

Державний вищий навчальний заклад  
“Прикарпатський національний університет імені Василя Стефаника”  
Кафедра інформаційних технологій

УДК 004

## ДИПЛОМНИЙ ПРОЕКТ

Тема: **Аплікація для криптування смс повідомлень на платформі  
Android**

Спеціальність 121 Інженерія програмного забезпечення  
код і назва спеціальності

### **ПОЯСНЮВАЛЬНА ЗАПИСКА ДП.ІІЗ-ІІ-15.ІЗ**

(позначення)

Рецензент

ст.викладач Пікуляк М.В.  
(посада) (підпис) (дата) (розшифровка підпису)

Студент

ІІЗ-41 Мрічко М.М.  
(шифр групи) (підпис) (дата) (розшифровка підпису)

Нормоконтролер

ст.викладач Пікуляк М.В.  
(посада) (підпис) (дата) (розшифровка підпису)

Керівник дипломного проекту

ст.викладач Іщеряков С.М.  
(посада) (підпис) (дата) (розшифровка підпису)

Допускається до захисту

Завідувач кафедри

доцент Козленко М.І.  
(посада) (підпис) (дата) (розшифровка підпису)

2020  
(рік)

Державний вищий навчальний заклад  
«Прикарпатський національний університет імені Василя Стефаника»  
Факультет математики та інформатики Кафедра інформаційних технологій  
Спеціальність 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Завідувач кафедри Козленко М.І.

\_\_\_\_\_” \_\_\_\_\_ 20\_\_р.

**ЗАВДАННЯ  
НА ВИКОНАННЯ ДИПЛОМНОГО ПРОЕКТУ**

Студенту \_\_\_\_\_ Мрічко Михайлу Михайловичу

(прізвище, ім'я, по батькові студента)

1. Тема проекту. Аплікація для криптування смс повідомлень на платформі Android

затверджена розпорядженням по факультету математики та інформатики від „25” жовтня 2019р. №7

2. Термін здачі студентом закінченого проекту 22 травня 2020 р

3. Вихідні дані до дипломного проекту: Android-додаток, Java, Gradle симетричне шифрування

4. Зміст пояснювальної записки (перелік питань, що їх належить опрацювати)

1. Аналіз предметної області

2. Моделювання та формування вимог до продукту

3. Програмна реалізація

4. Економічні розрахунки вартості програмного продукту

5. Перелік графічного матеріалу (з точним забезпеченням обов'язкових креслень) Титульний аркуш, Приклади месенджерів, Мова реалізації та її компоненти, алгоритм шифрування, Візуальне оформлення, приклад роботи програми

6. Дата видачі завдання

11.09.2019

Керівник

\_\_\_\_\_ (підпис)

Іщеряков С.М.

(розшифровка підпису)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

Мрічко М.М.

(розшифровка підпису)

## КАЛЕНДАРНИЙ ПЛАН

Номер і назва етапів дипломного проекту	Термін виконання етапів проекту	Примітка
Вивчення предметної області	02.12.2019	Виконав
Вивчення технологій розробки	15.02.2020	Виконав
Написання меседжеру	17.04.2020	Виконав
Розрахунок економічної вартості	11.05.2020	Виконав
Оформлення пояснювальної записки	18.05.2020	Виконав

Студент

Мрічко М.М.  
(підпис) (розшифровка підпису)

Керівник проекту

Іщеряков С.М.  
(підпис) (розшифровка підпису)

## РЕФЕРАТ

Розрахунково-пояснювальна записка: сторінок 57, рисунків 11, таблиць 1, додатків 4, посилань 10.

Об'єктом розробки є створення месенджера для телефону з використанням шифрування повідомлень.

Метою завдання було розробити месенджер, який шифрує повідомлення між двома користувачами.

Для досягнення мети було використано Java, Gradle.

Отриманим результатом є робочий месенджер, який відправляє шифровані повідомлення.

Виконано необхідні техніко-економічні розрахунки.

Галузевою сферою застосування є використання як інтернет ресурсу.

## **ABSTRACT**

Calculation and explanatory note: pages 57, figures 11, tables 1, appendices 4, links 10.

The object of development is to create a messenger for the phone using message encryption.

The goal was to develop a messenger that encrypts messages between two users.

Java and Gradle were used to achieve this goal.

The result is a working messenger that sends encrypted messages.

The necessary technical and economic calculations have been performed.

Sectoral scope is the use as an Internet resource.

## ЗМІСТ

ВСТУП.....	7
1 ОГЛЯД ІСНУЮЧИХ ВЕБ ДОДАТКІВ ДЛЯ ЛИСТУВАННЯ ШИФРОВАНИМИ ПОВІДОМЛЕННЯМИ .....	9
1.1 TextSecure .....	9
1.2 Handcent SMS .....	11
1.3 8sms .....	13
1.4 Постановка завдання.....	15
2 ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ПРОЕКТУ .....	16
2.1 Мова програмування Java .....	16
2.2 Gradle .....	19
2.3 Android Studio .....	21
2.4 Шаблон проектування MVC .....	24
2.5 Система контролю версій Git.....	26
2.6 Платформа для тестування роботи проекту Heroku.....	27
2.7 Математична модель шифрування.....	28
3 BACK-END ЧАСТИНА ПРОЕКТУ .....	31
3.1 Модель бази даних.....	31
3.2 Опис функціональної частини .....	32
3.2 Методи тестування.....	45
4 ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ ПРОЕКТУ .....	45
4.1 Економічна задача.....	45
4.2 Розрахунок витрат на розробку структури проекту .....	46
4.3 Розрахунок витрат на експлуатацію .....	49
4.4 Економічний ефект .....	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	57
Додатки.....	58

					<b>ДП.ІПЗ – 15.ПЗ</b>					
					Аплікація для криптування смс повідомлень на платформі Android					
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>						<i>Лім.</i>	<i>Ар.</i>	<i>Акрушів</i>
<i>Розроб.</i>		<i>Мрічко М.М</i>								
<i>Перевір.</i>		<i>Іщерков С.М</i>							6	64
<i>Реценз.</i>		<i>Пікуляк М.В</i>						<b>ПНУ ІПЗ-41</b>		
<i>Н. контр.</i>		<i>Пікуляк М.В</i>								
<i>Затверд.</i>		<i>Козленко М.І.</i>								

## ВСТУП

Інформаційні технології стали дуже важливою частиною процесу використання інформаційних ресурсів сучасного суспільства. До сьогоднішнього дня вони пройшли досить багато еволюційних етапів, зміна яких визначалася основним чинником розвитком науково-технічного прогресу, і є появою новітніх інформаційних засобів обробки, пошуку та різні типи інформації. У сучасному суспільстві головним засобом поширення інформації являється інтернет. В щоденній мові фраза Інтернет дедалі частіше використовується в значенні Всесвітнього павутиння і існуючій в ній інформації, а не у розумінні основи головної фізичної мережі. Також використовують науковий термін «Всесвітня мережа», «Глобальна мережа» чи заміняють вираз одним словом – «Міжмережжя», «Мережа», «Нет», «Тенета», «Браузер», «Інтернети». Все більш частим Інтернет вживається і з малої букви, що можливо пояснити паралелями з наукової термінології «радіо», «телебачення», які пишуть з малої.

INTERNET надає такі види послуг:

- e-mail – електронна ящик;
- список останніх новин;
- можливість поштової розсилки;
- доступ до віддалених серверів зберігання інформації;
- сеанси двохстороннього зв'язку з іншими комп'ютерами, , які мають свої адреса і підключені до INTERNET;
- пошук необхідної інформації в базах даних у швидкому режимі;
- можливість комунікувати з іншими юзерами шляхом використання онлайн-сервісу Internet Relay Chart, Viber;
- доступ до програмної системи World Wide Web

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		7

З додаткових послуг можливо представити наступні:

- швидка передача мультимедіа;
- відеоконференції;
- RadioInternet;
- аудіо конференційний зв'язок;
- безпроводне з'єднання.

В даний час WEB використовується у всіх сферах діяльності, починаючи від розваг (фільми, відеоігри, музика і т.д.) і закінчуючи науковою інформацією (дисертації, монографії, нові відкриття в галузі науки і т.д.). Глобальна мережа використовується для навчання, спілкування, розваг та розміщення різного роду інформації(наукових статей, реклами, вакансій, новин політики та спорту). Також інтернет дозволив значно полегшити передачу та поширення інформації між людьми, компаніями та країнами за допомогою різних сервісів.

Мета даного дипломного проекту полягає у створенні меседжеру, який шифруватиме повідомлення між двома користувачами використовуючи один із звичаних методів шифрування – шифрування симетричним ключем персональних повідомлень.

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		8



# 1 ОГЛЯД ІСНУЮЧИХ ВЕБ ДОДАТКІВ ДЛЯ ЛИСТУВАННЯ ШИФРОВАНИМИ ПОВІДОМЛЕННЯМИ

## 1.1 TextSecure

TextSecure – месенджер для Android з відкритим вихідним кодом, який використовує власний відкритий протокол з end-to-end шифруванням. TextSecure дозволяє безпечно передавати миттєві повідомлення, файли і мультимедіа іншим користувачами TextSecure і Signal. У програмі є можливість перевірити особу співрозмовника шляхом порівняння відбитків ключів. TextSecure також дозволяє відправляти звичайні незашифровані SMS і MMS і може працювати як заміна стандартному SMS-додатку Android. Локальну базу повідомлень можна зашифрувати пральний фразу.

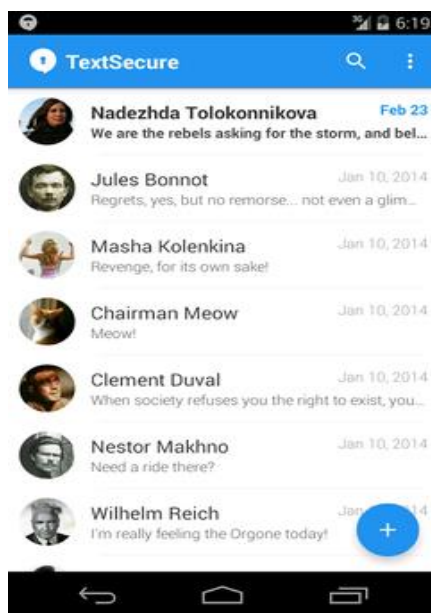


Рисунок 1.1 — TextSecure

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		9

TextSecure сумісний з Signal для iOS. Обидві програми розроблені проектом Open Whisper Systems і опубліковані під ліцензією GPLv3.

TextSecure дозволяє відправляти зашифровані текстові та аудіо-повідомлення, фотографії, відео, контакти і смайли через інтернет-з'єднання (наприклад, Wi-Fi, 3G або 4G) іншим користувачам TextSecure і Signal, а також обмінюватися звичайними незашифрованими SMS і MMS з людьми, які не використовують TextSecure або Signal

Повідомлення, відправлені за допомогою TextSecure або Signal, автоматично зашифровуються на пристрої відправника і розшифровуються на пристрої одержувача, таким чином, повідомлення може прочитати тільки той, кому воно адресоване.

Ключі, що використовуються для шифрування, зберігаються тільки на пристрої користувача і в разі, якщо включення до додатка секретною фразою, данні захищаються додатковим шаром шифрування. В інтерфейсі додатка зашифровані повідомлення відзначені значком замка.

Недоліками даного програмного продукту виступає його ціна, яка складає 10\$ на місяць, що є досить дорогим для додатку такого типу. Також до недоліків можна віднести неможливість видалити за допомогою віддаленого доступу свій персональний ключ безпеки або змінити його, якщо користувач загубив свій телефон. Також додаток не має можливості зробити копію свого ключа що є недоречним, оскільки при зміні пристрою або перевстановленні програми необхідно знову «передавати» свій персональний ключ безпеки, що завдає проблем при спілкуванні і викликає дискомфорт.

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		10

## 1.2 Handcent SMS

Для розробки цього додатку використовували Kotlin. Функціонал даного додатку є невеликим, але він виконує свої основні функції. Шифрування повідомлень можна як включати так і виключати по бажанню користувача. Шифрування в даному додатку використовується синхронне. Даний тип шифрування є одним із найпростіших:

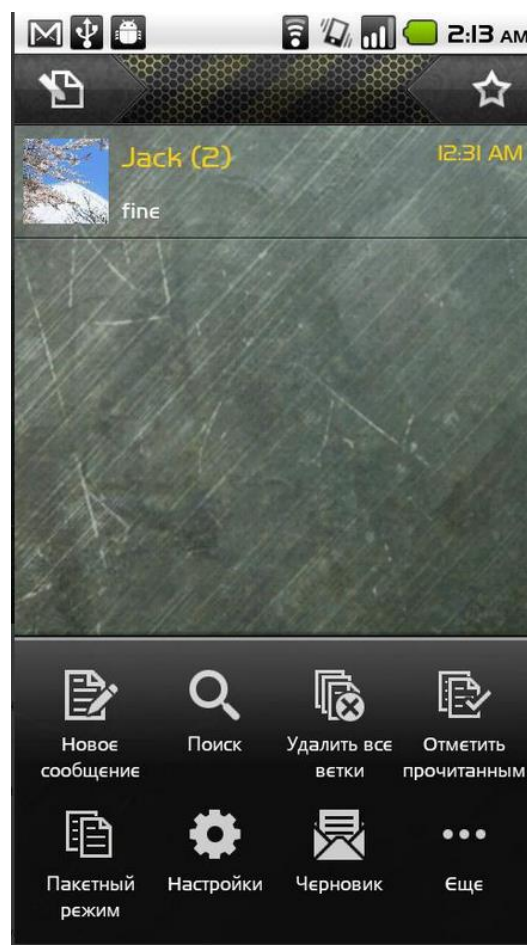


Рисунок 1.2 — Handcent SMS

#### Функціонал:

- Відправлення повідомлень (нешифрованих)
- Відправлення повідомлень (Шифрованих)
- Зміна зовнішнього вигляду
- Створення групи

#### Недоліки:

- Відсутність іншого типу шифрування
- Частий «виліт» програми через помилку ключів

Kotlin – статично типізована мова програмування, що працює поверх JVM і розроблений компанією JetBrains. Kotlin можна компілювати в JavaScript або в Native для запуску на iOS платформі.

Основними перевагами мови Kotlin є її бурхливий розвиток; легкий перехід з Java на Kotlin – потрібно просто встановити плагін Kotlin та їх сумісність; наявність extension functions для розробки чистих API; наявність null в системі типів; Kotlin лаконічна, що зменшує кількість помилок.

Проте існують і певні недоліки: Kotlin має меншу швидкість компілювання; середовища розробки (наприклад, Android Studio) працює повільніше за Kotlin. Продуктивність є одним з найважливіших аспектів, що необхідно враховувати при виборі мови програмування.

Структура байт-коду Kotlin майже ідентична структурі Java, що робить додатки так само швидкими. Однак, Kotlin може підтримувати вбудовані функції, які дозволяють коду, який містить лямбда вирази, працювати швидше, ніж той, який написано на Java. Клас, що складається з 50 рядків коду при використанні Java, може бути написаний тільки одним рядком Kotlin.

Наведені вище переваги та недоліки демонструють ефективність використання Kotlin, яка вимагає меншої кількості ліній коду, що робить її використання безпечнішим в порівнянні із Java. Завдяки цьому покращується

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		12

читабельність коду, ефективність його обробки, знижується ймовірність виникнення помилок при написанні коду ще на етапі компіляції

### 1.3 8sms

Для розробки додатку 8sms використано мову програмування Python. Даний додаток містить такий функціонал:

- перегляд повідомлень;
- відправка повідомлень;
- відправка шифрованих повідомлень;

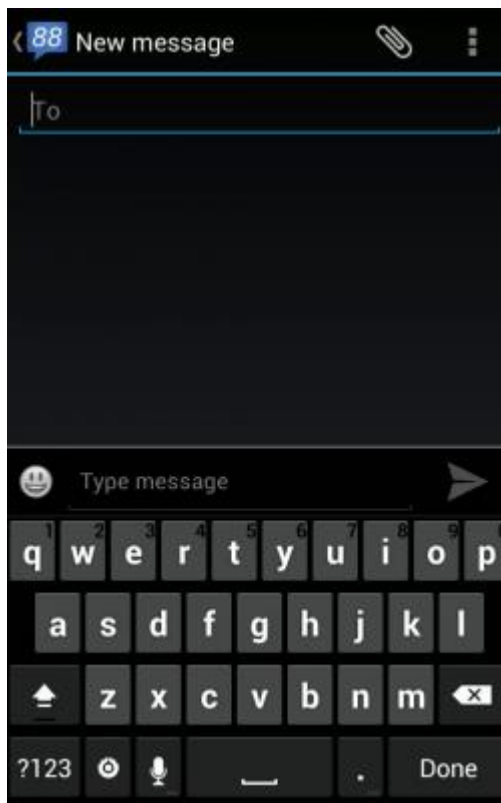


Рисунок 1.3 — 8sms

Python (найчастіше читається або "Пайтон", запозичено з назви британського телешоу Монті Пайтон) – інтерпретована об'єктно-орієнтована мова програмування високого рівня з строгою динамічною типізацією коду. Створена в далекому 1990 році Гвідо ван Россумом. Структури такого високого рівня включно з динамічною семантикою та динамічним зв'язуванням роблять її досить хорошим варіантом для швидкісної розробки програмного забезпечення, а також як один із засобів поєднання уже наявних компонентів. Python підтримує модульну структуру, що сприяє модульній розробці та з'єднюванню існуючих компонентів. Інтерпретатор Python та стандартні бібліотеки доступні або у виконанні як у вихідній формі на всіх екземплярів. В мові високого рівня програмування Python підтримується декілька базових концепцій програмування, зокрема: об'єктно - орієнтована, процедурна, функціональна та аспектно - орієнтована.

#### Переваги:

- чистий синтаксис (для виділення частин коду слід використовувати пропуски);
- переносність (що є «рідним» великій кількості інтерпретованих мов);
- стандартний софт має велику кількість корисних модулів (включно з модулем для вивчення графічного інтерфейсу);
- можливість формування Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих завдань);
- стандартний дистрибутив пайтон є простим, хоч і достатньо потужним середовищем розробки IDLE і створених на мові Python; – зручний для розв'язання математичних проблем (має способи виконання роботи з комплексною кількістю, може оперувати з цілими даними довільної величини).

У діалоговому режимі може використовуватися або здатний виконати функцію автозапуску виконуваного файлу); – відкритий шифр (можливість редагувати нього іншими користувачами).

					ДП. ІПЗ–15.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підп.	Дата		

Недоліки:

- низька швидкодія;
- відсутність статичної типізації;
- неможливість модифікації вбудованих класів;
- глобальне блокування інтерпретатора;

#### 1.4 Постановка завдання

Було поставлено завдання розробити Додаток на Андроїд для відправки і отримання шифрованих повідомлень. Він повинен надавати такі можливості:

- відправлення та отримання повідомлень;
- відправлення та отримання повідомлень (шифрованих);
- перегляд існуючих повідомлень;
- розшифровка наявних повідомлень;
- відправка сповіщень про надходження повідомлень;
- відправлення сповіщень про невідправлені повідомлення.

					ДП. ІПЗ–15.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підп.	Дата		

## 2 ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ПРОЕКТУ

### 2.1 Мова програмування Java

Java – це потужна універсальна мова програмування, призначена для вирішення різноманітних завдань.

Особливістю даної мови є незалежність від платформи – для всіх операційних систем існують віртуальні Java-машини – інтерпретатори, які виконують ваше java-додаток по можливості одним і тим же чином. Зрозуміло, машини для різних систем мають різний набір можливостей – у desktop-систем один набір, у серверних систем інший, у мобільних – третій, у вбудованих пристроїв - четвертий. Але в основному відмінності стосуються призначеного для користувача інтерфейсу, принципів роботи з пам'яттю і процесами. Синтаксис Java заснований на синтаксисі мови C ++, але можливості набагато більше. До них можна, наприклад, віднести достатню ізольованість виконуваної програми від основної операційної системи: java-машина відстежує дії, що виконуються програмою, і при спробі несанкціонованих операцій негайно перериває виконання.

За рахунок того, що java-машині доводиться перетворювати байт-код вашої програми в процесі її виконання в інструкції операційній системі, додаток працює повільніше, ніж якби він був відразу відкомпільоване. Але цей недолік перекривається:

- 1) Універсальністю коду для всіх платформ
- 2) Достатньо ефективною реалізацією java-машин для пристроїв, де важлива швидкість обробки (мобільні та вбудовані пристрої)
- 3) Набором бібліотек, орієнтованих на роботу конкретних платформ
- 4) Набором вбудованих інструментів, значно спрощують процес розробки ПО.

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		16



Програми на мові програмування Java транспонуються в двійковий байт-код Java, який виконується Java Virtual Machines (JVM) - програмою, обробний байт-код і процесу відлагодження як інтерпретатор.

Перевагою такого способу виконання програмного продукту подібних програм є повна самостійна реалізація двійкового байт-коду від ОС і устаткування, що дає можливість реалізувати Java-додатки на любых приладах, для якого є створені віртуальні машини. Також вагомою особливістю технології Java є “резинова ” система, яка відповідає за безпеку, в рамках якої створення програм всесторонньо контролюється Virtual Mashines. Любі операції, які виходять за рамки встановлені привілегій програми (наприклад, спроба несанкціонованого доступу до файлів або спроба коннекту до іншого ПК), викликають негайну зупинку.

До деяких недоліків даної концепції машин для віртуального відтворення відносять зниження продуктивності. Ряд удосконалень який має більшу швидкодію відлагодження програм, створених на мові Java:

- Використання технології генерації двійкового байт-коду в код, зрозумілий для машини під час самого створення проекту (JIT-технологія) з можливістю збереження усіх можливих варіантів бібліотек(класів) в машинному кодї,
  - Обширне використання частин орієнтованого коду (native-код) в типічних бібліотеках,
  - апаратні засоби, що відповідають за швидкодію обробки двійкового байт коду (наприклад, технологія Jazelle).

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		17

За даними сайту [shootout.alioth.debian.org](http://shootout.alioth.debian.org), для 7 завдань час потрібний на створення проекту і його запуск на Java становить в середньому в 1.5-2 рази більше, ніж для C / C ++, в певних моментах мова Java має більшу швидкодію, а в інших моментах приблизно в сім разів повільніше . З другого боку, для більшої кількості програм мова Java споживає велику кількість пам'яті ( в деяких випадках вона перевищує 10-30) на відміну від мов програмування C \ C ++. Також після проведення досліджень компанією Google було виявлено що Java має істотно нижчу продуктивність при використанні ідентичних ресурсів тестових прикладів на відміну від програм написаних на C ++, що істотно впливає на необхідність мати «краще залізо» для виконання ідентичних програм, написаних на Java

Ідеї, які пробували закласти в концептуальну модель і різні способи її реалізації в середовищі віртуальних машин Java дало поштовх різним типам ентузіастів на збільшення і написання нових мов, які можна було б виконувати на віртуальних машинах. Такі ідеї також знайшли вираз специфікації загальномовній структурі CLI, яку ставить у фундамент .NET компанія Microsoft.

Усередині Java існує кілька основних сімейств технологій:

Java SE – Java Standard Edition, основне видання Java, містить компілятори, API, Java Runtime Environment; підходить для створення призначених для користувача додатків, в першу чергу - для настільних систем.

Java EE – Java Enterprise Edition, являє собою набір специфікацій для створення програмного забезпечення рівня підприємства. У 2017-му проект Java EE був переданий Eclipse Foundation , після чого був перейменований в Jakarta EE . Модулі Java EE видалені з Java SE, починаючи з 11-ї версії .

Java ME – Java Micro Edition, створена для використання в пристроях, обмежених по обчислювальній потужності, наприклад, в мобільних телефонах, КПК, вбудованих системах;

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		18

Java Card – технологія надає безпечне середовище для додатків, що працюють на смарт-картах та інших пристроях з дуже обмеженим обсягом пам'яті і можливостями обробки.

Мова Java активно використовується для створення мобільних додатків під операційну систему Android. При цьому програми компілюються в нестандартний байт-код, для використання їх віртуальною машиною Dalvik (починаючи з Android 5.0 Lollipop віртуальна машина замінена на ART). Для такої компіляції використовується додатковий інструмент, а саме Android SDK (Software Development Kit), розроблений компанією Google.

Розробку додатків можна вести в середовищі Android Studio, NetBeans, в середовищі Eclipse, використовуючи при цьому плагін Android Development Tools (ADT), або в IntelliJ IDEA. Версія JDK при цьому повинна бути 5.0 або вище.

8 грудня 2014 року Android Studio визнана компанією Google офіційної середовищем розробки під ОС Android.

## 2.2 Gradle

Gradle – система автоматичного будування з інсуючих модулів, концепції якої працює на принципах Apache Ant і Apache Maven, але надає DSL на мовах Groovy і Kotlin заміняючи типічні XML форми подання конфігурації написаного проекту.

На відміну від Apache Maven, заснованого на концепції життєвого циклу проекту, і Apache Ant, в якому порядок виконання завдань (targets) визначається відносинами залежності (depends-on), Gradle використовує спрямований ациклічний граф для визначення порядку виконання завдань.

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		19

Gradle був розроблений для розширюваних багатопроєктних збірок, і підтримує інкрементальні збірки, визначаючи, які компоненти дерева збірки не змінилися і які завдання, залежні від цих частин, не вимагають перезапуску.

Основні плагіни призначені для розробки і розгортання Java, Groovy і Scala додатків, але готуються плагіни і для інших мов програмування.

Java плагін емулює життєві цикли Maven, у вигляді завдань в наведеному ациклічному графі залежностей для входів і виходів кожного завдання. У цьому прикладі виконання завдання build залежить від результату виконання завдань check і assemble. Також завдання check залежить від test, а assemble від jar.

Для додатків які складаються з модулів зручно використовувати мультипроєктний підхід Gradle.

Кожен модуль розташований в своїй директорії. Кожен модуль може включати дочірні, які будуть розташовані у вкладених директоріях. Опис мультипроєктної структури виконується в settings.gradle в кореневій директорії проєкту

У кореневій директорії проєкту є build.gradle який може налаштовувати себе і дочірні модулі.

Кожен дочірній модуль може мати такі ж два файли для настройки себе і своїх дочірніх модулів.

Такий підхід дозволяє зручно запускати завдання для конкретних модулів або всього проєкту в цілому.

Структура проєкту в Android Studio:

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докum.	Підп.	Дата		20

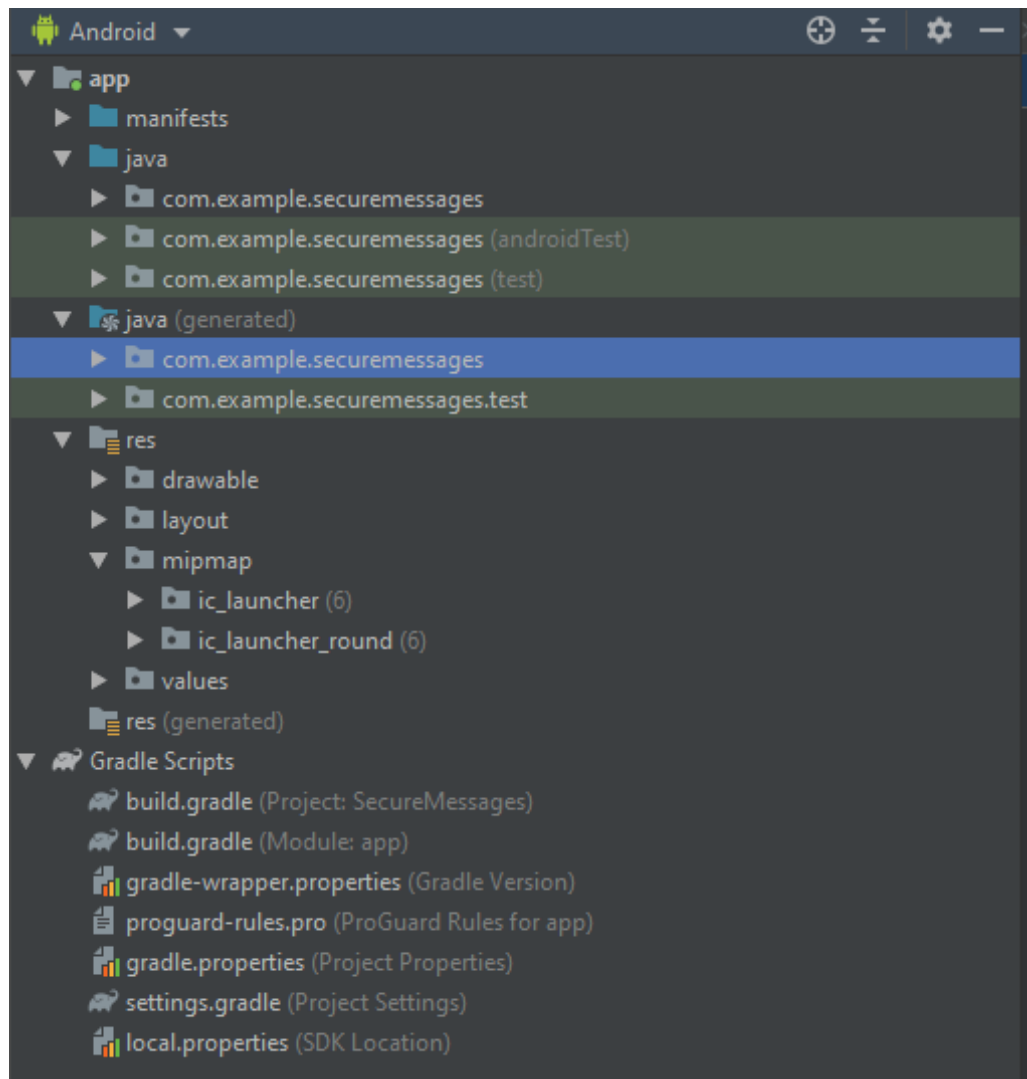


Рисунок 2.1 — Структура проєкту

## 2.3 Android Studio

Android Studio Прийшла прийшла замінити ADT для програми Eclipse. Середовище створене на базі вихідного коду продукту IntelliJ IDEA Community Edition, що розвиває компанією JetBrains. Android Studio розвивається в рамках Відкритої моделі розробки та поширюється під ліцензією Apache 2.0.

Бінарні складання підготовлені для Linux (для тестування Використання Ubuntu), Mac OS X и Windows. Середовище пропонує інструменти для розробки програм НЕ тільки для телефонів з Android, но і гаджетів на базі платформи на базі Android Wear, телевізорів (Android TV), окулярів Google

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		21

Glass и автомобільних інформаційно-розважальних систем (Android Auto). Для застосунків, спочатку розроблення з Використання Eclipse и ADT Plugin, підготовленій опції для автоматичного імпорту\експорту в уже створений проект Android Studio.

Середовище розробки адаптованості для виконання типових завдання, що вирішуються в процесі розробки програм на базі Android. Також сюди включені інструменти для тестування по спрощеній програмі на сумісність з різними версіями. Платформи та інструменти для проектування застосунків, що Працюють на прибудований з екранам різної роздільної здатності (DPI), яка відрізняється у планшетах, смартфонах, ноутбуках, годинниках, окулярів. Крім можливий, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька Додатковий новий функціонал для створення уніфікованих систем складання, тестування застосунків засновано на складальному інструментарії Gradle і підтримуюча використання ЗАСОБІВ безперервної інтеграції.

До складу також включені призупинені під особливості платформи Android розширені інструменти рефакторингу, перевірки сумісності з минулим випуском, виявлення проблем з продуктивністю, моніторингу використання оперативної пам'яті та оцінки ефективного її використання. У редакторі доданий режим швидкого внесення правок. Система давала підказки щодо аналізу та знаходження багів розширена підтримкою Android API. Інтегрована підтримка оптимізатора коду ProGuard. Вбудовані засоби генерації цифрових підписів. Надано інтерфейс для управління перекладом на інші мови. Особливості

Деякі особливості будуть пізніше реалізовані для юзерів так як програмне забезпечення іще на стадії розробки. Наразі, пропонується такий функціонал:

- 1) Живі макети (layout): редагувальник WYSIWYG – живе кодування - Подання (rendering) програми в реальному часі.
- 2) Консоль розробника: підказки по оптимізації, допомога по перекладу, відстеження можливих напрямків – метрики Google аналітики.

					ДП. ІПЗ-15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		22

3) Резерви бета релізів та покрокові релізи.  
4) Базування на Gradle.  
5) Android-орієнтований рефакторинг та Швидкі виправлення.  
6) Lint утіліті для охоплення продуктивності, юзабіліті, сумісності версій та інших проблем.

7) Використання можливостей ProGuard та підписів до програм.

8) Шаблони для створення поширення Android дизайнів та компонентів.

9) Багатий редактор макетів (layouts), що дозволяє користувачу перетягнути і покласти (drag-and-drop) компоненти користувацького інтерфейсу

Основними компонентами застосунків Android Studio є модель (model), вид (view) і контролер (controller).

Модель дає іншій частині класів об'єктно-орієнтоване представлення даних. Об'єкти моделі виконують загрузку і збереження даних в динамічній базі даних проекту.

Завдяки можливостям, які надає динамічна типізація в мові програмування Java, девелоперу достаньо наслідувати клас model від батьківського класу ActiveRecord::Base, Java автоматично зробить пов'язання цих класів моделі з існуючими у базі даних таблиць і надасть атрибути об'єктів класів

Вид (або View) згенерує інтерфейс для юзера для коректного відображення даних від класу-контроллера. Вид також передає запити користувача на маніпуляцію даними в контролер.

Контролер – основний компонент, що відповідає за взаємодію з користувачем. Контролер прочитує необхідні дані з моделі і готує їх для відображення, а також зберігає отримані від відображення дані в моделі.

## 2.4 Шаблон проектування MVC

Модель–вигляд-контро́лер (англ. Model-view-controller, MVC) – шаблон архітектури, який використовується для моделювання, проекту і його подальшої прозорки.

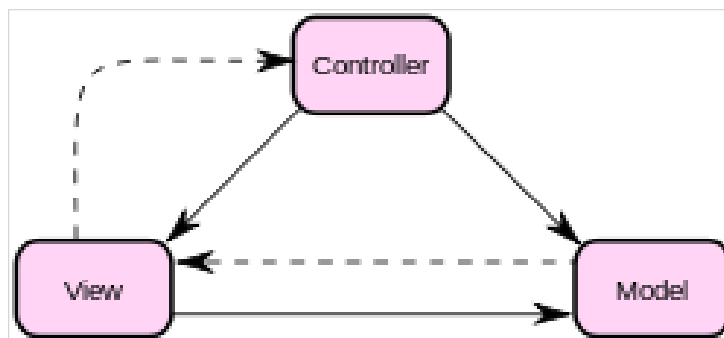


Рисунок 2.2 – Діаграма взаємодії між компонентами шаблону

Даний шаблон передбачає проектування моделі на три рівноцінні компоненти, які є зв'язані між собою: представлення, модель і вигляд та модуля керування.

Застосовується дана архітектура для чіткої розділеності даних від інтерфейсу користувача так, щоб коректування представлення якнайменше впливало на зміни в моделі даних і була можливість без зміни інтерфейсу користувача

Мета шаблону – гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми.

Основною перевагою цього шаблону при написанні великих проектів є чітка розподіленість структури проекту, що робить його ефективнішим за підібні або гібридні шаблони проектування.



У межах проектування шаблону модель–вигляд–контролер (MVC) проект розділяють на деякі етапи, але взаємопов'язані частини з чітким розділення функціональних компонентів. Модель (Model) – зберігання даних і коректного відображення. Вигляд (View) – представлення інтерфейсу для користувача. Контролер (Controller) відповідальний за керування компонентами і відповідної реакції на дію користувача (рух курсора, введення даних чи дії при натисканні певної кнопки) і передає дані у модель.

Модель є основоположною частиною шаблону MVC і відповідає за коректний вивід застосунків, незалежну від інтерфейсу користувача. Модель відповідає за керування даними, логікою програми та правилами аплікації.

Вигляд може бути довільним представленням інформації, отримуване на виході, наприклад текст чи картинку. Водночас можуть існувати декілька представлень однієї і тієї ж самої категорії інформації: інформація про новину та текст новини.

Контролер отримує данні і відправляє їх на команду для відображення іншого вигляду чи виконання іншої функції моделі.

Модель інкапсулює ядро даних і головні функції виконання і не має залежності від введених чи виведених даних.

Вигляд пов'язаний з декількома областями, наприклад, вивід різних таблиць з даними і формами до них.

До функцій контролера входить реагування на зміни подій, викликаних результатом дії користувача. Також контролер дає змогу створення структури коду шляхом групування однотипних дій в один клас

Події які пройшли реєстрацію в класі транслюються в запити, які спрямовуються компонентам моделі або об'єкту класу, які відповідають за відображення потрібних даних.

Відокремлення моделі, як вигляду даних надає можливість використовувати потрібні компоненти незалежно один від другого для

					ДП. ІПЗ–15.ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підп.	Дата		

перегляду інформації В такий спосіб через використання контролера може вносити зміни до моделі даних, і данні зміни подані декількома компонентами візуально відображеними, буде автоматично коректуватись відповідно до внесених користувачем змін.

## 2.5 Система контролю версій Git

Git – система спільного контролю версій над файлами для колективної роботи. Даний проект був створений американським програмістом Лінусом Торвальдсом для керування створення ядра Linux, а сьогодні підтримується Джуніо Хамано (англ. Junio C. Hamano). На сьогоднішній день Git рахується однією із самих надійних, ефективних та високопродуктивних систем для колективного створення проектів і верування версіями що дає можливість гнучкої розробки нелінійних програм які базуються на можливих розгалуженнях і при необхідності – злиття гілок. Прикладами проектів, що використовують Git, є ядро Linux, Android, LibreOffice, Cairo, GNU Core Utilities, Mesa 3D, Wine, багато проектів з X.org, XMMS2, GStreamer, Debian DragonFly BSD, Perl, Eclipse, GNOME, KDE, Qt, Ruby on Rails, PostgreSQL, VideoLAN, PHP, One Laptop Per Child (OLPC), АБІС Koha, GNU LilyPond та ELinks і деякі дистрибутиви GNU/Linux .

Програма є вільною і випущена під ліцензією GNU GPL версії 2.

Сама система проектувалась як певний набір програм для розробки ПЗ, спеціально створених з урахуванням її використання у скриптах, що дозволяло зручно будувати системи вузького напрямлення і управління версіями на на базі системи Git.

Інтерфейси, наприклад Cogito є наявним прикладом розробки фронт-енд частини до реалізації платформи Git, А SitGit використовує данну платформу для керування колекціями.

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		26

Для данної системи було розроблено цілий ряд користувацьких інтерфейсів – gitk а також git-gui, які знаходяться в «пакеті» набору програм Git.

Віддалене корсиування і доступ до репозиторіїв забезпечується за допомогою технології git-repo, ключами шифрування SSL або протоколу HTTP сервера. TCP – сервіс git-daemon є складовою дистрибутива Git і йде разм із ключем шифрування SSH, який рахують достатьо надійним, через що він є настільки поширеним.

## 2.6 Платформа для тестування роботи проекту Heroku

Heroku – хмарна PaaS-платформа, яка надає можливість запуску програм на різних мовах. Компанією Heroku володіє Salesforce.com. Heroku є однією із самих перших платформ хмарного середовища, з'явилась в червні 2007 року і спочатку могла підтримувати єдину мову програмування – Ruby, але на сьогоднішній день список підтримуваних мов також включає в себе Java, Node.js, Scala, Clojure, Python і PHP. На серверах Heroku використовуються операційні системи Debian або Ubuntu (яка також заснована на Debian).

Джеймс Лінденбаум, Адам Вігінс та Оріон Генрі є головними директорами ( по суміництву засновниками) «Heroku» 2007 року як підтримку проектів, створених на Rack (англ. web server interface "Rack").

У липні 2011 року Heroku почала підтримувати Node.js і Clojure.

15 вересня 2011 року Heroku та Facebook продемонстрували світу функцію «Heroku для Facebook». Heroku підтримує системи управління базами, такими як: CouchDB, Membase, MongoDB і Redis, крім основної — PostgreSQL.

Програми, що виконуються на платформі на Heroku, використовують також DNS-сервер Heroku (звичайним аплікаціям надається доменне ім'я по типу: «ім'я\_додатку.herokuapp.com»). Для любої створеної програми платформа виділяє декілька незалежних процесів які називаються «dynos». Дані процеси

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		27

поділяються по своїй рідній віртуальній мережі, яка компанується із декількох серверів. Heroku також підтримує систему контролю версій - Git.

## 2.7 Математична модель шифрування

Симетричні криптосистеми шифрування – спосіб кодування даних, в якому для кодування та розкодування використовуються однаковий криптографічний ключ. Ключ шифрування не повинен розголошуватись стороннім особам і має триматись в секреті, також він повинен бути обраний іще до початку відправлення першої частини даних. Процес розшифрування даного алгоритму базується на повторному складанні шифрованої послідовності з тим самим набором шифру.

Описаний метод не є надійним, оскільки отримання хоча б частини повідомлення надає змогу людині з набором потрібних навичок без проблем розшифрувати все інше повідомлення. Більшість симетричних шифрів базуються на створенні алгоритмів для більш сильного шифрування, використовуючи декількох комбінацій, повторного шифрування уже зашифрованого повідомлення, використовуючи при наступному проході той самий ключ шифрування .

Кількість «ключів проходів» для всіх виконаних проходів називається розкладом ключів.

Основними параметрами будь-якого алгоритму шифрування має бути:

- стійкість;
- довжина ключа;
- кількість циклічних проходів шифрування;
- довжина частини, яка шифрується;

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		28

- складність алгоритму шифрування;
- складність розшифровки частини даних.

До переваг симетричної системи можна віднести:

- порівняно велику швидкість шифрування даних і їх об'єм;
- проста реалізація (через використання коефіцієнтів);
- невелика довжина ключа;

Для зображення систем шифрування використовують як математичну модель так і структурну схему.



Рисунок 2.3 – Математична модель та структурна схема симетричного алгоритму шифрування

Саму процедуру шифрування динних та розшифрування можна представити формулою:

$$c = E_{k_e}(p);$$

$$p = D_{k_d}(c)$$

де  $p$  і  $c$  відповідно – відкритий і зашифрований тексти,  $k_e$  і  $k_d$  – ключі шифрування і дешифрування;  $E_{k_e}$ ,  $D_{k_d}$  – функції шифрування з ключем  $k_e$  і дешифрування з ключем  $k_d$  відповідно, причому для будьякого відкритого тексту справедливе співвідношення:

$$D_{k_d}(E_{k_e}(p)) = p$$

Шифрування і дешифрування повідомлень відбувається на вході і виході каналу передачі даних.

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		30

## 3 BACK-END ЧАСТИНА ПРОЕКТУ

### 3.1 Модель бази даних

Після вивчення предметної області було розроблено базу даних для додатку. Після чого була сформована структура таблиці:

- address – для збереження номеру телефону;
- date– для збереження дати;
- body – для збереження самого повідомлення;

Для підключення бази даних використано бібліотеку `android.database.sqlite.SQLiteDatabase` яка дозволяє згенерувати конфігураційний файл бази-даних `app.db`.

Конфігурація бази даних:

development:

clients:

default:

database: academy\_alpha\_development

hosts:

- localhost:27017

options:

options:

test:

clients:

default:

database: academy\_alpha\_test

hosts:

- localhost:27017

options:

					ДП. ІПЗ–15.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підп.	Дата		

```

read:
mode: :primary
max_pool_size: 1
production:
clients:
default:
uri:
'sqlite://heroku_7sbj3ftz:aeqrqhr7pdfa178ep41ohcncg@ds157057.mlab.com:57057/
heroku_7sbj3ftz'
options:
consistency: :strong
max_retries: 30
retry_interval: 1
timeout: 15
safe: true
refresh_interval: 10

```

### 3.2 Опис функціональної частини

Для того щоб підключити усі необхідні файли в додаток використовується build.gradle. Щоб завантажити вказані у build.gradle бібліотеки необхідно підключити усі файли

Код build.gradle:

```

buildscript {
    repositories {
        google ()
        jcenter ()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.5.3'
        // NOTE: Do not place your application dependencies here; they belong

```



```

        // in the individual module build.gradle files
    }
}
allprojects {
    repositories {
        google ()
        jcenter ()
    }
}

task clean (type: Delete) {
    delete rootProject.buildDir
}

```

Для перевірки підключення використовується функція ShowDialog (код файлу наведено в додатку А).

Код showDialog:

```

public void showDialog (String [] text) {
    final Dialog dialog = new Dialog (context, R.style.FilterDialogTheme);
    dialog setContentView (R.layout.dialog);
    dialog.setTitle (text [0]);
    TextView t = dialog.findViewById (R.id.textView);
    TextView date = dialog.findViewById (R.id.textDate);
    t.setMovementMethod (new ScrollingMovementMethod ());
    t.setText (text [1]);
    date.setText (text [2]);
    dialog.setCancelable (false);
    buttonClose = dialog.findViewById (R.id.btnClose);
    buttonDecode = dialog.findViewById (R.id.btnDecode);
    buttonClose.setOnClickListener ((v) -> {
        dialog.dismiss ();
    });
    buttonDecode.setOnClickListener ((v) -> {
        t.setText (Coder.coder (text [1]));
    });
}

```

					ДП. ІПЗ–15.ПЗ	Арк.
						33
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		

```

    dialog.show ();
}

```

Після перевірки підключення додаток спробує вивести усі смс, які були нам надіслані.

```

public ArrayList fetchInbox () {
    sms = new ArrayList ();
    Uri uriSms = Uri.parse ( "content: // sms / inbox");
    Cursor cursor = getContentResolver (). Query (uriSms, new String [] { "_ id", "address",
"date", "body"}, null, null, null);

    cursor.moveToFirst ();
    lastSms = cursor.getString (1) + "\ n" + cursor.getString (3);
    do {
        String address = cursor.getString (1);
        String body = cursor.getString (3);
        DateFormat dateFormat = new SimpleDateFormat ( "dd.MM.yyyy HH: mm");
        Date createdAt = new Date (Long.parseLong (cursor.getString (2)));
        String date = dateFormat.format (createdAt);
        String [] s = {address, body, date };
        arrayList.add (s);
        if (body.length ()> 70) {
            sms.add (Html.fromHtml ( "<h5>" + address + "</ h5>" + "\ n" + body.substring (0,
70) + "...", Html.FROM_HTML_MODE_COMPACT)); // прибрати і нормально
налаштувати
        } Else {
            sms.add (Html.fromHtml ( "<h5>" + address + "</ h5>" + "\ n" + body,
Html.FROM_HTML_MODE_COMPACT));
        }
    } While (cursor.moveToNext ());
    return sms;
}

```

Для взаємодії між бізнес логікою і клієнтським інтефейсом в MVC використовуються контролери (додаток Б). На даний час додаток містить такі контролери:

- ApplicationController – батьківський клас від якого наслідуються всі нище перчислені класи;
- CheckActivity – клас для роботи з активністю;
- CheckSentActivity – клас для роботи із повідомленнями;
- Coder – клас для шифрування;
- Database – клас для роботи з локальною базою;
- MainActivity – клас для роботи з вікнами.

Для прикладу візьмемо CheckSentActivity, в ньому оголошено такі функції для роботи з інформацією:

- onCreate – перевірка на пусту базу;
- showDialog – відображення вхідних шифрованих повідомлень;
- forListView - сортування усіх по часу;

Клас ConferencesController:

@Override

```
protected void onCreate (Bundle savedInstanceState) {
    super.onCreate (savedInstanceState);
    setContentView (R.layout.activity_check_sent);
    listSms = findViewById (R.id.listSms);

    try {
        ArrayAdapter adapter = new ArrayAdapter (this,
        android.R.layout.simple_list_item_1, forListView (db.getInfo ()));
        listSms.setAdapter (adapter);

        listSms.setOnItemClickListener (((parent, view, position, id) -> {
            showDialog (db.getInfo (). get (position));
        }));
    } Catch (android.database.sqlite.SQLiteException e) { // перевірити
```

```

        Toast toast = Toast.makeText (getApplicationContext (), "You have not sent any
messages yet", Toast.LENGTH_SHORT);
        toast.show ();
    }
}

```

```

public void showDialog (String [] text) {
    final Dialog dialog = new Dialog (context, R.style.FilterDialogTheme);
    dialog setContentView (R.layout.dialog);
    dialog.setTitle (text [0]);

```

```

    TextView t = dialog.findViewById (R.id.textView);
    TextView date = dialog.findViewById (R.id.textDate);
    t.setMovementMethod (new ScrollingMovementMethod ());
    t.setText (text [2]);
    date.setText (text [1]);
    dialog.setCancelable (false);
    buttonClose = dialog.findViewById (R.id.btnClose);
    buttonDecode = dialog.findViewById (R.id.btnDecode);

```

```

    buttonClose.setOnClickListener ((v) -> {
        dialog.dismiss ();
    });

```

```

    buttonDecode.setOnClickListener ((v) -> {
        t.setText (Coder.coder (text [2]));
    });
    dialog.show ();
}

```

```

private ArrayList <String> forListView (ArrayList <String []> arrayList) {
    ArrayList al = new ArrayList ();
    for (int i = 0; i < arrayList.size (); i++) {
        String address = ((String []) arrayList.get (i)) [0];

```

```

String body = ((String []) arrayList.get (i)) [2];
if (body.length ()> 70) {
    al.add (Html.fromHtml ( "<h5>" + address + "</ h5>" + "\ n" + body.substring (0,
70) + "...", Html.FROM_HTML_MODE_COMPACT)); // прибрати і нормально
налаштувати
} Else {
    al.add (Html.fromHtml ( "<h5>" + address + "</ h5>" + "\ n" + body,
Html.FROM_HTML_MODE_COMPACT));
}
}
return al;
}

```

Для відображення даних в зручному для користувача форматі MVC використовує представлення (додаток Г).

Всі вони служать для відображення даних і кожна з них містить стільки записів скільки і методів оголошено в контролерах, окрім захищених специфікатором private.

Для прикладу візьмемо клас Database, а точніше файл Database.java:

```

@Override
protected void onCreate (Bundle savedInstanceState) {
    super.onCreate (savedInstanceState);
    setContentView (R.layout.activity_check_sent);
    listSms = findViewById (R.id.listSms);

    try {
        ArrayAdapter adapter = new ArrayAdapter (this,
android.R.layout.simple_list_item_1, forListView (db.getInfo ());
        listSms.setAdapter (adapter);

        listSms.setOnItemClickListener (((parent, view, position, id) -> {
            showDialog (db.getInfo (). get (position));
        }));
    }
}

```

```

    ));
} Catch (android.database.sqlite.SQLiteException e) { // перевірити
    Toast toast = Toast.makeText (getApplicationContext (), "You have not sent any
messages yet", Toast.LENGTH_SHORT);
    toast.show ();
}
}
public void showDialog (String [] text) {
    final Dialog dialog = new Dialog (context, R.style.FilterDialogTheme);
    dialog setContentView (R.layout.dialog);
    dialog.setTitle (text [0]);
    TextView t = dialog.findViewById (R.id.textView);
    TextView date = dialog.findViewById (R.id.textDate);
    t.setMovementMethod (new ScrollingMovementMethod ());
    t.setText (text [2]);
    date.setText (text [1]);
    dialog.setCancelable (false);
    buttonClose = dialog.findViewById (R.id.btnClose);
    buttonDecode = dialog.findViewById (R.id.btnDecode);

    buttonClose.setOnClickListener ((v) -> {
        dialog.dismiss ();
    });

    buttonDecode.setOnClickListener ((v) -> {
        t.setText (Coder.coder (text [2]));
    });
    dialog.show ();
}
private ArrayList <String> forListView (ArrayList <String []> arrayList) {
    ArrayList al = new ArrayList ();
    for (int i = 0; i <arrayList.size (); i++) {
        String address = ((String []) arrayList.get (i)) [0];
        String body = ((String []) arrayList.get (i)) [2];
    }
}

```

```

if (body.length ()> 70) {
    al.add (Html.fromHtml ( "<h5>" + address + "</ h5>" + "\ n" + body.substring (0,
70) + "...", Html.FROM_HTML_MODE_COMPACT)); // прибрати і нормально
налаштувати
} Else {
    al.add (Html.fromHtml ( "<h5>" + address + "</ h5>" + "\ n" + body,
Html.FROM_HTML_MODE_COMPACT));
}
}
return al;
}

```

Щоб шифрувати повідомлення використовуємо простий метод синхронного шифрування, що дозволяє зробити мінімальну безпеку і конфіденційності, rails\_admin.rb (додаток Д).

Код класу шифрування:

```

public class Coder {
    private static StringBuilder sb;
    private static String temp, startPhase, key, phase = "";

    private static void phaseGenerate () {
        key = "111001111011";
        startPhase = "111111011111";
        sb = new StringBuilder (startPhase);

        while (true) {
            temp = "";
            for (int i = sb.length () - key.length (); i <sb.length (); i ++ ) {
                temp += sb.charAt (i);
            }
            if ((startPhase.equals (temp)) && (sb.length () != startPhase.length ())) {
                phase = sb.substring (0,
sb.length () - startPhase.length ());
                System.out.println (phase);
            }
        }
    }
}

```





Для формування тестових даних в java використовується бібліотека junit і файл seeds.rb. Щоб запустити створення тестових даних в консолі потрібно окремо запустити тест.

Код файлу ExampleUnitTest:

```
public class ExampleUnitTest {  
    @Test  
    public void addition_isCorrect () {  
        assertEquals (4, 2 + 2);  
    }  
}
```

Після вдалого тестування і написання усіх класів, і при запуску програми можна буде побачити наступне вікно:

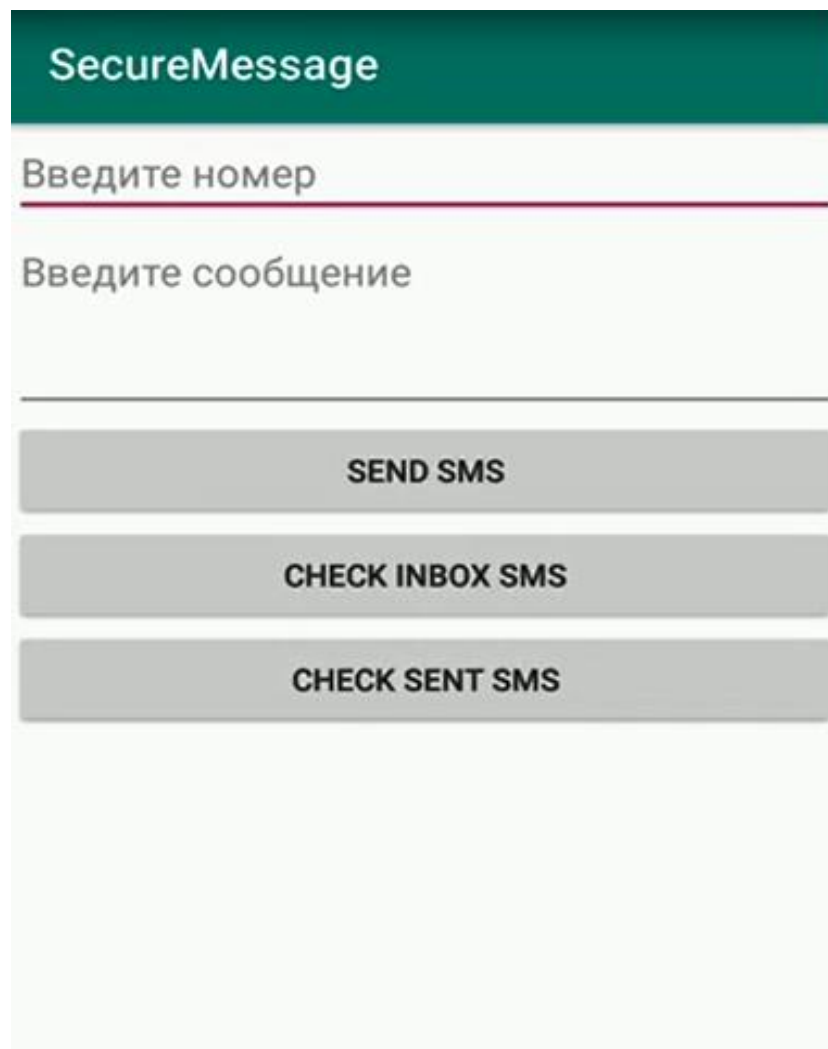


Рисунок 3.2 – Головне вікно

В даному вікні потрібно ввести номер телефону і повідомлення, яке необхідно відправити. Нижче (рисунок 3.2) неведений приклад з тестовими даними:

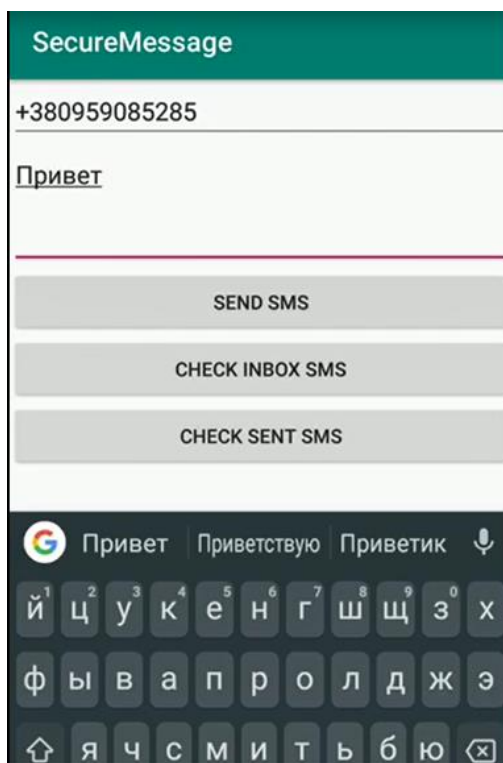


Рисунок 3.3 – Приклад відправлення повідомлення



Рисунок 3.4 – Відображення шифрованого повідомлення в телефоні

Після введення тестових даних і відправлення повідомлення, саме повідомлення в телефоні (не через програму) буде відображатись у вигляді набору незрозумілих символів, як у нас так і у користувача, якому відправили (рисунок 3.3)

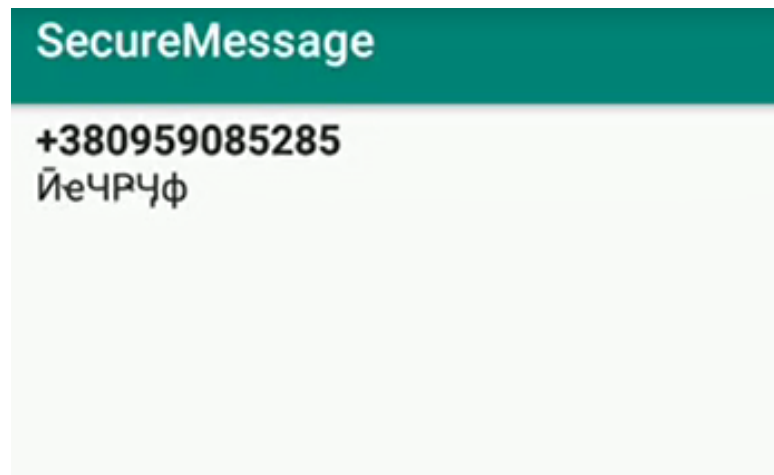


Рисунок 3.5 – Відображення повідомлення в додатку

При відкритті програми і відкритті повідомлення, яке ми відправили або отримали, вигляд буде ідентичний як і у телефоні без програми (рисунок 3.4).

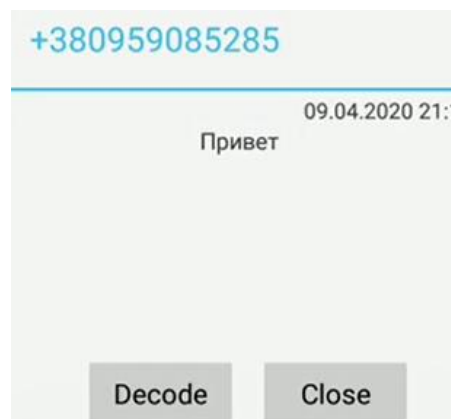


Рисунок 3.6 – Вигляд розшифрованого повідомлення

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		43

Для того що б прочитати повідомлення, яке було відправлено нам, потрібно відкрити повідомлення і натиснути функцію Decode, яка за допомогою алгоритму розшифрує повідомлення

### 3.3 Методики тестування

Для тестування даного додатку використовувався метод модульного тестування (Unit testing) – тести, завданням яких було перевірити кожен модуль системи окремо. Бажано, щоб це були мінімально подільні елементи системи, наприклад, модулі. Unit-модульні тести, що застосовуються в різних шарах додатку, що тестують найменшу подільну логіку програми: наприклад, клас, але найчастіше – метод. Ці тести, зазвичай, намагаються по максимуму ізолюватись від зовнішньої логіки, тобто створити ілюзію того, що інша частина програми працює в стандартному режимі.

Даних тестів завжди має бути багато (більше, ніж інших видів), так як вони тестують маленькі шматочки і вельми легковажні, не потребують багато ресурсів (під ресурсами я маю на увазі оперативну пам'ять і час). В даному проекті використовувався вбудований модуль Junit який відповідає за Unit тестування в Java, і який дозволяє покрити 100% коду тестами. Для прикладу відібрано тест, який показував справність виконання програмою шифрування тексту з публічними ключами:

```
public class ExampleUnitTest {
    @Test
    public void addition_isCorrect () {
        assertEquals (4, 2 + 2);
    }
}
```

## 4 ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ ПРОЕКТУ

### 4.1 Економічна задача

Завданням даної дипломної роботи є аплікація для криптування смс повідомлень на платформі Android .

Для визначення економічних показників доцільності впровадження програмного продукту чи математичної моделі існує декілька методів. Враховуючи стрімкий розвиток технологій на сьогоднішній день такий розрахунок є невід'ємною складовою попереднього аналізу створення програмного забезпечення і доцільністю, та рентабельністю його створення, оскільки саме автоматизація процесів, їх результат дає головне покращення в виборі технологій, або необхідному діагностуванню об'єктів, а фінансування, яке виділяється на даний проект мають бути еквівалентні тому ефекту, який є розрахований.

Для економічного обґрунтування розробки обчислюються наступні економічні показники:

витрати на розробку проекту;

економічний ефект;

річні експлуатаційні витрати.

### 4.2 Розрахунок витрат на розробку структури проекту

При обчисленні витрат на розробку комп'ютеризованої системи слід враховувати всі витрати від планування до кінцевого результату. Оскільки даний

Оскільки даний продукт розроблявся на ноутбуці, тому слід підрахувати собівартість ноутбука та кількість спожитої енергії.

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		45

На розробку програми було затрачено 3 місяці, при 5-ти годинному робочому дні та 3-ох робочих днях в тиждень 3 даних, наведених вище потрібно підрахувати скільки годин витрачено на створення продукту:

$$P_3 = 4 \cdot (P_m \cdot P_t \cdot P_d) \quad (4.1)$$

де  $P_3$  – час який був затрачений на розробку програми,

$P_m$  – кількість тижнів в місяць,

$P_t$  – кількість робочих днів в тиждень,

$P_d$  – кількість робочих годин в день.

Тоді згідно формули (4.1):

$$P_3 = 3 \cdot (4 \cdot 3 \cdot 5) = 180 \text{ год.}$$

Вартість комп'ютера ( $V_{eom}$ ) та його складових становить:

$$V_{eom} = 11500 \text{ грн.},$$

Термін експлуатації комп'ютера ( $T_{ек}$ ) становить 7448 робочих годин. Отже, ціна однієї години використання ноутбука ( $V_{геом}$ ) можна вирахувати за даною формулою формулою:

$$V_{геом} = V_{eom} / T_{ек}$$

Підставивши данні отримуємо:

$$V_{геом} = 11500 / 7448 = 1.54 \text{ грн.}$$

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		46

Витрати що припадають на одиницю години часу роботи машини можна вирахувати за витратами електроенергії.

$$V_{\text{ел}} = T_{\text{сп}} \cdot C_{\text{год}} \quad (4.2)$$

де  $T_{\text{сп}}$  - споживана потужність ЕОМ, Вт;

$C_{\text{год}}$  - вартість 1 кВт/год електроенергії.

Оскільки  $T_{\text{сп}} = 0.6$  кВт,  $C_{\text{год}} = 0.283$  грн., то згідно формули (4.2):

$$V_{\text{ел}} = 0.6 \cdot 0.283 = 0.17 \text{ грн/год}$$

Оскільки відома вся кількість годин, можемо вирахувати повну вартість витрат використання ноутбука та енергії під час створення продукту ( $V_{\text{в}}$ ):

$$V_{\text{в}} = (P_{\text{з}} \cdot V_{\text{геом}}) + (P_{\text{з}} \cdot V_{\text{ел}}) \quad (4.3)$$

Підставивши данні у формулу (4.3) маємо:

$$V_{\text{в}} = (180 \cdot 1.54) + (180 \cdot 0.17) = 307.8 \text{ грн.}$$

До ціни продукту також входить заробітня плата спеціаліста, котрий створював софт для ноутбука. До обрахунків повної вартості потрібно врахувати сплати у наступні фонди:

- нарахування в пенсійний фонд становить – 31,8%
- нарахування в фонд соціального страхування становить – 2,9%
- нарахування в фонд безробіття становить – 1,3%
- нарахування в фонд соціального страхування від нещасних випадків становить – 0,68%

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		47

Заробітна плата спеціаліста з розробки програмного забезпечення становить – 40 грн/год.

Користуючись даними, отриманими за допомогою обрахунків вище, отримуємо можливість розрахувати повну вартість із урахуванням заробітної плати спеціаліста. Заробітна плата розробника буде становити:

$$B_{зп} = P_3 \cdot ЗП_1 \quad (4.4)$$

Підставивши данні в формулу (4.4) отримаємо:

$$B_{зп} = 180 \cdot 40 = 7200 \text{ грн.}$$

Відрахування до пенсійного фонду ( $B_{пф}$ ) складає:

$$B_{пф} = B_{зп} \cdot 31.8\%$$

$$B_{пф} = 7200 \cdot 31.8\% = 2289.6 \text{ грн}$$

Відрахування до фонду соціального страхування ( $B_{сс}$ ) складає:

$$B_{сс} = B_{зп} \cdot 2.9\%$$

$$B_{сс} = 7200 \cdot 2.9\% = 208.8 \text{ грн.}$$

Відрахування до фонду безробіття ( $B_{фб}$ ) складає:

$$B_{фб} = B_{зп} \cdot 1.3\%$$

$$B_{фб} = 7200 \cdot 1.3\% = 86.4 \text{ грн.}$$

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		48



Відрахування до фонду соціального страхування від нещасних випадків ( $B_{ссч}$ ) становить:

$$B_{ссч} = B_{зп} \cdot 0.68\%$$
$$B_{ссч} = 7200 \cdot 0.68\% = 48,96 \text{ грн.}$$

За допомогою виконання розрахункових робіт вище, отримуємо можливість вирахувати загальну суму яка була нарахована для зарплати спеціалісту з розробки програми ( $B_{зп}$ ):

$$B_{зп} = B_{зп} + B_{пф} + B_{сс} + B_{фб} + B_{ссч}$$
$$B_{зп} = 7200 + 2289,6 + 208,8 + 86,4 + 48,96 = 9833.76 \text{ грн.}$$

Витрати на відлагодку програми включає в себе час витрачений спеціалістом для розповсюдження програми а також оплату для усунення неполадок. Для розповсюдження програми по мережі, яка включає в себе 2 ноутбуки, включно з сервером, була затрачена 1 год. Звідси маємо можливість вирахувати витрати на енергію::

$$B_{нел} = T_n \cdot K_m \cdot V_{ел} , \quad (4.5)$$

де  $T_n$  – час затрачений на налагодження програми;

$K_m$  – кількість комп'ютерів в мережі, тоді з формули (4.5):

$$B_{нел} = 2 \cdot 1 \cdot 0.17 = 0.34 \text{ грн.}$$

Витрати на усунення неполадків також входять до  $ЗП_2$  яка дорівнює 12 грн/год. Тоді загальні витрати на усунення неполадок проекту ( $B_n$ ) будуть наступні:

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		49

$$B_H = 2 \cdot 3П_2 + B_{нел}$$

$$B_H = 2 \cdot 12 + 0.34 = 24.34 \text{ грн.}$$

Коли відомі повністю усі данні є можливість порахувати повну сумму потрібну для реалізації розробки проекту ( $B_3$ ) включаючи вартість розробки структури програми ( $B_{cm}$ ):

$$B_3 = B_B + B_{ззп} + B_H + B_{ст}$$

$$B_3 = 307.8 + 9833.76 + 24.34 + 1012 = 11177.9 \text{ грн.}$$

Для розрахунку рентабельності  $P_p$  використовуємо таку формулу:

$$P_p = B_3 \cdot P \quad (4.6)$$

де  $P$  – коефіцієнт рентабельності і дорівнює 20%.

Тоді підставивши у формулу (4.6) отримаємо:

$$P_p = 11177.9 \cdot 20\% = 2235.58 \text{ грн.}$$

### 4.3 Розрахунок витрат на експлуатацію

Освоєння програми та її використання на протязі року розраховане на 200 годин ( $T_{pич}$ ). Для вираховання витрат на використання потрібно розрахувати зарплату оператора який буде впроваджувати систему, інженера, який розробив програму, а також вартість електроенергії та знос комп'ютерів які буде обслуговувати комп'ютерна система.

Заробітна плата оператора ( $3П_0$ ) який буде обслуговувати дану систему становить – 15 грн/год. Отже за весь рік експлуатації його зарплата буде

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		50

СТАНОВИТИ:

$$B_{зп} = T_{річ} \cdot ЗП_0. \quad (4.7)$$

Тоді підставивши значення в формулу (4.7) отримаємо:

$$B_{зп} = 200 \cdot 15 = 3000 \text{ грн.}$$

Тепер підрахуємо витрати у фонди:

нарахування до пенсійного фонду ( $B_{пф}$ ) складає:

$$B_{пф} = B_{зп} \cdot 31.8\%$$

$$B_{пф} = 3000 \cdot 31.8\% = 954 \text{ грн.},$$

нарахування в фонд соціального страхування ( $B_{сс}$ ) складає:

$$B_{сс} = B_{зп} \cdot 2.9\%$$

$$B_{сс} = 3000 \cdot 2.9\% = 87 \text{ грн.},$$

нарахування в фонд безробіття ( $B_{фб}$ ) становить:

$$B_{фб} = B_{зп} \cdot 1.3\%$$

$$B_{фб} = 3000 \cdot 1.3\% = 39 \text{ грн.},$$

нарахування в фонд соціального страхування від нещасних випадків ( $B_{ссч}$ ) становить:

$$B_{ссч} = B_{зп} \cdot 0.68\%$$

$$B_{ссч} = 3000 \cdot 0.68\% = 20.4 \text{ грн.}$$

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		51

За допомогою формул можна вирахувати повну сумму, яку необхідно нарахувати оператору для зарплати оператора який буде обслуговувати дану програму ( $B_{зпз}$ ):

$$B_{зпз} = B_{зп} + B_{пф} + B_{сс} + B_{фб} + B_{ссч}$$
$$B_{зпз} = 3000 + 954 + 87 + 39 + 20.4 = 4100.4 \text{ грн.}$$

Заробітна плата інженера ( $ЗП_4$ ) який буде обслуговувати дану програму становить – 13 грн./год. Отже за весь рік експлуатації його зарплата буде становити:

$$B_{зп} = T_{річ} \cdot ЗП_4 . \quad (4.8)$$

Тоді підставивши значення в формулу (4.8) отримаємо:

$$B_{зп} = 200 \cdot 13 = 2600 \text{ грн.}$$

Тепер підрахуємо витрати у фонди:

нарахування в пенсійний фонд ( $B_{пф}$ ) складає:

$$B_{пф} = B_{зп} \cdot 31.8\%$$
$$B_{пф} = 2600 \cdot 31.8\% = 826.8 \text{ грн.,}$$

нарахування в фонд соціального страхування ( $B_{сс}$ ) складає:

$$B_{сс} = B_{зп} \cdot 2.9\%$$
$$B_{сс} = 2600 \cdot 2.9\% = 75.4 \text{ грн.,}$$

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		52

нарахування в фонд безробіття ( $B_{фб}$ ) складає:

$$B_{фб} = B_{зп} \cdot 1.3\%$$

$$B_{фб} = 2600 \cdot 1.3\% = 33.8 \text{ грн. ,}$$

нарахування в фонд соціального страхування від нещасних випадків ( $B_{ссч}$ ) становить:

$$B_{ссч} = B_{зп} \cdot 0.68\%$$

$$B_{ссч} = 2600 \cdot 0.68\% = 17.68 \text{ грн.}$$

За допомогою обчислень, маємо можливість провести обрахунки на повну сумму, яку нараховали для зарплат інженера для обслуговування програми ( $B_{зп4}$ ):

$$B_{зп4} = B_{зп} + B_{пф} + B_{сс} + B_{фб} + B_{ссч}$$

$$B_{зп4} = 2600 + 826.8 + 75.4 + 33.8 + 17.68 = 3553.68 \text{ грн.}$$

Витрати на електроенергію ( $B_{елріч}$ ) за час використання можна порахувати підставивши значення у формулу (4.5), але за умови що  $T_n = T_{річ}$ , отже:

$$B_{елріч} = 200 \cdot 3 \cdot 0.17 = 102 \text{ грн.}$$

Знос комп'ютерів ( $B_{зк}$ ) можна порахувати за наступною формулою:

$$B_{зк} = B_{геом} \cdot T_{річ} \cdot K_M. \quad (4.10)$$

Підставивши значення у формулу (4.10) отримаємо:

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		53

$$B_{зк} = 1.66 \cdot 200 \cdot 3 = 996 \text{ грн.}$$

Коли стали відомі усі данні, моємо можливість вирахувати повну сумму витрат ( $B_e$ ):

$$B_e = B_{зпз} + B_{зп4} + B_{елріч} + B_{зк}. \quad (4.11)$$

Підставивши значення у формулу (4.11) отримаємо:

$$B_e = 4100.4 + 3553.68 + 102 + 996 = 8752.08 \text{ грн.}$$

#### 4.4 Економічний ефект

Для розрахунку економічного ефекту потрібно вирахувати витрати на розробку продукту та експлуатації машини, тобто:

$$E_e = B_з + B_e. \quad (4.12)$$

Підставивши значення у формулу (4.12) отримаємо:

$$E_e = 11177.9 + 8752.08 = 19929.98 \text{ грн.}$$

Тепер, коли є загальна сума витрат на розробку програми та витрат на експлуатацію додатку, можна порахувати економічний ефект. Для цього потрібно порівняти дані розробленого додатку з витратами на існуючі додатку приблизна ціна back-end яких становить 22000:

$$E_{еф} = 22000/19929.98 = 1.1$$

$$E_{еф} = 22000 - 19929.98 = 2070.02 \text{ грн}$$

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		54

Наведені вище розрахунки підтверджують доцільність застосування програми в процесі як з технічної так і з економічної точок зору.

В таблиці 4.1 перелічені зведені економічні показники проектного програмного забезпечення.

Таблиця 4.1 – Зведені економічні показники розробки

<b>Показник</b>	<b>Розмірність</b>	<b>Значення</b>
Витрати на розробку структури програми	грн.	8752.08
Загальну суму витрат пов'язаних з розробкою програми	грн.	11177.9
Загальна сума витрат пов'язаних з експлуатацією програми	грн.	19929.98
Економічний ефект	Разів	1.1
Грошовий економічний ефект	Грн	2070.02

З вище наведених розрахунків видно, що розробка та впровадження даного алгоритму є економічно доцільною, ефективність його впровадження в 1.1 разів краща за попередній.

## ВИСНОВКИ

В даній дипломній роботі спроектовано та реалізовано додаток шифрування повідомлень на андроїд.

В процесі дослідження предметної області було систематизовано вхідні дані та створено специфікацію системи. В якості бази даних було використано документоорієнтовану СУБД SQLite.

Розроблений додаток дозволяє шифрувати повідомлення і відправляти іншому користувачу, а також отримувати закодовані повідомлення і розшифровувати за допомогою програми.

При проектуванні виконано всі необхідні економічні розрахунки і досліджено, що розробка та впровадження системи є доцільною та економічно вигідною.

Отже в даній дипломній роботі спроектовано повністю функціональний, логічно побудований додаток, з використанням Java, який повністю задовільняє поставлені вимоги.

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		56



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

### REFERENCES

1. David A. Black The Well-Grounded Java / A. David. – Manning Publications Co., 2009. – 483 с.
2. David A. Black The Well-Grounded Java second edition / A. David. – Manning Publications Co., 2014. – 538 с.
3. Java -Doc.org: Documenting the Java Language [Електронний ресурс]. – Core API docs for Ruby 2.5.1 This is the current official release. – Режим доступу: <https://ruby-doc.org/core-2.5.1/>.
4. java – java Manual 7.0 [Електроннийресурс]. – java is the officially supported ODM (Object-Document-Mapper) framework for java in Java. – Режим доступу: <https://docs.java.com/java/master>
5. Subbu Allamaraju RESTful Web Services Cookbook / Subbu Allamaraju//O'Reilly Media / Yahoo Press. – 2010. – 316.
6. Martin Fowler UML Distilled: A Brief Guide to the Standard Object ModelingLanguage/ Martin Fowler // – Addison-Wesley Professional – 2003. – 208.
7. C.J. DateAn Introduction to Database / C.J. DateAn // – Pearson – 2003. – 1024.
8. Kathy Sierra Head First Java /Kathy Sierra// – O'Reilly Media – 2005. – 688.1.Ryan Hodson Android Programming Succinctly / Ryan Hodson // – SyncfusionInc. – 2014. – 113.
9. Кузь М., Новак В., Новак М., Сільва Р. Метод визначення вагових коефіцієнтів метрик якості програмних продуктів. Прикладні науково-технічні дослідження: матеріали III міжнар. наук.-практ. конф., м. Івано-Франківськ, 3-5 квіт. 2019 р., Івано-Франківськ, 2019. С. 35.
10. Kuz M., Shevliuk L., Ostafiichuk T., Mishagin R. Modeling of software time characteristics. Applied scientific and technical research: Proceedings of the IV International Scientific and Practical Conference, Ivano-Frankivsk, April 1-3, 2020, Ivano-Frankivsk, V.2. 2020. P. 39-42.

					ДП. ІПЗ–15.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		57

## Додаток А

### Код файлу CheckSentActivity

```

package com.example.securemessages;
import androidx.appcompat.app.AppCompatActivity;
import java.util.ArrayList;
public class CheckSentActivity extends AppCompatActivity {

    private Context context = this;
    private Button buttonClose;
    private Button buttonDecode;
    private ListView listSms;
    private DataBase db = new DataBase (this);

    @Override
    protected void onCreate (Bundle savedInstanceState) {
        super.onCreate (savedInstanceState);
        setContentView (R.layout.activity_check_sent);
        listSms = findViewById (R.id.listSms);
        try {
            ArrayAdapter adapter = new ArrayAdapter (this,
android.R.layout.simple_list_item_1, forListView (db.getInfo ());
            listSms.setAdapter (adapter);
            listSms.setOnItemClickListener (((parent, view, position, id) -> {
                showDialog (db.getInfo (). get (position));
            }));
        } Catch (android.database.sqlite.SQLiteException e) {// перевірити
            Toast toast = Toast.makeText (getApplicationContext (), "You have not sent any
messages yet", Toast.LENGTH_SHORT);
            toast.show ();
        }
    }
}

```

```

public void showDialog (String [] text) {
    final Dialog dialog = new Dialog (context, R.style.FilterDialogTheme);
    dialog setContentView (R.layout.dialog);
    dialog.setTitle (text [0]);
    TextView t = dialog.findViewById (R.id.textView);
    TextView date = dialog.findViewById (R.id.textDate);
    t.setMovementMethod (new ScrollingMovementMethod ());
    t.setText (text [2]);
    date.setText (text [1]);
    dialog.setCancelable (false);
    buttonClose = dialog.findViewById (R.id.btnClose);
    buttonDecode = dialog.findViewById (R.id.btnDecode);
    buttonClose.setOnClickListener ((v) -> {
        dialog.dismiss ();
    });
    buttonDecode.setOnClickListener ((v) -> {
        t.setText (Coder.coder (text [2]));
    });
    dialog.show ();
}

private ArrayList <String> forListView (ArrayList <String []> arrayList) {
    ArrayList al = new ArrayList ();
    for (int i = 0; i < arrayList.size (); i++) {
        String address = ((String []) arrayList.get (i)) [0];
        String body = ((String []) arrayList.get (i)) [2];
        if (body.length () > 70) {
            al.add (Html.fromHtml ( "<h5>" + address + "</ h5>" + "\ n" + body.substring (0,
70) + "...", Html.FROM_HTML_MODE_COMPACT)); // прибрати і нормально
налаштувати
        } Else {
            al.add (Html.fromHtml ( "<h5>" + address + "</ h5>" + "\ n" + body,
Html.FROM_HTML_MODE_COMPACT));
        }
        return al;    }
}

```

## Додаток Б

### Код класу шифрування

```

public class Coder {
    private static StringBuilder sb ;
    private static String temp, startPhase, key, phase = "";

    private static void phaseGenerate() {
        key = "111001111011";
        startPhase ="11111011111";
        sb = new StringBuilder(startPhase);
        while (true) {
            temp = "";
            for (int i = sb.length() - key.length(); i < sb.length(); i++) {
                temp += sb.charAt(i);
            }
            if ((startPhase.equals(temp)) && (sb.length() != startPhase.length())) {
                phase = sb.substring(0, sb.length() - startPhase.length());
                System.out.println(phase);
                break;
            }
            sb.append(Integer.bitCount((Integer.parseInt(key, 2) ^ Integer.parseInt(temp, 2)) &
Integer.parseInt(key, 2)) % 2);
        }
    }

    public static String coder(String text) {
        phaseGenerate();
        sb = new StringBuilder();
        for (int i = 0; i < text.length(); i++) {
            sb.append(((char) (text.charAt(i) ^ Integer.parseInt(phase.substring(i * 8, (i + 1) * 8), 2))));
        }
        return sb.toString();
    }
}

```

## Додаток В

### Код Інтерфейсу

```

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

<EditText
    android:id="@+id/txtPhoneNo"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="Введите номер"
    android:inputType="phone"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />

<EditText
    android:id="@+id/txtMessage"
    android:layout_width="fill_parent"
    android:layout_height="90dp"
    android:gravity="top"
    android:hint="Введите сообщение"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtPhoneNo"
    app:layout_constraintVertical_bias="0.0" />

<Button
    android:id="@+id/btnSendSMS"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Send SMS"
    app:layout_constraintBottom_toBottomOf="parent"

```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txtMessage"
app:layout_constraintVertical_bias="0.0" />

```

```
<Button
```

```

    android:id="@+id/btnCheckSMS"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Check Inbox SMS"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnSendSMS"
    app:layout_constraintVertical_bias="0.0" />

```

```
<Button
```

```

    android:id="@+id/btnCheckSentSms"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Check Sent Sms"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnCheckSMS"
    app:layout_constraintVertical_bias="0.0" />

```

```
<Button
```

```

    android:id="@+id/buttonExit"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Exit"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnCheckSentSms"
    app:layout_constraintVertical_bias="1.0" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Додаток Г

### Код Предствалення

```

package com.example.securemessages;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.widget.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

public class MainActivity extends AppCompatActivity {

    private EditText numberText;
    private EditText messageText;
    private Button buttonSend;
    private Button buttonCheck;
    private Button buttonCheckSent;
    private Button buttonExit;
    private DataBase db = new DataBase(this);

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

        numberText = findViewById(R.id.txtPhoneNo);
        messageText = findViewById(R.id.txtMessage);
        buttonSend = findViewById(R.id.btnSendSMS);
        buttonCheck = findViewById(R.id.btnCheckSMS);
        buttonCheckSent = findViewById(R.id.btnCheckSentSms);
        buttonExit = findViewById(R.id.buttonExit);

        buttonSend.setOnClickListener((v)->{
            if(!numberText.getText().toString().equals("") &&
!messageText.getText().toString().equals("")) {
                String toSms = "smsto:" + numberText.getText().toString();
                String textSms = messageText.getText().toString();
            }
        });
    }
}

```

```

        DateFormat dateFormat = new SimpleDateFormat("dd.MM.yyyy HH:mm");
        Date date = new Date();
        db.insert(numberText.getText().toString(), dateFormat.format(date), textSms);
        String[] s = {numberText.getText().toString(),
Coder.coder(messageText.getText().toString())};
        Intent sms = new Intent(Intent.ACTION_SENDTO, Uri.parse(toSms));
        sms.putExtra("sms_body", Coder.coder(textSms));
        startActivity(sms);
    }else{
        Toast toast = Toast.makeText(getApplicationContext(),"Enter a phone number and/or
write a message", Toast.LENGTH_SHORT);
        toast.show();
    }
});

buttonCheck.setOnClickListener((v)->{
    Intent intent = new Intent(MainActivity.this, CheckActivity.class);
    startActivity(intent);
});

buttonCheckSent.setOnClickListener((v)->{
    Intent intent = new Intent(MainActivity.this, CheckSentActivity.class);
    startActivity(intent);
});

buttonExit.setOnClickListener((v)->{
    finish();
});
}
}
}

```