НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)

"___" _____ 202___ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Спеціалізовані комп'ютерні системи»
спеціальності **123 «Комп'ютерна інженерія»**

на тему: «Додатковий модуль захисту мобільних пристроїв» _____
_____
_____

Виконав:

студент IV курсу, групи К В - 6 3
2(шифр групи)

Парфіненко Юрій Михайлович _____ _____
(прізвище, ім'я, по батькові) (підпис)

Керівник професор кафедри СПіСКС, д.т.н., професор Терейковський І.А. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2020 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»

Освітньо-професійна програма «Спеціалізовані комп’ютерні системи»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)
«___» _____20___ р.

ЗАВДАННЯ
на дипломний проєкт студента

Парфіненка Юрія Михайловича

1. Тема проєкту «Додатковий модуль захисту мобільних пристроїв» ,керівник проєкту професор кафедри СПіСКС, д.т.н., Терейковський І.А.
затверджені наказом по університету від «25» травня 2020 р. №N1181-С

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту Операційна система Windows 10, Частота процесору 2.4 ГГц, Середовище розробки – Android Studio, Мова програмування – Java.

4. Зміст пояснювальної записки постановка задачі, аналіз існуючих рішень, вибір засобів реалізації, реалізація, керівництво по використанню, висновки.

5. Перелік графічного матеріалу (із зазначенням обов’язкових креслеників, плакатів, презентацій тощо) зв’язки модулів програми, блок-схеми роботи програми, блок-схема роботи інтерфейсу користувача, алгоритм програмного забезпечення

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Видача завдання на дипломне проєктування	26.12.2019	
2.	Вивчення літератури за тематикою роботи	16.01.2020	
3.	Розроблення та узгодження технічного завдання	25.01.2020	
4.	Розроблення структури додатку	17.02.2020	
5.	Розроблення дизайну та графічних елементів	04.03.2020	
6.	Програмна реалізація додатку	15.03.2020	
7.	Тестування додатку	2.04.2020	
8.	Підготовка матеріалів текстової частини проєкту	30.04.2020	
9.	Підготовка матеріалів графічної частини проєкту	16.05.2020	
10.	Оформлення технічної документації проєкту	19.05.2020	

Студент _____ Юрій ПАРФІНЕНКО
(підпис) (Ім'я та ПРІЗВИЩЕ)

Керівник проєкту _____ Ігорь ТЕРЕЙКОВСЬКИЙ
(підпис) (Ім'я та ПРІЗВИЩЕ)

* Консультантом не може бути зазначено керівника дипломного проєкту.

АНОТАЦІЯ

Бакалаврській дипломний проєкт викладений на 50 сторінці, містить в собі 4 розділи, 19 ілюстрацій, 2 таблиці та 12 джерел в переліку посилань.

Даний дипломний проєкт посвячений захисту даних та інформації, що містить на телефонах.

Метою даного дипломного проєкту є створення додаткового модулю, за допомогою якого можна додатково захисти власний телефон.

В роботі було розглянуто існуючі рішення даної проблематика та широко розглянута загальна потреба в даному проєкту. В результаті було виділено основні переваги існуючих рішень, а також їх недоліки.

Було створено додаток, який компенсує основні недоліки існуючих рішень з схожим способом захисту даних. Також був даний стислий огляд засобів які використовувались для написання даного дипломного проєкту.

В першому розділі міститься описання проблеми та шляхи їх рішення. В другому розділі описані шляхи виконання поставленої задачі. В третьому розділі описанні використанні шаблони та класи й описана загальна структура програми. В четвертому розділі містяться вказівки, що до використання створеного додатку.

В результаті виконання проєкту зроблені висновки для кожного розділу та загальні.

Ключові слова: захист інформації, додаток, графічний ключ, Android Studio, Java, користувач.

ANNOTATION

The thesis consists of 50 pages, has 4 sections, 19 illustrations, 2 tables, 12 points in the link list.

This project dedicated to the protection of data and information contained on telephones.

The aim of this diploma project is to create module with which you can further protect your own phone.

The project considered the existing solutions of this problem and considered the general need for this project. As result the main advantages of existing solution as well as their disadvantages were highlighted.

As a result, an application was created that compensates for the main shortcomings of existing solution with a similar way of data protection. A brief overview of the tools used to write this thesis project was also given.

The first section contains a description of the problem and wats to solve them. The second section describes ways to accomplish this task. The third section describe the use of templates classes and describes the general structure of the program. The fourth section provides instructions for using the created application.

As a result of the project, conclusions are made for each section as well as general.

Keywords: information protection, application, graphic key, user, Android Studio, Java.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045470.002 ТЗ	Розробка додаткового	4		
			модулю захисту			
			мобільного пристрою			
			Технічне завдання			
	А4	ІАЛЦ.045470.003 ТП	Розробка додаткового	2		
			модулю захисту			
			мобільного пристрою			
			Відомість технічного			
			проєкту			
	А4	ІАЛЦ.045470.004 ПЗ	Розробка додаткового	50		
			модулю захисту			
			мобільного пристрою			
			Пояснювальна записка			
	А4	ІАЛЦ.045470.005 Д1	Схема взаємодії модулів	1		
			програми			
			Діаграми класів			

Змін.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.045470.001 ОА		
Розробив		Парфіненко Ю.М.			Додатковий модуль захисту мобільних пристроїв **Опис альбому**	Літ.	Аркуш 1 / Аркушів 2
Перевірив		Терейковський І.А.					
Консульт.						КПІ ім. Ігоря Сікорського, ФПМ КВ-63	
Н. контроль		Клятченко Я.М.					
Зав. каф.		Тарасенко В.П.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045470.006 Д2	Блок-схема алгоритм	1		
			роботи програми			
			Блок-схема алгоритму			
	А4	ІАЛЦ.045470.007 Д3	Схема роботи	1		
			Програмного забезпечення			
			Функціональна схема			
	А4	ІАЛЦ.045470.008 Д4	Блок-схема	1		
			функціонування			
			інтерфейсу користувача			
			Блок-схема алгоритму			
		Диск CD-ROM	Текст пояснювальної	1		
			записки.			
			Графічний матеріал			

					ІАЛЦ.045470.001 ОА	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		2

ЗМІСТ

					ІАЛЦ. 045470.002 ТЗ		
Змін	Арк.	№ докум.	Підпис	Дата			
Розробив		Парфіненко Ю.М.			*Додатковий модуль захисту мобільних пристроїв*	Літ.	Аркуш 1 / Аркушів 4
Перевірив		Терейковський І.А.					
Н. контроль		Клятченко Я.М.			***Технічне завдання***	КПІ ім. Ігоря Сікорського, ФПМ КВ-63	
Затвердив		Тарасенко В.П.					

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Додатковий модуль захисту мобільних пристроїв».

Галузь застосування: створення модулю, за допомогою якого можливо захистити дані на телефоні.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення мобільного додатку під операційну систему Android для захисту даних та інформації.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації в періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

1.1 Вимоги до програмного продукту, що розробляється

- сумісність з операційною системою Android;
- можливість працювати в фоновому режимі;
- можливість створювати пароль;
- можливість змінювати пароль;
- зберігання встановленого паролю в захищеному місці;
- наявність зручної системи вибору для захисту даних;

a. Вимоги до апаратного забезпечення

- оперативна пам’ять: 1 Гб;
- наявність доступу до мережі Wi-Fi (IEEE 802.11 b/g/n).

b. Вимоги до програмного та апаратного забезпечення користувача

- операційна система Android 9.0 та вище;
- наявність доступу до мережі Wi-Fi (IEEE 802.11 b/g/n).

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Видача завдання на дипломне проєктування	26.12.2019
2.	Вивчення літератури за тематикою роботи	16.01.2020
3.	Розроблення та узгодження технічного завдання	25.01.2020
4.	Розроблення структури додатку	17.02.2020
5.	Розроблення дизайну та графічних елементів	04.03.2020
6.	Програмна реалізація додатку	15.03.2020
7.	Тестування додатку	2.04.2020
8.	Підготовка матеріалів текстової частини проєкту	30.04.2020
9.	Підготовка матеріалів графічної частини проєкту	16.05.2020
10.	Оформлення технічної документації проєкту	19.05.2020

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045470.004 ПЗ	Додатковий	50		
			модуль захисту			
			мобільного пристрою			
			Пояснювальна записка			
	А4	ІАЛЦ.045470.005 Д1	Схема взаємодії модулів	1		
			Програми			
			Діаграми класів			
	А4	ІАЛЦ.045470.006 Д2	Блок-схема алгоритм	1		
			роботи програми			
			Блок-схема алгоритму			
	А4	ІАЛЦ.045470.007 Д3	Схема роботи	1		
			Програмного забезпечення			
			Функціональна схема			

Змін.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.045470.003 ТП		
Розробив		Парфіненко Ю.М.			Додатковий модуль захисту мобільних пристроїв **Відомість технічного проєкту**	Літ.	Аркуш 1 / Аркушів 2
Перевірив		Терейковський І.А.					
Консульт.						КПІ ім. Ігоря Сікорського, ФПМ КВ-63	
Н. контроль		Клятченко Я.М.					
Зав. каф.		Тарасенко В.П.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045470.008 Д4	Блок-схема	1		
			функціонування			
			інтерфейсу користувача			
			Блок-схема алгоритму			
		Диск CD-ROM	Текст пояснювальної	1		
			записки.			
			Графічний матеріал			

					ДП.045470.003 ТП	Арк. 2
	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

					ІАЛЦ. 045470.004 ПЗ		
Зм.	Лист	№ докум.	Підп.	Дата			
Розробив		Парфіненко Ю.М.			Додатковий модуль захисту мобільних пристроїв **Пояснювальна записка**	Літ.	Аркуш 1 / Аркушів 50
Перев.		Терейковський І.А.					
Н. контр.		Клятченко Я.М				НТУУ "КПІ" ім. Ігоря Сікорського, ФПМ, КВ-63	
Затвер.		Тарасенко В.П.					

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

Wi-Fi – технологія бездротового зв’язку локальної мережі, основана на стандартах IEEE 802.11.

Touch ID – сканер відбитків пальців, розроблений корпорацією Apple.

Face ID – сканер об’ємно-просторової форми обличчя людини, розроблений компанією Apple.

PIN (англ. Personal Identification Number — персональний ідентифікаційний номер) – аналог пароля.

ОС (OS) – операційна система.

ПК – персональний комп’ютер .

Ppi (англ, pixel per inch) – розширення пікселів на один дюйм.

Drag-and-drop (перетягування) — форма виконання певних дій у графічних інтерфейсах користувача, що передбачає використання комп’ютерної миші.

API (програмний інтерфейс додатка, інтерфейс прикладного програмування) (англ. Application programming interface, API) – опис способів (набір класів, процедур, функцій, структур або констант), якими одна комп’ютерна програма може взаємодіяти з іншою програмою.

RSS — це родина XML-форматів, що використовується для публікації та постачання інформації, що часто змінюється, наприклад, нових записів в блозі, заголовків новин, анонсів статей, зображень, аудіо і відео матеріалів. Документ в стандарті RSS (який також інколи називають «стрічкою», «веб-стрічкою» або «каналом») складається з повного або часткового тексту і метаданих (дата і авторство).

IDE – Integrated Development Environment або інтегроване середовище розробки.

Байт-код (англ. Bytecode, також іноді p-код, p-code від portable code) – стандартне проміжне представлення, в яке може бути переведена комп'ютерна програма автоматичними засобами.

RAM (англ. *Random Access Memory, ОЗП*, пам'ять с довільним доступом, оперативна пам'ять) – енергозалежна частина системи комп'ютерної пам'яті, в якій під час роботи комп'ютера зберігається виконуваний машинний код (програми), а також вхідні, вихідні та проміжні дані, що обробляються процесором.

GB (гігабайти) – одиниці вимірювання розміру інформації

SDK (від англ. Software development kit) – набір засобів розробки, який дозволяє фахівцям з програмного забезпечення створювати додатки для певного пакету програм.

ВСТУП

Проблема захисту особистих даних є однією з основних проблем сьогодення, оскільки існує багато способів та засобів їх крадіжки. В більшій мірі втрата особистих даних відбувається через недостатній їх захист, зокрема це можуть бути примітивні способи захисту, паролі, дуже короткі чи складаються з простих чисел, ну або взагалі просто набор цифр підряд.

Найпростіший спосіб крадіжки даних є: крадіжка персональних даних з телефону чи соціальних мереж, бо більшість людей мають на телефоні персональні фото чи відео, які б вони не бажали показувати іншим, а в тому числі й персональні аккаунти в соціальних мережах, де зберігаються персональні переписки з іншими людьми.

Тому, на даний момент існує багато способів блокування телефону, найбільш поширені це: Touch ID, Face ID, графічний ключ чи простий пароль. У всіх передових смартфонах є дані способи блокування/розблокування, однак, існують способи обійти дані способи захисту. Хоча виробники даних алгоритмів стверджують, що можливість цього прирівнюється один до мільйона, у випадку з Face ID, чи один до сто тисяч, у випадку з Touch ID, однак деякі з представлених алгоритмів не є повноцінними, і це доказано на практиці людьми, які намагалися найти всі не враховані в алгоритм випадки.

Тому, для того, щоб забезпечити більш надійний захист персональних даних на телефоні, не буде лишнім забезпечити захист даних вже розблокованого телефону.

Саме для цього я розробив програму додаткового захисту додатків на телефоні. Представлена в даному дипломному проєкті програма може вибірково додавати запит на заданий вами графічний ключ, перед запуском додатка.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМУ

1.1. Загальний опис проблеми

На сьогоднішній день трафік інформації, завдяки інтернету, є дуже великий, кожен день ми переглядаємо та зберігаємо різні її види, здебільшого це текстова інформації отримана з різних джерел. Оскільки пам’ять носіїв, зокрема телефонів, збільшується з великою швидкістю, то і кількість інформації яку ми можемо зберігати збільшується.

Існує також, так звані, хмарні сховища інформації, тобто ви можете зберігати інформацію не на носії, а на сервері компанії, яка пропонує даний сервіс, наприклад Icloud, Google disk, Яндекс диск та інші. Даний спосіб зберігання даних є дуже простий в використанні, однак доволі часто дані користувачів бувають вкрадені. Більшість випадків крадіжки даних відбувається саме з таких сервісів, тому що обсяг користувачів даних сервіс доволі великий і слідкувати за безпекою кожного є неможливим. В такому випадку від користувача нічого не залежить, хіба що він розповів комусь персональні дані для входу, і він ні як не захищений від крадіжки даних таким чином.

Інколи зловмисники використовують розсилку повідомлень для крадіжки ваших персональних даних, в більшості випадків це банківські дані. Звісно від цього надає захист спам фільтр який влаштований в більшості додатків для повідомлень та електронних скриньках. Він розпізнає спосіб поширення повідомлення та його вміст, і відштовхуючись від отриманих даних вирішує чи відправляти повідомлення в спам. В більшості спам фільтрах можуть стояти розпізнавання ключових слів, які найчастіше використовують зловмисники, наприклад в більшості випадків вони намагаються повідомити отримувача про виграш великої суми грошей та просять передати дані картки чи перейти за посиланням. Раніше дані

розсилання подібних повідомлень посилались відразу на велику кількість користувачів, що відразу і видавало спам, тому вони і блокувалися. На сьогодні деякі зловмисники також використовують розсилання повідомлень через телефони. Звісно для таких випадків також існують додатки які захищають від такого спаму, але не всі їх використовують. Потрібно бути обережним та не довіряти повідомленням які вітають вас з перемогою в будь-якій лотереї, про яку ви навіть не знали, крім цього зловмисники можуть видавати себе менеджером банку якого ви користуєтесь й намагатися змусити вас видати персональні дані банківського рахунку.

Також існує риск небезпечного трафіку при загрузці чого-небудь з інтернету. Наприклад, це може бути вірус який подається у вигляді файлу установки, або просто міститись зашифрованим в якомусь вигляді. З даною проблемою можна зустрітися доволі часто, оскільки всі ми використовуємо безплатні точки Wi-Fi, що може бути небезпечним. Розглянемо чим може бути небезпечний даний тип підключення.

Підключитися до такої мережі може будь-хто, в тому числі і зловмисник. Підключишся до безплатної мережі Wi-Fi зловмисник може загрузити вірусну програму, і потім під виглядом загрузки оновлень популярних додатків, в більшості випадків від всім відомих компаній, таких яких Google чи Adobe, встановити на ваш пристрій сторонній додаток, який буде шукати конфіденційну інформацію користувача, дані паспорту бухгалтерські документи та таке подібне. Також цей додаток може запам’ятовувати введені вами паролі та логіни і передавати їх зловмиснику. Деякі зловмисники також можуть перехоплювати ваші дані підключившись до публічної мережі. Є ймовірність що зловмисники можуть створити псевдо публічну мережу, тобто може існувати точка доступу Wi-Fi, і така ж сама лише з поміткою _Free, щоб підтвердити безплатний допуск до неї, і за допомогою даної мережі будуть вкрадені ваші дані, оскільки весь трафік даних буде йти через зловмисника. Зловмисники також можуть створити

копію добре відомого сайту для крадіжки ваших логінів та паролів, це можна уникнути просто необхідно перевіряти сайт на наявність захисних протоколів hhtps.

Для захисту від такого виду загроз можна частково захиститися антивірусом, який можна завантажити та встановити з офіціального сайту чи маркетів з додатками, Google Play чи App Store, в залежності від операційної систем встановленої на телефоні.

Через втрату персональних даних ми ставите себе в небезпечну ситуацію, оскільки зловмисники які отримали ваші персональні дані можуть використовувати їх для різних цілей. В більшості випадків це робиться для власної наживи злодія або шантажу отриманими даними, або просто для забруднення репутації персони, в більшості випадків звичайно з відомих та публічних людей.

Можлива також цілеспрямована кібератака на веб-додаток чи веб-сайт, для отримання даних які там розміщені. Найбільша кількість таких атак направлені на державні установи, транспорті управління, компанії в сфері послуг й звичайно ж фінансові компанії та банки, оскільки там міститься найбільша кількість персональних даних. Інша ж частина атак направлена на ІТ-компанії з цілю отримати контроль над додатком та даних його користувачів.

Ми все частіше використовуємо телефон для зберігання інформації, оскільки він компактний, і з часом стає все потужніше, тому потрібно задуматися над захистом даних які містяться на вашому телефоні. Як варіант можна встановити захист який буде складніше обійти (сканер відбитку пальців чи об’ємний сканер обличчя), та додатково, у разі все таки злому захисту, додаткові паролі на важні для вас додатки. Глянувши на статистку втрати персональних даних зображені на рисунку 1.1, можна побачити як збільшується кількість подібних випадків.

Рисунок 1.1 – Статистика втрат персональних даних

1.2. Способи блокування телефонів для їх захисту

З моменту виходу перших телефонів способи їх захисту доволі різняться, адже технології доволі просунулися вперед, і наразі телефони не уступають по потужності персональним комп'ютерам початку 2010 року. Це можна легко зрозуміти глянувши на характеристики які мають телефони.

1.2.1 Комбінація клавіш

Самим простим способом розблокування/блокування телефонів була комбінація натиснутих в певному порядку клавіш. Це було доволі давно, адже це було ще на кнопкових телефонах, якій вийшли з обігу доволі давно. Звичайно такий спосіб не є безпечним, адже в більшості випадків телефон сам нагадував в якому порядку потрібно натиснути клавіші. Даній спосіб можна порівняти з здвигом екрану блоку в смартфонах. Такі способи не надають безпеки адже вони ніяк не прив'язані до особи і використовувати їх зможу будь-хто. Ставити такий тип «паролю» не рекомендовано, але якщо у вас немає конфіденційної інформації на телефоні і вам важлива швидкість розблокування, то звичайно даний спосіб є найшвидшим, якщо не брати в

рахунок найновіші способи, оскільки там також розблокування відбувається за долі секунд.

1.2.2 PIN-код та паролі

Далі по складності та ступеню захисту йде PIN-код. Він являє собою набір цифр які користувач задає собі сам, для легко запам’ятовування та використання в майбутньому. В більшості випадків являє собою набір саме з 4 цифр, в деяких випадках 6. Кількість всіх можливих варіантів можна побачити на формулах 1 та 2 відповідно.

$$A_{10}^{4} = \frac{10!}{(10-4)!} = 5040 \quad (1)$$

$$A_{10}^{6} = \frac{10!}{(10-6)!} = 151200 \quad (2)$$

Однак, на відміну від попереднього способу, тепер у разі вводу PIN-коду певну кількість разів не правильно, то телефон буде заблоковано. Для того щоб зняти цю блокіровку прийдеться прибігти до одного з способів які будуть запропоновані, адже існує ймовірність що ви просто забули код. У таких випадках у телефонах в залежності від ОС у вас є аккаунт чи Google чи Apple. Наступним є простий пароль який може складатись й з букв, однак оскільки більшість людей використовують слова які легко запам’ятати, та які асоціюються з ними, а не хаотичний набір букв та цифр, то він не є дуже ефективнішим за PIN-коди.

1.2.3 Графічний ключ

Далі йде так званий графічний ключ. Він являє собою пароль для розблокування телефона заданий у вигляді зображення. Для того щоб зняти блокування з екрану та отримати доступ до інформації необхідно повторити зображення яке було задано при встановленні даного способу блокування. Він може забезпечити більш високий рівень захисту чим паролі чи PIN-коди. Графічний ключ був представлений на андроїд в 2008 році. Однак тут також є

свої недоліки, оскільки більшість людей при його зображенні не використовують всі можливі 9 точок (рисунок 1.2). але в цьому й свої переваги перед простим паролем чи PIN-кодом, в плані початкової точки і подальших направлень. Максимальна кількість всіх можливих комбінацій рівна 389112, що більше майже в сімдесят сім разів за простий PIN-код та майже в два з половиною рази за PIN-код який складається з шести цифр. Обрахувати це можливо додавши всі можливі варіації вводу паролю, а саме це повинно бути не менше 4 та не більше 9.

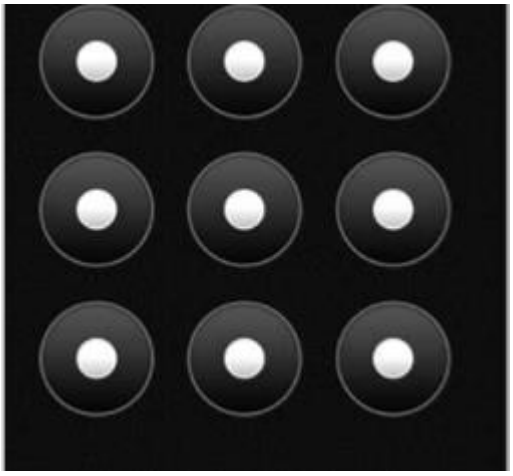

Рисунок 1.2 – графічний ключ

1.2.4 Сканер відбитку пальців

Наступний по ступені захисту являється так званий Touch ID, але оскільки це придуманий Apple алгоритм і бренд то в інших моделях телефону він називається просто, сканер відбитків пальців. Вперше був представлений в телефонах Apple в 2013 році. Потім інші компанії побачивши цю ідею зробили свої аналогії які були трохи гірше в плані швидкості.

Сканер відбитку пальців являється самим популярним біометричним замком, якщо брати загальну статистику не тільки телефонів. Даний метод являє собою датчик який розпізнає зображення відпечатка який був заданий, при створені даного типу паролю. Даний датчик сканує подушечку пальця при прикладанні її то сканера, використовуючи розширення 550 ppi (розмір

одного пікселя 50 мкм), після чого розпізнає зображення навіть під різним кутом. Дана технологія запам’ятовує зображення навіть з рельєфом.

Зберігається таке зображення в зашифрованому вигляді в спеціальному чіпі, доступ до якого немає ніхто, і при необхідності розпізнати відбиток розшифровується і всі процеси відбуваються в операційній пам’яті пристрою. Розблокування відбувається майже миттєво, інколи він може не спрацювати з першого разу, але таки випадки велика рідкість. Зламати даний захист є доволі складним, оскільки в результаті розвитку технології сучасні пристрої можуть сканувати такі дані як пульс чи піт на кінчиках пальців.

В сучасних версіях Android зустрічається сканер відбитку пальців під екраном (рисунок 1.3), зокрема це зроблена для збільшення розширення екрану відносно рамок телефону, однак в даного способу також є недоліки. Ці недоліки є дуже прості й випливають вони саме з місця розташування сканеру, а саме під екраном. В результаті такого розташування є ймовірність не спрацювання сканеру, викликати це можуть різні чинники. Найбільш поширена проблема – це захисний екран, що клеїться поверх екрану та може заважати скануванню. Також були випадки, коли сканер працював через три або чотири слоїв такої плівки, що також не може рахуватися нормальним. І доволі рідкісні випадки коли сканер не працював через наявність жиру на відбитку. Однак ця технологія нова, бо була представлена в 2019 році, тому наявність неточності не є критичним, вони можуть бути вирішені з часом.

Рисунок 1.3 – сканер відбитку пальців під екраном

Вчені стверджують, що шанс ідентичного відбитку пальців у людей є один до 64 мільярдів, тобто не існує однакових відбитків. Однак доказати це не можливо, в наслідок не достатнього матеріалу для досліджень, не можливо взяти відбиток пальця у кожної з сімох мільярдів людей.

Для підтримання даної функції потрібно також встановити пароль або графічний ключ, який потрібно вводи раз в 72 години, щоб в разі якщо сканер не спрацює можливе було розблокування за допомогою паролю чи графічного ключа, а також для додаткового захисту.

1.2.5 Об’ємний сканер обличчя

Найновіша технологія яка була представлена на ринку являє собою Face ID, та всі її аналоги які були після представлення його на публіці, знову ж таки компанією Apple. Дана технологія була представлена на презентації нового смартфону від компанії Apple в 2017 році. До представлення даної технології також існували технології на подобі цієї, але вони не враховували об’ємність обличчя, а працювали скоріше як розпізнавання фото користувача, що на сьогоднішній день може зробити велика кількість нейронних систем.

Перевага технології представленої компаніє Apple також була саме запам’ятовування, саме об’ємної моделі обличчя, тобто ви могли міняти зачіску, головний убір, чи надягати окуляри і пристрій міг вас розпізнати. Інші ж алгоритми мали більше недоліків, перед алгоритмом компанії Apple, вони могли розпізнавати надруковані 3D обличчя власників, а деякі і зовсім як і раніше розблокування фото обличчя користувача пристрою.

Дана технологія представляє собою інфрачервоний сканер, який сканує обличчя (рисунок 1.4) за допомогою проекції більше ніж тридцяти тисяч точок й зберігає дані в процесорі системи, і при повторному розблокуванні бере інформацію звідти. За заявами компанії в процесорі зберігається зашифрована математична модель, а не зображення. Технологія також навчається, вона запам’ятовує зміну в обличчі за допомогою нейронних

мереж та зберігає їх також в процесорі системи.

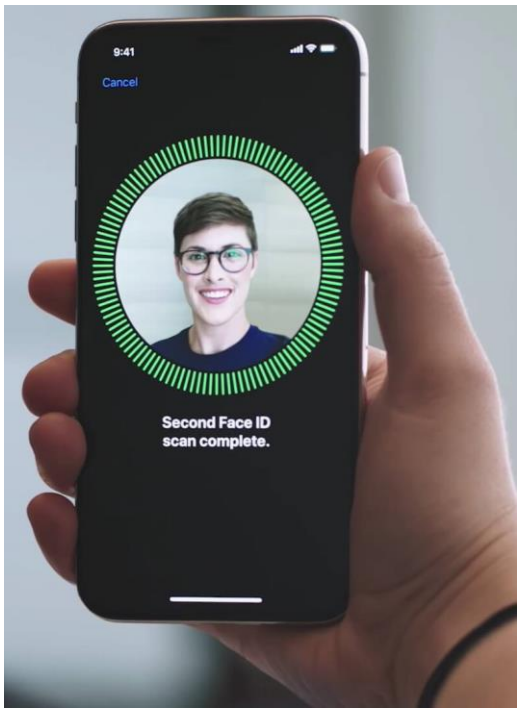

Рисунок 1.4 – сканер обличчя Face ID

Дана технологія також потребує вторинної ступені захисту по типу паролю чи графічного ключа, такий додатковий тип захисту також необхідний для підтвердження особистості у разі зміни іміджу, тобто зачіски чи макіяжу і таке подібне. Шанс того, що інша людина зможе розблокувати ваш телефон, за заявами компанії, один до мільйона, що набагато менше ніж шанс співпадіння відбитку пальців, але втричі більше за кількість можливих варіантів вводу графічного ключа. Ситуація з іншими брендами по типу Samsung, який рахується прямим конкурентом, є гірше нікуди оскільки, як вже й писалось, така ж технологія на пристроях Samsung розпізнає власника навіть по фото чи відео з ним та не взагалі не розпізнає його при зміні іміджу. Тому таку технологію можна рахувати просто баловством, оскільки захист вона надає гірший ніж сканер пальців, але дана технологія появилась в наслідок розвитку технологій, а саме стремлінням зробити максимально великий екран при максимально малих рамках.

1.3 Аналіз існуючих рішень по захисту даних

Серед існуючих рішень для захисту даних на телефоні можна виділити декілька типів, зокрема: антивіруси (програми спрямовані на захист всього телефону, від вірусів та крадіжки персональних даних), менеджери паролів (тобто програми які зберігають всі ваші паролі, для різних соціальних мереж та такі подобні), шифрування даних й останній тип це додатки які надають додатковий тип захисту, для заданого додатку, який можна вибрати в налаштуваннях.

Найбільш поширені антивірусами є: «антивірус Dr.Web Light», антивірус Avast, антивірус AVG, Kaspersky Internet Security: антивірус і захист, Avira Security. Всі дані додатки пропонують вам захист персональних даних та захист від вірусів, оскільки Android має відкритий код то написання вірус не є великою проблемою, але як і всі інші безплатні додатки за безплатно вам дають не повну версію продукту і за певну суму грошей вам дають повну можливість даного антивірусу, в більшості випадків це додатковий захист від завантажених з інтернету файлів. Також дані програми надають змогу сканування пристрою да виділення кеш файлів з пристрою які займають лишню пам’ять. З інтерфейсом з деяких з них можна ознайомитися на рисунках наведених нижче (рисунок 1.5 та рисунок 1.6). Також деякі антивіруси можуть надавати вам захист від рекламних додатків які можуть нашкодити вашому мобільному пристрою. Деякі антивіруси, серед представлених більшість, надають так звані проти воровські можливості, наприклад, відслідковування телефону або, як Avast, при зміні Sim-карти перехвачувати фото та аудіо. Майже всі антивіруси надають можливість прискорити роботу телефону за рахунок закриття непотрібних задач в процесорі, за рахунок цього також можливо зменшення температури. У деяких є функція вбудованого, так званого, чорного списку, для блокування небажаних повідомлень чи викликів, у новіших моделей телефонів така функція є за замовчуванням. Але знову ж таки більшість з таких

можливостей надаються користувачу саме після оформлення підписки, тому розраховувати на їх відразу не завжди буде правильно.

Рисунок 1.5

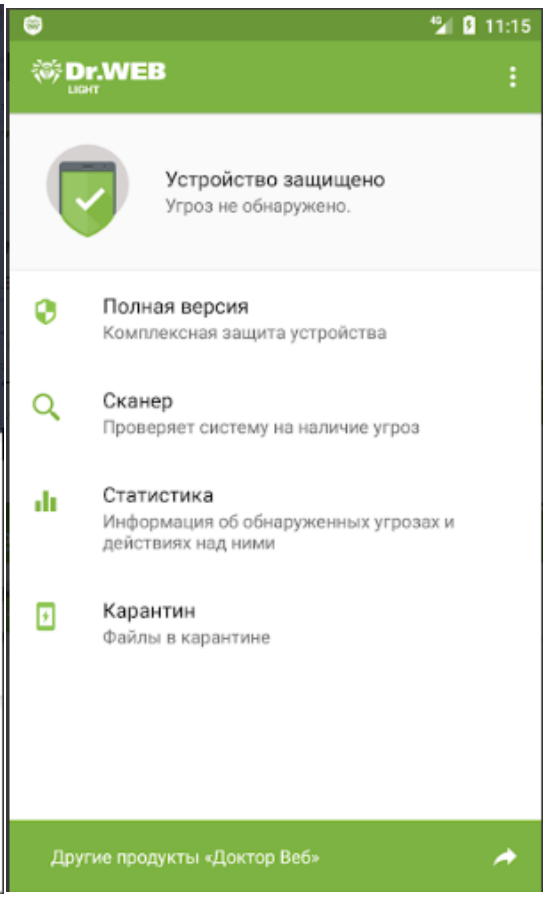

Рисунок 1.6

Найбільш поширені додатки серед другого типу є: SafeInCloud, LastPass Password Maneger, My Password, SecureX. Avast паролі –менеджер паролів. Дані додатки представляють собою сховище всіх вашів паролів і логінів до додатків та соціальних мереж чи інших ваших аккаунтів. Деякі з представлених додатків також надають можливість входити до всіх додатків за допомогою відбику пальця, що на мою думку не є дуже безпечним, оскільки якщо у телефона був захист за допогою відубитку пальця, то в результаті вдачного взлому зловмисники застосують вже раніше спрацьований спосіб для того, щоб зайти в додаток пароль від якого є в менеджеру паролів.

Шифрування даних – це доволі розповсюджена річ, оскільки, на

сьогоднішній день, захист даних при їх передачі є основним способом їх захисту. Шифрування даних відбувається за рахунок перетворення вже існуючих даних в, так званий, шифротекст, що робить прочитання даних неможливим, якщо особа не має ключів шифру. Шифрування на рівні стовпців БД, на відміну від інших способів шифрування, даний метод шифрує кожен стовпець окремо та для кожного зашифрованого стовпця додає свій ключ, що надає додатку гнучкість при шифруванні даних. Також до кожного ключа додається пароль для запобігання автоматичної розшифровки, однак це ускладняє роботу з БД. Серед способів шифрування можна виділити два типа, а саме:

- **Симетричне шифрування** (рисунок 1.7) – являється найстарішим й відповідно найвідомішим методом. Головна ідея даного методу розміщення в базі данних закритого ключа, який потім і використовується для шифровки та дешифровки. Перевага даного метода швидкодія
- **Асиметричне шифрування** (рисунок 1.8) – на відміну від симетричного шифрування має два пов’язані між собою ключа. Один ключ є відкритим та може передаватися, навіть, по не захищеному каналу зв’язку, він потрібен для шифровки даних. Другий же ключ є закритим та ідивідуальний для кожного користувача. Первагою є більша надійність на відміну від симетричного, але втрата в часі.

Розглянемо метод шифрування даних в додатках. Шифрування додатком здійснюється при зміні даних самим додатком, тобто перед записом в базу даних. Цей спосіб має більшу гнучкість, оскільки додатку відомі ролі та права доступу користувачів, а також інформації про конфеденційні дані. Перевагою такого способу є те, що необхідність у використанні додаткового рішення для захисту даних, що передаються по каналам відкритого зв’язку,

не є необхідим, так як вони передаються вже зашифрованими додатком. Недоліків у використанні даного методу також чимало, основною проблемою може стати реалізація шифрування, тому що для її повноцінної роботи необхідно внести зміни не тільки в додаток, а й в базу даних. Проблемою також може бути одночасне використання кількома програмами бази даних, в насалідок чого ключі зберігаються у багатьох місцях, що може призвести до крадіжки інформації. Ну і для зміни ключа необхідно провести розшифрову даних, а потім знову їх шифрувати, але вже з новим ключем. Така операція по зміні ключа є дуже довгою та не зручною.

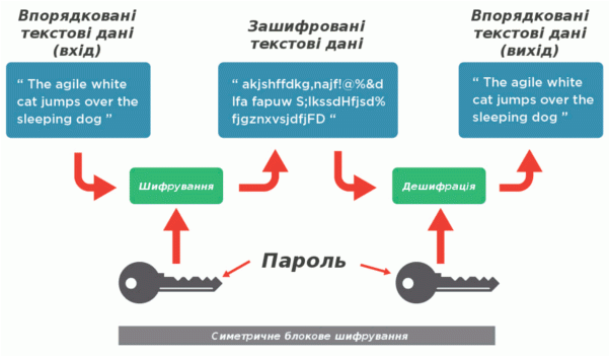

Рисунок 1.7

Рисунок 1.8

Й остання група додатків направлена на захист даних являє собою додаток який встановлює попердню аутинефікацію, перед запуском того чи іншого додатку. Майже всі додатки даного типу мають однотипну назву, а саме AppLock, що відображує їх основну ідею, а саме блокування додатків. Майже всі з них пропонує лише один спосіб блокування додатків, а саме пароль чи PIN-код, і деякі також графічний ключ та/або відбиток пальця. Більшість з них являє собою блокування лише галареї (фото та відео). Деякі додатки, в основному додатки банків, вже мають додатковий захист, оскільки вони мають конфідейну інформацію про користувача, тому такий захист є просто необхідним.

1.4. Постановка задачі

Головна мета дипломного проєкту – створити додаток за допомогою якого, можливо будете додатково, окрім оснвих способів перерахованих вище захистити свій пристрій.

Проаналізуваши всі додатки які можуть надати дану можливість, я прийшов до висновку що буде доцільно створини додаток, за допомогою якого можливо буде можливим додати додатковий рівень захисту додаткам на телефоні. Проаналізувавши всі вище приведені в приклад додатки було віділено основні програмні вимоги для реалізації цього проєкту.

Ключовими вимогами є:

- Надійність збереження інформації
- Простота в використанні
- Швидкість роботи програми
- Простий та зрозумілий інтерфейс

Однак основною перевагою, на мою думку, даного проєкту являється, відстуність реклами, коли в інших програграмах можлива функції відключення її по принципу місячної підписки. Для когось даний фактор може стани вирішальним, оскільки не всім подобається постійно вимикати реклами в ручну, та платити гроші, за те щоб її не було, немає бажання чи фінансів.

Висновки до розділу

Проблема захисту інформації є одною з передових на сьогодні, оскільки у зловмисників які бажають отримати персональні дані та зловживати ними по своєму бажанню, які були описанні вище. Способів за допомогою яких можливо крадіжка персональних даних все більше, оскільки прогрес не стоїть на місці.

Наразі існує багато додатків направлених на збереження інформації, які використовують більшість користувачів мобільних телефонів. Дані додатку в певній мірі забезпечують захист особистих даних користувачів. Також існує різноманіття можливостей блокування телефону, для його захисту від інших. Більшість людей находить додаток чи спосіб який підходить саме ним.

Серед наведених прикладів можна звернути увагу на антивіруси, оскільки вони дають найбільш можливий захист від вірусів, та на додатки попередньої ідентифікації перед запуском додатків, оскільки вони надають додатковий захист перед допуском до ваших персональних даних.

Проте є велика кількість користувачів які не можуть найти додаток який підходить саме ним, через ті чи інші причини. Такими причинами можуть стати:

- Присутність реклами в додатках
- Розблокування повних можливостей додатка лише після внесення коштів у сам додаток
- Складний у використанні інтерфейс

2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Платформи для розробки додатків на телефони

Оскільки додаток буде спеціалізований для телефону, то потрібно підібрати платформу яка надає можливість створити дані додатки. Найбільш поширені платформи спрямовані на створення додатків є: Xamarin, jQuery Mobile, Android Studio, Como, Appery.io, Bizness Apps. Dojo Mobile, GameSalad, AppMachine, AppMakr. Тепер проаналізуємо дані платформи для вибору однієї, яка буде підходити саме мені.

Xamarin – американська компанія яка спеціалізується на розробці програмного забезпечення. Дана платформа є конструктором, і надає змогу будувати програму, а для написання програми потрібно пізнання в C#. Виходячи з даної умови, платформа мені не підходить оскільки мої пізнання в C# не є достатніми для написання повноцінної програми.

jQuery Mobile – сенсорно-орієнтований веб фреймворк. Розробка додатків сфокусована на кроссплатформені браузери й адаптаційні додатки з інших платформ. Для написання програми використовується JavaScript та CSS та HTML. Дана платформа не підходить для написання проєкту, оскільки метою даного проєкту не являється написання веб-браузера.

Платформа Como націлена на написання вузькоспрямованних додатків для бізнесу, здебільшого в сфері громадського харчування та розваг. По функціональності вона є середньою, оскільки основні додатки написані на даній платформі є додатки для касових апаратів, тому можна відразу сказати що дана платформа нам не підходить.

Appery.io являє собою візуальний онлайн додаток на веб ресурсі, на якому користувачі самостійно створюють додаток та проводять тести. Для даного випадку не потрібні навички в програмуванні, оскільки все що від вас необхідно для даної платформи – це завантажувати фото та відео матеріалів

для подальшого їх використання. Цей веб ресурс також надає зворотній зв’язок по створеним вашим проєктам. Ця платформа підходить людям які мають бажання створювати власні додатки для телефонів, але з певних причин не мають часу на вивчення мов програмування. Саме тому даний ресурс нам не підходить оскільки ми не зможемо створити запланований додаток.

Bizness apps виходячи з назви платформа для створення додатків для бізнесу. Вона має в собі готові шаблони для створення нових додатків, однак ці шаблони є з малих та вже добре опрацьованих сегментів бізнесу, тому додатки іншого спрямування не можливо сконструювати на даній платформі, тому вона нам не підходить.

Dojo Mobile це пакет пропонує ряд віджетів, які можна використовувати для створення веб-додатків для мобільних пристроїв, таких як iPhone, Android, BlackBerry або Windows Phone 8. Ці віджети найкраще працюють із браузерами на базі веб-програм, наприклад Safari чи Chrome або з IE10, оскільки функції CSS3 широко використовуються. Однак віджети повинні працювати «витонченою деградацією» навіть у браузерах, що не мають CSS3, таких як IE (до IE10) або (старший) Firefox. У цьому випадку модні ефекти, такі як анімація, колір градієнта або прямокутник прямокутника, можуть не працювати, але ви все одно можете керувати вашою програмою. Крім того, як окремий файл доступний модуль сумісності, dojox / mobile / compat, який імітує деякі функції CSS3, використовувані в цьому модулі. Dojo Mobile розроблений таким чином, щоб бути максимально легким для досягнення кращого користувальницького досвіду. Знову дана платформа нам не підходить, оскільки вона спрямована лише створення веб ресурсів.

GameSalad – вже виходячи з назви можна сказати, що дана платформа спрямована на створення саме ігор. Даний додаток був розроблений компанією Gendai Games. Додаток оснований на принципі Drag-and-Drop спеціально для людей, що не знають мов програмування. Додаток має повістю зрозумілий та інтуїтивний інтерфейс, завдяки чому його можуть використовувати й прості люди, а не лише ігрові дизайнери. Виходячи з направленості даного додатка він нам не підходить, тому що нам не потрібно створювати гру.

AppMachine також являє собою конструктор з готовими шаблонами для легшого та швидшого створення додатків. Даний конструктор являє собою 35 готових налагоджених блоків, які при їх об'єднанні надають змогу створювати додатки. Дана платформа нам також не підходить оскільки нам потрібно створити додаток для захисту інформації.

AppMakr – веб ресурс який надає змогу конструювати власні додатки. Платформа AppMakr використовує інтеграцію RSS та API для живлення вмісту в додатку творця. Цей вміст може включати текст, зображення, аудіо та відео. Творці також можуть додавати RSS-канали GeoRSS та медіа-файлів, що знають місцеположення. Крім того, платформа дозволяє використовувати додатки HTML5 та CSS3, які обслуговуються локально з мобільного пристрою і можуть отримати доступ до своєї камери, акселерометра та інших вбудованих функцій. Інші функції платформи включають push-сповіщення, нативні фотогалереї, спільні календарі, чати в реальному часі та інтеграцію мобільної рекламної мережі.

Android Studio – інтегроване середовище розробки (IDE) для платформи Android, представлене 16 травня 2013 року. Дане середовище підходить нам більше всього оскільки в повній мірі надає змогу самостійно

написати додаток, на одній з запропонованих мов, що підтримуються, а саме Java, Kotlin та C++.

2.2. Середовище Android Studio

Android Studio прийшло на зміну плагіну ADT для платформи Eclipse. Середовище побудоване на базі вихідного коду продукту IntelliJ IDEA Community Edition, що розвивається компанією JetBrains. Середовище надає засоби для розробки застосунків не тільки для смартфонів і планшетів, але і для пристроїв на базі Android Wear, телевізорів (Android TV), окулярів Google Glass і автомобільних інформаційно-розважальних систем (Android Auto). Для застосунків, спочатку розроблених з використанням Eclipse і ADT Plugin, підготовлений інструмент для автоматичного імпорту існуючого проєкту в Android Studio. Android Studio підтримує написання коду на таких мовах як: Java, C++ та Kotlin.

Одним з найбільших переваг даної платформи є емулятор, розглянемо його детальніше. За домогою даного емулятору можна створити не лише емуляцію телефону, а й інших платформ з операційною системою Android. Ступень створення зразку є доволі вислокий, якщо не брати в рахунок можливі девайси для емуляції, оскільки всі вони розроблені компанією Google. При створені емуляції телефону можна вибрати девайс та версію Android, на які буде праюцвати телефон. Нас цікавить саме емуляція телефону тому розглянемо саме її (рисунок 3.1). Панель керування яка знаходиться зправа можна розбити на 8 розділів, розглянему функції які виконують кожна з них. Під цифрою один знаходиться, так звана, кнопка power, яка відповідає за включення емулятора та його перезавантаження, під цифрою 2 та 3 знаходяться кнопки громкості, працюють вони як і в звичайному телефоні, під цифрою чотири розміщені клавіші повороту екрану, нажавши на кнопку під номером п’ять буде зроблений скріншот та збережений на робочому столі, кнопка під цифрою 6 відповідає за

збільшення зображення, під цифрами сім знаходяться кнопки керування системою, нажавши цифру вісім можна відрити меня для детальнішого налаштування емулятору в тому числі і тим мережі, і останнім елементом є сам екран телефону, розмір якого налаштовується при створені емулятору.

Рисунок 2.1

Середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки застосунків для платформи Android. У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проєктування

застосунків, 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 застосунків, заснована на складальному інструментарії Gradle і підтримуюча використання засобів безперервної інтеграції.

Набір інструментів Gradle побудована на принципах Apache Ant Apache Maven, але представлена на мовах Groovy та Kotlin. Даний набір інструментів визначає які компоненти дерева збору не мінялися, і які задачі залежні від цих частин не потребують перезапуску. Для додатків які складаються з великої кількості модулів є режим мультипроєкту. Опис даної структури відбувається в setting.grable в коронній директорії проєкту. У кореневій папці також знаходиться файл в якому описані налаштування для всього проєкту для вкладених модулів. Кожен модуль програми також може мати такі файли де буде на налаштований сам модуль та дочірні модулі. Необхідний він також для збору андроїд файлів, за допомогою яких можливо потім встановити створений додаток на телефон. Перевагою даного інструмента збору додатка є його незалежність від Android Studio, тобто ви можете збирати проєкт і за допомогою командного рядка де студія не встановлена, при цьому на вихід програми це ніяк не вплине. Розглянемо процес збірки проєкту для цього глянемо на рисунок 2.2.

Рисунок 2.2 – схема роботи grable

Для прискорення розробки застосунків представлена колекція типових елементів інтерфейсу і візуальний редактор для їхнього компонування, що надає зручний попередній перегляд різних станів інтерфейсу застосунку (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.

До складу також включені пристосовані під особливості платформи Android розширені інструменти рефакторингу, перевірки сумісності з минулими випусками, виявлення проблем з продуктивністю, моніторингу споживання пам’яті та оцінки зручності використання (рисунок 2.2). У редактор доданий режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API.

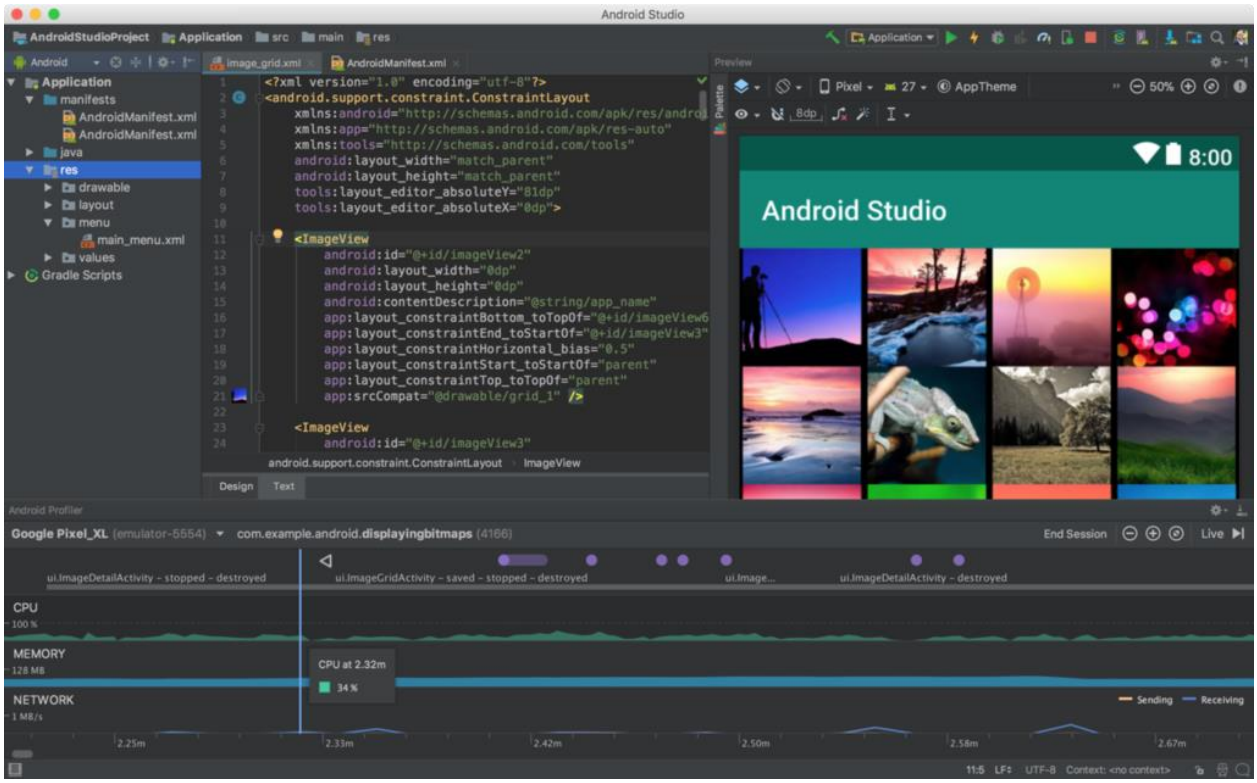

Рисунок 2.3 – інтерфейс Android Studio та відстеження продуктивності

Проаналізувавши всі вище вказані переваги можна виділити наступні особливості, які допоможуть при написані програми:

- Живі макети (layout): редагувальник WYSIWYG — живе кодування — подання програми в реальному часі.
- Консоль розробника: підказки по оптимізації, допомога по перекладу.
- Android-орієнтований рефакторинг та швидкі виправлення.

- Lint утиліти для охоплення продуктивності, сумісності версій та інших проблем.
- Багатий редактор макетів (layouts) що дозволяє користувачам перетягнути і покласти (drag-and-drop) компоненти користувацького інтерфейсу, як варіант, переглянути одночасно макети (layouts) на різних конфігураціях екранів.

При роботі з таким насиченим на функціонал середовищем необхідно ознайомитися з системними вимогами (таблиця 2.1), адже не всі комп'ютери можуть витримати навантаження й потужності як необхідні даній програмі.

Таблиця 2.1 – системні вимоги до Android Studio

OS	**Windows**	**OS X/macOS**	**Linux**
Версія OS	Microsoft Windows 10/8/7 (32- чи 64-bit)	Mac OS X 10.9.5 або вище, аж до 10.11.6 (El Capitan) чи 10.14 (Mojave)	графічне середовище GNOME чи KDE
RAM	3 GB RAM мінімум, 8 GB RAM рекомендовано		
Disk space	500 MB простору для Android Studio, щонайменше 1.5 GB для SDK, зображень емульованих систем, та кешів		
Версія Java	Java Development Kit (JDK) 8		
Роздільність дисплею	1280x800 мінімум		

2.3. Вибір мови програмування

Оскільки платформа Android Studio надає можливість писати програму на одному з трьох, підтримуваних, мов програмування, то потрібно вибрати одну з них для подальшого використання, під час написання проєкту. Розглянемо всі представлені мови й зупинимося на одній з них після аналізу.

2.3.1 Мова C++

Для початку розглянемо мову програмування з якої були запозичені деякі елементи іншими мовами, а саме C++. C++ — типізована мова програмування загального призначення. Підтримує такі парадигми програмування, як процедурне програмування, об’єктно-орієнтоване програмування, загально спрямоване програмування.

Мова має велику стандартних бібліотек, які включають в себе розкидані контейнери та алгоритми, вводу-виводу, константі вирази, підтримує роботу з паралельними потоками та інші можливості. Головним середовищем для даної мови рахується Microsoft Visual Studio, оскільки від компанії яка підтримує та вдосконалює дану мову програмування. Розвиток даної мови почався ще в 1980-ті й відбувається до сих пір, з часом з’являлися нові функції які заміняли або вдосконалювали вже існуючі. На відміну від мови C, від якої було взято основу, в даній уклон зроблений на об’єктно-орієнтоване програмування.

Синтаксис ж повністю унаслідуваний від мови C, що полегшувало перехід на дану мову, перевагою такого також була часткова підтримка проєктів написаних мовою C, при запуску їх на компіляторі призначеного для C++. Серед основних же відмінностей від мови C можна виділити:

- Поліпшена робота з написання об’єктно-орієнтованих програм за рахунок класів.
- Підтримка шаблонів.
- Додаткові типи даних (bool).
- Розширені бібліотеки з стандартними функціями.

- Посилання та більш гнучке управління пам’яттю.
- Оголошення у вигляді виразів.
- Перетворення типів завдяки функція.

2.3.2 Мова Kotlin

Kotlin – статично типізований, об’єктно-орієнтована мова програмування, що працює поверх Java Virtual Machine і розробляється компанією JetBrains. Також компілюється в JavaScript і в виконуваний код ряду платформ через інфраструктуру LLVM. Автори ставили за мету створити мову більш лаконічний, ніж Java. Наслідком спрощення стали також більш швидка компіляція і найкраща підтримка мови в IDE. Мова повністю сумісна з Java, що дозволяє java програмістам поступово перейти до використання цієї мови.

Синтаксис мови використовує елементи з Паскаля, TypeScript, Haxe, PL / SQL, F #, Go і Scala, C ++, Java, C #, Rust і D. Можна сказати, що дана мова програмування є спрощеною версію мови Java. Мова програмування Kotlin є дуже гнучкої оскільки, вона дозволяє вставки з інших мов.

Одним з головних переваг даної мови може стати його визнання компанією Google, компанії розробки ОС Андроїд, одним з факторів для того могло стати ідентичність з мовою Java, бо всі програми писалися раніше саме на ній. Через це в середовищі Android Studio, є функція автоматично переписати ваші проєкти на мову Kotlin. Програми написаної на даній мові не мають прив’язки до операційної системи, як і Java від якої пішла дана мова. Дана мова розвивається відкрито, тобто код проєкту можна переглянути в інтеренеті.

2.3.3 Мова Java

І остання мова програмування за допомогою якої можливо працювати в Android Studio – це Java. Java – строго типізований об’єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems (в подальшому

придбаної компанією Oracle). Дата офіційного представлення – 23 травня 1995 року.

Синтаксис мови був запозичений з таких мов як C та C++, в основному було взято об'єктну частину C++. В мові Java полегшено написання об'єктно-орієнтованих програм, а саме за рахунок спрощення деяких необхідних дій для цього та перекладення них на віртуальну машину.

Однією з основних переваг даної мови можна назвати незалежність від платформи написання. Програми написані мовою Java транслюються в байт-код, завдяки чому вони можуть працювати на любій комп'ютерній системі, для якої існує реалізацію, так званої, віртуальної Java-машини(JVM) – програми, яка обробляє байт-код і передає інструкції обладнанню як інтерпретатор. Перевагою подібного способу виконання програм є повна незалежність байт-коду від операційної системи і устаткування. Це можна зробити за допомогою існуючого байт-коду та віртуальної машини. Також завдяки цьому є можливим виклик вкладених підпрограми написані іншими мовами програмування.

Перевага звичайно існує і це портативність, але час який може затрачуватися на повторну інтерпретацію може бути значно більшим, ніж скомпільовані у машинний код, через що говорять, що Java «повільна» мова, однак у сучасних способах реалізації віртуальної машини цей недолік був суттєво зменшений.

Для уникнення, так званого, витіку пам'яті, у Java присутній збирач сміття, який працює під час життєвого циклу об'єкта. Після того як об'єкт стає непотрібним, відповідальна за це частина віртуальної машина звільняє пам'ять. Однак витік пам'яті все ж таки можливий, у разі якщо програміст звертається до непотрібних об'єктів, що зберігалися в пам'яті. Збирання сміття відбувається в будь-який час, але в більшості випадків сам в період не активності програми, або для звільнення пам'яті під новий об'єкт.

2.4. XML стилі

Перед початком роботи в Android Studio необхідно ознайомитися ще з розміткою стилів в XML, оскільки всі візуальні зміни в нашому додатку відбувають за рахунок зміни XML файлів.

XML (англ. eXtensible Markup Language) – розширювана мова розмітки. XML розроблявся як мова з простим формальним синтаксисом, зручний для створення, з підкресленням націленості на використання в Інтернеті. Мова називається розширюваним, оскільки ним не фіксується розмітку, яка використовується в документах: розробник вільний створити розмітку відповідно до потреб конкретної області, будучи обмеженим лише синтаксичними правилами мови. Розширення XML – це конкретна граматика, створена на базі XML і представлена словником тегів і їх атрибутів, а також набором правил, що визначають які атрибути і елементи можуть входити до складу інших елементів.

Стандарт XML визначає набір лексичних так синтаксичних правил для побудови мови опису інформації завдяки застосуванню тегів. Завдяки такому формату можливе застосування його в різних галузях. Таким чином він формує метамову на основі якої будуються стилі для різних потреба та програм. XML-документи мають ієрархію у вигляді дерева з одним кореневим елементом.

Для роботи в Android Studio необхідні лише базові пізнання в XML, оскільки замість вас програма, в більшості випадків, робить все сама. Можливе так за допомогою налаштування програми власноручно за допомогою так званої drag-and-drop функції та візуального бачення майбутнього проєкту, під час його створення (рисунок 2.2).

Окрім графічної складової файл XML файл також використовується для заповнення, так званого, AndroidManifest. Всі дані розбиті на вкладені елементи, основним є **<manifest>** де знаходяться всі інші елементи. В елементі, під назвою, **<application>** находиться дані проєкту, тобто тут

описані головні його візуальні особливості, такі як: назва додатка, тема, посилання на рисунок (який буде встановлений як значок додатка), а також запускає задекларовані програмою дії (які в свою чергу описані в елементі під назвою <**activity**>).

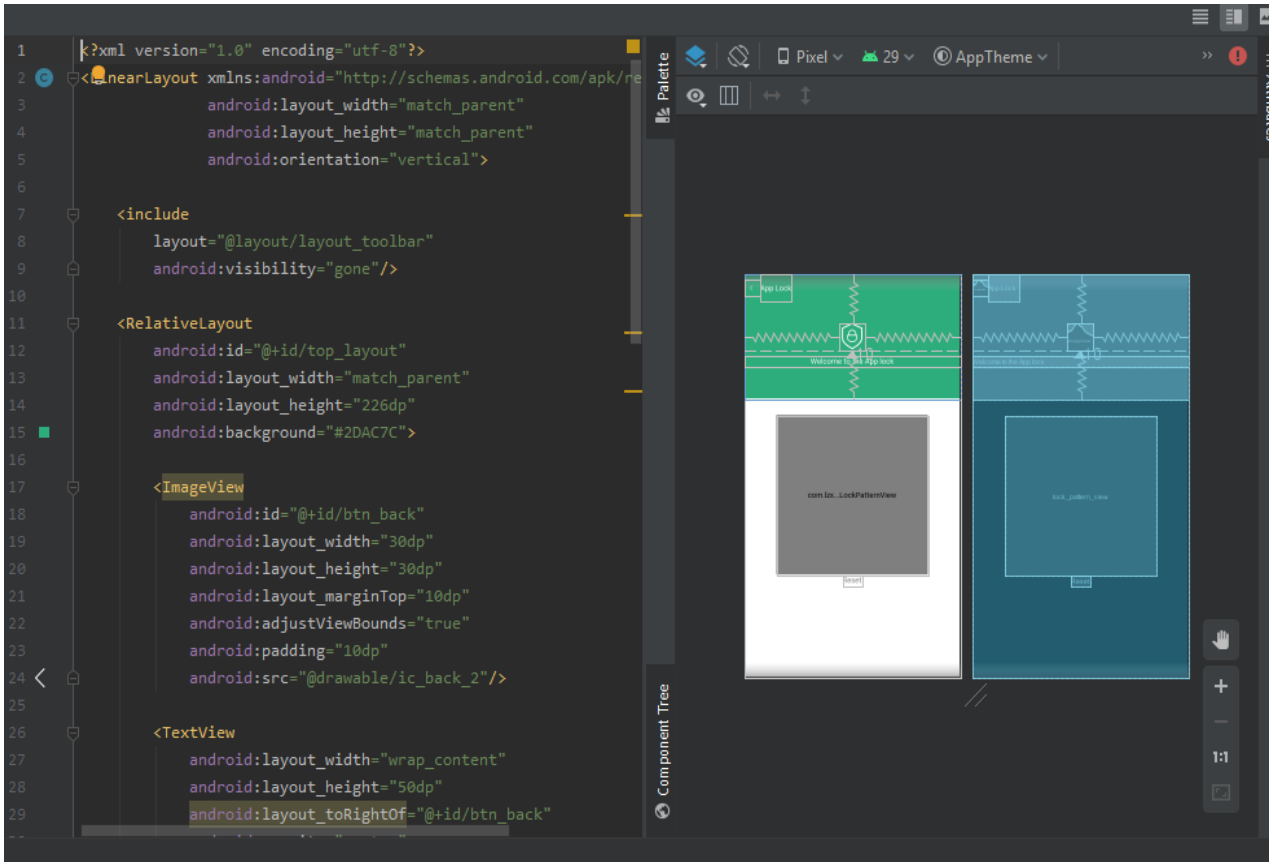

Рисунок 2.4– поєднання XML код та візуальної частини

Висновки до розділу

Проаналізувавши всі найбільш відомі платформи та ресурси, для створення додатків на телефон, можна прийти до висновку що більшість являє собою готові конструктори якими можуть користуватися як професіонали так прості люди, які бажають створювати власні додатки. Також можна прийти до висновку що більшість з них є вузькоспрямовані, в основному в сфері бізнесу. Тому після аналізу всі можливих варіантів було вирішено вибрати Android Studio, оскільки вона є середовищем розробки, яке було створено й підтримується компанією Google.

Після аналізу всіх можливих мов написання майбутнього проєкту був зроблений вибір зупинитися сам не Java. Для цього вибору було декілька причин зокрема це були: моє індивідуальне ознайомлення з даною мовою та написання інших робіт, саме цією мовою, більша швидкодія в порівняні з іншими варіантам.

Виходячи з усього перерахованого вище, було вибрано оптимальні шляхи для реалізації поставленої задачі, розроблення додатку для телефона, а саме середовище для запуску проєкту Android Studio та мова програмування Java для написання самого коду.

3. ОСОБЛИВОСТІ РОЗРОБКИ ДОДАТКУ ДЛЯ ЗАХИСТУ ДАНИХ

3.1 Розроблена архітектура програмного забезпечення

Враховуючи всі фактори, які були перераховані вище, необхідно створити додаток який задовольняв наступні потреби користувача:

- Простота використання.
- Надійний захист даних.
- Доступність.
- Швидкодія.

Перед початком роботи з додатком необхідно розробити його алгоритм роботи, з яким можна ознайомитися в додатках. Після розробки додатка який би відповідав поставленими вимогам, була складена архітектура даного додатку, яка зображена на рисунку 3.1. В даній схемі є декілька елементів, а саме «інтерфейс користувача», «блок пам’яті», «ядро системи».

Інтерфейс користувача – відповідає за взаємодію між користувачем додатка та самою системою. В інтерфейсі користувач вводить попередньо поставлений графічний ключ, для подальшого користуванням додатком, після чого він може вибирати на які додатки буде поставлений цей графічний ключ. Також він може змінити раніше поставлений графічний ключ в налаштуваннях.

Блок пам’яті – у цей блок входить пам’ять додатка в якому будуть записані додатки на які необхідно поставити графічний ключ, а також буде збережений сам графічний ключ для його подальшого порівняння при запуску інших додатків.

Ядро системи або блок процесора – відповідає за всі процеси взаємодії між інтерфейсом та блоку пам’яті, а також за розпізнавання графічного ключа при повторному його вводі.

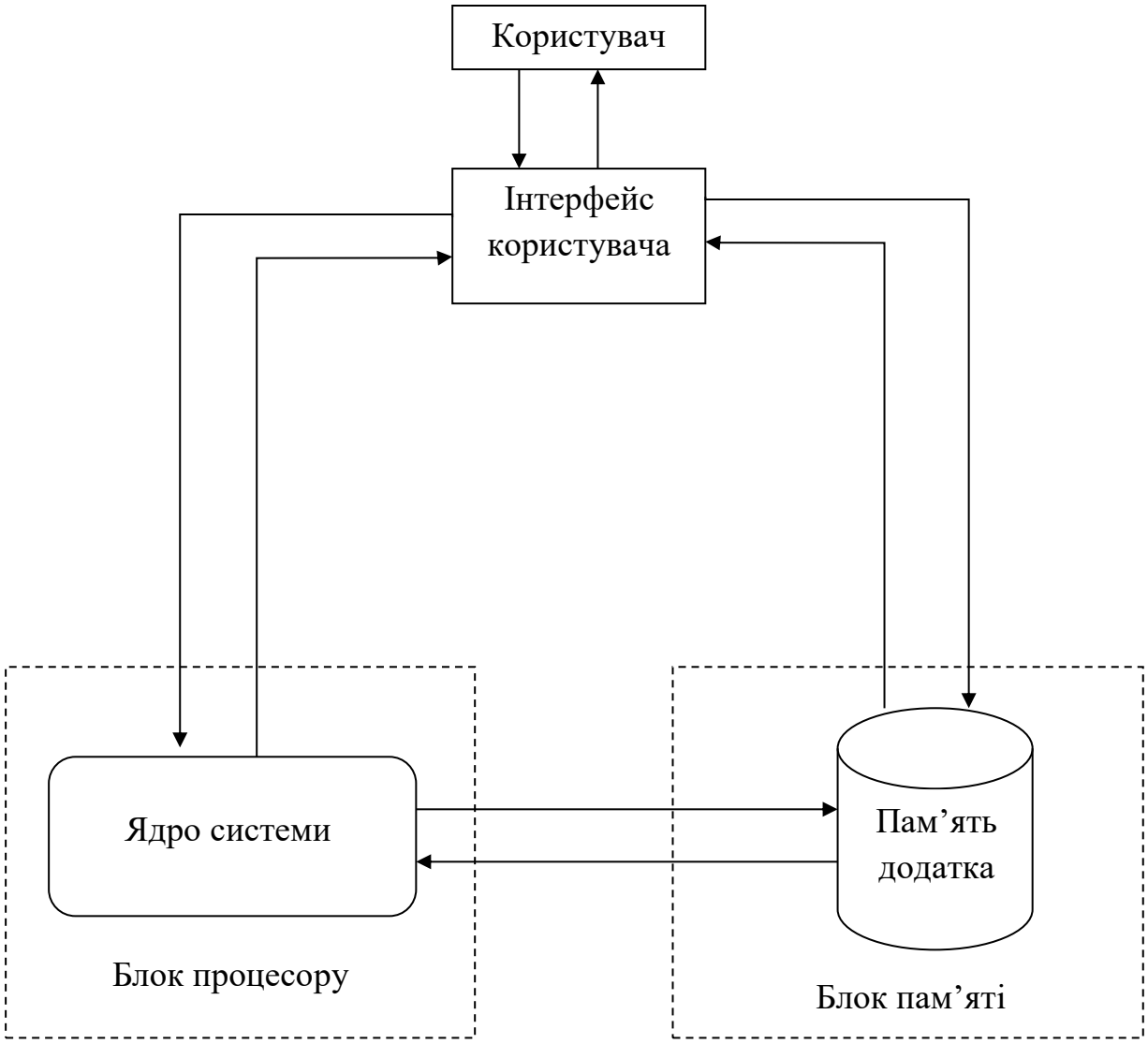

Рисунок 3.1 – архітектура додатка

3.2. Особливості написання коду програми та використані бібліотеки

При написанні програми були використані специфічні бібліотеки для виконання поставленої задачі, розглянемо їх подрібніше.

Бібліотека android.os надає основні послуги операційної системи, передачу повідомлень та міжпроцесорний зв’язок на пристрої. Дана бібліотека містить в собі множину класів, однак використовується лише клас Bundel. Даний клас необхідний для збереження різних значень string, окрім

цього вона забезпечує доступ до відліку часу пристрою, а також для синхронного виконання процесів.

Бібліотека android.content містить класи для доступу та публікації даних на пристрої. Вона включає три основні категорії API:

- **Content sharing** (обмін змістом) слугує для обміну інформації між компонентами додатка.
- **Package management** (управління пакетами) для доступу до інформації про Android-пакет (.apk), включаючи інформацію про його дозволи.
- **Resource management** (управління ресурсами) для отримання даних про ресурси пов’язаних з програмою.

Дана бібліотека є важливою для даного проєкту, оскільки потрібно отримати дозвіл для виконання програми в фоні. Серед класів можна виділити BroadcastReceiver, який відсилає та приймає дані або набір команд які потрібно виконати, Intent та IntentFilter являють собою набір команд та фільтр для їх структуризації.

Одною з головних бібліотек, в даному проєкту, є android.graphics, оскільки для захисту даних був вибраний графічний ключ. Дана бібліотека надає графічні інструменти низького рівня, такі як полотна, кольорові фільтри, точки та прямокутники, які дозволяють безпосередньо керувати малюванням на екрані. Класи які використовувались під час роботи це Bitmap, Point, Color, Canvas, Matrix, Paint, RectF, вже судячи з назви ,більшості з класів, можна сказати для чого необхідні дані класи, а саме для створення бітового поля, на якому потім буде зображуватися графічний ключ вибору кольору, створення матриці 3x3 задля поля вводу паролю. Бібліотека android.graphics.drawable допомагає в візуалізації необхідних елементів, однак більшість з них можна створити за допомогою стилів XML.

Наступна бібліотека це – android.widget містить в собі готові елементи інтерфейсу, переважно візуальних, які можна використовувати при створенні додатка. Також можливе створення власних віджетів для цього необхідно створити файл реалізації Java, в якому буде описано поведінку віджета, та файл визначення XML, в якому будуть записані його атрибути, які він підтримує.

Так звана бібліотка підртимки або android.support необхідна для підтримки та сумісності більш ранніх версій операційної системи android, та розширені класи з різними функціями. Можна виділити такі способи застосування для бібліотек підтримки:

- **Backward Compatibility for newer APIs** (зворотна сумісність для новіших версій API) відповідно до назви містить бібліотеки та класи та підтримки попередніх версій API.
- **Convenience and Helper Classes** (додаткові класи підтримки) додають допоміжні класи для розробки інтерфейсу.
- **Debugging and Utilities** (допоміжні утиліти) надає додаткову перевірку для коду.

Одною з необхідних бібліотек являється android.view, вона слугує для побудови інтерфейсу програми та взаємодії з користувачем. Одним з головних класів слугує клас Display, оскільки він передає інформацію про розширення екрану та його розміри, що необхідно для подальшого моделювання зовнішнього вигляду програми. Для забезпечення зовнішнього вигляду та поведінки вікон верхніх рівнів необхідний клас Window, а за взаємодію да роботу з усіма рівнями вікон необхідний клас WindowManager.

Наступна необхідна бібліотека це android.app, необхідна вона для роботи додатка в фоні, щоб він виконував свою основну функцію, тобто забезпечував додатковий рівень захисту для додатків, а також дана бібліотека необхідна для отримання інформації про додатки на пристрою, для подальшого використання даної інформації. Крім вище перерахованих

функцій дана бібліотека також слугує для допуску до оповіщень системи, а також вона необхідна для передачі команд для їх виконання.

Для плавного переходу між розділами додатка використано android.animation, яка, відповідно до назви, надає можливість додати анімації до елементів будь-якого типу. За допомогою даної бібліотеки є можливим встановити кадру або групову анімацію, для подальшого управління поведінкою анімацією. За допомогою класу Animator, який забезпечує основну підтримку анімації, можливо додавати анімацію до списку анімації для подальшого їх використання та запуску. За анімацію об'єктів відповідає клас з навою ObjectAnimator.

Щоб налаштувати ієрархію різних вкладених розділів додатка використано бібліотеку android.provider, яка також дає доступ до інформації розміщений, такій як контактна інформація, інформація календаря та медіа-файли. Класи даної бібліотеки надають спрощені метода додавання або отримання даних від цих постачальників вмісту, а клас Settings містить глобальні налаштування пристрою, навіть на рівні системи.

Для використання загальних корисних методів, таких як маніпуляції з датою та часом, конвертування чисел та букв, використовується бібліотека android.util. Вона також може відсилати звіт про роботу на ядро системи. Ще однією корисною функцією є зберігання колекції атрибутів, які пов'язані з файлом XML.

Окрім бібліотек які належать Android Studio були також використанні бібліотеки Java. Одною з таких бібліотек була – List. Він необхідний для реалізації списку та виконання всіх операцій притаманних даній бібліотеці, також дана бібліотека підтримує перемінну null. Ця бібліотека надає можливість використовувати елементи списку подібно до вектору, але на відміну від повноцінного вектору даний тип структури даних не є синхронізованим. Кожен член списку, створеного за допомогою такої бібліотеки, містить ємність – це розмір масиву який використовується для

зберігання елементів у списку. В більшості випадків величина ємності така ж велика як і розмір самого списку та при додаванні нового елементу списку його розмір автоматично збільшується. Реалізація списку за допомогою даної бібліотеки не є синхронізована. Тобто якщо допуск до списку отримують одночасно декілька потоків, і хоча б один з потоків занесе структурну зміну до списку то це може призвести до збою програми, якщо список не синхронізований, а гіршому випадку це може призвести до збою програми пізніше. Синхронізувати список можна за допомогою методу Collections#synchronizedList.

Для використання хеш-таблиць необхідно підключити бібліотеку HashMap. Дана бібліотека являє собою реалізації карти на хеш таблицях. Тобто ми записуємо дані на так звану карту, а потім додаємо до таблиць. Дана структура даних є дуже компактною та зручною у використанню, оскільки у випадку коли нам необхідно дані ми звертаємось за ключем та отримуємо всі дані розташовані за даним ключем у хеш-таблиці. При створені даної структури, без вказаних розмірів, за замовченням створиться картотека на 16 комірок. Додавання нових даних до такої структури відбувається доволі просто, а саме за допомогою команди put яка потребує додати ключ та зміст. На потужність HashMap структури впливають два фактори, а саме початкова потужність (початкова ємність) та коефіцієнт навантаження. У даному випадку ємність – це кількість комірок у хеш-таблицях, а початкова ємність, відповідно, кількість комірок які були при створенні даної структури. Коефіцієнт навантаження – це межа того скільки хеш-таблиця може отримувати даних до автоматичного збільшення її розміру. У випадку коли добуток цих двох змінних перевищує поточну ємність таблиці, то дані будуть перебудовані таким чином, щоб збільшити розмір комірок в двічі. У разі використання для однакових хеш значень різних ключів зменшить продуктивність даної структури. У випадку з хеш-таблицями, як і з списками, можливий одночасний доступ до таблиці різних

потоків, що знову ж таки може призвести до збоїв в програмі. Тому для уникнення цього використовують Collections#synchronizedMap.

Для того щоб синхронізувати доступ до структур даних перерахованих вище та виклику функції також перерахованих вище використовують клас Collection. Він містить в собі прості статичні методи які керують колекціями або повертають їх. У разі якщо методу або класу був переданий недійсна колекція чи об'єкти класу які є недійсними, вони видають NullPointerException. Алгоритми деструктора у даних класах та методах у разі спроби модифікації колекції, яку вона не підтримує, видають у відповідь unSupportedOperationException.

Для повноцінного функціонування потоків необхідна бібліотека Timer. Класи та методі даної бібліотеки слугують для планування черги майбутнього виконання операцій у фоні, це є необхідним для повноцінного функціонування нашого додатка. Кожному об'єкту таймера відповідає один потік, в якому записані всі дії які необхідно виконати. У разі якщо ж буде завдання з довгим часом виконання, то воно затримує виконання подальших задач у планувальнику. У разі завершення всіх задач у потоці він переходить у «режим збирання сміття», тобто режим сну. Через це він може відкладати завершення програми. Якщо ж користувачу необхідно терміново завершити процес то для цього існує відповідні методи скасування таймера. У випадку неочікуваного завершення виконання програми, наприклад виклик методу його зупини, то подальше планування завдань на таймері є не можливим і призведе до помилки.

Для полегшення написання коду було вирішено розбити програму на різні розділи за їх призначенням відповідно, а саме за їх функціями. В папці widget знаходяться файли які відповідають за різні функції інтерфейсу, які викликаються в XML файлах. В папці utils знаходяться реалізації деяких функцій необхідних іншим частинам коду. В каталозі db знаходяться функції для виводу інформації.

3.3 Особливості розробки інтерфейсу

Враховуючи поставлену задачу потрібно розробити простий і зрозумілий інтерфейс, для легко використання даного додатку та швидкого освоєння.

Інтерфейс програми було вирішено розділити на дві частини (рисунок 3.2). Верхня части головного екрану являє собою простий фон з рисунком щита, який символізує захист, та назвою програми. Нижня ж частина це матриця точок 3x3, яка необхідна для вводу графічного ключа, а також в даній частині інтерфейсу відбувається вивід інформації, якщо були допущені помилки при вводі ключа.

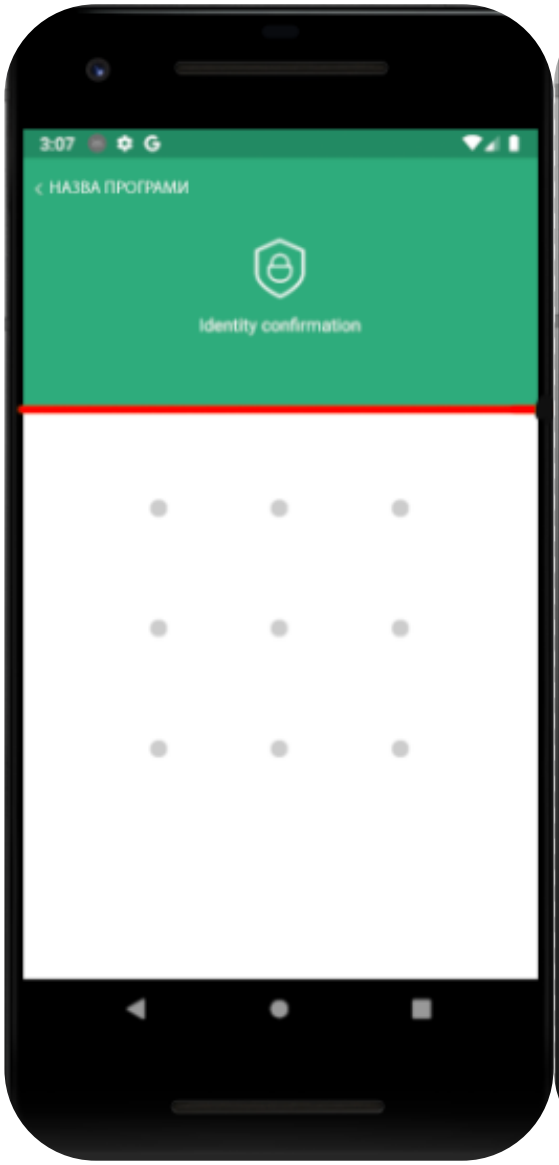

Рисунок 3.2

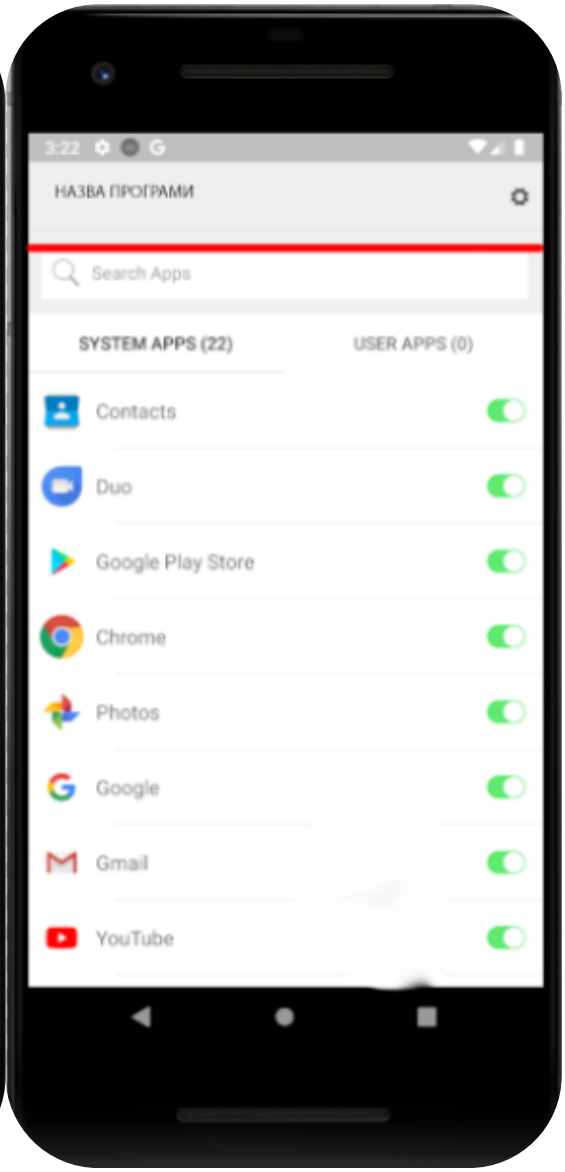

Рисунок 3.3

Після введення графічного ключа, та ввійшовши в сам додаток, потрібно було зробити меню (рисунок 3.3), де б розміщувалися всі додатки встановлені на телефон. Даний інтерфейс користувача також складається з двох частин, у верхній частині розміщені назва додатка, а також кнопки налаштувань. В нижній же частині додатку були розміщені всі додатку та панель пошуку, для випадку якщо додатків багато.

У разі використання ж програми по прямому назначені, а тобто запуску додатка з додатковим рівнем захисту, то при запуску додатку з’являється ,схоже до головного меню, вікно де необхідно ввести графічний ключ, після чого можна буде продовжити використовувати додаток.

Висновки до розділу

При виконанні даного проєкту було задіяно велику кількість бібліотек доступних на Android Studio, оскільки було необхідним створити програму з дозволом працювати в фоновому режимі та викликатися перед запуском додатків, яка б могла будувати та запам’ятовувати зображений рисунок, а саме графічний ключ, та мала доступ до пам’яті пристрою, задля навігації по доступним додаткам.

Був розроблений простий та зрозумілий інтерфейс, як і було визначено в проєкті. Даний інтерфейс не має зайвих елементів, що сприяє легшому його використанню. У разі помилок при вводі графічного паролю були зроблені відповідні повідомлення.

Всі тести проводилися на емуляторі від Android Studio, тестовою платформою слугував Pixel 2 API 28, тобто android 9.0, й розширенням екрану 1080x1920 420dpi.

4. КЕРІВНИЦТВО ПО ВИКОРИСТННЮ ДОДАТКА

Для легкого використання розробленого додатка, було розроблено керівництво для ще більшого спрощення його використання. В даному розділі буде детально описано, як використовувати даний додаток.

При першому запуску додатка ви побачите наступне зображення (див. рис.4.1). При першому запуску програми необхідно придумати та ввести графічний ключ, він повинен містити в собі не менше 4 точок, у разі якщо їх буде менше на екрані з'явиться відповідне повідомлення й попросить повторити спробу створення графічного ключа (рисунок 4.2). Після введення графічного ключа перший раз, при його створенні, необхідно його підтвердити, тобто ввести повторно.

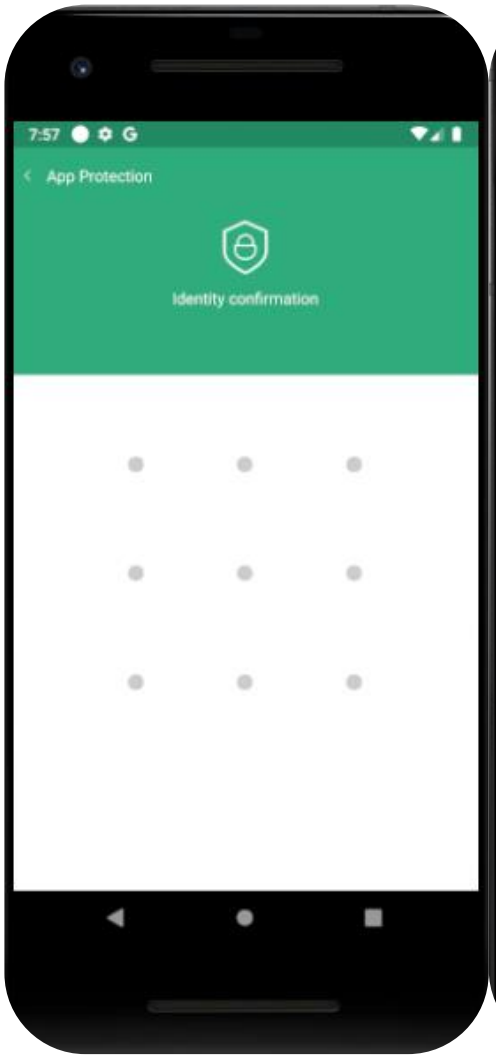

Рисунок 4.1

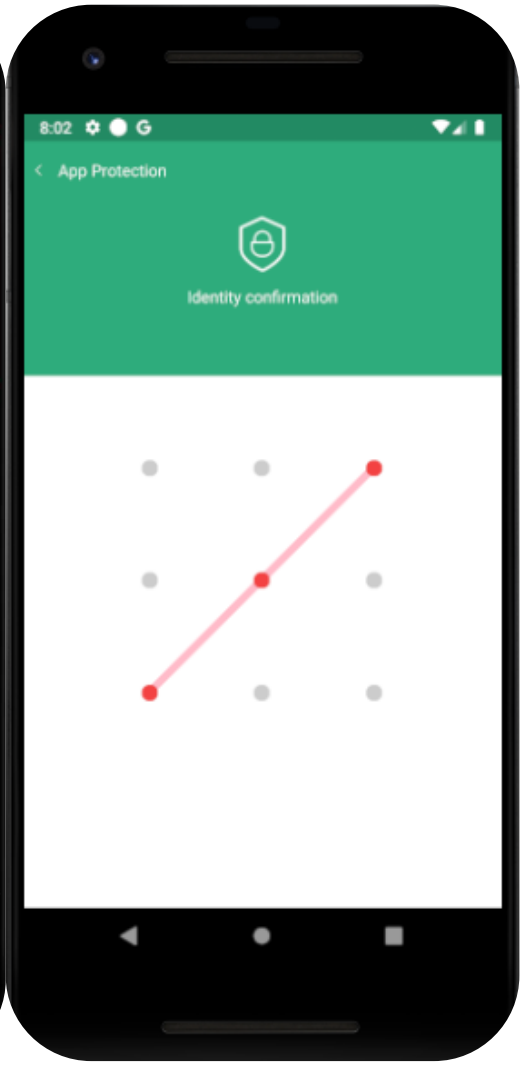

Рисунок 4.2

Після вдалого створення графічного ключа та його підтвердження перед вами постане меню (рисунок 4.3) для вибору додатків які будуть

використовувати функцію яку надає даний додаток. Розглянемо більше детальніше розділи меню які можна запримітити. Під цифрою один знаходиться розділ налаштувань (рисунок 4.4), де є наявні три функції, а саме вимкнути функцію додатку, у разі якщо користувач забажає зберегти налаштування додатків, для яких потрібно вмикати додатковий захист, а також ще дві функції, а саме змінити графічний ключ та функція вимкнення анімації при вводі графічного ключа. Під цифрою два можна побачити розділ пошуку додатків. Під функцією три ж можна побачити перелік всіх додатків, які розподілені на системні та додатки користувача.

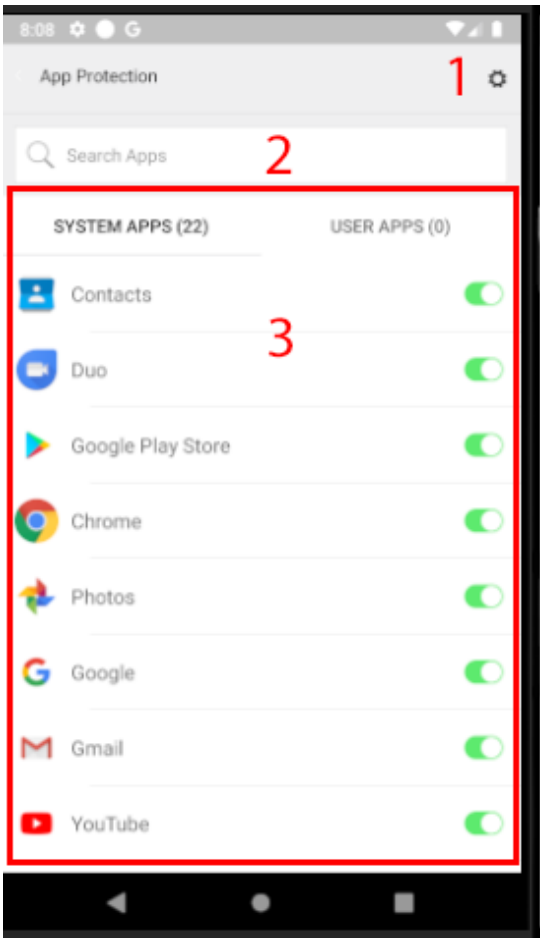

Рисунок 4.3

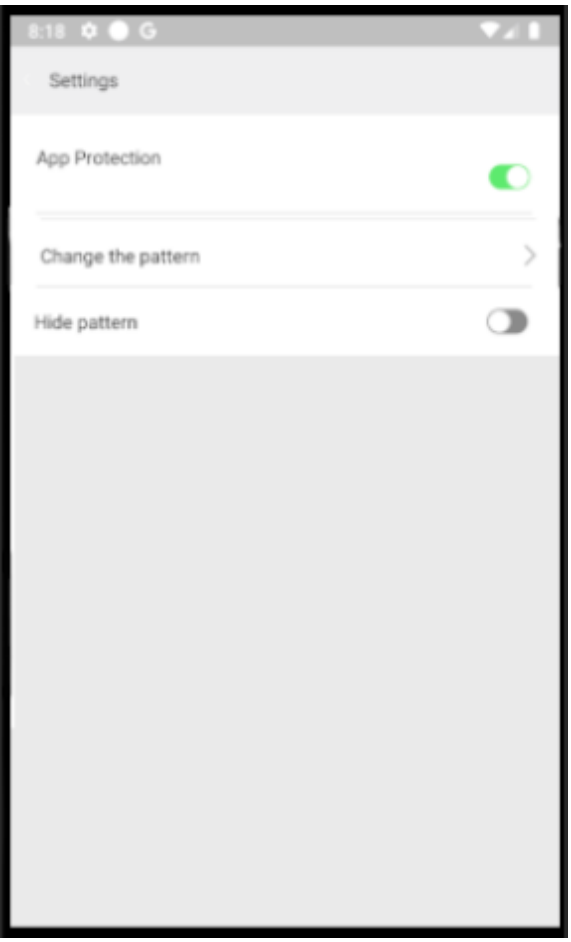

Рисунок 4.4

При включені додатків, для яких включена функція додаткового захисту, з'являється меню ідентичне до головного меню, яке продемонстроване на рисунку 4.1, але на відміну від нього замість зеленого фону там знаходиться зображення додатка який вмикається. Для отримання доступу до додатка необхідно ввести графічний пароль заданий раніше.

ВИСНОВОКИ

Розроблений додаток надає можливість додати додатковий рівень захисту для додатків на телефоні, шляхом введення графічного ключа перед запуском додатків. В результаті тестів розробленої програми не було виявлено неочікуваних збоїв в її роботі. Розроблений додаток надає змогу власноруч вибирати додатки для яких буде встановлено графічний ключ, для спрощення даної функції було розроблено строку пошуку по всім встановленим додаткам, а також можливість вибирати додатки лише з встановлених користувачем.

У разі якщо користувач забажає змінити заданий ним графічний ключ було додано вкладку налаштувань. Також в даній вкладці була добавлена функція схову введення графічного ключа, тобто анімацію вводу. І остання функція яка є в даній вкладці являє собою, вимкнення додатку, тобто не потрібно вводити графічний ключ при запуску додатків на яких він встановлений. Зроблено це з метою спрощення використання, щоб не потрібно було вимикати цю функцію для кожного додатку окремо.

Дана система може використовуватися на OS Android 9.0 та вище, а завдяки бібліотеці підтримки і на більш старіших версіях.

Переваги: доступність та легкість у використанні. На відміну від інших подібних додатків не має реклами, оскільки не є комерційним проєктом.

Недоліки: не має більш сучасніших методів блокування, таких як сканер відбитка пальців та об’ємний сканер обличчя.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Характеристика джерела	Назва видання чи посилання
Книги	
Один автор	1. Бьёрн Страуструп. Дизайн и эволюция C++ = The Design and Evolution of C++. — СПб.: Питер, 2007. — 445 с. 2. Buyya. Object-oriented Programming with Java: Essentials and Applications. — Tata McGraw-Hill Education, 2009. — 678 с.
Два автори	1. Коробейников А. Г., Гатчин Ю. А. Математические основы криптографии. Учебное пособие. — СПб : СПб ГУ ИТМО, 2004. — 106 с.
П’ять і більше авторів	1. Дэвид Хантер, Джефф Рафтер, Джо Фаусетт, Эрик ван дер Влист, и др. XML. Работа с XML, 4-е издание = Beginning XML, 4th Edition. — М.: «Диалектика», 2009. — 1344 с.
Електронні видання	
Періодичні видання	1. Андрей Бреслав. Язык программирования Kotlin // Открытые системы. — 2011. — № 09. URL: https://www.osp.ru/os/2011/09/13011550/ (дата звернення: 12.10.2018).

Характеристика джерела	Назва видання чи посилання
Сторінки з веб-сайтів	1. Відритий код мови Kotlin та коментарі авторів. URL: https://github.com/JetBrains/kotlin 2. Опис бібліотек та керівництва по написанню програми від компанії Google. URL: https://developer.android.com/reference/packages (дата звернення: 12.10.2018). 3. Статистика крадіжки персональних даних. URL: https://www.ptsecurity.com/ru-ru/research/analytics/web-application-attacks-2019/ (дата звернення: 12.10.2018). 4. Стаття про сканер відбитку пальців. URL: https://meduza.io/feature/2019/09/11/biometricheskaya-identifikatsiya-eto-tochno-nadezhno (дата звернення: 12.10.2018). 5. Опис роботи технології Face ID від компанії Apple. URL: https://support.apple.com/ru-ru/HT208108 6. Прозоре шифрування даних (TDE) URL: https://docs.microsoft.com/ru-ru/sql/relational-databases/security/encryption/transparent-data-encryption?redirectedfrom=MSDN&view=sql-server-ver15 (дата звернення: 12.10.2018). 7. Стаття про можливі злодіяння з публічним Wi-Fi. URL: https://news.finance.ua/ua/news/-/345683/chym-nebezpechni-bezkoshtovni-wi-fi-merezhi (дата звернення: 12.10.2018).