МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____ С.В. Казмірчук
«_____»_____ 20__ р.

На правах рукопису
УДК 004.056.5:510.22(043.3)

КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Програмне забезпечення захисту персональної інформації на ОС Android

Виконавець: Кобка М.В.
Науковий керівник: к.т.н., доцент Кардовський С.Є.
Нормоконтролер: к.т.н., доцент Кардовський С.Є.

Київ 2020

ЗМІСТ

Вступ Персональні дані це будь-які відомості, за якими ідентифікується або може бути ідентифікована фізична особа. Зокрема, прізвище, ім’я, по батькові, адреса, телефони, паспортні дані, національність, освіта, сімейний стан, релігійні та світоглядні переконання, стан здоров’я, матеріальний стан, дата і місце народження, місце проживання та перебування тощо. Також, це дані про особисті майнові та немайнові відносини цієї особи з іншими особами, зокрема членами сім’ї. До персональних даних відносяться також відомості про події та явища, що відбувалися або відбуваються у побутовому, інтимному, товариському, професійному, діловому та інших сферах життя особи (за винятком даних стосовно виконання повноважень особою, яка займає посаду, пов’язану із здійсненням функцій держави або органу місцевого самоврядування) тощо. І цей перелік не є вичерпним.

Актуальність В сучасному цифровому світі, де значна частина наших персональних даних зберігається в звичайному смартфоні, з’явилося таке явлення, як кіберзлочин і з кожним днем воно набуває все більшої гостроти. Втрата смартфону із всією важливою для нас інформацією, починаючи від улюблених фотографій і закінчуючи паспортними даними, може стати дуже великою проблемою. В деяких випадках, зловмиснику достатньо просто на кілька хвилин отримати доступ до вашого пристрою і дістати потрібну йому інформацію. Постає питання як можна додатково забезпечити захист персональним даним. Існує багато методів, і додаткове приховування інформації на пристрої користувача може стати хорошим і нескладним виходом із ситуації.

Метою даної дипломної роботи є розробка програмного модулю забезпечення захисту інформації на ОС Android.

Об’єкт дослідження: удосконалення методів і підходів зfхисту персональних даних користувача на ОС Android.

Предметом дослідження є: існуючі методи і підходи забезпечення захищеності персональних даних користувача.

Метод дослідження: проведені дослідження базуються на практичному порівнянні існуючих аналогів, що забезпечують захищеність персональних даних користувача та виявленні їх недоліків.

Наукова новизна даної дипломної роботи полягає в удосконаленні підходів до забезпечення захищеності персональних даних користувача, і впровадженню нових можливостей для мобільного додатку.

Практичне значення роботи полягає у створенні удосконаленого мобільного додатку, який би мав актуальні можливості захисту персональних даних, а також містив би у собі нові, удосконалені можливості які б сприяли захисту інформації.

РОЗДІЛ 1. АНАЛІЗ ОСНОВНИХ ПОЛОЖЕНЬ ПРО ПЕРСОНАЛЬНІ ДАНІ

1. Інформація про особу та персональні дані

Перш за все варто згадати поняття особистої інформації. Така інформація включає будь-яку інформацію, що стосується особистості, таку як біографічна інформація, уподобання (наприклад, улюблені книги або кольори), погляди тощо. Це дуже великий обсяг інформації. Але чи є будь-яка інформація про особу її особистими даними?

Закон України про інформацію визначає ці два поняття. Стаття 11 закону містить таке визначення: "Інформація про фізичну особу (персональні дані) - інформація або група інформації про ідентифіковану або конкретно ідентифіковану фізичну особу ...". Стаття 1 закону про "Захист персональних даних" визначає особисті дані аналогічним чином: "Інформація про персональні дані або група інформації про ідентифіковану або конкретно визначену особу". Однак закон чітко не визначає, що вся інформація про людину - це її особисті дані.

Головне зрозуміти, яку особу можна ідентифікувати чи ідентифікувати. Як правило, не викликає сумнівів ім’я, адреса, персональний податковий номер та інша інформація. Ці дані можна використовувати окремо або в поєднанні, щоб власник даних міг чітко ідентифікувати конкретну особу. Ситуація непрямого ідентифікації даних людини є більш складною. Наприклад, ви купуєте певні товари в магазині та користуєтесь дисконтною карткою. Незважаючи на те, що інформація про придбані товари стосується фізичних осіб, сама по собі не є особистими даними. Зрештою, кожен може придбати один і той же товар у магазині чи деінде.

Однак, якщо ви купуєте та використовуєте дисконтну карту, продавець може використовувати її для ідентифікації конкретної особи, тому ваша історія покупок та інформація про картку стануть особистими даними. Так, можна сказати, що персональними даними буде інформація про номер картки, ім’я власника картки,

дата та час придбання, її вартість та інформація про придбані предмети. І ці дані будуть захищені законом і повинні збиратися для законних цілей. Іншими словами, якщо певна інформація дозволяє власникові вибрати конкретну особу з групи людей, вона може трактуватися як особисті дані. Тому в деяких випадках (коли вони можуть ідентифікувати особу) ними стануть дані, які не є персональними даними.

Однак, якщо всі деякі дані не можуть ідентифікувати особу, їх обробка не захищена Законом про захист персональних даних. Ми хотіли б звернути вашу увагу на наступний факт: Регламент Європейського Парламенту та Ради (ЄС) 2016/679 (головний документ ЄС щодо захисту особистих прав при обробці персональних даних) передбачає таку ж позицію. Особливо це стосується даних, які були знеособлені (стали анонімними). Отже, пункт 26 цього регламенту передбачає: «Відповідно, принцип захисту даних не застосовується до анонімної інформації, особливо до інформації, яка не має нічого спільного з ідентифікованими особами чи персональними даними, які таким чином стають анонімними і унеможливлюють ідентифікацію особи. Тому цей регламент не застосовується до обробки такої анонімної інформації, в тому числі для статистичних або дослідницьких цілей ".

Персональні дані - це інформація про фізичних осіб. Не вся особиста інформація - це особиста інформація. Все залежить від того, чи може ця інформація ідентифікувати людину.

2. Проблеми визначення конфіденційної інформації

Як особисті дані, так і конфіденційна інформація стосуються фізичних осіб. Але ці поняття не зовсім однакові. Відповідно до " Про доступ до публічної інформації", конфіденційна інформація стосується інформації, обмеженою для

доступу фізичних або юридичних осіб, за винятком суб'єкта владних повноважень, і може поширюватися у встановленому порядку відповідно до їх вимог та умов, передбачених ними. "Закон про інформацію" у статті 21 також дає подібне визначення, тобто: "Конфіденційною є інформація про фізичних осіб та інформація, до якої фізичним або юридичним особам може бути обмежений доступ, крім суб'єкта владних повноважень. Поширення у встановленому ним порядку відповідно до передбачених ним умов, а також може поширюватися за інших обставин, передбачених відповідним законодавством.

Стаття 7 Закону «Про доступ до публічної інформації»	Стаття 11 і 21 Закону «Про інформацію»	Стаття 2 Закону «Про захист персональних даних»
Конфіденційна інформація - інформація, доступ до якої обмежено фізичною або юридичною особою, крім суб'єктів владних повноважень, та яка може поширюватися у визначеному ними порядку за їхнім бажанням відповідно до передбачених ними умов.	Інформація про фізичну особу (персональні дані) - відомості чи сукупність відомостей про фізичну особу, яка ідентифікована або може бути конкретно ідентифікована... До конфіденційної інформації про фізичну особу належать, зокрема, дані про її національність, освіту, сімейний стан, релігійні переконання, стан здоров'я, а також адреса, дата і місце народження. ... Конфіденційною є інформація про фізичну особу, а також інформація, доступ до якої обмежено фізичною або юридичною особою, крім суб'єктів владних повноважень. Конфіденційна інформація може поширюватися за бажанням (згодою) відповідної особи у визначеному нею порядку відповідно до передбачених нею умов, а також в інших випадках, визначених законом	Персональні дані - відомості чи сукупність відомостей про фізичну особу, яка ідентифікована або може бути конкретно ідентифікована; Персональні дані **можуть бути віднесені** до конфіденційної інформації про особу законом або відповідною особою

Рис. 1.1 Витяги із Закону України про персональні дані.

Можна зазначити, що обидва закони вказують, що лише особи приватного права можуть вирішувати, яка інформація про них є конфіденційною, а яка публічною. Однак є багато прикладів того, що закон класифікує певну інформацію як конфіденційну, тобто закон за замовчуванням встановлює "конфіденційну" систему. Наприклад, "Закон про інформацію" передбачає, що конфіденційна інформація про фізичних осіб включає: національність, рівень освіти, сімейний стан, релігійні переконання, стан здоров'я та адреса, дата та місце народження. Національний закон про статистику передбачає, що конфіденційна інформація

повинна включати оригінальні дані, отримані національним статистичним агентством від респондентів протягом періоду статистичного спостереження, та інформацію про респондента, отриману національним статистичним агентством від установ, що займаються пов'язаною діяльністю, пов’язаною зі збором та використанням управлінських даних.

Варто також зазначити, що не вся інформація може бути класифікована як конфіденційна за індивідуальним запитом. У багатьох випадках різні закони передбачають відкритість певної інформації, наприклад, пов’язані посади та робочі контакти, розпорядження бюджетними коштами, розкриття інформації в реєстрі тощо. Тому закон може заборонити будь-кому обмежувати доступ до певної інформації. Насправді особа, пов'язана з інформацією, не має права визначати, як отримати доступ до такої інформації.

Тому, можемо зробити висновок, що:

- Інформація про фізичну або юридичну особу є конфіденційною, за винятком суб’єкта владних повноважень, який обмежений доступом особи та раніше обмежувався законом, доки особа не розкриє таку інформацію відповідно до власних вимог.
- Така інформація може поширюватися відповідно до вимог (згоди) відповідних осіб на умовах, передбачених ними, у встановленому ними порядку (за інших обставин, передбачених законодавством)
- Закон може заборонити певну інформацію класифікувати як інформацію з обмеженим доступом, включаючи конфіденційну.

Слід також додати, що відповідно до судової практики, якщо інша інформація, пов’язана з посадою (наприклад, інформація про освіту, досвід роботи, знання іноземної мови, відсутність судимості тощо), не може бути віднесена до категорії конфіденційних. Для виконання функцій національної або місцевої автономної

організації були встановлені відповідні кваліфікаційні вимоги або інші обов'язкові вимоги до посади.

Не вся особиста інформація є конфіденційною. За обставин, передбачених законом, певні персональні дані є відкритою інформацією. У той же час конфіденційна інформація включає не лише персональні дані.

Рис. 1.2 Персональні дані і конфіденційна інформація.

3. Персональні дані та конфіденційна інформація

Перше, що слід пам’ятати про особисті дані, це завжди інформація про окремих людей і лише про живих людей. Відповідно до статей 24 та 25 Цивільного кодексу України особи, які беруть участь у цивільних відносинах, вважаються фізичними особами. Її здатність до цивільної поведінки з’явилася при народженні та припинилася після смерті. Отже, згідно із "Законом про захист персональних даних" та "Цивільним законом" інформація про померлого не є його

персональними даними. Однак конфіденційна інформація може також включати інформацію про юридичних осіб, наприклад, "комерційну таємницю". Відповідно до статті 505 Цивільного кодексу України, це може бути інформація технічного, організаційного, комерційного, промислового та іншого характеру (за винятком інформації, яка не може бути віднесена до комерційної таємниці в законі), і юридична особа вжила заходів проти цієї інформації Заходи щодо збереження конфіденційності. Іншу інформацію також можна назвати конфіденційною юридичною особою.

Водночас закон може забороняти класифікацію певних персональних даних осіб як конфіденційну інформацію. Це може стосуватися такої інформації:

- Прізвища, імена та спонсори осіб, які отримали бюджетні кошти, володіють державним та / або громадським майном, користуються ними чи розпоряджаються (пункт 5 статті 6 Закону про доступ до публічної інформації);
- Персональні дані, пов’язані з виконанням особами, які виконують свої обов’язки в рамках функцій держави чи місцевого самоврядування, посадових осіб чи службових повноважень (частина 2 статті 5 Закону про захист персональних даних);

В обох випадках прізвище, ім’я чи по-батькові особи, яка отримала бюджетні кошти або займає певну посаду, є її персональними даними, оскільки ця інформація дозволяє ідентифікувати конкретну особу. Однак будь-які акти щодо розкриття персональних даних, такі як збір, розповсюдження тощо, не захищені Законом про захист персональних даних. Більше того, хоча цей закон прямо не передбачає цього, це розуміння логічно з його положення, оскільки всі дії з обробки (збір, реєстрація, накопичення, зберігання, адаптація, модифікація, оновлення, використання та розповсюдження) (випуск, продаж, передача), скасування

персоналізації, знищення персональних даних, включаючи використання інформаційних (автоматичних) систем) призначені для захисту персональних даних, які класифікуються як конфіденційна інформація. В іншому випадку власник нестиме відповідальність (наприклад, суб'єкт даних буде повідомлений про зміну, видалення або знищення своїх персональних даних відповідно до статті 21 Закону про захист персональних даних), що на практиці неможливо. Це ефективно паралізує роботу будь-якого власника таких даних.

Слід також додати, що відповідно до судової практики, якщо інша інформація, пов'язана з посадою (наприклад, інформація про освіту, досвід роботи, знання іноземної мови, відсутність судимості тощо), не може бути віднесена до категорії конфіденційних. Для виконання функцій національної або місцевої автономної організації були встановлені відповідні кваліфікаційні вимоги або інші обов'язкові вимоги до посади.

4. Загальні поняття та сфера застосування

Визначення термінів:

- Персональна база даних - організована група персональних даних, названих в електронній формі та / або у формі файлів персональних даних;
- Відповідальна особа - це призначена особа, яка організовує роботу, пов'язану із захистом персональних даних, у межах, передбачених законодавством;
- Власник персональної бази даних - фізична або юридична особа, яка має право обробляти ці дані відповідно до законодавства або за згодою суб'єкта

персональних даних, яка затверджує мету обробки персональних даних у цій базі даних, визначає склад та процедури обробки цих даних (якщо Так) Закон не визначає;

- Реєстр стану персональних баз даних - єдина система інформації про статус, яка використовується для збору, накопичення та обробки інформації про зареєстровані бази персональних даних;
- Загальнодоступні джерела-каталоги персональних даних, адресні книги, реєстри, списки, довідники та інші колекції публічної інформаційної системи, ця публічна інформація містить персональні дані, опубліковані та опубліковані з відома суб’єкта персональних даних.

Соціальна мережа та інтернет-ресурси, де суб’єкти персональних даних зберігають свої персональні дані, не вважаються загальнодоступними ресурсами (якщо суб’єкт персональних даних чітко не вказує, що надання персональних даних здійснюється для безкоштовного розповсюдження та використання).

Згода суб'єкта персональних даних - це будь-який письмовий запис і готовність, висловлена особою, яка бажає висловити свою готовність обробляти свої персональні дані відповідно до заявленої мети обробки. Неособиста ідентифікація персональних даних стосується видалення особистої інформації.

Обробка персональних даних - будь-яка дія або сукупність дій та розповсюдження, що стосуються набору збору, реєстрації, накопичення, зберігання, адаптації, модифікації, оновлення, використання, що виконуються повністю або частково в інформаційній (автоматизованій) системі та / або файлі персональних даних (Поширення, продаж, передача), знеособлення, знищення інформації про осіб.

Інформація про особисті дані або група інформації про особу, яка була ідентифікована або може бути конкретно ідентифікована. Контролер персональних

даних - фізична або юридична особа, яка є власником персональної бази даних або юридично уповноваженою на обробку цих даних. Власник та / або адміністратор персональної бази даних вказує, що особа, яка використовує персональну базу даних для виконання технічних робіт без доступу до змісту персональних даних, не є адміністратором персональної бази даних. Суб'єктом персональних даних є фізична особа, і його персональні дані будуть оброблятися відповідно до законодавства. Третя сторона - будь-хто, крім суб'єкта персональних даних, власника або адміністратора персональної бази даних та уповноваженого національного органу з захисту персональних даних, власник або адміністратор персональної бази даних передаватиме їм персональні дані відповідно до законодавства. До спеціальних категорій даних належать особисті дані, пов'язані з расою чи етнічним походженням, політичними, релігійними чи ідеологічними переконаннями, політичними партіями та членством у профспілках, а також дані, пов'язані зі здоров'ям чи статевим життям.

Це положення є обов'язковим для відповідальних осіб та працівників продавця, які безпосередньо обробляють та / або мають доступ до персональних даних, пов'язаних із виконанням своїх обов'язків.

5. Мета обробки персональних даних

Відповідно до статей 6 та 7 "Закону про захист персональних даних" України, метою обробки персональних даних у системі є зберігання та підтримка даних учасника операції.

Метою обробки персональних даних є забезпечення встановлення цивільних правовідносин відповідно до українського податкового законодавства, українського «Закону про бухгалтерський облік та фінансову звітність» України для надання / отримання та оплати придбаних товарів / послуг.

6. Порядок обробки персональних даних: отримання згоди, повідомлення про права та дії з персональними даними суб’єкта персональних даних.

Згода суб'єкта персональних даних повинна виражати готовність особи добровільно погодитися на обробку його персональних даних відповідно до встановленої цілі обробки. Згода суб'єкта персональних даних може мати наступні форми: паперовий документ із детальною інформацією, яка може ідентифікувати документ та особу; електронний документ, який повинен містити обов'язкову детальну інформацію, який може ідентифікувати документ та особу. . Бажання добровільно виразити волю особи дозволити обробку її персональних даних повинно бути доведено електронним підписом суб’єкта персональних даних.

Відповідно до задокументованих програмних та апаратних рішень, позначайте на електронній сторінці документа або в електронному файлі, обробленому в інформаційній системі. Відповідно до чинного законодавства, згода суб’єкта персональних даних буде отримана під час процесу реєстрації цивільних відносин.

Під час реєстрації цивільних відносин суб’єкти персональних даних отримують повідомлення про включення своїх персональних даних до бази персональних даних, права, передбачені українським «Законом про захист персональних даних», мету збору даних та законодавство про повідомлення особи, якій передаються їхні персональні дані.

Обробка персональних даних, що стосуються расового чи етнічного походження, політичних, релігійних чи ідеологічних переконань, членства в політичних партіях та профспілках, а також даних, що стосуються здоров'я або статевого життя (спеціальні категорії даних), забороняється.

7. Умови розкриття інформації про персональні дані третім особам.

Порядок доступу до персональних даних третіх осіб залежить від умов згоди суб'єкта персональних даних, наданих власнику персональної бази даних для обробки таких даних, або від вимог законодавства.

Якщо особа відмовляється обіцяти забезпечити виконання вимог українського «Закону про захист персональних даних» або не надає цю інформацію, особі не буде надано доступ як третій особі.

Суб’єкт відносин, пов’язаних із персональними даними, подає запит на доступ до персональних даних (далі - запит) власнику персональної бази даних. У запиті було зазначено:

- прізвище, ім’я та по батькові, місце проживання (місце перебування) і реквізити документа, що посвідчує фізичну особу, яка подає запит (для фізичної особи — заявника);
- ім’я, місцезнаходження, посада, прізвище юридичної особи, яка подає запит, а також ім’я та особисті дані особи, яка засвідчує запит; підтвердити, що зміст запиту відповідає повноваженням юридичної особи (для юридичної особи-заявника);
- Інформація про персональну базу даних, на яку подається запит, або інформація про власника або адміністратора цієї бази даних;
- Перелік необхідних персональних даних;
- мета запиту.

З дати отримання запиту задовільний період для запиту на дослідження не повинен перевищувати десяти робочих днів.

Протягом цього періоду власник персональної бази повідомляє особу, яка подає запит, що вона задовольнить запит або не надасть відповідні персональні дані, та вказує причини, зазначені у відповідному правовому акті.

Якщо законом не передбачено інше, запит задовольняється протягом тридцяти календарних днів з дати отримання запиту. Усі працівники власників персональних

баз даних зобов'язані дотримуватися вимог конфіденційності персональних даних та інформації про рахунки у цінних паперах та обігу цінних паперів.

Якщо необхідні дані не можуть бути надані протягом тридцяти календарних днів з дати отримання запиту, можна відкласти доступ до персональних даних третьої сторони. У цьому випадку загальний термін вирішення питань, порушених у запиті, не повинен перевищувати сорока п’яти календарних днів.

У повідомленні про розширення письмово повідомляється третя сторона, що подає запит, та пояснюється процедура оскарження рішення.

У повідомленні про відстрочення зазначаються:

- прізвище, ім’я та по батькові посадової особи;
- дата відправки повідомлення;
- причина відкладення;
- Кінцевий термін для задоволення запиту;

Якщо закон забороняє доступ до персональних даних, заборонено доступ. У повідомленні про відмову слід вказати: прізвище, ім’я, особисту особу чиновника, який відмовився від відвідування, дату відправлення повідомлення та причину відмови. Рішення про відкладення або відмову у доступі до персональних даних можуть бути оскаржені до уповноважених національних органів з питань захисту персональних даних, інших державних органів та органів місцевого самоврядування (до повноважень яких належить захист персональних даних) або судів.

8. Захист персональних даних: способи захисту та її суб’єкти

Власник персональної бази даних оснащений системами, програмним та апаратним забезпеченням та засобами зв'язку, які можуть запобігти втратам, крадіжкам, несанкціонованому знищенню, спотворенням, підробці та копіюванню інформації, а також відповідати вимогам міжнародних та національних стандартів.

Відповідальна особа організовує роботу, пов’язану із захистом персональних даних, відповідно до законодавства. Відповідальна особа визначається наказом окремих власників бази даних.

Посадова інструкція передбачає відповідальність керівника організації праці щодо захисту персональних даних під час обробки персональних даних. Відповідальна особа зобов’язана:

- Розуміти українське законодавство у галузі захисту персональних даних;
- Розробити процедури доступу до персональних даних працівників відповідно до їх професійних чи службових чи службових обов’язків;
- Забезпечити, щоб захист персональних даних та управління внутрішніми документами власника персональної бази даних щодо обробки та захисту персональних даних у персональній базі відповідали вимогам законодавства України;
- Встановити процедури (процедури) внутрішнього контролю для забезпечення відповідності українському законодавству про захист персональних даних та внутрішнім документам, що регулюють діяльність із захисту персональних даних у власників персональних даних та базах персональних даних, особливо про правила частоти контроль;
- Не пізніше одного робочого дня повідомляти власника бази персональних даних про те, що працівник порушив українські нормативні акти щодо захисту персональних даних та внутрішні документи власника бази персональних даних щодо обробки та захисту персональних даних у базі персональних даних. . порушення;
- Забезпечити зберігання документів, щоб підтвердити, що суб’єкт персональних даних погоджується на обробку своїх персональних даних, та повідомити призначеного суб’єкта про свої права;

З метою виконання своїх обов’язків відповідальна особа має право:

- Отримувати необхідні документи, включаючи замовлення, видані власником персональної бази даних, та інші документи управління, пов’язані з обробкою персональних даних;
- Робити копії отриманих документів, включаючи копії документів, усіх записів, що зберігаються в локальних мережах та незалежних комп’ютерних системах;
- Брати участь в дискусіях про його обов'язки, пов'язані із захистом персональних даних, у процесі обробки персональних даних;
- Подавати пропозиції щодо вдосконалення діяльності та методів роботи, вносити зауваження та варіанти усунення недоліків, виявлених у процесі обробки персональних даних;
- Отримувати вказівки щодо обробки персональних даних;
- Підписувати документи в межах своїх повноважень;

Працівники, які безпосередньо обробляють та / або отримують доступ до персональних даних під час виконання службових (трудових) обов’язків, повинні дотримуватись вимог законодавства України щодо захисту персональних даних та внутрішніх документів, пов’язаних з обробкою та захистом персональних даних у базі персональних даних.

Працівники, які мають доступ до персональних даних, включаючи працівників, які обробляють персональні дані, зобов'язані запобігати передачі їм будь-яких персональних даних або персональних даних, які добре відомі при виконанні професійних або службових обов'язків або службових обов'язків. Мова. Якщо інше не передбачено законом, це зобов'язання набуває чинності лише після припинення діяльності, пов'язаної з персональними даними.

Ті, хто має право доступу до персональних даних, у тому числі ті, хто здійснює обробку з порушенням вимог "Закону України про захист персональних даних", несуть відповідальність відповідно до законодавства України.

Час зберігання персональних даних не повинен перевищувати часу, необхідного для цілей їх зберігання, але в будь-якому випадку час їх зберігання не повинен бути довшим за час збереження даних, визначений за згодою персональних даних, що обробляються цими даними.

9. Права суб'єкта персональних даних

Суб'єкт персональних даних має право:

- Знати місцезнаходження персональної бази даних, яка містить її персональні дані, її призначення та ім'я, ім'я та / або місце проживання (проживання) власника бази даних або адміністратора, або видавати відповідні розпорядження щодо отримання цієї інформації від уповноваженого персоналу;
- Отримувати інформацію про умови надання доступу до персональних даних, особливо інформацію про передачу третій стороні третьої сторони, чиї персональні дані містяться у відповідній базі персональних даних;
- Отримати доступ до персональних даних, що містяться у відповідних базах персональних даних;
- Якщо законом не передбачено інше, не пізніше 30 календарних днів з дати отримання запиту на отримання відповіді щодо того, чи зберігаються персональні дані у відповідній персональній базі даних, та зміст персональних даних, що ними зберігаються;

- Подавати обґрунтовані запити до органів державної влади, а органи місцевого самоврядування здійснюють свої повноваження щодо обробки їх персональних даних відповідно до закону;
- Якщо будь-який власник та адміністратор бази даних обробляв дані незаконно або є неточними, вони повинні надати обґрунтовані причини, щоб вимагати зміни або знищення своїх персональних даних;
- Захищати свої персональні дані від незаконної обробки та випадкової втрати, знищення, пошкодження, спричинених навмисним приховуванням, ненаданням або ненаданням вчасно, та запобігати тим, хто надає інформацію, яка є недостовірною або завдає шкоди честі, гідності та діловій репутації організації
- звертатися до уряду штату та місцевого самоврядування щодо захисту їхніх прав на особисті дані, і їх повноваження включають захист персональних даних;
- Якщо ви порушили закони про захист персональних даних, скористайтеся засобами правового захисту.

10. Порядок роботи з запитами суб’єкта персональних даних

Суб'єкт персональних даних має право отримувати будь-яку інформацію про себе від будь-якого суб'єкта у відносинах, пов'язаних з персональними даними, якщо інше не передбачено законом, без зазначення мети запиту. Суб'єкт персональних даних має вільний доступ до даних про себе. Суб’єкт персональних даних подає запит на доступ до персональних даних власнику персональної бази даних (далі - запит). У запиті зазначаються:

- Дані про прізвище, ім’я та ім’я, місце проживання (місце проживання) та документи, що посвідчують особу суб’єкта персональних даних

- Інша інформація, яка може ідентифікувати суб’єкта персональних даних;
- Інформація про запитувану персональну базу даних або інформація про власника або адміністратора цієї бази даних;
- Перелік запитуваних персональних даних.

З дати отримання запиту задовільний період запиту на дослідження не повинен перевищувати десяти робочих днів. Протягом цього періоду власник бази даних персональних даних повідомляє суб’єкта персональних даних про те, що він задовольнить запит або не надасть відповідні персональні дані, та вказує причини, зазначені у відповідних юридичних діях. Якщо законом не передбачено інше, вимоги повинні виконуватися протягом тридцяти календарних днів з дати отримання запиту.

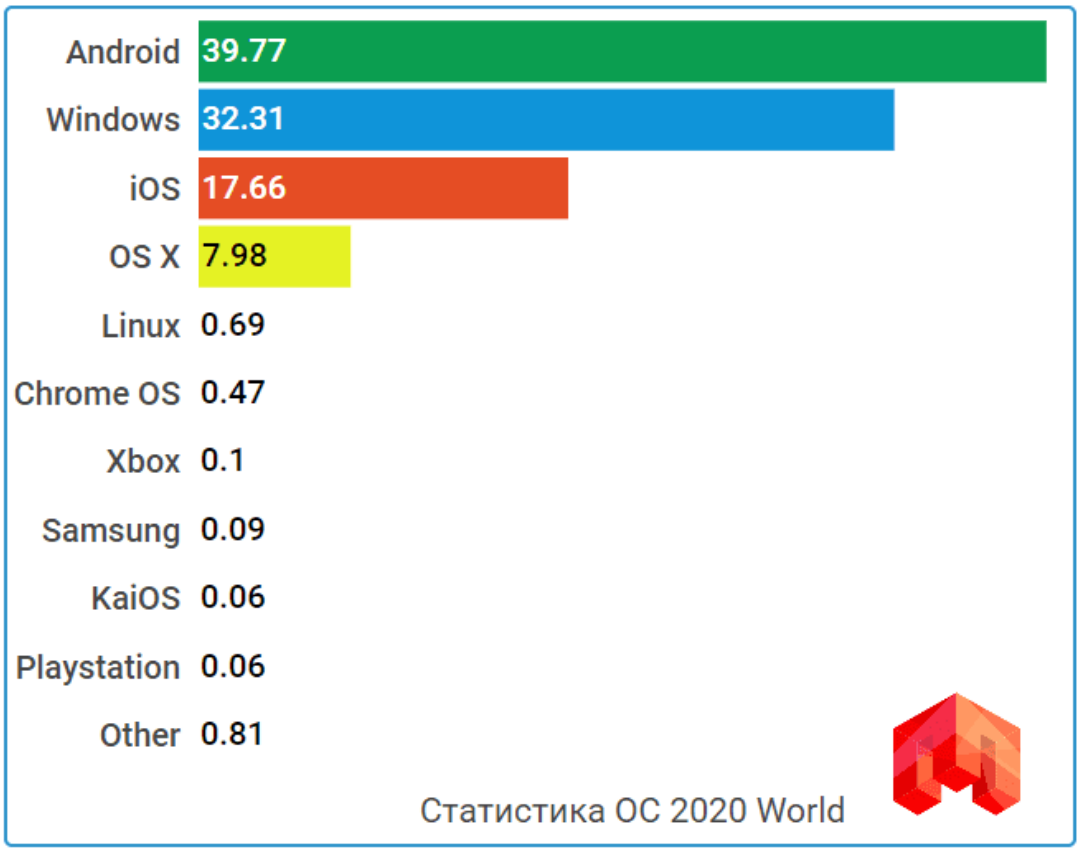
Висновок: в даному розділі було проведено детальне вивчення поняття «персональні дані» та проаналізовано його характеристики. Було ознайомлено з інформацією про особу, метою та порядком обробки персональних даних. Було визначено за яких умов інформація про персональні дані розкривається третім особам, та розглянуто способи захисту персональних даних. Також було розглянуто таке поняття як «конфіденційна інформація», та чим вона відрізняється від персональних даних. На основі вивченої інформації можна зробити висновок, що питання щодо персональних даних є дуже актуальним в сьогоднішньому цифровому світі і їх захист є дуже важливим фактором у житті кожної людини, яка зберігає їх у цифровому вигляді.

Статистика операційних систем у світі.

У сучасному світі більшість персональних даних людей зберігається у цифровому варіанті. Це водночас зручно і не потребує додаткових зусиль для зберігання інформації. На заміну флешкам, у вигляді цифрових носіїв інформації прийшли смартфони. Вони набагато гнучкіші у використанні, і на відміну від звичайих карт пам’яті надають змогу не тільки зберігати, а водночас і редагувати файли. В той же час, смартфон залишається в першу чергу засобом для комунікації, а не захищеним вмістилищем для інформації користувача. А це означає, що з’являються ситуації, при яких ваш пристрій котактує із третіми особами. Вихід в інтернет, добровільна передача смартфону у користування іншого, його втрата, це ніщо інше як збільшення ризику втратити інформацію про свої персональні дані. Перед тим як, розпочати розробку програмного модуля, який би забезпечував захищеність персональних даних, потрібно провести дослідження платформи, для якої дане програмне забезпечення і розробляється.

За даними Statcounter.com, загальний рейтинг операційних систем, включаючи настільні комп'ютери, ноутбуки, смартфони і навіть ігрові консолі, показує, що лідером є Android, який встановлений на 39,77% пристроїв, що вказує на кількість мобільних гаджетів у всьому світі що постійно росте.

Якщо розглядати окремо операційну систему, встановлену на комп'ютері, Windows домінує - 76,58% усіх комп'ютерів. OS X-18.93 та Linux-1.62%. Серед мобільних телефонів Android утримує лідируючі позиції - 70,43%, а мобільна операційна система Apple iOS - 29,06% - майже кожна третина смартфонів. Хоча відтворення на планшеті iOS становить 59,97, на планшеті Android - 39,79. Операційна система Xbox в основному встановлюється на ігрових приставках - 62,69% та Playstation - 36,9%.



Операційні системи, смартфони світ (червень 2020)	%
Android	70,43
iOS	29,06
Samsung	0,16
KaiOS	0,11
Windows	0,07
Tizen	0,02
Series 40	0,02
Nokia	0,02
BlackBerry OS	0,01
Linux	0,01
Other	0,09

Рейтинг операційних систем в Україні в 2020 році.

Загальний рейтинг операційних систем України, в який входять комп'ютери, мобільні телефони, планшети та ігрові консолі, показує, що лідером є Windows, яка встановлена на 59,81% пристроїв. Якщо брати окремо операційну систему, встановлену на вашому комп’ютері, то лідером в Україні також є Windows - 83,73%, OS X - 13.1 та Linux - 1.82%.

Серед мобільних телефонів Android посідає перше місце - 82,09%, а операційна система iOS - 17,46%, майже кожен шостий смартфон. Більше того, Android також займає лідируючі позиції в планшетних комп’ютерах, складаючи 54,93% усіх планшетних комп’ютерів і 44,85% у iOS. Серед ігрових консолей в Україні найулюбленішими є Playstation-71,52% та Xbox-28,42%.

З проаналізованих даних можна зробити простий, але дуже важливий висновок. Розробка програмного забезпечення саме для ОС Android є фокусною задачею, адже це найпопулярніша операційна система у світі, і хоча в Україні вона посягає друге місце, на мобільних пристроях і на планшетах вона залишається найбільш затребуваною.

Аналіз існуючих додатків.

Тож перед нами стоїть задача розробити додаток, який міг би забезпечити захищеність персональним даним свого користувача. Перш за все потрібно проаналізувати уже існуючі варіанти які пропонуються. В онлайн магазині Google Play Market міститься безліч додатків, серед яких деякі з них пропонують нам можливість захистити особисту інформацію про користувача. На прикладі чотирьох найпопулярніших додатків Google Play Market проведемо наше дослідження, в результаті якого сформуємо висновок про всі плюси і мінуси даного програмного забезпечення і доцільності розробки власного програмного модуля.

Основною ціллю даного дослідження буде розібратися в можливостях додатків які стосуються саме забезпечення захищеності інформації.

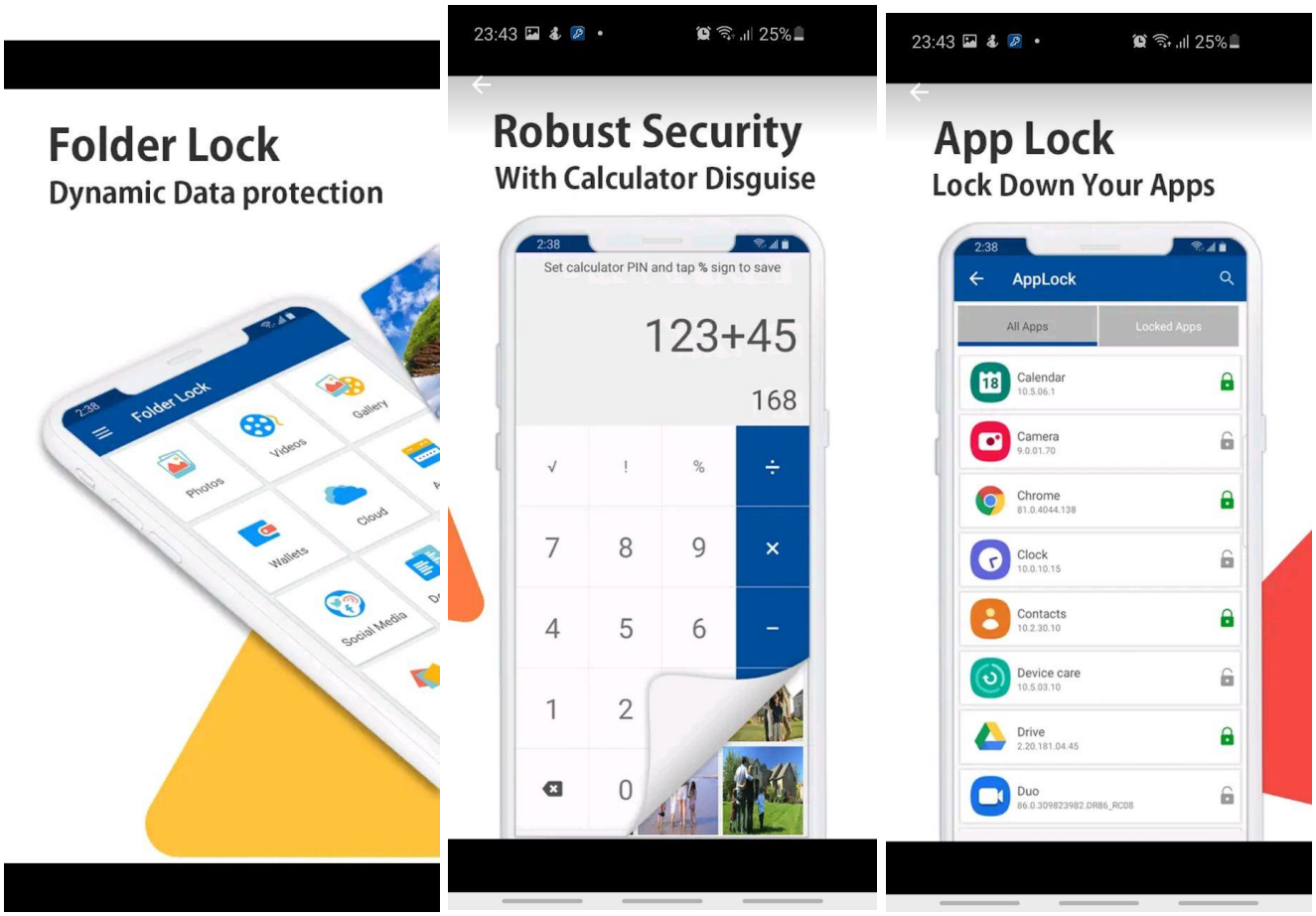
Folder Lock

Одне із найпопулярніших додатків в мережі Google Play Market В рамках програмного забезпечення для забезпечення захищеності персональних даних. Володіє досить широким спектром можливостей для свого користувача. Функціонал, що стосується забезпечення приватності даних:

- Функція доступу до додатку по PIN-коду
- Можливість резервного копіювання інформації в хмарний сервіс
- Розробник заявляє повну конфіденційність і можливість шифрування даних, при завантаженні їх у додаток.
- Можливість встановити PIN-код на вхід в сторонні додатки через калькулятор.

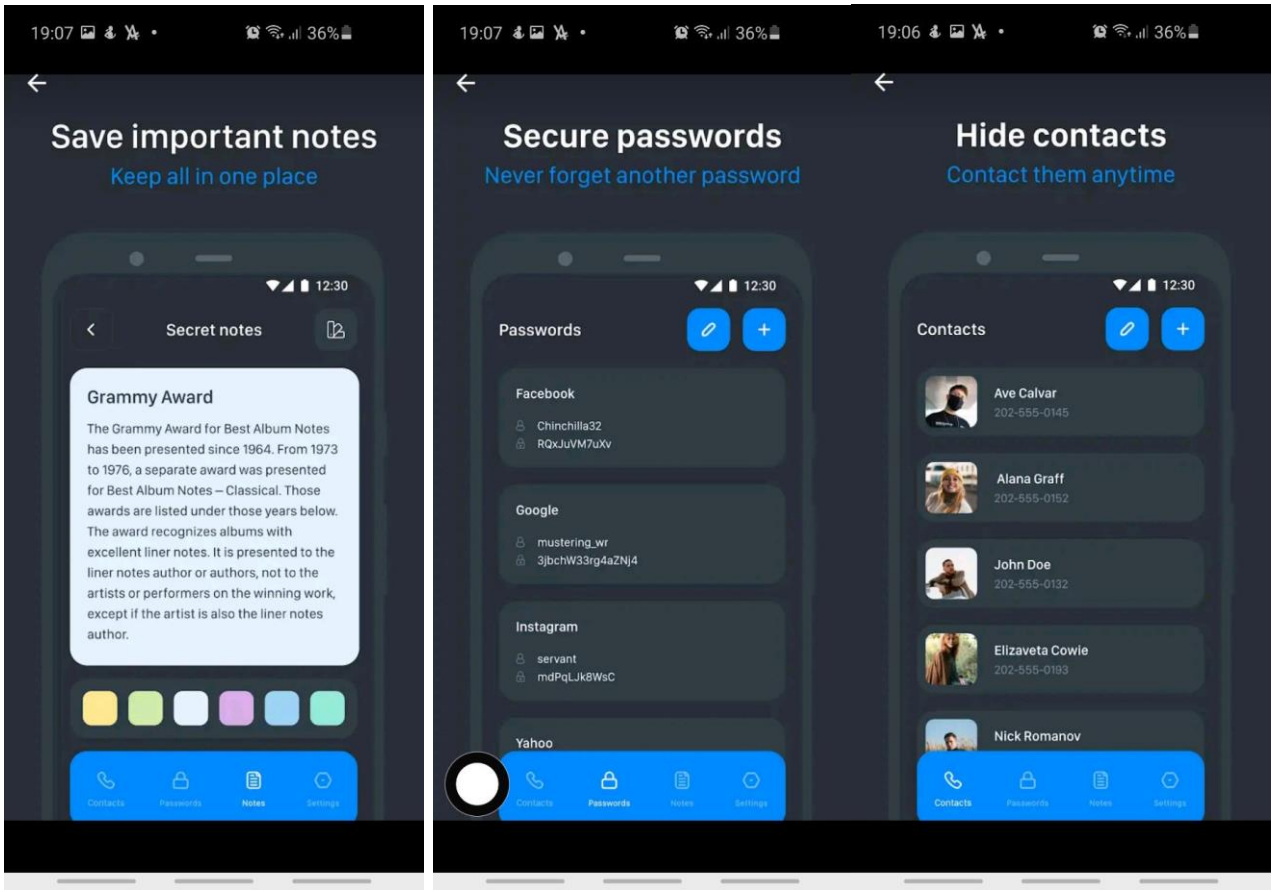
В цілому досить непоганий додаток, особливо варто відмітити його можливість працювати не тільки з фотографіями, але і відео, аудіозаписами, документами, тощо. Щодо самих можливостей забезпечення захищеності даних, існують недоліки, як наприклад:

1. Відсутній повторний запит на введення PIN-коду при згортанні додатку, або при блокуванні пристрою.
2. Є можливість зробити скріншот екрану і записати відео з екрану, при цьому знаходячись у самому додатку.
3. При згортанні додатку, в меня диспетчера запущених додатків на прев’ю секретної папки проглядається особиста інформація користувача.



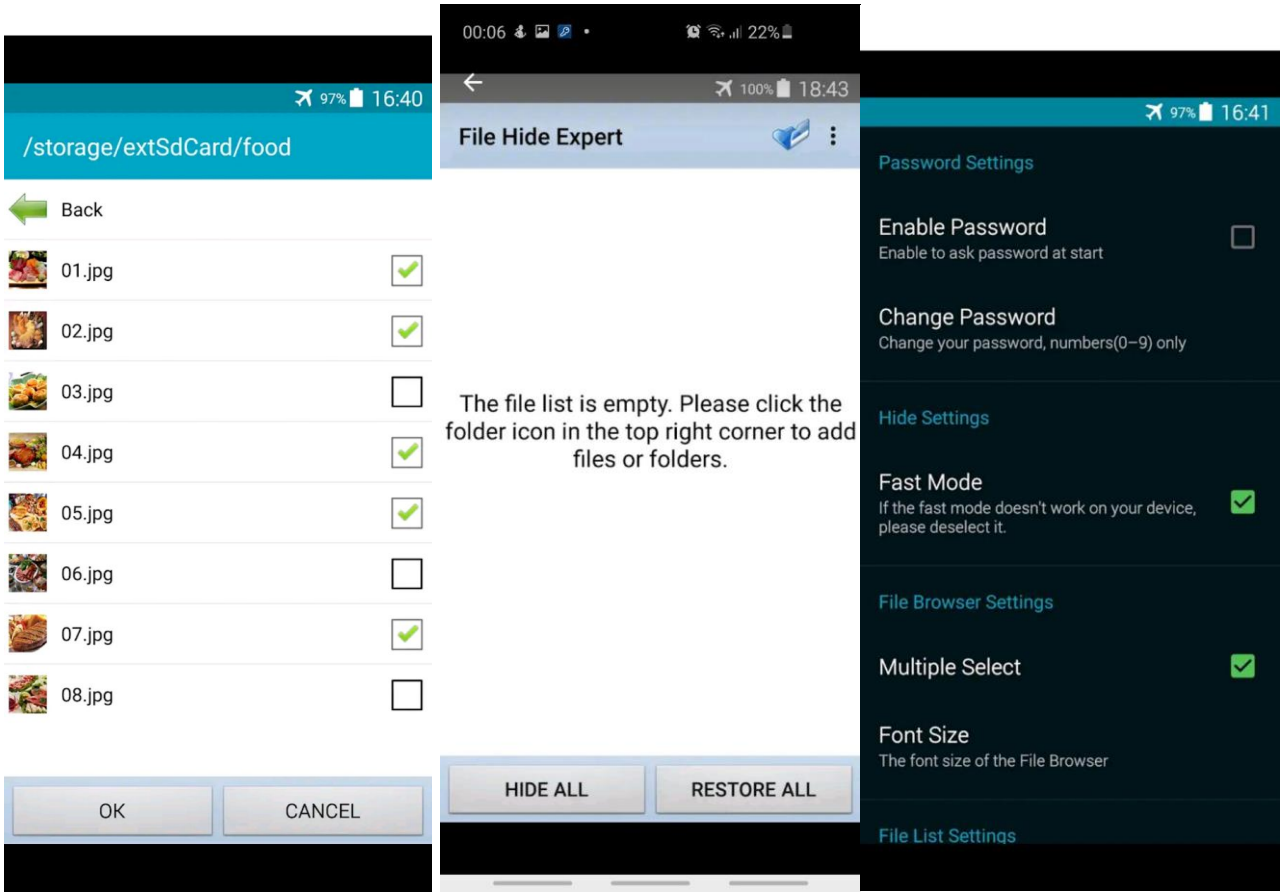
Secret Folder Vault

По своєму функціоналу даний програмний модуль значно програє своєму попередньому конкуренту, через набагато менший функціонал, відсутність резервного копіювання, відсутність входу по PIN- коду, що взагалі робить його доступним для сторонніх очей. Із можливостей є взаємодія із нотатками і контактами, також розробник заявляє про наявність певної ступені захищеності і шифрування даних, що зберігаються всередині додатку.



File Hide Expert

Ще один аналог програмного забезпечення здатного забезпечувати захищеність даних. На ділі ж, працює досить примітивно: заходить в корінь папки і переміщує його в свою кореневу папку. Є вхід по PIN-коду, на цьому плюси закінчуються, дані не шифруються, присутні усі ті ж уразливості що і були описані вище.



Secure Folder-Secure File

Додаток сам по собі і своїм функціоналом схожий на попередні приклади. Із приємних особливостей є режим секретного входу в браузер, хоча зараз практично у кожному браузері є режим інкогніто, що робить дану функцію більше зручною в користуванні, чим корисною і захищеною. Вхід за PIN- кодом присутній. Щодо недоліків:

1. Відсутній повторний запит на введення PIN-коду при згортанні додатку, або при блокуванні пристрою.
2. Є можливість зробити скріншот екрану і записати відео з екрану, при цьому знаходячись у самому додатку.
3. При згортанні додатку, в меня диспетчера запущених додатків на прев'ю секретної папки проглядається особиста інформація користувача.

4. Відсутня можливість використання біометричних даних для входу в додаток.

Для підведення підсумків сформуємо таблицю, в якій зазначимо, весь функціонал, який присутній у додатках, та функціонал, який нас цікавить.

Назва можливості/вимога	Folder Lock	Secret Folder Vault	File Hide Expert	Secure Folder-Secure File
PIN код	+	+	+	-

Можливість входу по біометричним даним	-	-	-	-
Резервне копіювання в хмарний сервіс	+	-	-	-
Забезпечення шифрування даних	+	+	-	-
Маскування самого додатку на пристрої	-	-	-	-
Можливість зробити ScreenShot/ Запис екрану	-	-	-	-
Унеможливлення перегляду вмісту секретної папки в диспетчері запущених задач	-	-	-	-
Унеможливлення перегляду вмісту папки під час трансляції екрану	-	-	-	-
Розлогін при блокуванні смартфону	-	-	-	-
Розлогін, при згортанні додатку	-	-	-	-

Висновок

Отже, проаналізувавши дане програмне забезпечення можна зробити висновок, що подібні аналоги уже існують на ринку. Очевидним недоліком є те, що ці додатки в основному відштовхуються від ситуації і пропрацьовують сценарій користування додатком на одинці, без присутності сторонніх людей поруч і без варіанту передачі смартфону у чужі руки. І це є досить суттєвою проблемою, адже така річ як смартфон є водночас як і дуже корисним унікальним девайсом, який містить у собі всю корисну інформацію для користувача, так і досить звичним пристроєм, який ми часто даємо стороннім лицям, щоб вони подзвонили, відправили повідомлення, зайшли в інтернет. Та чи зможемо ми бути впевнені, що в цей момент, третя особа не зможе краєм ока побачити те що їй не призначалося побачити? Чи можемо ми бути впевнені, що людина, яка сидить поруч з вами, зовсім не зацікавлена в тому, що ви переглядаєте на вашому смартфоні? Звичайно ні, саме тому і було прийнято рішення розробити програмний модуль, з яким було б набагато зручніше взаємодіяти в умовах оточення іншими людьми, в умовах короткострокової передачі пристрої в чужі руки. Даний додаток нам і допоможуть розробити попередньо проведені дослідження.

ЧАСТИНА 3

Етап реалізації програмного модулю.

Перш за все потрібно чітко визначити задачу і розібратися якими саме функціями повинна володіти наша програма. Оскільки програмне забезпечення буде створюватися для ОС Android, мовою програмування буде Java.

В додатку планується реалізувати:

- Даний програмний модуль повинен включати у собі всі найосновніші способи забезпечення захищеності персональних даних користувача, які ми змогли виявити в результаті дослідження сторонніх додатків, а також функції, які не були передбачені у попередніх розробників.
- Перша, і одна із основних функцій, яка буде присутня в програмному модулі, це реалізація паролю для входу в додаток. Даний спосіб буде представлений у вигляді PIN-коду, який користувачеві потрібно буде задати. Також, потрібно додати можливість входу в додаток, по відбитку пальців, таким чином у нас уже буде 2 способи входу в додаток, і користувач зможе обирати для себе найбільш зручніший спосіб.
- Далі нам потрібно додати функцію зберігання фотографій у додатку, і можливість завантажувати в нього фотографії із стандартного додатку галереї смартфону.
- Завантажуючись в додаток фотографії будуть шифруватися, і таким чином їх буде неможливо переглянути за межами додатку.
- Також буде реалізована функція резервного копіювання в хмарний сервіс. В якості хмарного сервісу ми будемо використовувати FireBase. Вводячи свою електронну адресу користувачеві буде надано можливість провести резервне копіювання фотографій, що знаходяться в секретній папці в хмарний сервіс.
- Відповідно потребується реалізація можливості завантаження фотографій із хмарного сервісу.
- Одною із цікавих особливостей програмного модулю стане можливість створити ярлик із кастомною картинкою та назвою. Ця функція потрібна для

додатково ступеня маскування самої іконки додатку, і щоб водночас користувачу було зручно користуватися програмним забезпеченням.

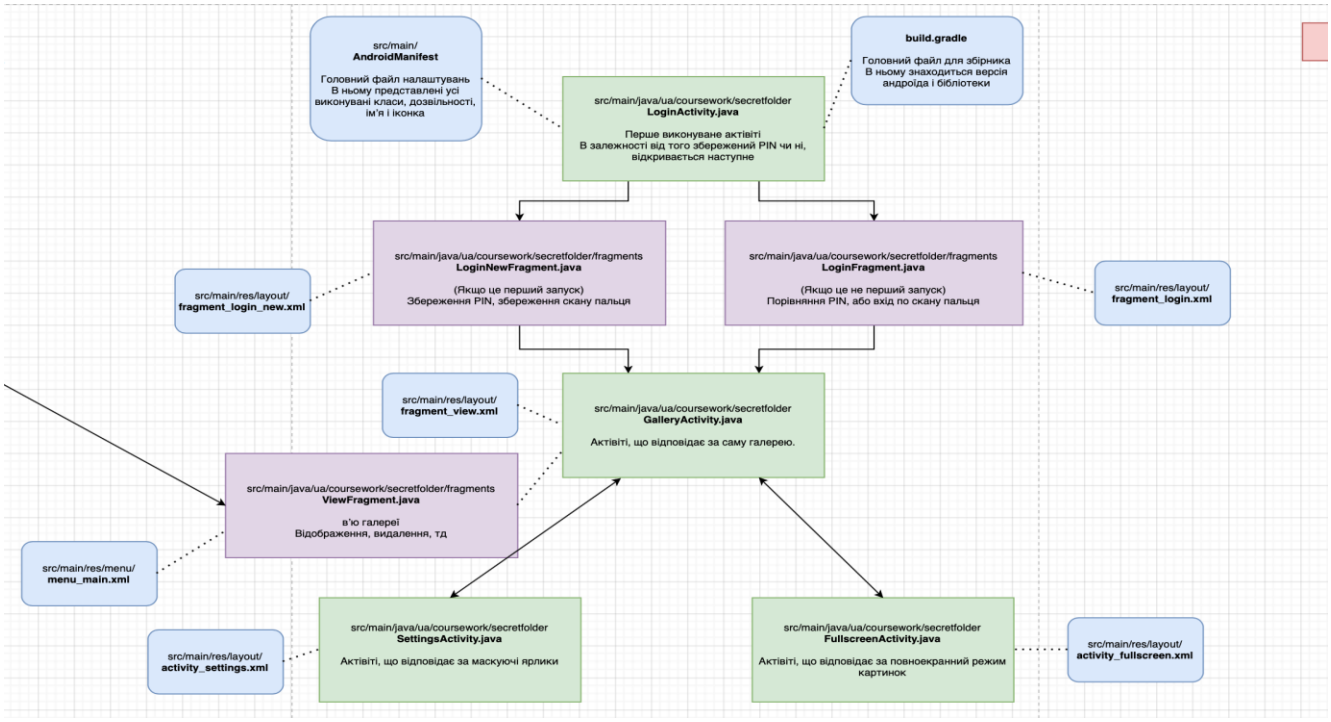
Далі мова піде про політику конфіденційності додатку, сценарії які можуть відбуватися при маніпуляціях зі смартфоном і функції, які будуть впроваджені опираючись на унеможливлення відтворювання нижчеописаних сценаріїв.

1. Ситуація, коли юзер знаходився всередині додатку, і йому знадобилося різко заблокувати телефон. В такому випадку, пілся того коли він розблокує його, останній запущений додаток, який був відкритий, висвітиться на екрані. Тому нам потрібна функція, яка б. виконувала розлогін нашого додатку, і спонукала до введення PIN-коду або прикладання відбитку пальців, з метою підтвердження особистості користувача. Те ж саме повинно відбуватися, коли користувач згортає додаток, а потім через диспетчер запущених завдань знову заходить в секретну папку.
2. Ситуація, коли додаток згортається, для того щоб перейти в інший додаток. У такому випадку, у диспетчері запущених завдань іконка додатку запам’ятовує останнє що відображалося в цьому додатку. Тому з’являється імовірність того, що стороння людина, зайшовши в диспетчер запущених задач, хоч і не зможе зайти в секретну папку, бо на ній стоїть пароль, але все ж зможе побачити частину інформації, яку містить в собі секретна папка. То ж потрібна функція, яка б унеможливлювала попередній перегляд вмісту додатку.
3. Ситуація, коли особа, яка не є володільцем пристрою, і все ж отримала доступ до вмісту додатку, файли вона вивантажити не зможе, вони за межами додатку зберігаються у зашифрованому вигляді, тому зловмисник може спробувати зробити скріншот екрану або відеозапис вмістимого екрану, і таким чином отримати файл, який містить у собі інформацію користувача.

Тому запис відео і можливість робити скріншоти потрібно вимкнути в рамках користування секретною папкою.

4. Ситуація, коли відбувається трансляція екрану смартфона на інший пристрій через сервіси по типу TeamViewer і т.д. Секретна папка - є секретна папка, відповідно сценарій, коли користувачу знадобиться показати її вміст віддалено, транслюючи екран – відпадає. Залишається імовірність випадкового, ненавмисного проступку з боку користувача, коли він по неуважності може показати вміст своєї секретної папки. Для такого випадку і потрібно розробити функцію унеможливлення перегляду вмісту секретної папки під час трансляції екрану смарфона.

Нижче представлена схема, яка відображає логіку додатку та послідовність відтворення дій.



Етап програмування

Основне з чого варто почати це встановленя PIN-коду та можливості входу по відбитку пальців. Що конкретно відбувається, коли користувач, задає PIN-код. Він шифрується за допомогою методу шифрування AES. Даний зашифрований PIN-код зберігається в підрутовій чатині пам’яті пристрою, куди можна отримати доступ лише отримавши права суперкористувача, що є додатковою ступінню захисту. Але перед тим як почати шифрування піна, ми беремо його md5 і далі ми уже шифруємо його md5. При наступній спробі входу, користувач вводить пін, ми отримуємо його md5, порівнюємо із дешифрованим md5 нашого ключа і йкщо вони співпадають, то викликається функція OnSuccess і ми отримуємо доступ до галереї.

```
package ua.coursework.secretfolder.utils;

import android.content.Context;
import android.content.SharedPreferences;
import android.os.Build;

import androidx.annotation.Nullable;
import androidx.annotation.RequiresApi;
import ua.coursework.secretfolder.utils.md5Calculator;

import javax.crypto.Cipher;

import co.infinum.goldfinger.crypto.CipherCrypter;
import co.infinum.goldfinger.crypto.CipherFactory;
import co.infinum.goldfinger.crypto.impl.Base64CipherCrypter;
import co.infinum.goldfinger.crypto.impl.UnlockedAesCipherFactory;
import ua.coursework.secretfolder.BuildConfig;
```

```
@RequiresApi(Build.VERSION_CODES.M)
public class PINCrypter {

    private static final md5Calculator md5 = new md5Calculator();
    private static final String PREF_FILE = BuildConfig.APPLICATION_ID.replace(".",
"_");
    private static SharedPreferences PREFS;
    private static CipherCrypter CRYPTER;
    private static CipherFactory FACTORY;

    @Nullable
    public static String getPin() {
        String encryptedPin = PREFS.getString("MmzD5LnDLfnhu8Q8", "");
        if ("".equals(encryptedPin)) {
            return "";
        }

        Cipher cipher = FACTORY.createDecryptionCrypter("MmzD5LnDLfnhu8Q8");
        if (cipher == null) {
            return "";
        }

        String decrypted = CRYPTER.decrypt(cipher, encryptedPin);

        return decrypted;
    }
```

```
public static void setPin(String pin) {
    Cipher cipher = FACTORY.createEncryptionCrypter("MmzD5LnDLfnhu8Q8");
    if (cipher == null) {
        return;
    }

    String passMD5 = md5.md5Apache(pin);
    String encryptedPin = CRYPTER.encrypt(cipher, passMD5);
    PREFS.edit().putString("MmzD5LnDLfnhu8Q8", encryptedPin).apply();

}

public static void init(Context context) {
    PREFS = context.getSharedPreferences(PREF_FILE, Context.MODE_PRIVATE);
    CRYPTER = new Base64CipherCrypter();
    FACTORY = new UnlockedAesCipherFactory(context);
}

public static boolean getFingerAuth() {
    String encryptedPin = PREFS.getString("RFu49REaA8EUVx2v", "");
    if ("".equals(encryptedPin)) {
        return false;
    }

    Cipher cipher = FACTORY.createDecryptionCrypter("RFu49REaA8EUVx2v");
    if (cipher == null) {
        return false;
```

```
        }

        if (CRYPTER.decrypt(cipher, encryptedPin).equals(String.valueOf(true))){
            return true;
        }else{
            return false;
        }
    }

    public static void setFingerAuth(boolean bool) {
        Cipher cipher = FACTORY.createEncryptionCrypter("RFu49REaA8EUVx2v");
        if (cipher == null) {
            return;
        }

        String encryptedBool = CRYPTER.encrypt(cipher, String.valueOf(bool));
        PREFS.edit().putString("RFu49REaA8EUVx2v", encryptedBool).apply();
    }
}
```

Це клас, який відповідає за шифрування по методу AES

```
package ua.coursework.secretfolder.utils;

import android.content.Context;

import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
```

```
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class CryptoHandler {

    Cipher cipher;
    md5Calculator md5 = new md5Calculator();

    public byte[] encrypt(Context context, String data) {

        // AES = симметричный алгоритм шафрования
        // CBC = режим алгоритма AES
        // PKCS5Padding = режим обработки последних байт данных
        try {
            cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        } catch (NoSuchAlgorithmException | NoSuchPaddingException e) {
            e.printStackTrace();
        }

        String stringKey = md5.md5Apache(PINCrypter.getPin());
```

```
    byte[] byteKey16 = Arrays.copyOfRange(stringKey.getBytes(), 0, 16);

    SecretKeySpec key = new SecretKeySpec(
            byteKey16,
            "AES");
    try {
        cipher.init(Cipher.ENCRYPT_MODE, key);
    } catch (InvalidKeyException e) {
        e.printStackTrace();
    }

    byte[] preparedData = data.getBytes();
    byte[] cipherText = new byte[0];
    try {
        cipherText = cipher.doFinal(preparedData);
    } catch (BadPaddingException | IllegalBlockSizeException e) {
        e.printStackTrace();
    }
    return cipherText;
}

public String decrypt(Context context, byte[] data) {

    // AES = симметричный алгоритм шафрования
    // CBC = режим алгоритма AES
    // PKCS5Padding = режим обработки последних байт данных
    try {
```

```
    cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
} catch (NoSuchAlgorithmException | NoSuchPaddingException e) {
    e.printStackTrace();
}

String stringKey = md5.md5Apache(PINCrypter.getPin());
byte[] byteKey16 = Arrays.copyOfRange(stringKey.getBytes(), 0, 16);

SecretKeySpec key = new SecretKeySpec(
        byteKey16,
        "AES");
try {
    cipher.init(Cipher.DECRYPT_MODE, key);
} catch (InvalidKeyException e) {
    e.printStackTrace();
}

byte[] preparedData = data;
byte[] cipherText = new byte[0];

try {
    cipherText = cipher.doFinal(preparedData);
} catch (BadPaddingException | IllegalBlockSizeException e) {
    e.printStackTrace();
}

String test = new String(cipherText);
```

```
        return test;
    }
}
```

В класові CryptoHandler також описаний процес шифрування фотографій, які ми завантажуємо із галереї. Фотографії шифруються усе тим же методом шифрування AES. В якості ключа для шифрування у нас виступають перші 16 байт md5 хеша криптованого піна.

Відповідно відбувається це наступним чином:

1. Ми перетворюємо нашу картинку в бітмапу
2. Потім цю бітмапу ми кодуємо за алгоритмом Base64(це потрібно для того щоб упорядкувати хаотичні символи)
3. Далі у нас відбувається шифрування за методом AES
4. Після чого ми знову упорядковуємо наш файл за допомогою Base64 (це потрібно для можливості нормального збереження файлу і подальшої роботи з ним)

Дешифрування відповідно відбувається наступним чином:

1. У нас є прочитаний файл Base64
2. Далі ми проводимо дешифрування за методом AES іна виході отримуємо закодований файл Base64
3. Далі ми проводимо декодування Base64 і отримуємо Бітмапу
4. І власне бітмапу ми і відображаємо в нашій галереї

Підводячи короткий підсумок, збережений на виході після шифрування файл Base 64 у нас зберігається, результати самого дешифрування ми ніде не зберігаємо як тільки в оперативній пам’яті пристрою.

AndroidManifest

Це головний файл налаштувань додатку. Ту ми задаємо іконку додатків, лейбли, і тут у нас викликаються активіті (певні сценарії, за якими працює програма). Тут також прописані і базові налаштування додатку, деякі серед них слугують як DefaultActivity тобто вони за замовчуванням постійно відтворюються, одна із таких активіті – це LoginActivity яка викликається кожен раз, при вході в додаток. Тут також і прописані дозволи які які є у програмі, по своїй суті вони поділяться на безпечні і небезпечні. До безпечних відносяться ті, які відбуваються автоматично, і не поребують додаткового дозволу від користувача, як приклад можна навести можливість створення маскувальних ярликів для нашого додатку. Небезпечні дозволи, це відповідно дозволи, які потребують підтвердження від користувача. До таких дозволів можна віднести дозвіл на надання доступу до пам’яті пристрою.

```
package ua.coursework.secretfolder;

import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Base64;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
```

```
import android.view.View;
import android.widget.ProgressBar;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import com.firebase.ui.auth.AuthUI;
import com.firebase.ui.auth.IdpResponse;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.storage.FileDownloadTask;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.ListResult;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.io.ByteArrayOutputStream;
import java.io.File;
```

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.util.Arrays;
import java.util.List;

import ua.coursework.secretfolder.fragments.ViewFragment;
import ua.coursework.secretfolder.utils.CryptoHandler;
import ua.coursework.secretfolder.utils.PreferencesHandler;
import ua.coursework.secretfolder.utils.ProgressBarHelper;
import ua.coursework.secretfolder.utils.permissionsHandler;

public class GalleryActivity extends AppCompatActivity {

    AppCompatActivity activity;
    Context context;
    File mApplicationDirectory;
    File mApplicationDirectoryData;
    ProgressBarHelper progressBarHelper;
    FirebaseUser user;
    private StorageReference mStorageRef;
    private FirebaseAuth mAuth;
    private Menu menu;
    FloatingActionButton fabBtn;

    CryptoHandler cryptoHandler;
```

```
String filename = null;
String picturePath = null;

boolean isFirstGalleryOpen = true;
boolean werePermissionsGranted = true;

@Override
public void onStart() {
    super.onStart();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    context = getApplicationContext();

    cryptoHandler = new CryptoHandler();

    setContentView(R.layout.activity_main);
    openFragment(R.id.nav_host_fragment, new ViewFragment());

    clearBackStackExclusive();

    if (!permissionsHandler.checkPermissions(this, context)){
        werePermissionsGranted = false;
    } else {
```

```
        werePermissionsGranted = true;
    }

    activity = this;

    mApplicationDirectory = context.getExternalFilesDir(null);
    mApplicationDirectoryData = new File(mApplicationDirectory + "/data");

    mStorageRef = FirebaseStorage.getInstance().getReference();

    mAuth = FirebaseAuth.getInstance();
}



@Override
protected void onResume() {
    super.onResume();

    if (fabBtn == null){
        FloatingActionButton fab = activity.findViewById(R.id.fabAdd);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent i = new Intent(
```

```
                    Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                startActivityForResult(i, 699);
                isFirstGalleryOpen = true;


            }
        });
        fabBtn = fab;
    }
    fabBtn.show();

    if (!werePermissionsGranted){
        werePermissionsGranted = true;
    } else {



        if (isFirstGalleryOpen) {
            isFirstGalleryOpen = false;
        } else {
            if (permissionsHandler.checkPermissions(this, context)) {
                werePermissionsGranted = true;
                lockApp();
            } else {
                isFirstGalleryOpen = true;
            }
        }
    }
```

```
}

@Override
protected void onPause() {
    super.onPause();
}

private void lockApp() {

    Intent intent = new Intent(context, LoginActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    androidx.core.content.ContextCompat.startActivity(context, intent, null);
    this.finish();

}

@Override
protected void onStop() {
    super.onStop();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);

    mAuth.addAuthStateListener(new FirebaseAuth.AuthStateListener() {
        @Override
```

```
        public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {
            if (firebaseAuth.getCurrentUser() == null){
                restrictMenu();
            }else{
                unRestrictMenu();
            }
        }
    });

    this.menu = menu;
    return true;
}

private void restrictMenu() {
    try {
        menu.getItem(1).setEnabled(false);
        menu.getItem(2).setEnabled(false);
        menu.getItem(5).setEnabled(false);
    }catch (NullPointerException e){
        e.printStackTrace();
    }
}

private void unRestrictMenu(){
    try {
        menu.getItem(1).setEnabled(true);
        menu.getItem(2).setEnabled(true);
```

```
            menu.getItem(5).setEnabled(true);
        }catch (NullPointerException e){
            e.printStackTrace();
        }
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        FragmentManager fm = getSupportFragmentManager();
        ViewFragment fragment = (ViewFragment) fm.findFragmentById(R.id.nav_host_fragment);
        ProgressBar progressBarTemp = fragment.getProgressBar();
        progressBarHelper = new ProgressBarHelper(progressBarTemp, getApplicationContext());

        File[] files = mApplicationDirectoryData.listFiles();

        if (id == R.id.action_settings) {

            List<AuthUI.IdpConfig> providers = Arrays.asList(
                    new AuthUI.IdpConfig.Builder(AuthUI.EMAIL_PROVIDER).build(),
                    new AuthUI.IdpConfig.Builder(AuthUI.PHONE_VERIFICATION_PROVIDER).build(),
                    new AuthUI.IdpConfig.Builder(AuthUI.GOOGLE_PROVIDER).build());
```

```
            isFirstGalleryOpen = true;
            startActivityForResult(
                    AuthUI.getInstance()
                            .createSignInIntentBuilder()
                            .setAvailableProviders(providers)
                            .build(),
                    69);
            isFirstGalleryOpen = true;

            return true;

        } else if (id == R.id.action_download) {

            mStorageRef = FirebaseStorage.getInstance().getReference();
            StorageReference gsReference = FirebaseStorage.getInstance().getReferenceFromUrl("gs://secretfolder-dc714.appspot.com/" + PreferencesHandler.getValue(getApplicationContext(), "userID", "ND"));

            Task listAllTask = gsReference.listAll();

            listAllTask
                    .addOnSuccessListener(new OnSuccessListener<ListResult>() {
                        @Override
                        public void onSuccess(ListResult listResult) {

                            progressBarHelper.setMax(listResult.getItems().size());
```

```
                Log.i("Firebase Download List Success", listResult.toString());

                for (StorageReference prefix : listResult.getItems()) {

                    try {

                        String fileName = prefix.getName();

                        if (!new File(mApplicationDirectoryData + "/" +
fileName).exists()) {

                            File localFile = new File(mApplicationDirectoryData + "/" +
fileName);
                            if (!localFile.createNewFile()) {
                                break;
                            }
                            prefix.getFile(localFile).addOnSuccessListener(new
OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
                                @Override
                                public void onSuccess(FileDownloadTask.TaskSnapshot
taskSnapshot) {

                                    progressBarHelper.add();
                                    Log.i("Firebase File Download", "SUCCESS");

                                }
```

```
                        }).addOnFailureListener(new OnFailureListener() {
                            @Override
                            public void onFailure(@NonNull Exception e) {

                                progressBarHelper.add();
                                Log.e("Firebase File Download", "FAILURE: \n\n" +
e.toString());
                            }
                        });
                    } else {

                        progressBarHelper.add();
                        Log.i("Firebase Download", "FILE EXISTS");

                    }
                } catch (IOException e) {
                    progressBarHelper.abort();
                    e.printStackTrace();
                }
            }
        }
    })
    .addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {

            progressBarHelper.abort();
```

```
                Log.e("Firebase Download List failure", e.toString());
            }
        });

    } else if (id == R.id.action_upload) {

        progressBarHelper.setMax(files.length);

        for (File mFile : files) {

            Uri file = Uri.fromFile(mFile);
            StorageReference riversRef = mStorageRef.child(PreferencesHandler.getValue(getApplicationContext(), "userID", "ND") + "/" + file.getLastPathSegment());
            UploadTask uploadTask = riversRef.putFile(file);

            uploadTask.addOnFailureListener(new OnFailureListener() {

                @Override
                public void onFailure(@NonNull Exception exception) {

                    progressBarHelper.add();
                    Log.e("Firebase", "Upload onFailure");

                }
```

```
            }).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {

                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {

                    progressBarHelper.add();
                    Log.i("Firebase", "Upload onSuccess");

                }

            });
        }
    } else if (id == R.id.action_refresh) {
        Intent intent = new Intent(getApplicationContext(), GalleryActivity.class);
        startActivity(intent);
        activity.finish();
    } else if (id == R.id.logout) {
        signOut();
        delete();
        Toast.makeText(getApplicationContext(),
                "Logout successful", Toast.LENGTH_LONG).show();

    } else if (id == R.id.settings) {
        Intent intent = new Intent(getApplicationContext(), SettingsActivity.class);
        startActivity(intent);
    }
```

```
        return super.onOptionsItemSelected(item);
    }

    public void openFragment(int fragmentID, Fragment fragment) {
        FragmentTransaction tx = getSupportFragmentManager().beginTransaction();
        tx.replace(fragmentID, fragment);
        tx.addToBackStack(null);
        tx.commit();
    }

    public void clearBackStackExclusive() {
        getSupportFragmentManager().popBackStack("content_main",
getSupportFragmentManager().POP_BACK_STACK_INCLUSIVE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == 69) {
            IdpResponse response = IdpResponse.fromResultIntent(data);

            if (resultCode == RESULT_OK) {
                user = FirebaseAuth.getInstance().getCurrentUser();
                PreferencesHandler.setValue(getApplicationContext(),                "userID",
user.getUid());
```

```
            Log.i("Firebase", "Login Successful");
        } else {
            Log.e("Firebase", "Login Failed: " + response.getErrorCode());
        }
    }else if(requestCode == 699){
        if (resultCode == RESULT_OK && null != data) {

            final Uri selectedImage = data.getData();
            String[] filePathColumn = {MediaStore.Images.Media.DATA};
            Cursor cursor = getContentResolver().query(selectedImage,
                    filePathColumn, null, null, null);
            cursor.moveToFirst();
            int columnIndex = cursor.getColumnIndex(filePathColumn[0]);

            picturePath = cursor.getString(columnIndex);
            filename = picturePath.substring(picturePath.lastIndexOf("/") + 1);

            cursor.close();

            Bitmap bMap = null;
            try {
                InputStream in = getContentResolver().openInputStream(selectedImage);
                bMap                                                                    =
BitmapFactory.decodeStream(getContentResolver().openInputStream(selectedImage));
                in.close();
            } catch (IOException e) {
```

```
            e.printStackTrace();
        }

        writeFileOnInternalStorage(filename, convert(bMap));

    }
  }
}

public void writeFileOnInternalStorage(String sFileName, String sBody) {
  File dir = mApplicationDirectoryData;
  if (!dir.exists()) {
    dir.mkdir();
  }

  try {
    File gpxFile = new File(dir, sFileName);
    FileWriter writer = new FileWriter(gpxFile);
    writer.append(sBody);
    writer.flush();
    writer.close();
  } catch (Exception e) {
    e.printStackTrace();
  }
}

public void signOut() {
```

```
        AuthUI.getInstance()
                .signOut(this)
                .addOnCompleteListener(new OnCompleteListener<Void>() {
                    public void onComplete(@NonNull Task<Void> task) {
                        // ...
                    }
                });
    }

    public void delete() {
        AuthUI.getInstance()
                .delete(this)
                .addOnCompleteListener(new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        // ...
                    }
                });
    }

    public String convert(Bitmap bitmap) {
        ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
        bitmap.compress(Bitmap.CompressFormat.PNG, 100, outputStream);
        String base64 = Base64.encodeToString(outputStream.toByteArray(),
Base64.DEFAULT);
        byte[] encrypted = cryptoHandler.encrypt(context, base64);
        String encryptedString = Base64.encodeToString(encrypted, Base64.DEFAULT);
```

```
        return encryptedString;
    }


}
```

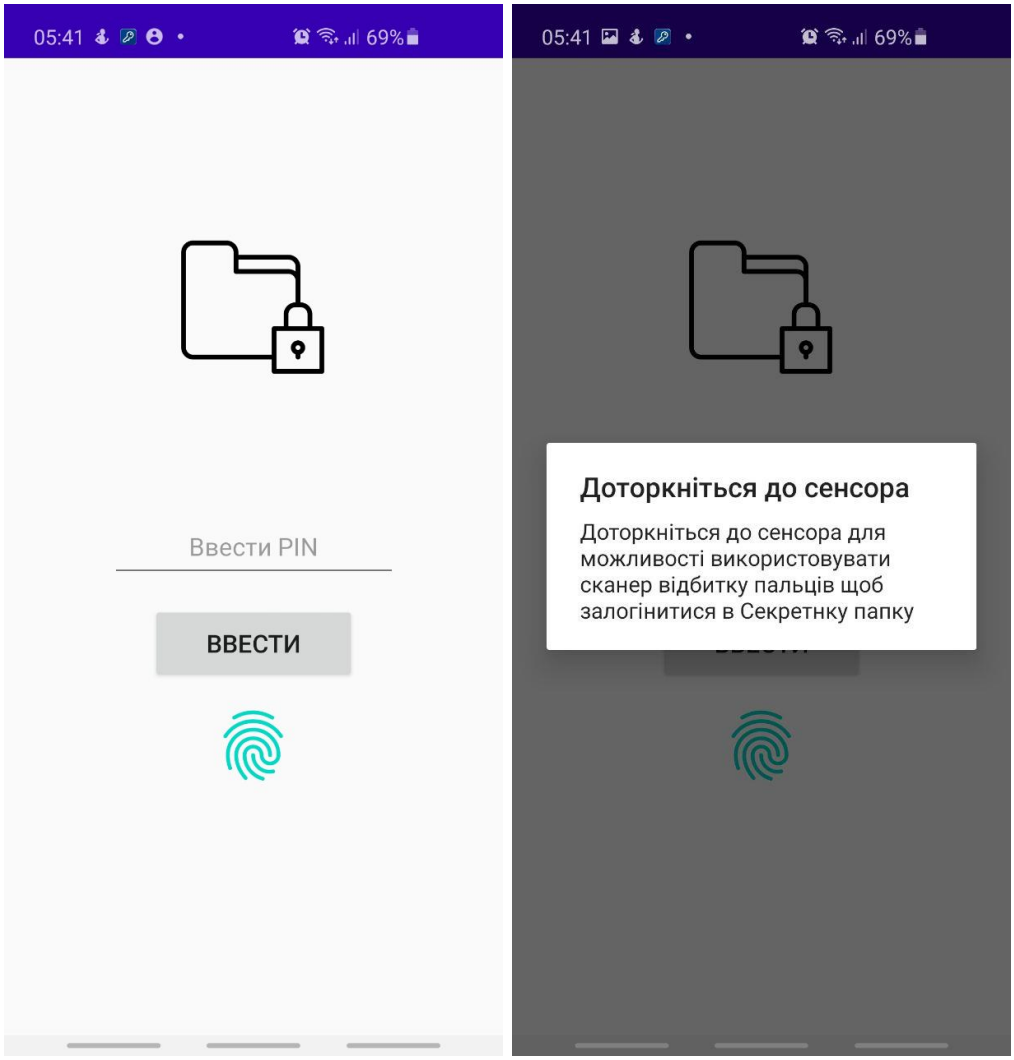
Для нормального функціонування додатку, варто розглянути також BuidGradle файл, який містить у собі усі id додатку, інформацію про версію Android про версію самого додатку і усі бібліотеки, які ми використовували в даній програмі.

```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 29
    buildToolsVersion "29.0.2"

    defaultConfig {
        applicationId "ua.coursework.secretfolder"
        minSdkVersion 26
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
```

```
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
        }
    }

}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.2.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'androidx.navigation:navigation-fragment:2.3.1'
    implementation 'androidx.navigation:navigation-ui:2.3.1'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'androidx.gridlayout:gridlayout:1.0.0'
    testImplementation 'junit:junit:4.13.1'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
    implementation 'commons-codec:commons-codec:20041127.091804'

    implementation 'com.google.firebase:firebase-storage:19.2.0'
    implementation 'com.google.firebase:firebase-auth:20.0.1'
```

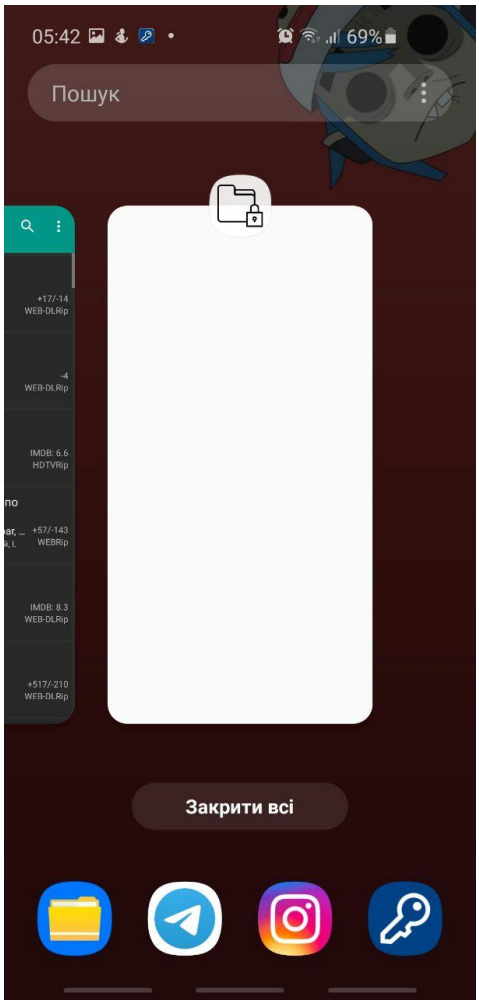
```
    implementation 'com.firebaseui:firebase-ui-auth:2.3.0'

    implementation 'com.google.android.gms:play-services:12.0.1'
    implementation 'com.google.android.gms:play-services-auth:19.0.0'

    implementation 'com.snatik:storage:2.1.0'
    implementation 'co.infinum:goldfinger:2.0.1'
    implementation 'androidx.biometric:biometric:1.0.1'
    implementation 'pub.devrel:easypermissions:1.3.0'
    implementation 'com.wei.android.lib:fingerprintidentify:1.2.6'

}
```

Коротка демонстрація додатку і його можливостей.



Реалізована можливість входу в додаток через PIN-код або по відбиткам пальців

Функція блокування



Висновок:

Проаналізувавши існуючі додатки, які є у відкритому доступі, нам удалося виявити усі плюси і недоліки даного програмного забезпечення. При розробці ж власного додатку було відтворено весь основний функціонал, що сприяє захисту інформації користувача, що представлений у додатках сторонніх розробників, та також були запропоновані і реалізовані нові підходи до забезпечення захищеності персональних даних користувача. Серед таких підходів були запропоновані: функція доступу в додаток через використання відбитку пальців, цей метод є більш захищеним та зручнішим для користування; функція унеможливлення запису відео екрану чи скріншоту під час користування додатком, це досить хороший підхід з точки зору не дати можливості зловмисникові зробити із зашифрованих даних

користувача файл-носій інформації. Також за політикою безпеки додатку, було унеможливлено транслювання екрану пристрою при користуванні додатком, це потрібно на випадок, коли користувач ненавмисно, по неуважності забув згорнути додаток, або випадково його відкрив. Функція резервного копіювання даних у хмарний сервіс, потрібна на випадок утрати смартфону, але при цьому була можливість відновити дані. Функція приховування вмісту секретної папки, при згортанні додатку, це потрібно для випадку, коли користувач згорнув додаток, зайшов у диспетчер запущених додатків і ненароком не показав вміст стороннім людям. Також однією із основних функцій є вихід із додатку і його блокування при згортанні самого додатку або блокуванні пристрою, в такому випадку, програма заново запросить ввод PIN-коду або вхід по відбитку пальців.

Наукові цінність даної розробки полягає у створенні нових підходів, які б допомогли користувачеві удосконалити захист його персональних даних.

Список літератури

1. Интернет вещей – аналитика вещей? – Режим доступу : http://channel4it.com/blogs/ Internet-veshchey-analitika-veshchey-7403.html

2. Интернет вещей. – Режим доступу : http://igate.com.ua/tag/internet-veshhej/4; http://revo lverlab.com/chto-takoe-internet-veschey

3. Адам Тернер. Интернет вещей и носимые технологии : решение тайны частной жизни и безопасности, не сорвать инноваций. – 21 Rich. – JL & Технология. – No 6 (2015), – Режим доступу : http://jolt.richmond.edu/v21i2/article6.pdf;

4. Интернет вещей. – Режим доступу: //www.Users/Home85/AppData/Local/Temp/EPIC%20 - %20%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%20%D0%B2% D0%B5%D1%89%D0%B5%D0%B9%20%28IoT%29.html

5. Будущее Интернета вещей, или Как будут управляться огромные объемы данных. – Режим доступу : http://broadcast.net.ua/show/Infrastruktura/6155-buduweeinternetaveweijilikakbudutupravli atsiaogromnyeobemydannyh13.04.2016

6. The Societal Impact of the Internet of Things. A report of a workshop on the Internet of Things organized by BCS – The Chartered Institute for IT, on Thursday 14 February 2013. The Chairs were Jeremy Crump (BCS) and Ian Brown (Oxford Internet Institute, University of Oxford). Режим доступу : https://www.bcs.org/upload/pdf/societal-impact-report-feb13.pdf

7. Интернет вещей уже в Украине. Кто заработает на роботах? – Режим доступу : http://www. epra vda.com.ua/rus/publications/2015/08/13/554178

8. е-будущее и информационное право / В. Брыжко, А. Орехов, О. Гальченко ; ред. д.ю.н., проф. Р. Калюжного и д.э.н, проф. Н. Швеца. – К. : Інтеграл, 2002. – 264 с.

“Інформація і право” No 2(17)/2016 91

9. Как технологии будут выглядеть через пять лет. – Режим доступу : http://igate.com.ua/ news/11402-kak-tehnologii-budut-vyglyadet-cherez-pyat-let

10. Интернет вещей : всё подключается к сети. – Режим доступу : http://igate.com.ua/news/ 6309-internet-veshhej-vse-podklyuchaetsya-k-seti

11. 6 новостей из мира IT, которые нужно знать сегодня. – Режим доступу : http://igate.com. ua/news/10123-6-novostej-iz-mira-it-kotorye-nuzhno-znat-segodnya

12. K. Rose, S. Eldridge, L. Chapin The Internet of Things : An Overview. Understanding the Issues and Challenges of a More Connected World / The Internet Society (ISOC), October 2015. – 50 P. – Режим доступу : http://www.internetsociety.org/sites/default/files/ISOC-IP-Overview-201510 22.pdf

13. Charlie Hawes. Hogan Lovells assists Internet of Things policy group in Brussels, 28 October 2015. – Режим доступу : http://www.hlmediacomms.com/2015/10/28/hogan-lovells-assists-internet- of-things-policy-group-in-brussels

14. Интернет вещей : чем угрожает будущее. – Режим доступу : http://igate.com.ua/news/31 69-internet-veshhej-chem-ugrozhaet-budushhee

15. Как в 2015 году был взломан Интернет вещей. – Режим доступу : http://igate.com.ua/ news/12342-kak-v-2015-godu-byl-vzloman-internet-veshhej

16. Eric Barbry. The Internet of Things, Legal Aspects: What Will Change (Everything) / Communications & Strategies, No. 87. – Pp. 83-100. – Quarter 2012. – Режим доступу : http://pape rs.ssrn.com/sol3/papers.cfm?abstract_id=2304137

17. S. Chen et al. A Vision of IP : Applications, Challenges and Opportunities With China Perspective IEEE / Internet of Things Journal, vol. 1, No. 4. – Pp. 349-359. – August 2014. – Режим доступу : http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6851114

18. Recommendation CM/Rec(2010)13 of the Committee of Ministers to member states on the protection of individuals with regard to automatic processing of personal data in the context of profiling (Adopted by the Committee of Ministers on 23 November 2010 at the 1099th meeting of the Ministers' Deputies). – Режим доступу : https://search.coe.int/cm/Pages/result_details.aspx?ObjectID=09000016805cdd00

19. Internet of Things: Privacy & Security in a Connected. – World Federal Trade Commission (FTC) Staff Report. – January 2015. – Режим доступу : https://www.ftc.gov/system/files/documents /reports/federal-trade-commission-staff-report-november-2013-workshop-entitled-internet-things-priva cy/150127IPrpt.pdf

20. Системна інформатизація правоохоронної діяльності : європейські нормативно-правові акти та підходи до упорядкування інформаційних відносин у зв'язку з автоматизованою обробкою даних : посіб. / В. Брижко, М. Швець [та ін.]. – Кн. 2. – К. : ТОВ "ПанТот", 2006. – 509 с.

21. Convention for the Protection of Individuals with regard to Automatic Processing of Personal Data Amendment to Convention ETS No. 108 allowing the European Communities to accede. – Strasbourg, 28.1.1981. – (Article 5 – Quality of data). – Режим доступу : http://www.convention. coe.int/treaty/en/Treaties/Html/108.htm

22. Права человека и защита персональных данных / А. Баранов. В. Брыжко, Ю. Базанов. –(Финансовая помощь и содействие в издании Харковской правозащитной группы и Национального фонда поддержки демократии (США). – Харьков : Фолио, 2000. – 280 с.

23. On the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) : Regulation (EU) 2016/679 of the European Parlament and of the council of 27 April 2016. – Режим доступу : http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN

24. Про захист персональних даних : Закон України від 01.06.10 р. No 2297-VI ; із змін. та доп. // Відомості Верховної Ради України (ВВР). – 2010. – No 34. – Ст. 481