGL, режим повноекранного перегляду;
- консоль, яка дозволяє вам взаємодіяти з вашою віртуальною машиною, використовуючи командний рядок;
- підтримка Android Debug Bridge;
- емуляція передньої і задньої камери;
- функції скринкаста і віддаленого управління (в комерційній версії);
- модулі до Eclipse і Android Studio;
- підтримує Linux, Windows і Mac;
- у майбутніх релізах очікується поява нових сенсорів: гіроскоп, яскравість екрану, температура, вектор повороту і багато іншого;
- працює на порядок швидше емуляторів від Google (не тільки на процесорах Intel);
- загальний буфер обміну віртуального пристрою і ПК;
- можливість редагування IMEI/MEID віртуального пристрою (в комерційній версії);
- Pixel Perfect - режим відображення, при якому 1 піксель екрану пристрою будуть відображатися з використанням тільки одного пікселя на вашому моніторі (в комерційній версії);
- Factory Reset - повне скидання віртуального пристрою (в комерційній версії);
- дублювання образу в списку віртуальних пристроїв (в комерційній версії);
- редагування кількості процесорів (CPU) і кількості оперативної пам'яті (RAM) в настройках віртуального пристрою;
- контроль датчиків в коді програми через Genymotion Java API (в комерційній версії);
- автоматичне оновлення образів для нової версії програми (в комерційній версії);

- генерування архіву log файлів при зверненні в технічну підтримку;
- збереження налаштувань GPS віджета до наступного запуску (в комерційній версії);
- виправлення помилок падіння емулятора, додано інформацію в логах;
- віджет емуляція голосових дзвінків (в комерційній версії);
- емуляція роботи Інтернету через GPRS, Edge, 3G, 4G і ін.

### 3.3 Приклади реалізації додатку

Для прикладу покажемо пару основних вікон розробленого мобільного додатку для демонстрації його функціоналу та працездатності.

Перше вікно з яким зустрічається користувач запустивши мобільний додаток – це вікно авторизації та реєстрації користувача, зображено на рисунку 3.4.

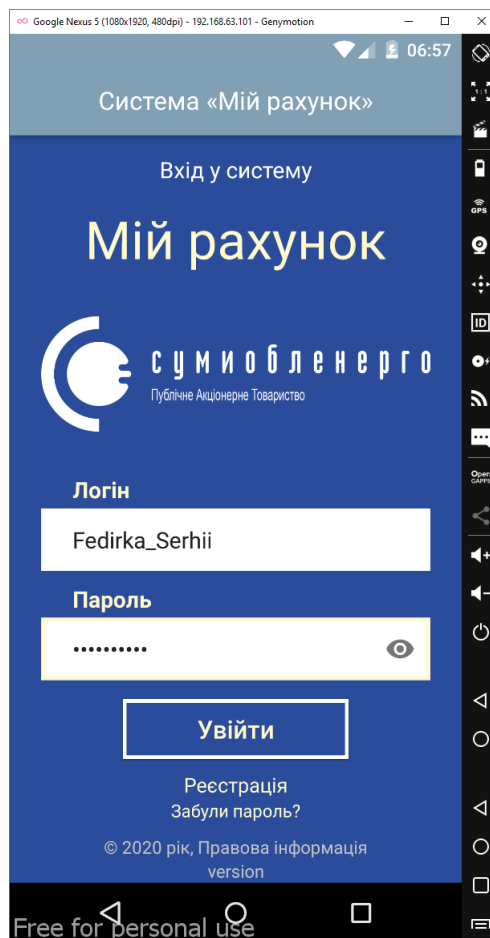


Рисунок 3.4 – Вікно авторизації користувача

У другому вікні користувач може обрати свій особистий рахунок чи створити інший, якщо в нього їх декілька. Дане вікно зображено на рисунку 3.5.

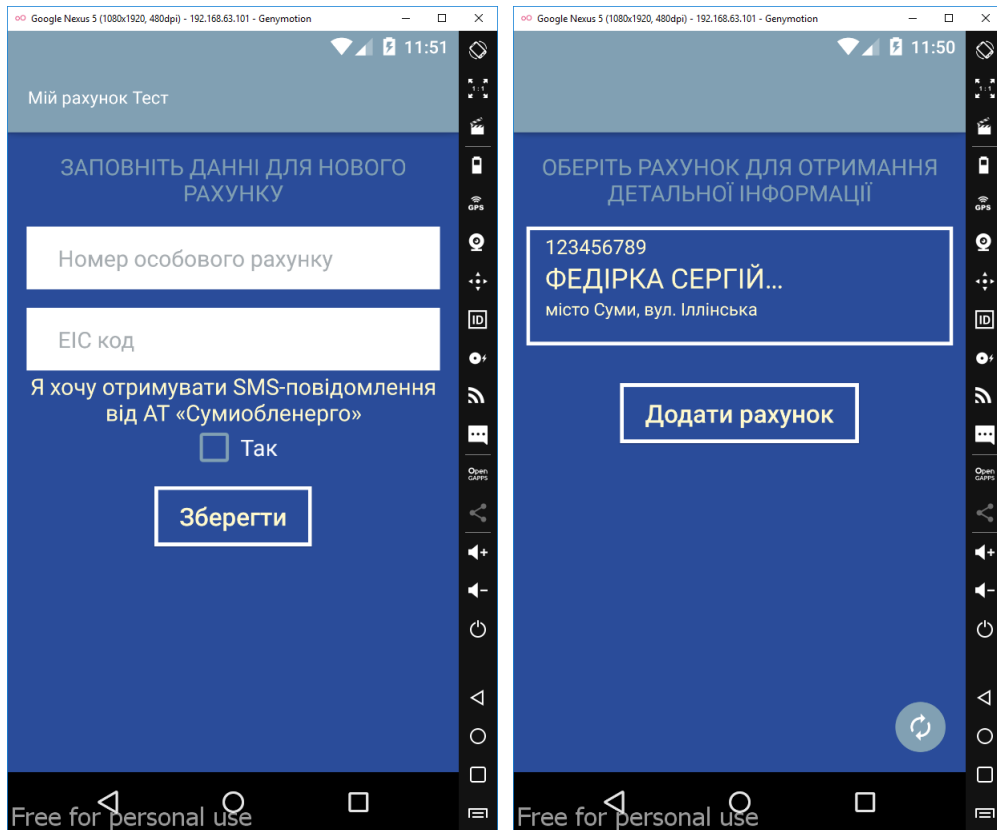


Рисунок 3.5 – Вікно вибору особового рахунку

Головним вікном виступає вікно навігації, адже саме в ньому користувач обирає відповідний розділ додатка для отримання чи занесення інформації. В даному вікні можна переглянути договірну потужність та споживання за відповідний період, а також можна вибрати такі розділи: внести покази, компенсація, послуги, графік споживання, історія внесених показів. На рисунку 3.6 зображено вікно навігації.

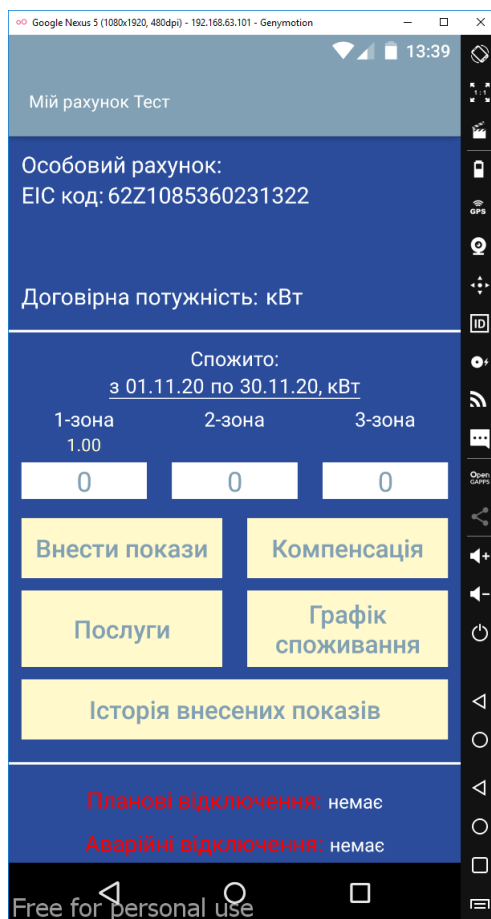


Рисунок 3.6 – Вікно навігації

Для того щоб передати дані користувач повинен обрати вікно внесення показників лічильника, адже саме в ньому користувач заносить дані споживання електроенергії згідно встановленого тарифу. На рисунку 3.7 зображено вікно внесення показників лічильника. Для прикладу було внесено показники лічильника 112 кВт.

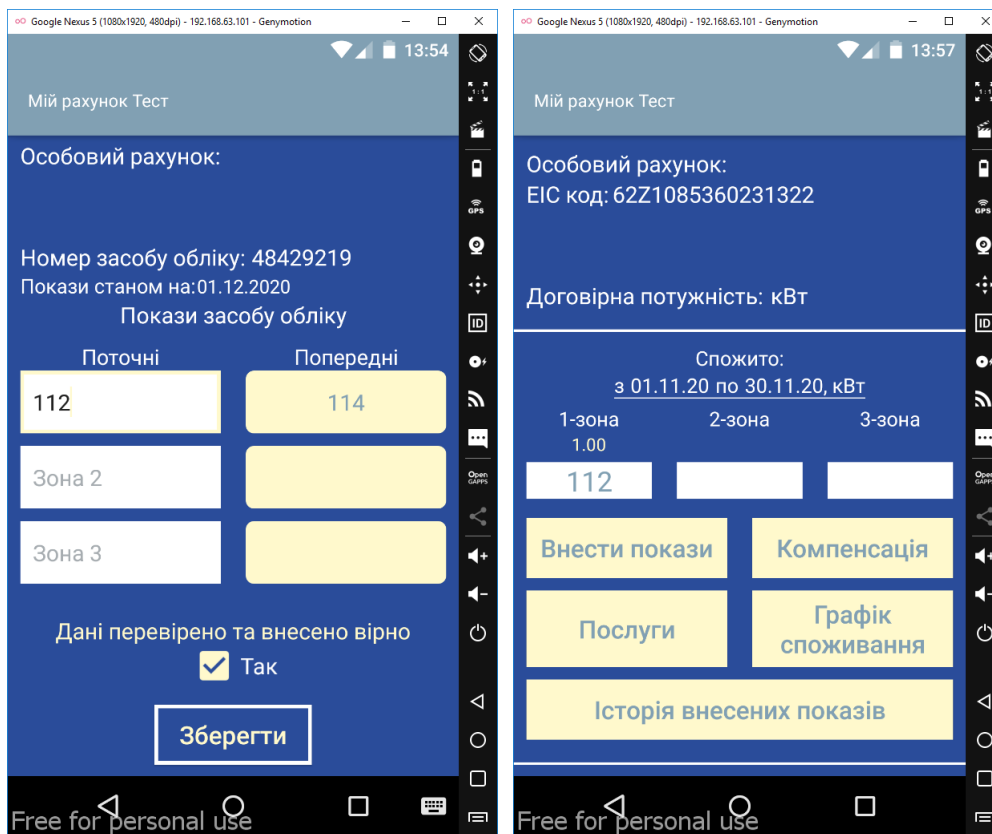


Рисунок 3.7 – Вікно внесення показників лічильника

Якщо користувач захоче переглянути внесені дані тоді йому потрібно зайти у вікно перегляду внесених даних. Також в цьому вікні відображаються внесені показники контрольного огляду. Для прикладу були обрані внесені дані попереднього місяця. Дані вносилися 02.11.2020 показники, що було занесено становили 114 кВт відповідні дані відображаються у вікні перегляду внесених даних зображено на рисунку 3.8.

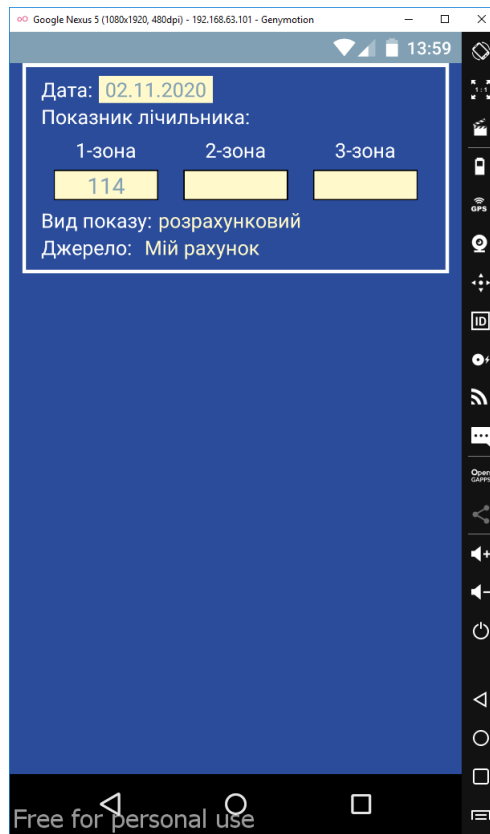


Рисунок 3.8 – Вікно перегляду внесених даних

Для замовлення користувачем додаткової послуги в тому числі зміна тарифного плану використовується вікно послуги зображено на рисунку 3.9.



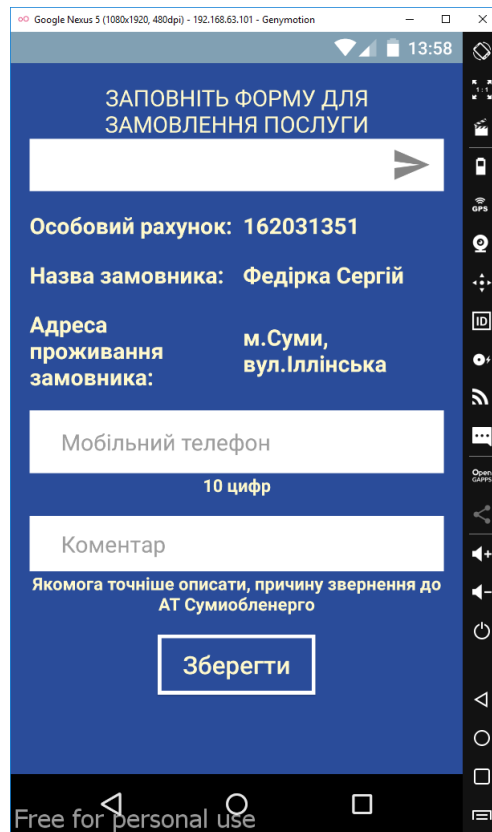


Рисунок 3.9 – Вікно замовлення послуг

Вікна компенсації та графіка споживання підв'язані до бази даних Товариства. Особиста інформація користувача не використовується в даній роботі.

### 3.4 Висновки

У розділі розроблений інтуїтивно зрозумілий інтерфейс та дизайн мобільного додатку з приведеною ілюстрацією вікна авторизації користувача. Вибрана розмітка екрану Android-додатку з наведенням частини коду вибраного рішення, а також реалізована логіка програмного продукту з дотриманням всіх взаємозв'язків між всіма станами кожного методу.

Вибрано та проведено тестування мобільного додатку за допомогою емулятора Genymotion. Програмний продукт відповідає всім сучасним нормам, легкий у використанні має інтуїтивно зрозумілий інтерфейс. Приведені основні активності, що реалізовані в даному додатку для демонстрації працездатності та функціональності.

## ВИСНОВКИ

У даній роботі було розроблено Android-додаток для користувачів послуг АТ "Сумиобленерго" по передачі показів лічильника електроенергії.

У ході виконання цієї роботи було розглянуто і вивчено різні методології та технології розробки Android-додатку. Виконаний аналіз та вибрано мову програмування та середовище розробки.

Проаналізовано найбільш популярні існуючі мобільні аналоги, щоб врахувати всі недоліки і взяти всі переваги для проектування свого додатку зробивши користування ним максимально зручним та водночас простим та зрозумілим.

Проаналізовано основні інструменти які призначені для створення мобільного додатку. Розроблено інтуїтивно-зрозумілий інтерфейс користувача, розроблено розмітку вікон авторизації Android-додатку та логіку програмного продукту.

Вибрано метод тестування та виконано тестування додатку. Тестування підтвердило правильність та ефективність розробленого додатка по внесенню даних спожитої електроенергії користувача за розрахунковий період. Додаток дозволяє заощаджувати час користувача за рахунок мінімального введення даних. Дає можливість не тільки вносити дані але й переглядати як свої так і дані внесені під час контрольного огляду за вибраний період. Також можна переглянути аварійні відключення та самому подавати заявки про відсутність електропостачання та аварійні події.

Додаток значно полегшує користувачеві процедуру передачі даних адже для цього не потрібно користуватися ПК чи ноутбуком достатньо дістати свій смартфон підключити його до Інтернету і витратити кілька хвилин будучи в будь-якому місці навіть під час поїздки.

## СПИСОК ЛІТЕРАТУРИ

1. Дэрси Л., Кондер Ш. Android за 24 часа. Программирование приложений под операционную систему Google; Рид Групп - Москва, 2011. - 464 с.
2. Android Development Tutorial. Часть 2. [Электронный ресурс]. – Режим доступа до ресурсу: <https://habrahabr.ru/post/115122/>
3. Android. [Электронный ресурс]. – Режим доступа до ресурсу: <https://uk.wikipedia.org/wiki/Android>.
4. Мобільний додаток. [Электронный ресурс]. – Режим доступа до ресурсу:  
[https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B1%D1%96%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9\\_%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA/](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B1%D1%96%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA/)
5. Медникс З., Дорнин Л., Мик Б., Накамура М. Программирование под Android. Издательский дом "Питер" – 496 с.
6. Аналіз методів і технологій розробки мобільних додатків для платформи Android: навч. посіб./ О.В. Шматко, А.О. Поляков, В.М. Федорченко. – Харків: НТУ "ХПІ", 2018. – 284 с.
7. Добряк П. Модели жизненного цикла / П. Добряк. – СПб.: Символ-Плюс, 2016. – 132 с.
8. Голян В.В., Кравченко О.К., Порівняння моделей життєвих циклів програмного забезпечення з метою виявлення найефективнішого. – Харків: ХНУ, стаття, 2019 – 8 с.
9. Орлов С. А. Теория и практика языков программирования: учебник для вузов. 2-е изд. Стандарт третьего поколения. СПб.: Питер, 2017. 688 с.
10. Горбань А. Г. Програмування в Java, 2008. – 310 с.
11. Філліпс Б., Стюарт К., Марсикано К. Android. Програмування для професіоналів. 3-е изд. СПб.: Питер, 2017. – 688 с.

12. Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. 3-е изд. СПб.: Питер, 2016. – 512 с.

13. Android Studio. [Электронный ресурс]. – Режим доступа до ресурсу: [https://uk.wikipedia.org/wiki/Android\\_Studio](https://uk.wikipedia.org/wiki/Android_Studio)

14. Методологія проектування та інструментарій для створення мобільних додатків. [Електронний ресурс]. – Режим доступу до ресурсу: [http://www.kpi.kharkov.ua/archive/%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B0\\_%D0%BF%D0%B5%D1%80%D1%96%D0%BE%D0%B4%D0%B8%D0%BA%D0%B0/vestnik/%D0%9D%D0%BE%D0%B2%D1%8B%D0%B5%20%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%B2%20%D1%81%D0%BE%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D1%8B%D1%85%20%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F%D1%85/2013/56/%D0%9C%D0%95%D0%A2%D0%9E%D0%94%D0%9E%D0%9B%D0%9E%D0%93%D0%86%D0%AF%20%D0%9F%D0%A0%D0%9E%D0%95%D0%9A%D0%A2%D0%A3%D0%92%D0%90%D0%9D%D0%9D%D0%A%D0%9F%20%D0%A2%D0%90%20%D0%86%D0%9D%D0%A1%D0%A2%D0%A0%D0%A3%D0%9C%D0%95%D0%9D%D0%A2%D0%90%D0%A0%D0%86%D0%99.pdf](http://www.kpi.kharkov.ua/archive/%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B0_%D0%BF%D0%B5%D1%80%D1%96%D0%BE%D0%B4%D0%B8%D0%BA%D0%B0/vestnik/%D0%9D%D0%BE%D0%B2%D1%8B%D0%B5%20%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%B2%20%D1%81%D0%BE%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D1%8B%D1%85%20%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F%D1%85/2013/56/%D0%9C%D0%95%D0%A2%D0%9E%D0%94%D0%9E%D0%9B%D0%9E%D0%93%D0%86%D0%AF%20%D0%9F%D0%A0%D0%9E%D0%95%D0%9A%D0%A2%D0%A3%D0%92%D0%90%D0%9D%D0%9D%D0%A%D0%9F%20%D0%A2%D0%90%20%D0%86%D0%9D%D0%A1%D0%A2%D0%A0%D0%A3%D0%9C%D0%95%D0%9D%D0%A2%D0%90%D0%A0%D0%86%D0%99.pdf)

15. Layout. [Електронний ресурс]. – Режим доступу до ресурсу: <http://developer.alexanderklimov.ru/android/theory/layout.php>

16. Activity. [Електронний ресурс]. – Режим доступу до ресурсу: <http://metanit.com/java/android/2.1.php>

17. Тестування мобільних додатків для Android. [Електронний ресурс]. – Режим доступу до ресурсу: <https://web24.pro/android-rozrobka/testuvannya-mobilnyh-dodatkov-dlya-android/>

## ДОДАТОК А

### Основний програмний код Android-додатку

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="local.soe.itps02.soebilling"
    android:installLocation="auto">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.FLASHLIGHT" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <uses-feature
        android:name="android.hardware.camera.flash"
        android:required="false" />
    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/logo512x512"
        android:label="@string/app_name"
        android:roundIcon="@drawable/logo"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity
            android:name=".LoginActivity"
            android:theme="@style/AppTheme.LoginNoActionBarFullScreen"
            android:configChanges="orientation|screenSize">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".ShortListActivity"
            android:label="@string/my_accounts"
            android:theme="@style/AppTheme.NoActionBar"
            android:launchMode="singleTop"
            android:configChanges="orientation|screenSize">
        </activity>

        <activity android:name=".AddActivity">
            android:label="@string/my_accounts"
            android:theme="@style/AppTheme.NoActionBar"
            android:launchMode="singleTop"
            android:configChanges="orientation|screenSize">
        </activity>
        <activity android:name=".DetailActivity">
            android:label="@string/my_accounts"
            android:theme="@style/AppTheme.NoActionBar"
            android:launchMode="singleTop"
        </activity>
    </application>
</manifest>
```

```

        android:configChanges="orientation|screenSize">
    </activity>
    <activity android:name=".IndexActivity"
        android:label="@string/my_accounts"
        android:theme="@style/AppTheme.NoActionBar"
        android:launchMode="singleTop"
        android:configChanges="orientation|screenSize">
    </activity>
    <activity android:name=".HistoryListActivity"
        android:label="@string/my_accounts"
        android:theme="@style/AppTheme.NoActionBar"
        android:launchMode="singleTop"
        android:configChanges="orientation|screenSize">
    </activity>
    <activity android:name=".RegistrationActivity"
        android:label="@string/my_accounts"
        android:theme="@style/AppTheme.NoActionBar"
        android:launchMode="singleTop"
        android:configChanges="orientation|screenSize">
    </activity>
    <activity android:name=".ServicesListActivity"
        android:label="@string/my_accounts"
        android:theme="@style/AppTheme.NoActionBar"
        android:launchMode="singleTop"
        android:configChanges="orientation|screenSize">
    </activity>
    <activity android:name=".IndexListActivity">
        android:label="@string/my_accounts"
        android:theme="@style/AppTheme.NoActionBar"
        android:launchMode="singleTop"
        android:configChanges="orientation|screenSize">
    </activity>
</application>
</manifest>

```

### LoginActivity.java

```

package local.soe.itps02.soebilling;
import android.content.Intent;
import android.nfc.Tag;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONObject;
import java.lang.reflect.Method;

public class LoginActivity extends AppCompatActivity {
    Toast toast;
    EditText editBillingLogin;
    EditText editBillingPassword;
    String nameLogin;
    String namePassword;
    @Override

```

```

protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_edit_settings);
    editBillingLogin = findViewById(R.id.editBillingLogin);
    editBillingPassword = findViewById(R.id.editBillingPassword);
    Button btnSave = findViewById(R.id.btnSave);
    TextView btnRegistration = findViewById(R.id.btnRegistration);
    btnSave.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            nameLogin = editBillingLogin.getText().toString();
            namePassword = editBillingPassword.getText().toString();
            if (nameLogin.equals("Fedirka_Serhii") &&
namePassword.equals("1234567890")) {
                Intent intent = new Intent(getBaseContext(),
ShortListActivity.class);
                startActivity(intent);
            } else {
                toast = Toast.makeText(LoginActivity.this, "Не вірний Логін чи
Пароль", Toast.LENGTH_LONG);
                toast.show();
            }
        }
    });
    btnRegistration.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(getBaseContext(),
RegistrationActivity.class);
            startActivity(intent);
            String login = inputLogin.getText().toString().trim();
            String password = inputLogin.getText().toString().trim();
            if (!login.isEmpty() && !password.isEmpty()) {
                checkLogin(login, password);
            } else {
                Toast.makeText(getApplicationContext(), "Please enter the login
and password",
                Toast.LENGTH_LONG).show();
            }
        }
    });
    private void checkLogin (final String login, final String password) {
        String req = "req_login";
        StringRequest strReq = new StringRequest(Method.POST,
AppConfig.URL_LOGIN, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Log.d(TAG, "Login Response: " + response.toString());
                hideDialog();
                try {
                    JSONObject job = new JSONObject(response);
                    boolean error = job.getBoolean("Error");
                    if (!error) {
                        session.setLogin(true);
                        String id = job.getString("id");
                        JSONObject user = job.getJSONObject("user");
                        String name = user.getString("name");
                        String login = user.getString("login");
                        String email = user.getString("email");
                        String phone = user.getString("phone");
                        String created = user.getString("created");
                        db.addUser(name, login, id, email, phone, created);
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

```

ShortListActivity.class);
Intent intent = new Intent(getBaseContext(),
startActivity(intent);
finish());
    }
};
}
});
}
}
}
}

```

#### activity\_edit\_settings.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:context=".ActivityEditSettings">

<com.google.android.material.appbar.AppBarLayout
android:id="@+id/app_bar"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:theme="@style/AppTheme.AppBarOverlay">

<androidx.appcompat.widget.Toolbar
android:id="@+id/main_toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:title="@string/my_accounts"
app:popupTheme="@style/AppTheme.PopupOverlay">

<TextView
android:id="@+id/toolbar_title_edit_settings"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:text="@string/login_label"
android:textColor="@color/white"
android:textSize="20sp" />
</androidx.appcompat.widget.Toolbar>

</com.google.android.material.appbar.AppBarLayout>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
app:layout_behavior="@string/appbar_scrolling_view_behavior">

<ScrollView
android:id="@+id/my_scrollview"
android:layout_width="match_parent"
android:layout_height="wrap_content">

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

```



```

android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="25dp"
android:layout_marginRight="25dp"
android:orientation="vertical">

<TextView
    android:id="@+id/txtViewEnter"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:gravity="center"
    android:text="@string/enter"
    android:textColor="@color/white"
    android:textSize="18sp"
/>

<TextView
    android:id="@+id/txtViewMyAccount"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:gravity="center"
    android:text="@string/my_account"
    android:textColor="@color/sumy_yellow"
    android:textSize="42sp"
/>

<ImageView
    android:layout_width="wrap_content"
    android:layout_height="160dp"
    android:layout_gravity="center"
    android:focusable="true"
    android:focusableInTouchMode="true"
    android:src="@drawable/logo_soe_new_design" />

<TextView
    android:id="@+id/txtViewLogin"
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_marginBottom="3dp"
    android:paddingLeft="25dp"
    android:text="@string/login"
    android:textColor="@color/sumy_yellow"
    android:textSize="18sp"
    android:textStyle="bold" />

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/BillingLogin"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    app:hintEnabled="false">

    <EditText
        android:id="@+id/editBillingLogin"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:background="@drawable/border"
        android:hint="@string/login_hint"
        android:imeOptions="actionNext|flagNoExtractUi"
        android:inputType="textEmailAddress|textMultiLine"

```

```

        android:paddingLeft="25dp"
        android:paddingRight="10dp"
        android:singleLine="true"
        android:maxLines="2"
        android:textColorHint="@color/placeholder" />
</com.google.android.material.textfield.TextInputLayout>

<TextView
    android:id="@+id/txtViewPassword"
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_marginBottom="3dp"
    android:layout_marginTop="10dp"
    android:paddingLeft="25dp"
    android:text="@string/password"
    android:textColor="@color/sumy_yellow"
    android:textSize="18sp"
    android:textStyle="bold" />

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/BillingPassword"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    app:hintEnabled="false"
    app:passwordToggleEnabled="true"
    app:errorTextAppearance="@drawable/border_error">

    <EditText
        android:id="@+id/editBillingPassword"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:background="@drawable/border"
        android:hint="@string/password_hint"
        android:imeOptions="actionDone"
        android:inputType="textPassword"
        android:paddingLeft="25dp"
        android:paddingRight="10dp"
        android:password="true"
        android:singleLine="true"
        android:textColorHint="@color/placeholder"
        />
</com.google.android.material.textfield.TextInputLayout>

<Button
    android:id="@+id/btnSave"
    android:layout_width="180dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="15dip"
    android:background="@drawable/border_btn_login"
    android:onClick="OnSettingSaveButtonClick"
    android:text="@string/sign_in"
    android:textAllCaps="false"
    android:textColor="@color/sumy_yellow"
    android:textSize="20sp" />

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

```

```

        android:layout_gravity="center"
        android:background="@drawable/transparent_color"
        android:onClick="OnSettingSaveButtonClick"
        android:text="@string/account_registration_btn_registration"
        android:textAllCaps="false"
        android:textColor="@color/sumy_yellow"
        android:textSize="16sp"
    />
    <TextView
        android:id="@+id/btnForgot"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:layout_gravity="center"
        android:background="@drawable/transparent_color"
        android:onClick="OnSettingSaveButtonClick"
        android:text="@string/link_forgot"
        android:textAllCaps="false"
        android:textColor="@color/sumy_yellow"
        android:textSize="14sp"
    />

    <TextView
        android:id="@+id/footer"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentBottom="true"
        android:gravity="bottom|center"
        android:text="@string/footer"
        android:textColor="@color/labelGray"
        android:textSize="14sp" />
    <TextView
        android:id="@+id/version"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentBottom="true"
        android:gravity="bottom|center"
        android:text="version"
        android:textColor="@color/labelGray"
        android:textSize="14sp" />
    </LinearLayout>
</ScrollView>

</LinearLayout>
</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

#### RegistrationActivity.java

```

package local.soe.itps02.soebilling;
import android.os.Bundle;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class RegistrationActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);
    }
}

```

## activity\_registration.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:mask="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/scrollReg"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/reg"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="15dp">

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/wrapperEmail"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp"
            app:hintEnabled="false">

            <EditText
                android:id="@+id/Email"
                android:layout_width="fill_parent"
                android:layout_height="50dp"
                android:background="@drawable/border"
                android:hint="@string/account_registration_email"
                android:imeOptions="flagNoExtractUi"
                android:inputType="textEmailAddress"
                android:paddingLeft="25dp"
                android:paddingRight="10dp"
                android:singleLine="true"
                android:textColorHint="@color/placeholder" />
        </com.google.android.material.textfield.TextInputLayout>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/registration_or"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_marginBottom="5dp"
                android:layout_marginTop="5dp"
                android:gravity="center"
                android:text="@string/account_registration_or"
                android:textColor="@color/sumy_yellow"
                android:textSize="18sp" />

            <ImageButton
                android:id="@+id/hint_icon_btn"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

```

```

        android:background="@color/transparent"
        android:onClick="tip"
        app:srcCompat="@drawable/button_hint"
        android:focusableInTouchMode="true"
        android:focusable="true" />
</LinearLayout>

<br.com.sapereaude.maskedEditText.MaskedEditText
    android:id="@+id/Mobile"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:background="@drawable/border"
    android:hint="YYXXXXXXX"
    android:imeOptions="actionNext|flagNoExtractUi"
    android:inputType="phone"
    android:paddingLeft="25dp"
    android:paddingRight="25dp"
    android:typeface="monospace"
    app:keep_hint="true"
    mask:allowed_chars="1234567890"
    mask:mask="+380(##)###-##-##" />

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/wrapperName"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    app:hintEnabled="false">

    <EditText
        android:id="@+id/Name"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:background="@drawable/border"
        android:hint="@string/account_registration_name"
        android:imeOptions="flagNoExtractUi"
        android:inputType="textCapSentences"
        android:paddingLeft="25dp"
        android:paddingRight="10dp"
        android:singleLine="true"
        android:textColorHint="@color/placeholder" />
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/wrapperSurname"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    app:hintEnabled="false">

    <EditText
        android:id="@+id/Surname"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:background="@drawable/border"
        android:hint="@string/account_registration_surname"
        android:imeOptions="flagNoExtractUi"
        android:inputType="textCapSentences"
        android:paddingLeft="25dp"
        android:paddingRight="10dp"
        android:singleLine="true"

```

```

        android:textColorHint="@color/placeholder" />
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/wrapperPassword"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="15dp"
    android:layout_marginTop="15dp"
    app:hintEnabled="false"
    app:passwordToggleEnabled="true">

    <EditText
        android:id="@+id/Password"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:background="@drawable/border"
        android:hint="@string/account_registration_password"
        android:imeOptions="flagNoExtractUi"
        android:inputType="textPassword"

        android:paddingLeft="25dp"
        android:paddingRight="10dp"
        android:singleLine="true"
        android:textColorHint="@color/placeholder" />
</com.google.android.material.textfield.TextInputLayout>

<com.ms.square.android.expandabletextview.ExpandableTextView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:expandableTextView="http://schemas.android.com/apk/res-auto"
    android:id="@+id/expand_text_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    expandableTextView:animDuration="200"
    expandableTextView:maxCollapsedLines="3">

    <TextView
        android:id="@id/expandable_text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAlignment="center"
        android:textColor="@color/sumy_yellow"
        android:textSize="16sp" />

    <ImageButton
        android:id="@id/expand_collapse"
        android:layout_width="match_parent"
        android:layout_height="15dp"
        android:layout_gravity="end"
        android:background="@color/sumy_gray"
        android:padding="16dp" />
<!--</FrameLayout>-->
</com.ms.square.android.expandabletextview.ExpandableTextView>

<CheckBox
    android:id="@+id/checkboxAgreement"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="10dp"
    android:checked="false"

```

```

        android:scaleX="1.3"
        android:scaleY="1.3"
        android:textColor="@color/pplaceholder"
        android:textIsSelectable="false"
        android:theme="@style/MyCheckBox" />

<Button
    android:id="@+id/btnRegistration"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="15dip"
    android:background="@drawable/border_btn_login"
    android:onClick="OnRegistrationButtonClick"
    android:paddingBottom="10dp"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:paddingTop="10dp"
    android:text="@string/account_registration_btn_registration"
    android:textAllCaps="false"
    android:textColor="@color/sumy_yellow"
    android:textSize="20sp" />

</LinearLayout>
</ScrollView>

```

#### ShortListActivity.java

```

package local.soe.itps02.soebilling;
import android.content.Intent;
import android.os.Bundle;
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.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class ShortListActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        RecyclerView rv = findViewById(R.id.account_list);
        setupRecyclerView(rv);
    }
    private void setupRecyclerView(@NonNull RecyclerView recyclerView) {
        RecyclerViewAdapter adapter = new RecyclerViewAdapter(
            new ArrayList<ShortInfo>(
                Arrays.asList(
                    new ShortInfo("", 12345679, "місто Суми, вул.
Іллінська", 123456789, "Федірка Сергій Володимирович")
                )
            );
    }
}

```

```

        recyclerView.setAdapter(adapter);
    }
    public class RecyclerViewAdapter
        extends RecyclerView.Adapter<RecyclerView.ViewHolder> {
        private static final int TYPE_HEADER = 0;
        private static final int TYPE_FOOTER = 1;
        private static final int TYPE_ITEM = 2;
        private final List<ShortInfo> mValues;
        public RecyclerViewAdapter(List<ShortInfo> items) {
            mValues = items;
        }
        @Override
        public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
            if (viewType == TYPE_ITEM) {
                View itemView =
LayoutInflater.from(parent.getContext()).inflate(R.layout.account_list_content,
parent, false);
                return new ItemViewHolder(itemView);
            } else if (viewType == TYPE_HEADER) {
                View itemView =
LayoutInflater.from(parent.getContext()).inflate(R.layout.activity_main_header,
parent, false);
                return new HeaderViewHolder(itemView);
            } else if (viewType == TYPE_FOOTER) {
                View itemView =
LayoutInflater.from(parent.getContext()).inflate(R.layout.activity_main_footer,
parent, false);
                return new FooterViewHolder(itemView);
            } else return null;
        }
        @Override
        public void onBindViewHolder(final RecyclerView.ViewHolder holder, final int
position) {
            if (holder instanceof HeaderViewHolder) {
                HeaderViewHolder headerHolder = (HeaderViewHolder) holder;
                ((HeaderViewHolder) holder).headerTitle.setText("ОБЕРІТЬ РАХУНОК ДЛЯ
ОТРИМАННЯ ДЕТАЛЬНОЇ ІНФОРМАЦІЇ");
            } else if (holder instanceof FooterViewHolder) {
                FooterViewHolder footerHolder = (FooterViewHolder) holder;
                footerHolder.btnAdd.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        Intent intent = new Intent(getBaseContext(),
AddActivity.class);
                        startActivity(intent);
                    }
                });
            } else if (holder instanceof ItemViewHolder) {
                final ItemViewHolder itemViewHolder = (ItemViewHolder) holder;
                ShortInfo shortInfo = mValues.get(position - 1);
                itemViewHolder.mItem = shortInfo;
                itemViewHolder.mLSView.setText(shortInfo.getLs().toString());
                itemViewHolder.mFIOView.setText(shortInfo.getFio());
                itemViewHolder.mAddressView.setText(shortInfo.getAddress());
                itemViewHolder.mView.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Intent intent = new Intent(getBaseContext(),
DetailActivity.class);
                        startActivity(intent);
                    }
                });
            }
        }
    }
}

```



```

        });
    }
}
@Override
public int getItemViewType(int position) {
    if (position == 0) {
        return TYPE_HEADER;
    } else if (position == mValues.size() + 1) {
        return TYPE_FOOTER;
    }
    return TYPE_ITEM;
}
@Override
public int getItemCount() {
    return mValues.size() + 2;
}
private class HeaderViewHolder extends RecyclerView.ViewHolder {
    TextView headerTitle;
    public HeaderViewHolder(View view) {
        super(view);
        headerTitle = (TextView) view.findViewById(R.id.textViewInstruction);
    }
}
private class FooterViewHolder extends RecyclerView.ViewHolder {
    Button addBtn;
    public FooterViewHolder(View view) {
        super(view);
        addBtn = (Button) view.findViewById(R.id.addBtn);
    }
}
private class ItemViewHolder extends RecyclerView.ViewHolder {
    View mView;
    TextView mLsView;
    TextView mFIOView;
    TextView mAddressView;
    ShortInfo mItem;
    public ItemViewHolder(View view) {
        super(view);
        registerForContextMenu(view);
        mView = view;
        mLsView = (TextView) view.findViewById(R.id.ls);
        mFIOView = (TextView) view.findViewById(R.id.fio);
        mFIOView.setAllCaps(true);
        mAddressView = (TextView) view.findViewById(R.id.address);
    }
}
public void clear() {
    int size = this.mValues.size();
    this.mValues.clear();
    notifyItemRangeRemoved(0, size);
}
}
}
}

```

activity\_mail.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"

```

```

android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true">
<androidx.swiperefreshlayout.widget.SwipeRefreshLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/swipeRefreshLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    >
    <FrameLayout
        android:id="@+id/frameLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:visibility="visible">

        <include layout="@layout/account_list" />
    </FrameLayout>

    <RelativeLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/loading"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:visibility="gone"
        android:layout_gravity="center">

        <RelativeLayout
            android:id="@+id/rrprog"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center">

            <ProgressBar
                android:id="@+id/progressBar"
                android:layout_width="230dp"
                android:layout_height="230dp"
                android:minHeight="5dp"
                android:maxHeight="5dp"
                android:layout_gravity="center" />
        </RelativeLayout>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="18dp"
            android:text="@string/loading"
            android:textColor="@color/mainBlue"
            android:layout_alignBottom="@id/rrprog"
            android:layout_alignTop="@id/rrprog" />
    </RelativeLayout>

</LinearLayout>
</androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

```

```

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="@dimen/fab_margin"
    app:borderWidth="0dp"
    app:elevation="0dp"
    app:fabSize="mini"
    app:srcCompat="@drawable/ic_autorenew"
    app:backgroundTint="@color/sumy_gray"/>

<com.google.android.material.appbar.AppBarLayout
    android:id="@+id/app_bar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/main_toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:title="@string/my_accounts"
        app:popupTheme="@style/AppTheme.PopupOverlay">

    </androidx.appcompat.widget.Toolbar>

</com.google.android.material.appbar.AppBarLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

#### account\_list.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.recyclerview.widget.RecyclerView
    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/account_list"
    android:name="local.soe.itps02.soebilling.AccountListFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="0dp"
    android:scrollbars="vertical"
    android:clickable="true"
    android:overScrollMode="ifContentScrolls"
    app:layoutManager="LinearLayoutManager"
    tools:context="local.soe.itps02.soebilling.ActivityMain"
    tools:listitem="@layout/account_list_content"
/>

```

#### ShortInfo.java

```

package local.soe.itps02.soebilling;
public class ShortInfo {
    public ShortInfo(String user_abon_id, Integer a, String address, Integer ls,
String fio) {
        this.user_abon_id = user_abon_id;
        this.a = a;
        this.address = address;
        this.ls = ls;
        this.fio = fio;
    }
}

```

```

}
String user_abon_id;
Integer a;
String address;
Integer ls;
String fio;
public String getUser_abon_id() {
    return user_abon_id;
}
public void setUser_abon_id(String user_abon_id) {
    this.user_abon_id = user_abon_id;
}
public Integer getA() {
    return a;
}
public void setA(Integer a) {
    this.a = a;
}
public String getAddress() {
    return address;
}
public void setAddress(String address) {
    this.address = address;
}
public Integer getLs() {
    return ls;
}
public void setLs(Integer ls) {
    this.ls = ls;
}
public String getFio() {
    return fio;
}
public void setFio(String fio) {
    this.fio = fio;
}
}}

```

#### account\_list\_content.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:clickable="true"
    android:background="@drawable/border_main_list"
    android:orientation="vertical"
    android:layout_marginBottom="20dp"
    android:layout_marginRight="10dp"
    android:layout_marginLeft="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:paddingTop="5dp"
    android:paddingBottom="20dp">
    <TextView
        android:id="@+id/ls"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textColor="@color/sumy_yellow"
        />

```

```

<TextView
    android:id="@+id/fio"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/l1"
    android:textSize="22sp"
    android:textColor="@color/mainBlue"
    android:ellipsize="end"
    android:maxLines="1"
/>

<TextView
    android:id="@+id/address"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:textColor="@color/sumy_yellow"
    android:textSize="14sp"
    android:ellipsize="end"
    android:maxLines="1"
/>

```

```
</LinearLayout>
```

activity\_main\_header.xml

```

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

    <TextView
        android:id="@+id/textViewInstruction"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:layout_marginBottom="15dp"
        android:layout_marginRight="0dp"
        android:layout_marginLeft="0dp"
        android:gravity="center_horizontal"
        android:text="@string/mainInstruction"
        android:textColor="@color/sumy_gray"
        android:textSize="18sp"
    />

</LinearLayout>

```

activity\_main\_footer.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <Button
        android:id="@+id/addBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="15dp"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:paddingLeft="20dp"
    />

```

```

        android:paddingRight="20dp"
        android:background="@drawable/border_btn_login"
        android:onClick="OnTransmitMeterReadingButtonClick"
        android:text="@string/btn_add_account"
        android:textAllCaps="false"
        android:textColor="@color/sumy_yellow"
        android:textSize="20sp" />
</LinearLayout>

```

#### AddActivity.java

```

package local.soe.itps02.soebilling;
import android.os.Bundle;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class AddActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_account_add);
    }
}

```

#### activity\_account\_add.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="local.soe.itps02.soebilling.ActivityAccountAdd">

    <LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="15dp"
        android:focusable="true"
        android:focusableInTouchMode="true"
        >

        <TextView
            android:id="@+id/add_instruction"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/account_add_instruction"
            android:textAllCaps="true"
            android:gravity="center_horizontal"
            android:textColor="@color/sumy_gray"
            android:textSize="18sp"/>

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/addLS"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp"
            app:hintEnabled="false">

            <EditText
                android:id="@+id/editLS"

```

```

        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:background="@drawable/border"
        android:hint="@string/hint_ls"
        android:imeOptions="actionNext"
        android:inputType="number"
        android:paddingLeft="25dp"
        android:paddingRight="10dp"
        android:singleLine="true"
        android:textColorHint="@color/placeholder" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/addInvoiceNumber"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    app:hintEnabled="false">

    <EditText
        android:id="@+id/editInvoiceNumber"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:background="@drawable/border"
        android:hint="@string/hint_last_receipt"
        android:paddingLeft="25dp"
        android:paddingRight="10dp"
        android:singleLine="true"
        android:textColorHint="@color/placeholder" />
</com.google.android.material.textfield.TextInputLayout>
<TextView
    android:id="@+id/txtSendSMS"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/send_sms"
    android:textSize="18sp"
    android:textColor="@color/mainBlue"/>
<CheckBox
    android:id="@+id/checkboxSendSMS"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="@string/yes"
    android:textColor="@color/white"
    android:scaleX="1.3"
    android:scaleY="1.3"
    android:theme="@style/MyCheckBox"
    android:focusable="false"/>

<Button
    android:id="@+id/btnAddAccount"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="15dp"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:background="@drawable/border_btn_login"

```

```

        android:onClick="OnAddAccountButtonClick"
        android:text="@string/save"
        android:textAllCaps="false"
        android:textColor="@color/sumy_yellow"
        android:textSize="20sp"
    />
</LinearLayout>

```

```
</ScrollView>
```

#### DetailActivity.java

```

package local.soe.itps02.soebilling;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class DetailActivity extends AppCompatActivity implements View.OnClickListener
{
    TextView fkvt1;
    TextView fkvt2;
    TextView fkvt3;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.account_detail);
        TextView consumedValueFrom = findViewById(R.id.consumed_value_from);
        consumedValueFrom.setText("01.11.20");
        TextView consumedValueTo = findViewById(R.id.consumed_value_to);
        consumedValueTo.setText("30.11.20");
        fkvt1 = findViewById(R.id.fkvt1);
        fkvt1.setText("0");
        fkvt2 = findViewById(R.id.fkvt2);
        fkvt2.setText("0");
        fkvt3 = findViewById(R.id.fkvt3);
        fkvt3.setText("0");
        Button enterMeters = findViewById(R.id.enter_meters);
        enterMeters.setOnClickListener(this);
        Button services = findViewById(R.id.services);
        services.setOnClickListener(this);
        Button historyOfIndicators = findViewById(R.id.historyOfIndicators);
        historyOfIndicators.setOnClickListener(this);
    }
    @Override
    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.enter_meters:
                Intent intentEnterMeters = new Intent(this, IndexListActivity.class);
                startActivityForResult(intentEnterMeters, 1);
                break;
            case R.id.services:
                Intent intentServices = new Intent(this, ServicesListActivity.class);
                startActivityForResult(intentServices, 1);
                break;
            case R.id.historyOfIndicators:
                Intent intentHistoryOfIndicators = new Intent(this,
HistoryListActivity.class);
                startActivityForResult(intentHistoryOfIndicators, 1);

```



```

                break;
            default:
                break;
        }
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable
@org.jetbrains.annotations.Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (data == null) {
            return;
        }
        String meter_readings1 = data.getStringExtra("meter_readings1");
        String meter_readings2 = data.getStringExtra("meter_readings2");
        String meter_readings3 = data.getStringExtra("meter_readings3");
        fkvt1.setText(meter_readings1);
        fkvt2.setText(meter_readings2);
        fkvt3.setText(meter_readings3);
    }
}

```

#### account\_detail.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal">
    <RelativeLayout android:layout_width="match_parent"
        android:layout_height="match_parent"
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:visibility="gone"
        android:id="@+id/load">
        <!--android:layout_marginTop="65dp"-->
        <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:tools="http://schemas.android.com/tools"
            android:id="@+id/loading"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_gravity="center">

            <RelativeLayout
                android:id="@+id/rrprog"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:gravity="center">

                <ProgressBar
                    android:id="@+id/progressBar"
                    android:layout_width="230dp"
                    android:layout_height="230dp"
                    android:layout_gravity="center"
                    android:maxHeight="5dp"
                    android:minHeight="5dp" />
            </RelativeLayout>

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_alignBottom="@id/rrprog"
                android:layout_alignTop="@id/rrprog"
                android:gravity="center"

```

```

        android:text="@string/loading"
        android:textColor="@color/mainBlue"
        android:textSize="18dp" />
    </RelativeLayout>
</RelativeLayout>
<androidx.core.widget.NestedScrollView
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/my_scrollview"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:visibility="visible">

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/fullinfo_layout"
        android:visibility="visible"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:layout_marginTop="10dp"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">

                <TextView
                    android:id="@+id/personal_account_colon"
                    android:layout_width="wrap_content"
                    android:layout_height="match_parent"
                    android:text="@string/personal_account_colon"
                    android:textColor="@color/white"
                    android:textSize="18sp"
                    tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
                />

                <TextView
                    android:id="@+id/ls"
                    android:layout_width="wrap_content"
                    android:layout_height="match_parent"
                    android:gravity="bottom"
                    android:paddingLeft="3dp"
                    android:textColor="@color/white"
                    android:textSize="18sp"
                    tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
                />
            </LinearLayout>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="match_parent"
                    android:text="@string/personal_account_colon"
                    android:textColor="@color/white"
                    android:textSize="18sp"
                    tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
                />
            </LinearLayout>
        </LinearLayout>
    </LinearLayout>
</NestedScrollView>
</RelativeLayout>
</RelativeLayout>

```

```

        android:id="@+id/eic_label"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:text="@string/eic"
        android:textColor="@color/white"
        android:textSize="18sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
    />

    <TextView
        android:id="@+id/eic"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="bottom"
        android:paddingLeft="3dp"
        android:textColor="@color/white"
        android:textSize="18sp"
        android:text="62Z1085360231322"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
    />

</LinearLayout>

<TextView
    android:id="@+id/fio"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/mainBlue"
    android:textSize="24sp"
    tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

<TextView
    android:id="@+id/address"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/white"
    android:textSize="18sp"
    tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/power_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/power"
        android:textColor="@color/white"
        android:textSize="18sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
    />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

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

```

```

        android:paddingLeft="3dp"
        android:textColor="@color/white"
        android:textSize="18sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

        <TextView
            android:id="@+id/power_kw"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:paddingLeft="3dp"
            android:text="@string/power_kw"
            android:textColor="@color/white"
            android:textSize="18sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />
    </LinearLayout>
</LinearLayout>

</LinearLayout>

<View
    android:layout_width="fill_parent"
    android:layout_height="2dp"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="15dp"
    android:background="@color/white" />

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/text_underline"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/consumed1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/consumed"
            android:textColor="@color/white"
            android:textSize="16sp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/consumed2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/consumed_from"

```

```

        android:textColor="@color/white"
        android:textSize="16sp" />

<TextView
    android:id="@+id/consumed_value_from"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="5dp"
    android:paddingRight="5dp"
    android:text="01.08.17"
    android:textColor="@color/white"
    android:textSize="16sp" />

<TextView
    android:id="@+id/consumed3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/consumed_to"
    android:textColor="@color/white"
    android:textSize="16sp" />

<TextView
    android:id="@+id/consumed_value_to"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="5dp"
    android:text="31.08.17"
    android:textColor="@color/white"
    android:textSize="16sp" />

<TextView
    android:id="@+id/consumed4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/comma_kw"
    android:textColor="@color/white"
    android:textSize="16sp" />
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:gravity="center">

    <LinearLayout
        android:id="@+id/LLzone1"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:orientation="vertical">

        <TextView
            android:id="@+id/zone1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="@string/zone1"

```

```

        android:textColor="@color/white"
        android:textSize="16sp" />

<TextView
    android:id="@+id/kzon1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center"
    android:text="1.00"
    android:textColor="@color/sumy_yellow"
    android:textSize="@dimen/koeff" />

<TextView
    android:id="@+id/fkvt1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="5dp"
    android:layout_marginTop="5dp"
    android:layout_weight="1"
    android:background="@color/white"
    android:text="123"
    android:gravity="center"
    android:textColor="@color/sumy_gray"
    android:textSize="22sp" />
</LinearLayout>

<LinearLayout
    android:id="@+id/LLzone2"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:orientation="vertical">

    <TextView
        android:id="@+id/zone2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/zone2"
        android:textColor="@color/white"
        android:textSize="16sp" />

    <TextView
        android:id="@+id/kzon2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:textColor="@color/sumy_yellow"
        android:textSize="@dimen/koeff" />

    <TextView
        android:id="@+id/fkvt2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginBottom="5dp"

```

```

        android:layout_marginTop="5dp"
        android:layout_weight="1"
        android:background="@color/white"
        android:text="1234567890"
        android:gravity="center"
        android:textColor="@color/sumy_gray"
        android:textSize="22sp" />
</LinearLayout>

<LinearLayout
    android:id="@+id/LLzone3"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:orientation="vertical">

    <TextView
        android:id="@+id/zone3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/zone3"
        android:textColor="@color/white"
        android:textSize="16sp" />

    <TextView
        android:id="@+id/kzon3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:textColor="@color/sumy_yellow"
        android:textSize="@dimen/koef" />

    <TextView
        android:id="@+id/fkvt3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginBottom="5dp"
        android:layout_marginTop="5dp"
        android:layout_weight="1"
        android:background="@color/white"
        android:text="1234567890"
        android:gravity="center"
        android:textColor="@color/sumy_gray"
        android:textSize="22sp" />
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"

```

```

android:layout_marginTop="10dp"
android:orientation="horizontal">

<Button
    android:id="@+id/enter_meters"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:background="@color/sumy_yellow"
    android:gravity="center"
    android:onClick="OnActionButtonClick"
    android:text="@string/enter_metrics"
    android:textAllCaps="false"
    android:textColor="@color/sumy_gray"
    android:textSize="20sp" />
<Button
    android:id="@+id/compensation"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:background="@color/sumy_yellow"
    android:gravity="center"
    android:onClick="OnActionButtonClick"
    android:text="@string/compensation"
    android:textAllCaps="false"
    android:textColor="@color/sumy_gray"
    android:textSize="20sp" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/services"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:background="@color/sumy_yellow"
        android:gravity="center"
        android:onClick="OnActionButtonClick"
        android:text="@string/services"
        android:textAllCaps="false"
        android:textColor="@color/sumy_gray"
        android:textSize="20sp" />
    <Button
        android:id="@+id/consumptionSchedule"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_weight="1"

```



```

        android:background="@color/sumy_yellow"
        android:gravity="center"
        android:onClick="OnActionButtonClick"
        android:text="@string/consumptionSchedule_btn"
        android:textAllCaps="false"
        android:textColor="@color/sumy_gray"
        android:textSize="20sp" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="3dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/historyOfIndicators"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:background="@color/sumy_yellow"
        android:gravity="center"
        android:onClick="OnActionButtonClick"
        android:text="@string/historyOfIndicators_btn"
        android:textAllCaps="false"
        android:textColor="@color/sumy_gray"
        android:textSize="20sp" />
</LinearLayout>

<View
    android:layout_width="fill_parent"
    android:layout_height="2dp"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="15dp"
    android:background="@color/white" />

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="10dp"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/scheduled_shutdown_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/scheduled_shutdown"
        android:textColor="@color/red"
        android:textSize="18sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
    />

    <TextView
        android:id="@+id/scheduled_shutdown_value"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="5dp"
        android:text="@string/scheduled_shutdown_absent"

```

```

        android:textColor="@color/white"
        android:textSize="15sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
    />
</LinearLayout>

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/list_scheduled_shutdown"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:orientation="vertical">

</androidx.recyclerview.widget.RecyclerView>

<!--emergency begin-->

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="10dp"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/emergency_shutdown_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/emergency_shutdown"
        android:textColor="@color/red"
        android:textSize="18sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
    />

    <TextView
        android:id="@+id/emergency_shutdown_value"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="5dp"
        android:text="@string/scheduled_shutdown_absent"
        android:textColor="@color/white"
        android:textSize="15sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
    />

</LinearLayout>

<LinearLayout
    android:id="@+id/emergency_shutdown_reason"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:orientation="horizontal">

    <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:orientation="horizontal">

    <TextView
        android:id="@+id/emergency_label_beginning"
        android:layout_width="0dp"
        android:layout_weight=".50"
        android:layout_height="wrap_content"
        android:text="@string/date"
        android:textColor="@color/white"
        android:textSize="18sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

    <TextView
        android:id="@+id/emergency_reason_beginning"
        android:layout_width="0dp"
        android:layout_weight=".50"
        android:layout_height="wrap_content"
        android:paddingLeft="5dp"
        android:text="B2018-12-27 07:00:00"
        android:textColor="@color/white"
        android:textSize="18sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/emergency_label_plan_end"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight=".50"
        android:text="@string/plan_avar"
        android:textColor="@color/white"
        android:textSize="18sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

    <TextView
        android:id="@+id/emergency_reason_plan_end"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight=".50"
        android:layout_gravity="bottom"
        android:paddingLeft="5dp"
        android:text="P2018-12-27 07:00:00"
        android:textColor="@color/white"
        android:textSize="18sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"

```

```

        android:layout_height="match_parent"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/emergency_label_fact_end"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight=".50"
            android:text="@string/fact_avar"
            android:textColor="@color/white"
            android:textSize="18sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

        <TextView
            android:id="@+id/emergency_reason_fact_end"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight=".50"
            android:layout_gravity="bottom"
            android:paddingLeft="5dp"
            android:paddingRight="5dp"
            android:text="F2018-12-27 09:00:00 test test test test"
            android:textColor="@color/white"
            android:textSize="18sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/emergency_label_reason"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight=".50"
            android:text="@string/reason"
            android:textColor="@color/white"
            android:textSize="18sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

        <TextView
            android:id="@+id/emergency_reason_value"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight=".50"
            android:paddingLeft="5dp"
            android:paddingRight="5dp"
            android:text="Аварийный ремонт РКТП-817, замена шкафа КТП-
817"

            android:textColor="@color/white"
            android:textSize="18sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />
    </LinearLayout>
</LinearLayout>

```

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    </LinearLayout>
</LinearLayout>

<!--TARIFF-->
<!-- <TextView
    android:id="@+id/tariff"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="@string/tariff"
    android:textColor="@color/hardGray"
    android:textSize="16sp"
    tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:orientation="horizontal">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="vertical">

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

```

```

        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/tariff_to"
        android:textColor="@color/hardGray"
        android:textSize="16sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

    <TextView
        android:id="@+id/tariff_including"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/tariff_including"
        android:textColor="@color/hardGray"
        android:textSize="16sp"

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />
</LinearLayout>

    <TextView
        android:id="@+id/tariff_to_value"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="2dp"
        android:paddingRight="2dp"
        android:textColor="@color/hardGray"
        android:textSize="16sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
/>

    <TextView
        android:id="@+id/kopecks_to"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/kopecks"
        android:textColor="@color/hardGray"
        android:textSize="16sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
/>

</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/tariff_from"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tariff_from"
        android:textColor="@color/hardGray"
        android:textSize="16sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
/>

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

```

```

        android:layout_height="wrap_content"
        android:paddingLeft="2dp"
        android:paddingRight="2dp"
        android:textColor="@color/hardGray"
        android:textSize="16sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
    />

    <TextView
        android:id="@+id/kopecks_from"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/kopecks"
        android:textColor="@color/hardGray"
        android:textSize="16sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"
    />

</LinearLayout-->
<!--TARIFF-->

</LinearLayout>

</LinearLayout>
</androidx.core.widget.NestedScrollView>
</LinearLayout>

```

#### IndexListActivity.java

```

package local.soe.itps02.soebilling;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class IndexListActivity extends AppCompatActivity implements
View.OnClickListener {
    EditText editZone1;
    EditText editZone2;
    EditText editZone3;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_account_transmit_meter_reading);
        editZone1 = findViewById(R.id.editZone1);
        editZone2 = findViewById(R.id.editZone2);
        editZone3 = findViewById(R.id.editZone3);
        TextView indicatorsAsOfValue = findViewById(R.id.indicators_as_of_value);
        indicatorsAsOfValue.setText("01.12.2020");
        TextView previousReadings1 = findViewById(R.id.previousReadings1);
        previousReadings1.setText("114");
        Button btnTransmitMeterReading = findViewById(R.id.btnTransmitMeterReading);
        btnTransmitMeterReading.setOnClickListener(this);
    }
    @Override
    public void onClick(View view) {
        Intent intent = new Intent();
        intent.putExtra("meter_readings1", editZone1.getText().toString());
        intent.putExtra("meter_readings2", editZone2.getText().toString());
    }
}

```

```

        intent.putExtra("meter_readings3", editZone3.getText().toString());
        setResult(RESULT_OK, intent);
        finish();
    }
}

```

#### activity\_account\_transmit\_meter\_reading.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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/scrollview_transmit"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/lltransmitMeterReading"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:orientation="vertical"
        android:focusable="true"
        android:focusableInTouchMode="true">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/personal_account_colon"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/personal_account_colon"
                android:textColor="@color/white"
                android:textSize="18sp"
                tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"/>

            <TextView
                android:id="@+id/ls"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textSize="18sp"
                android:textColor="@color/white"
                tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

        </LinearLayout>

        <TextView
            android:id="@+id/fio"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="@color/mainBlue"
            android:textSize="24sp"
            tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />

        <TextView
            android:id="@+id/address"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="@color/white"
            android:textSize="18sp"

```



```

tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/counter_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/counter"
        android:textColor="@color/white"
        android:textSize="18sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"/>
    <TextView
        android:id="@+id/counter_value"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingLeft="5dp"
        android:text="48429219"
        android:textColor="@color/white"
        android:textSize="18sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/indicators_as_of_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/indicators"
        android:textColor="@color/white"
        android:textSize="16sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail"/>
    <TextView
        android:id="@+id/indicators_as_of_value"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="1dp"
        android:text="01.01.2018"
        android:textColor="@color/white"
        android:textSize="15sp"
        tools:context="local.soe.itps02.soebilling.FragmentAccountDetail" />
</LinearLayout>
<ImageButton
    android:id="@+id/flashlight"
    android:layout_width="128dp"
    android:layout_height="128dp"
    android:layout_gravity="center"
    android:src="@drawable/flashlight_off"
    android:scaleType="fitCenter"
    android:background="@drawable/transparent_color"
    android:visibility="gone"
/>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
>

```

```

<TextView
    android:id="@+id/titleReadings"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:text="@string/titleIndicators"
    android:textSize="18sp"
    android:textColor="@color/white"
    android:gravity="center"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/currentReadingsTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:layout_marginRight="10dp"
        android:text="@string/currentIndicators"
        android:textSize="17sp"
        android:textColor="@color/white"
        android:layout_weight="1"/>
    <TextView
        android:id="@+id/previousReadingsTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:layout_marginLeft="10dp"
        android:text="@string/previousIndicators"
        android:textSize="17sp"
        android:textColor="@color/white"
        android:layout_weight="1"/>
</LinearLayout>

<LinearLayout
    android:id="@+id/llZone1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:orientation="horizontal">

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/TextInputLayoutZone1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        app:hintEnabled="false">

        <EditText
            android:id="@+id/editZone1"
            android:layout_width="match_parent"
            android:layout_height="50dp"
            android:background="@drawable/border"
            android:hint="@string/placeholderZone1"
            android:imeOptions="actionNext"
            android:inputType="number"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:singleLine="true"

```

```

        android:textColorHint="@color/placeholder"
    />
</com.google.android.material.textfield.TextInputLayout>

<TextView
    android:id="@+id/previousReadings1"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:gravity="center"
    android:layout_marginLeft="10dp"
    android:layout_weight="1"
    android:background="@drawable/border_previous"
    android:textColor="@color/sumy_gray"
    android:textSize="18sp"
    android:focusable="false"/>
</LinearLayout>

<LinearLayout
    android:id="@+id/llZone2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:orientation="horizontal">

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/TextInputLayoutZone2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        app:hintEnabled="false">

        <EditText
            android:id="@+id/editZone2"
            android:layout_width="fill_parent"
            android:layout_height="50dp"
            android:background="@drawable/border"
            android:hint="@string/placeholderZone2"
            android:imeOptions="actionNext"
            android:inputType="number"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:singleLine="true"
            android:textColorHint="@color/placeholder" />
    </com.google.android.material.textfield.TextInputLayout>

    <TextView
        android:id="@+id/previousReadings2"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:gravity="center"
        android:layout_marginLeft="10dp"
        android:layout_weight="1"
        android:background="@drawable/border_previous"
        android:textColor="@color/sumy_gray"
        android:textSize="18sp"
        android:focusable="false"/>
</LinearLayout>

<LinearLayout
    android:id="@+id/llZone3"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:orientation="horizontal">
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/TextInputLayoutZone3"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    app:hintEnabled="false">

    <EditText
        android:id="@+id/editZone3"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:background="@drawable/border"
        android:hint="@string/placeholderZone3"
        android:imeOptions="actionDone"
        android:inputType="number"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:singleLine="true"
        android:textColorHint="@color/placeholder"
    />
</com.google.android.material.textfield.TextInputLayout>
    <TextView
        android:id="@+id/previousReadings3"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:gravity="center"
        android:layout_marginLeft="10dp"
        android:layout_weight="1"
        android:background="@drawable/border_previous"
        android:textColor="@color/sumy_gray"
        android:textSize="18sp"
    />
</LinearLayout>
</LinearLayout>
<TextView
    android:id="@+id/data_check_label"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center"
    android:layout_marginTop="15dp"
    android:text="@string/data_check_label"
    android:textSize="18sp"
    android:textColor="@color/sumy_yellow"
/>

<CheckBox
    android:id="@+id/checkboxData"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:scaleX="1.3"
    android:scaleY="1.3"
    android:text="@string/yes"
    android:textColor="@color/white"
    android:theme="@style/MyCheckBox"

```

```

        />

        <Button
            android:id="@+id/btnTransmitMeterReading"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginBottom="20dp"
            android:layout_marginTop="15dip"
            android:paddingTop="10dp"
            android:paddingBottom="10dp"
            android:paddingLeft="20dp"
            android:paddingRight="20dp"
            android:background="@drawable/border_btn_login"
            android:onClick="OnTransmitMeterReadingButtonClick"
            android:text="@string/save"
            android:textAllCaps="false"
            android:textColor="@color/sumy_yellow"
            android:textSize="20sp"
        />

    </LinearLayout>

</ScrollView>

ServicesListActivity.java
package local.soe.itps02.soebilling;
import android.os.Bundle;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class ServicesListActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.services_add);
    }
}

services_add.xml
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="local.soe.itps02.soebilling.ActivityAccountAdd">

    <LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="15dp"
        android:focusable="true"
        android:focusableInTouchMode="true"
    >

        <TextView
            android:id="@+id/add_instruction"

```

```

android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/services_add_instruction"
android:textAllCaps="true"
android:gravity="center_horizontal"
android:textColor="@color/sumy_yellow"
android:textSize="18sp"/>

```

```
<Spinner
```

```

    android:id="@+id/spinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingBottom="10dp"
    android:paddingTop="10dp"
    android:paddingLeft="25dp"
    android:paddingRight="25dp"
    android:entries="@array/send_only_electronically_list"
    android:spinnerMode="dropdown"
    android:textAlignment="center"
    android:background="@drawable/test_border_spinner"

```

```
>
```

```
<LinearLayout
```

```

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="15dp"
    android:orientation="horizontal">

```

```
<TextView
```

```

    android:id="@+id/ls_label"
    android:layout_width="165dp"
    android:layout_height="wrap_content"
    android:layout_marginRight="5dp"
    android:text="@string/services_ls"
    android:textColor="@color/sumy_yellow"
    android:textSize="18sp"
    android:textStyle="bold" />

```

```
<TextView
```

```

    android:id="@+id/ls"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="162031351"
    android:textColor="@color/sumy_yellow"
    android:textSize="18sp"
    android:textStyle="bold" />

```

```
</LinearLayout>
```

```
<LinearLayout
```

```

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="15dp"
    android:orientation="horizontal">

```

```
<TextView
```

```

    android:id="@+id/name_label"
    android:layout_width="165dp"
    android:layout_height="wrap_content"
    android:layout_marginRight="5dp"
    android:text="@string/services_name"
    android:textColor="@color/sumy_yellow"

```

```

        android:textSize="18sp"
        android:textStyle="bold" />

<TextView
    android:id="@+id/name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Федірка Сергій"
    android:textColor="@color/sumy_yellow"
    android:textSize="18sp"
    android:textStyle="bold"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="15dp"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/address_label"
        android:layout_width="165dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="5dp"
        android:text="@string/services_address"
        android:textColor="@color/sumy_yellow"
        android:textSize="18sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/address"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="м.Суми, вул.Іллінська"
        android:textColor="@color/sumy_yellow"
        android:textSize="18sp"
        android:textStyle="bold"/>
</LinearLayout>
<EditText
    android:id="@+id/services_add_mobile_phone"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:paddingLeft="25dp"
    android:paddingRight="10dp"
    android:hint="@string/services_add_mobile_phone"
    android:singleLine="true"
    android:background="@drawable/border"
    android:inputType="phone"
    android:layout_marginTop="15dp"
    android:maxLength="10"
    />
<TextView
    android:id="@+id/services_add_mobile_phone_hint"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/services_add_mobile_phone_hint"
    android:textColor="@color/sumy_yellow"
    android:textSize="14sp"
    android:textStyle="bold" />

```

```

<EditText
    android:id="@+id/services_add_comment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingLeft="25dp"
    android:paddingRight="10dp"
    android:paddingBottom="10dp"
    android:paddingTop="10dp"
    android:hint="@string/services_add_comment"
    android:background="@drawable/border"
    android:layout_marginTop="15dp"
/>
<TextView
    android:id="@+id/services_add_comment_hint"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/services_add_comment_hint"
    android:textColor="@color/sumy_yellow"
    android:textSize="14sp"
    android:textStyle="bold" />

<Button
    android:id="@+id/btnAddServices"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="15dip"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:background="@drawable/border_btn_login"
    android:onClick="OnAddAccountButtonClick"
    android:text="@string/save"
    android:textAllCaps="false"
    android:textColor="@color/sumy_yellow"
    android:textSize="20sp"
/>

</LinearLayout>

</ScrollView>

```

#### HistoryListActivity.java

```

package local.soe.itps02.soebilling;
import android.os.Bundle;
import android.widget.TextView;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class HistoryListActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.history_of_indicators_list_content);
        TextView date = findViewById(R.id.date);
        date.setText("02.11.2020");
        TextView fkvt1 = findViewById(R.id.fkvt1);
        fkvt1.setText("114");
    }
}

```



```

        TextView fkvt2 = findViewById(R.id.fkvt2);
        fkvt2.setText(" ");
        TextView fkvt3 = findViewById(R.id.fkvt3);
        fkvt3.setText(" ");
    }
}

```

#### history\_of\_indicators\_list\_content.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:clickable="true"
    android:background="@drawable/border_main_list"
    android:orientation="vertical"
    android:layout_marginBottom="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="0dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:paddingTop="10dp"
    android:paddingBottom="10dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/label_date"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Дата:"
            android:textColor="@android:color/white"
            android:textSize="16sp"
            android:layout_marginRight="5dp"/>

        <TextView
            android:id="@+id/date"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="@color/sumy_yellow"
            android:text="01.11.2020"
            android:textColor="@color/sumy_gray"
            android:textSize="16sp"
            android:paddingRight="5dp"
            android:paddingLeft="5dp"
            />
    </LinearLayout>

    <TextView
        android:id="@+id/label_indicators"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/ls"
        android:textSize="16sp"
        android:textColor="@color/white"
        android:text="Показник лічильника:"
        />

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:gravity="center"
    android:layout_gravity="center">

    <LinearLayout
        android:id="@+id/LLzone1"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:orientation="vertical">

        <TextView
            android:id="@+id/zone1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="@string/zone1"
            android:textColor="@color/white"
            android:textSize="16sp" />

        <TextView
            android:id="@+id/fkvt1"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_gravity="center"
            android:layout_marginBottom="5dp"
            android:layout_marginTop="5dp"
            android:layout_weight="1"
            android:background="@drawable/border_of_indicators"
            android:gravity="center"
            android:textColor="@color/sumy_gray"
            android:textSize="18sp"
            android:text="12345"/>
    </LinearLayout>

    <LinearLayout
        android:id="@+id/LLzone2"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:orientation="vertical">

        <TextView
            android:id="@+id/zone2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="@string/zone2"
            android:textColor="@color/white"
            android:textSize="16sp" />

        <TextView
            android:id="@+id/fkvt2"

```

```

        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_gravity="center"
        android:layout_marginBottom="5dp"
        android:layout_marginTop="5dp"
        android:layout_weight="1"
        android:background="@drawable/border_of_indicators"
        android:gravity="center"
        android:textColor="@color/sumy_gray"
        android:textSize="18sp"
        android:text="123456" />
</LinearLayout>

<LinearLayout
    android:id="@+id/LLzone3"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:orientation="vertical">

    <TextView
        android:id="@+id/zone3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/zone3"
        android:textColor="@color/white"
        android:textSize="16sp" />

    <TextView
        android:id="@+id/fkvt3"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_gravity="center"
        android:layout_marginBottom="5dp"
        android:layout_marginTop="5dp"
        android:layout_weight="1"
        android:background="@drawable/border_of_indicators"
        android:gravity="center"
        android:textColor="@color/sumy_gray"
        android:textSize="18sp"
        android:text="123456789" />
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/label_type_indicators"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/white"
        android:textSize="16sp"
        android:text="Вид показу:"
        android:layout_marginRight="5dp"
        />
    <TextView

```

```

        android:id="@+id/type_indicators"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="@color/mainBlue"
        android:textSize="16sp"
        android:text="розрахунковий"
    />
    <!--або переданий-->
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/label_source"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/white"
        android:textSize="16sp"
        android:text="Джерело:"
        android:layout_marginRight="5dp"
    />
    <TextView
        android:id="@+id/source"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/mainBlue"
        android:textSize="16sp"
        android:text="Мій пахунок"
        android:background="@color/transparent"
        android:paddingRight="5dp"
        android:paddingLeft="5dp"
    />
    <!--або СМС або Коллцентр-->
</LinearLayout>

</LinearLayout>

```