

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет ракетно – космічної техніки

Кафедра вищої математики та системного аналізу

Пояснювальна записка до дипломної роботи

магістра

(освітньо-кваліфікаційний рівень)

на тему «Розробка iOS-додатку для безпечного зберігання контактних даних»

ХАІ.405.463м.124.1404037.190

Виконав: студент 6 курсу групи № 463м
спеціальності
124 «Системний аналіз»

(шифр і назва спеціальності)

Могозіна К. В.

(прізвище й ініціали студента)

Керівник: Макарічев В. О.

(прізвище й ініціали)

Рецензент: Колодяжний В. М.

(прізвище й ініціали)

Харків – 2019

НАЦІОНАЛЬНИЙ АЕРОКОСМІЧНИЙ УНІВЕРСИТЕТ
ім. М.Є. ЖУКОВСЬКОГО
«Харківський авіаційний інститут»

Факультет ракетно-космічної техніки
Кафедра вищої математики та системного аналізу
Освітнього-кваліфікаційний рівень: магістр
Напрямок підготовки 124 «Системний аналіз»

ЗАТВЕРДЖУЮ

Завідувач кафедри,
професор, д.ф.-м.н, О.Г.Ніколаєв

«__» _____ 20__ року.

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ
Могозіній Катерині Володимирівні
(прізвище та ініціали)

1. Тема проекту (роботи) Розробка iOS-додатку для безпечного зберігання контактних даних

керівник проекту(роботи) к.ф.-м.н., доцент каф. 405 В. О. Макарічев
(прізвище ,ім'я,по-батькові,науковий ступінь,вчене звання)

затверджені наказом вищого навчального закладу від

2. Строк подання студентом проекту (роботи) _____

3. Вихідні дані до проекту (роботи): створення програмного продукту: мобільного додатку, для безпечного зберігання контактних даних.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібні розробити):

1. Огляд апріорної інформації безпечного зберігання даних.
2. Системний аналіз розробки безпечного мобільного додатку.
3. Аналіз існуючих рішень інформаційної безпеки.
4. Проектування мобільного додатку.
5. Розробка мобільного додатку.
6. Оцінка економічної ефективності програмного продукту.

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

6. Консультанти розділів проекту (роботи)

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

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Огляд апріорної інформації безпечного зберігання даних		
2	Системний аналіз розробки безпечного мобільного додатку		
3	Аналіз існуючих рішень інформаційної безпеки		
4	Проектування мобільного додатку		
5	Розробка мобільного додатку		
6	Оцінка економічної ефективності програмного продукту		

Студент _____
 (підпис) (прізвище та ініціали)

Керівник проекту(роботи) _____
 (підпис) (прізвище та ініціали)

Нормоконтролер _____
 (підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи: 64 ст., 18 рис, 16 табл., 44 джерела, додаток і глосарій.

ДАНИ, БЕЗПЕКА ДАНИХ, ІНФОРМАЦІЙНА БЕЗПЕКА, МОБІЛЬНИЙ ДОДАТОК, ІНФОРМАЦІЯ.

Об'єкт дослідження – мобільні додатки.

Предмет дослідження – розробка безпечного мобільного додатку.

Мета роботи – розробка програмного забезпечення для безпечного зберігання контактних даних.

Методи дослідження – аналіз інформаційних джерел про проведені дослідження, методи системного аналізу об'єкта дослідження та теорія алгоритмів.

В сучасному суспільстві саме інформація стає найважливішим стратегічним ресурсом, основною виробничою силою, що забезпечує його подальший розвиток. Ось чому задача захисту інформації набуває все більшої актуальності.

У роботі описані підходи до захисту інформації та розробки безпечних мобільних додатків, на основі створених рекомендацій розроблено мобільний додаток для платформи iOS.

ABSTRACT

The explanatory note to the graduation project contains: 64 pages, 18 pictures, 16 tables, 1 appendix, 44 sources.

DATA, DATA SECURITY, INFORMATION SECURITY, MOBILE APPENDIX, INFORMATION.

Object of research – mobile applications.

Subject of research – developing a secure mobile app.

The purpose of the work is the software development for secure contact data storage.

Methods of research – analysis of information sources about the conducted research, methods of system analysis of the object of research and the theory of algorithms.

In today's society, information itself becomes the most important strategic resource, the main productive force, which ensures its further development. That is why the task of protecting information is becoming increasingly relevant.

The work describes approaches to protecting information and developing secure mobile applications, and based on the recommendations created, developed a mobile application for the iOS platform.

ЗМІСТ

СКРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ.....	9
ВСТУП.....	10
1 ОГЛЯД АПРІОРНОЇ ІНФОРМАЦІЇ	12
1.1 Актуальність завдання інформаційної безпеки	12
1.2 Завдання інформаційної безпеки.....	12
1.3 Методи аналізу захищеності мобільних додатків	13
1.4 Результати огляду і виділенні проблеми	14
1.5 Цілі і завдання дослідження.....	14
2 СИСТЕМНИЙ АНАЛІЗ РОЗРОБКИ БЕЗПЕЧНИХ МОБІЛЬНИХ ДО-	
ДАТКІВ.....	16
2.1 Об'єкт дослідження	16
2.2 Морфологічний опис об'єкта	16
2.3 Функціональний опис системи	17
2.4 Інформаційний опис	19
2.5 Класифікація системи	20
3 КОНЦЕПЦІЇ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ	23
3.1 Вибір мови програмування	24
3.1.1 Objective - C	24
3.1.2 Swift	25
3.1.3 Порівняння розглянутих мов програмування	25
3.2 Вибір архітектурного паттерну	28
3.2.1 MVC.....	28
3.2.2 Координатор	29
3.3 Core Data.....	30

	7
3.4 Keychain.....	31
3.5 Додаткові методи захисту	31
3.6 Архітектура додатку	32
3.6.1. Рівень бази даних	32
3.6.2 Рівень розпізнання телефонних номерів	32
3.6.3 Рівень презентації	33
3.6.4 Додаткова логіка	34
3.7 Результати розробки	35
4 ЕКОНОМІЧНА ЧАСТИНА.....	39
4.1 Опис програмного продукту	39
4.2 Аналіз ринків збуту.....	39
4.3 Визначення трудовитрат робіт	39
4.4 Перелік робіт для створення програмного продукту	40
4.5 Склад виконавців роботи та розрахунок заробітної плати.....	42
4.6 Розрахунок витрат на матеріал, комплектуючі, оренду офісу та електроенергію	44
4.7 Розрахунок собівартості та повної вартості програмного продукту	46
4.8 Альтернативний процес розробки програмного продукту.....	48
4.9 Склад виконавців роботи та розрахунок заробітної плати.....	50
4.10 Розрахунок витрат на матеріал, комплектуючі, оренду офісу та електроенергію	51
4.11 Розрахунок собівартості та повної вартості програмного продукту	52
Висновок:	54
ВИСНОВКИ	55
ГЛОСАРІЙ	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	57

ДОДАТОК А.....	61
А.1. ВСТУП.....	61
А.1.1 Найменування програми	61
А.1.2 Призначення та область застосування	61
А.2. ВИМОГИ ДО ПРОГРАМИ	61
А.2.1 Вимоги до функціональних характеристик.....	61
А.2.2 Вимоги до нефункціональних характеристик.....	62
А.3. УМОВИ ЕКСПЛУАТАЦІЇ	63
А.3.1 Кліматичні умови експлуатації.....	63
А.3.2 Вимоги до складу та параметрів технічних засобів	63
А.3.3 Вимоги до вихідного коду та мов програмування.....	63
А.4. ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	63
А.4.1 Попередній склад програмної документації	63
А.5. ТЕХНІКО-ЕКОНОМІЧНІ ПОКАЗНИКИ	63
А.5.1 Економічні переваги розробки	63
А.6. СТАДІЇ ТА ЕТАПИ РОЗРОБКИ.....	63
А.6.1 Стадії розробки.....	63
А.6.2 Етапи розробки	63

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

БД – база даних.

ТЗ – технічне завдання.

SDK – Software Development Kit.

ВСТУП

В сучасному світі досить швидко розвивається автоматизація інформаційних систем, заснована на використанні комп'ютерних і телекомунікаційних засобів. Це призводить до того, що люди довіряють персональні дані технологіям, і саме тому **інформаційна безпека** набуває особливої значущості.

Основні цілі інформаційної безпеки – це захист персональних даних.

Таким чином, **об'єкт дослідження** – мобільні додатки.

Предмет дослідження – розробка безпечного мобільного додатку.

Мета роботи – розробка програмного забезпечення для безпечного зберігання контактних даних.

Методи дослідження – аналіз інформаційних джерел про проведені дослідження, методи системного аналізу об'єкта дослідження та теорія алгоритмів.

Задачі роботи:

1. визначити необхідні інструменти для розробки;
2. визначити та обрати оптимальні методи захисту даних;
3. розробити мобільний додаток для безпечного зберігання контактних даних.

Результатом дипломної роботи є розроблений мобільний додаток, що дозволяє безпечно зберігати контактні дані.

У **першому розділі** зроблено огляд апріорної інформації. Приведені ключові поняття інформаційної безпеки, проаналізовано стан досліджень на тему даної роботи. За результатом аналізу інформаційних джерел були визначені основні проблеми технологічної безпеки, актуальні на сьогодні.

У **другому розділі** приведено аналіз об'єкта дослідження. Наведені докази того, що об'єкт є об'єктом з точки зору системного аналізу. Приведені структурний, функціональний, інформаційний види аналізу досліджуємого об'єкту. Всі результати описані у таблицях. Проведена класифікація об'єкта, наведено основні ознаки.

У **третьому розділі** описано процес проектування і розробки програмного забезпечення, методи захисту інформації. Було прийнято рішення про використання Keychain та мови Swift, як основних інструментів при розробці додатку. Обґрунтовано вибір БД та вибір архітектури програмного забезпечення. Також представлені результати розробки ПЗ.

У **четвертому розділі** представлено оцінку економічної ефективності програмного продукту. Проведено аналіз двох варіантів команд, визначені трудовитрати робіт створення програмного продукту, призначені виконавці роботи, а також проведені розрахунки заробітної плати та соціального внеску. Також складено перелік робіт для кожного з виконавців, складено перелік необхідного обладнання, для яких розраховані витрати на амортизацію та проведений розрахунок вартості електроенергії: технологічної та освітлювальної. Проведені розрахунки собівартості, прибутку та повної вартості розробленого програмного продукту.

У **висновку** представлені результати дипломної роботи – складене ПЗ і описані отримані для цього результати. Так само описані перспективи подальшого застосування результатів даної дипломної роботи.

1 ОГЛЯД АПРІОРНОЇ ІНФОРМАЦІЇ

1.1 Актуальність завдання інформаційної безпеки

Мобільні телефони в сучасному світі є не просто засобом зв'язку, а пристроєм, який містить вразливі персональні дані, несанкціонований доступ до яких може привести до непередбачуваних результатами. На даний момент сучасні засоби захисту не дозволяють в повній мірі розв'язувати питання безпеки мобільних систем і оцінити можливі ризики потенційних злочинних дій.

При розробці мобільного додатку слід враховувати, що дані, якими оперує цей додаток, можуть представляти цінність для третіх осіб. Ступінь важливості даних варіюється в широких межах, проте, навіть найбільш проста приватна інформація, наприклад, номер телефону чи адреса, вимагає збереження конфіденційності.

Перші повідомлення про факти несанкціонованого доступу до інформації (рисунок 1.1) були пов'язані, в основному, з хакерами, або «електронними розбійниками». Останнім десятиліттям порушення захисту інформації прогресує з використанням програмних засобів і через глобальну мережу Інтернет. Досить поширеною загрозою інформаційної безпеки стало також зараження систем так званими вірусами.

Наведений огляд показує **актуальність** дослідження питання інформаційної безпеки.

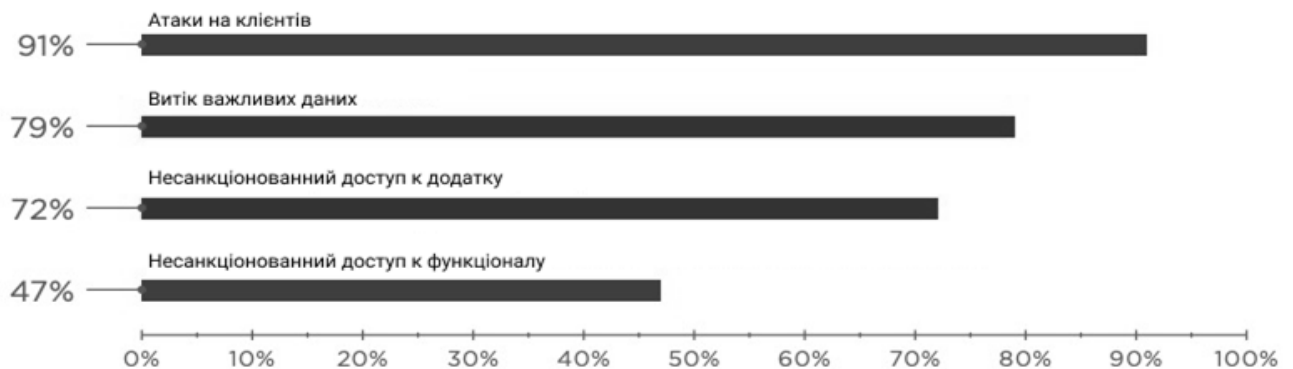


Рисунок 1.1 – Найбільш поширені загрози

1.2 Завдання інформаційної безпеки

Інформаційна безпека — це захищеність інформації від доступу сторонніх осіб, що можуть нанести шкоди власникам або користувачам інформації. Інформаційна безпека не тільки про захист інформації. Суб'єкт може постраждати не тільки від вкрадених даних, але й від не санкційного доступу, що може призвести до поломки системи.

Основні задачі інформаційної безпеки:

- своєчасне виявлення і усунення загроз даним і ресурсам, причин і умов, що можуть нанести матеріальні, моральні чи фізичні збитки;
- створення механізму і умов оперативного реагування на будь які загрози даним.

1.3 Методи аналізу захищеності мобільних додатків

Існують різні методи оцінювання загроз **інформаційній безпеці** в додатках для мобільних систем, які застосовуються як окремо, так і в сукупності. Розділити їх можна на дві великі категорії: статистичні і динамічні.

В якості методів динамічного аналізу використовуються:

- стрес-тестування;
- аналіз мережевого трафіку мобільного додатка;
- аналіз пам'яті програми;
- аналіз взаємодії додатка з файловою системою.

До методів статистичного аналізу відносяться:

- аудит безпеки коду програми;
- Reverse Engineering;
- дизасемблювання;
- декомпіляція.

Для комплексної оцінки стану захищеності мобільної системи необхідно дослідити три складових: клієнтську частину, серверну частину і безпосередньо

канал зв'язку. Для цього застосовують такі методи:

- комплексний аналіз архітектури клієнтської і серверної частини програми;
- моделювання загроз відповідно до логіки додатка;
- проектування моделі.

1.4 Результати огляду і виділенні проблеми

Проведений огляд і аналіз наявної інформації дозволив визначити, що одна з основних **проблем** досліджуваної області полягає у *відсутності вказівок по методам захисту мобільних додатків*.

Ключову роль при обиранні методів захисту мобільного додатку грають способи отримання даних (власноруч, з серверу) та специфіка даних.

1.5 Цілі і завдання дослідження

Об'єкт дослідження – мобільні додатки.

Предмет дослідження – розробка безпечного мобільного додатку.

Виходячи з існуючих проблем інформаційної безпеки мобільних додатків, була сформульована мета. **Мета дослідження** – розробка програмного забезпечення для безпечного зберігання контактних даних.

Спираючись на сформульовану мету і предмет дослідження, сформульовані **задачі дипломної роботи**:

- визначити необхідні інструменти для розробки;
- визначити та обрати оптимальні методи захисту даних;
- розробити мобільний додаток для безпечного зберігання контактних даних.

В ході виконання дипломної роботи будуть використані наступні методи дослідження:

- аналіз інформаційних джерел про проведені дослідження в сфері інформаційної безпеки;
- методи системного аналізу об'єкта дослідження;
- теорія алгоритмів.

2 СИСТЕМНИЙ АНАЛІЗ РОЗРОБКИ БЕЗПЕЧНИХ МОБІЛЬНИХ ДОДАТКІВ

Вхідними даними для розробки безпечного мобільного додатку є інформація про специфіку даних, які потребують захисту. Наприклад, для додатку по розпізнаванню інформації на візитних картках такими даними будуть номери телефонів, адреса та ФІО.

2.1 Об'єкт дослідження

Об'єктом дослідження дипломного проекту була обрана розробка безпечного мобільного додатку. У термінах системного аналізу предмет розглядається як перетворювач вхідних величин у вихідні. Можна виділити:

1. вхідні змінні: інформація про специфіку даних;
2. вихідні змінні: мобільний додаток.

2.2 Морфологічний опис об'єкта

Виділимо загальну структуру розробки безпечного додатку (рисунок 2.2):

1. Розробка безпечного мобільного додатку:
 - 1.1. Процес створення ТЗ на основі вхідних даних.
 - 1.2. Процес обирання методів захисту на основі ТЗ.
 - 1.3. Процес розробки.
 - 1.4. Процес тестування.

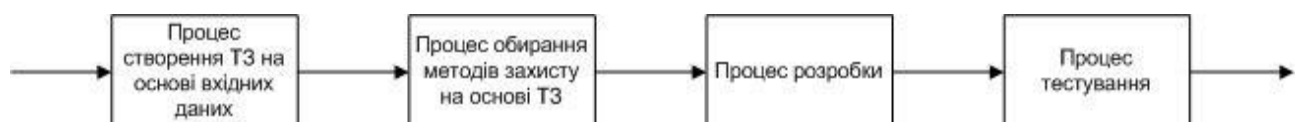


Рисунок 2.1 – Структурна схема об'єкта дослідження

Об'єкт є системою, так як він володіє наступними ознаками:

- емерджентність - функції системи відмінні від функцій її складових;

- цілісність - помилка у виконанні кожного з етапів призведе до помилки в визначення типу об'єкта;
- прогресуюча факторизація - на кожному з етапів виконується вузькоспеціалізоване завдання.

Тип елементного складу системи гомогенний, елементи інформаційні. Між елементами існують як інформаційні, так і прямі зв'язки. Структура системи лінійна.

2.3 Функціональний опис системи

Функції елементів системи представлені в таблиці 2.1:

Таблиця 2.1 – Функціональний опис

Код	Елемент	Функція
1	Розробка безпечного мобільного додатку	Перетворення інформації о специфіці даних у безпечний мобільний додаток
1.1	Процес створення ТЗ на основі вхідних даних	Виявлення специфіки вхідних даних
1.2	Процес обирання методів захисту на основі ТЗ	Обиранням методів для захисту даних
1.3	Процес розробки	Розробка мобільного додатку
1.4	Процес тестування	Тестування додатку та відладка помилок

Параметри елементів системи вказані в таблиці 2.2:

Таблиця 2.2 – Параметри системи

Код	Елемент	Параметри
1	Розробка безпечного мобільного додатку	Специфіка даних Час розробки
1.1	Процес створення ТЗ на основі вхідних даних	Специфіка даних Необхідна ступінь захисту
1.2	Процес обирання методів захисту на основі ТЗ	ТЗ Час розробки Необхідна ступінь захисту
1.3	Процес розробки	Час розробки Методи захисту
1.4	Процес тестування	Необхідна якість

Загальні характеристики системи описані в таблиці 2.3:

Таблиця 2.3 – Загальні характеристики систем

Функціональність	Ранг	Фактори	
		Системоруйнуючі	Системоформуючі
Однофункціональна	Обслуговування	Невірний формат вхідних даних Недостатня якість Недостатній час розробки	Відповідні вхідні дані Достатня якість Достатній час розробки

2.4 Інформаційний опис

1. елементами системи є наступні етапи алгоритму:

- процес створення ТЗ на основі вхідних даних;
- процес обирання методів захисту на основі ТЗ;
- процес розробки;
- процес тестування.

2. властивості елементів, таблиця 2.4:

Таблиця 2.4 – Властивості елементів системи

№ з/п	Наіменовання	Означення	Кількість властивостей	Примітка
1	Процес створення ТЗ на основі вхідних даних	a_1	1	1 (1) - забезпечує створення ТЗ
2	Процес процес обирання методів захисту на основі ТЗ	a_2	1	1 (2) - забезпечує обирання методів
3	Процес розробки	a_3	1	1 (3) – забезпечує розробку додатку
4	Процес тестування	a_4	1	1 (4) - забезпечує якість додатку

3. Середньогометричне число властивостей на один елемент:

$$\bar{a} = \sqrt[4]{a_1 * a_2 * a_3 * a_4} = \sqrt[4]{1 * 1 * 1 * 1} = \sqrt[4]{1} = 1$$

4. Структура об'єкта – лінійна.

5. Зв'язки системи між елементами:

а) Перетворюючі:

1. навколишнє середовище – створення ТЗ;
2. створення ТЗ – обирання методів;
3. обирання методів – розробка додатку;
4. розробка додатку – збільшення якості додатку;
5. збільшення якості додатку - навколишнє середовище.

6. Кількість зв'язків на елемент, таблиця 2.5:

Таблиця 2.5 – Кількість зв'язків на елемент.

№	Назва елементі	Кількість зв'язків
1	Процес створення ТЗ на основі вхідних даних	2
2	Процес процес обирання методів захисту на основі ТЗ	2
3	Процес розробки	2
4	Процес тестування	2

7. Середньгеометричне число зв'язків на елемент:

$$\bar{Y} = \sqrt[4]{2 * 2 * 2 * 2} = \sqrt[4]{16} = 2$$

2.5 Класифікація системи

Класифікація системи представлена в таблиці 2.6:

Таблиця 2.6 – Класифікація системи

№ п/п	Признак класифікації	Тип системи по признаку	Определение
1	За зв'язком системи з навколишнім середовищем	Відкрита	В процесі діяльності обмінюється із середовищем інформацією
2	За походженням	Штучна	Створена людиною
3	За об'єктивністю існування	Абстрактна	Модель реального об'єкту
4	За типом опису законів функціонування	Система типу «білий ящик»	Повністю відомі закони функціонування
5	По способу управління системою	Без зворотнього зв'язку	Строго підпорядковується керованим діям і не може змінювати управління при зміні зовнішніх умов
6	По централизації	Децентралізована	Немає головного елемента
7	За однорідністю структури	Різнорідна	Елементи не володіють властивістю взаємозамінності
8	За типом складності	Проста	Має мало елементів і лінійні зв'язки

Продовження таблиці 2.6

9	За вимірністю	Багатовимірна	Входів і виходів більше одного
10	За лінійністю	Нелінійна	Не описується лінійними рівняннями
11	За неперервністю	Дискретна	Містить хоча б один дискретний елемент
12	За обумовленістю дії	Казуальна	Мета внутрішньо не властива

3 КОНЦЕПЦІЇ РОЗРОБКИ БЕЗПЕЧНИХ МОБІЛЬНИХ ДОДАТКІВ

Кожного року операційна система iOS обходить свого головного суперника, Android, за ступенем інформаційної безпеки. Під безпекою мається на увазі, наскільки важко стороннім особам отримати доступ до персональних даних користувача системи. Це пов'язано з тим, що система iOS має закритий код, також вона більш прискіплива до програм, які користувач встановлює. А система Android є опенсорсною, тобто кожен має доступ до її коду, а це полегшує пошук вразливостей. Також Android має значно більше актуальних версій ОС, тому їх важче покрити захистом.

Проте кількість коли-небудь знайдених вразливостей у 50 найпопулярніших операційних систем, браузерів і інших продуктів за станом на 11 березня 2019 року:

- Android - 2146 вразливостей, 3 місце;
- iOS - 1514 вразливостей, 7 місце.

Саме тому було прийнято рішення розробляти мобільний додаток для операційної системи iOS. Дана система було створена у 2007 році для мобільного пристрою iPhone.

iOS розроблена на основі операційної системи OS X (з того часу перейменованої в macOS) і використовує той же набір основних компонентів Darwin, сумісний зі стандартом POSIX.

Ця операційна система оптимізована під невеликий екран та має сенсорне управління пальцем. Процес написання і налагодження програм полегшують бібліотеки розроблені Apple - iPhone SDK. Це пакет з декількох програм, які можуть знадобитися для створення додатка на iPhone. В тому числі:

- Середовище розробки Xcode - написання коду, налагодження, емуляція роботи iPhone для тестування додатку.
- Interface Builder - створення інтерфейсу програми, зв'язок між інтерфейсом користувача і кодом програми.

- Instruments - аналіз навантаження на процесор; роботи з пам'яті і аналіз коду в кожен момент його виконання для подальшого аналізу і оптимізації.

3.1 Вибір мови програмування

Для розробки мобільних додатків під операційну систему iOS існує дві мови програмування: Objective-C та Swift.

3.1.1 Objective - C

Objective-C - об'єктно-орієнтована мова програмування, побудована на основі мови C, і парадигм Smalltalk. Ця мова використовується корпорацією Apple, для операційних систем OS X і iOS, і їх програмних інтерфейсів сумісних з Cocoa і Cocoa Touch. Файли з кодом реалізації на мові Objective-C має розширення .m, а заголовки та файли інтерфейсів мають розширення .h, це взято з C. Файли з кодом реалізації на мові Objective-C ++ має розширення .mm.

Objective-C було розроблено Бредом Коксом на початку 1980-х в компанії Stepstone.

Однією з відмінних рис Objective-C є його динамічність: рішення, що у інших мовах програмування приймаються на етапі компіляції, у цій мові відкладаються до етапу виконання.

Прикладом динамічності є й той факт, що Objective-C оперує повідомленнями, а не функціями як інші мови. Об'єктам можна послати повідомлення, навіть якщо воно не належить даному об'єкту. Далі об'єкт може передати отримане повідомлення іншому об'єкту.

Також дана мова програмування має підтримку протоколів (це значить, що об'єкт та інтерфейс чітко розділені). Підтримує спадкування (проте не множинне, для реалізації множинного спадкування можна використовувати протоколи). Об'єкт може бути успадкованим від іншого об'єкта і одночасно підтримувати декілька протоколів.

3.1.2 Swift

Swift – це відносно нова мова програмування, що було створена для розробки додатків під iOS та OS X. Ця мова була основана на C та Objective-C, взяла краще від них та позбавлена деяких обмежень. У Swift використовуються сучасні практики та патерни програмування, також додані сучасні функції, що перетворюють створення додатка в простий, більш гнучкий процес. Swift працює з фреймворками Cocoa і Cocoa Touch і сумісний з основною кодовою базою Apple, написаної на Objective-C.

Не дивлячись на те, що Swift побудований на основі Objective-C, він визначається не вказівниками, а типами змінних, які обробляє компілятор. У той же час, він надає розробникам багато функцій, які раніше були доступні в C ++ і Java, такі як визначаються найменування, так звані узагальнення і перевантаження операторів.

Одним з плюсів даної мови є й те, що вона досі розвивається. Кожні декілька років Apple представляє нову версію Swift, яка має більше зручних та цікавих підходів, та позбавляється застарілих практик.

Також Swift значно швидший за найпопулярніші мови програмування, наприклад, згідно з офіційною документацією Apple алгоритм пошуку у глибину виконується на Swift у 2.6 разів швидше, ніж на Objective-C, та у 8.4 разів швидше ніж на Python 2.7.

3.1.3 Порівняння розглянутих мов програмування

Порівняння приведених мов програмування наведено у таблиці 3.1.

Таблиця 3.1 – Порівняння мов програмування

Критерій	Swift	Objective-C
Синтаксис	Дана мова взяла в себе найкращі практики та підходи вже існуючих мов програмування (JavaScript, Go, Ruby, Python), тому має легкий та досить схожий на інші синтаксис.	Має досить високий рівень входження через специфіку синтаксису, основу на вже не популярному SmallTalk.
Актуальність	Продовжує розвиток, кожні декілька років виходить нова версія з підтримкою найпопулярніших парадигм програмування та позбавленням застарілих підходів.	Підтримує нові бібліотеки, проте вже досить давно не оновлювалось.
Швидкість	Після проведення порівняльних експериментів на основних колекціях даних мов (Swift: Array, Dictionary, Set; Objective-C: NSArray, NSDictionary, NSSet), можна зробити висновок, що масив (рисунок 3.1) робить швидше у Objective-C, проте словники (рисунок 3.2) та сети (рисунок. 3.3) показали кращі результати у Swift.	

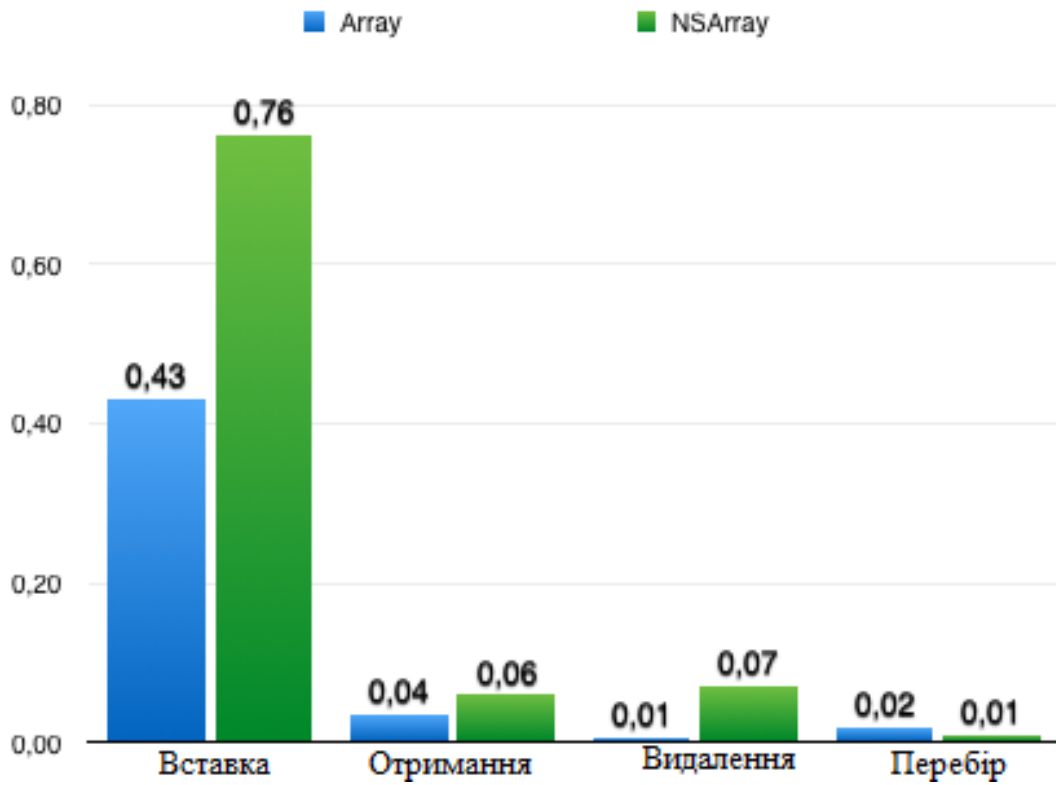


Рисунок 3.1 – Порівняння Array та NSArray

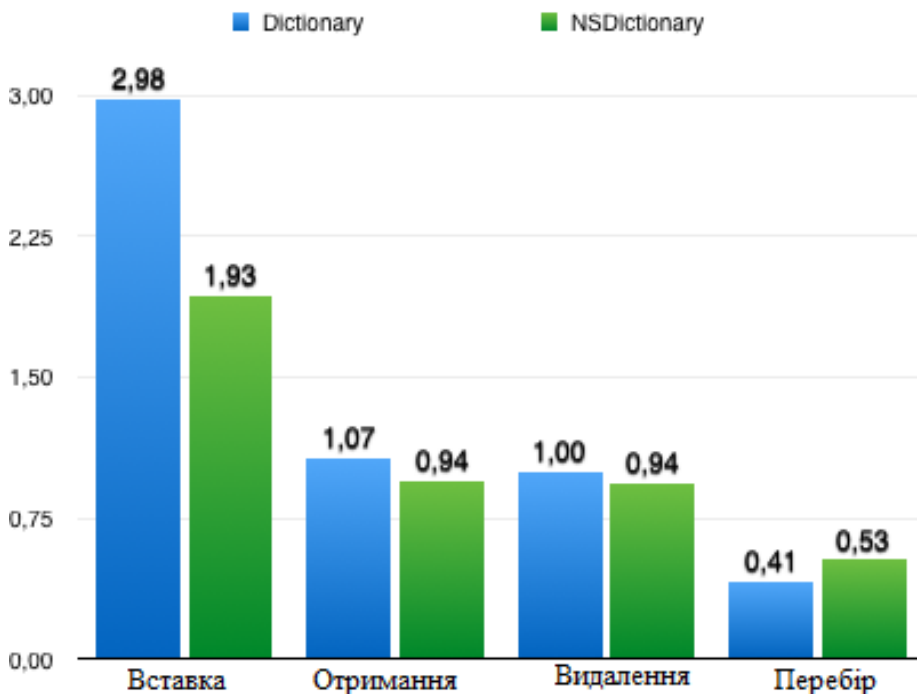


Рисунок 3.2 – Порівняння Dictionary та NSDictionary

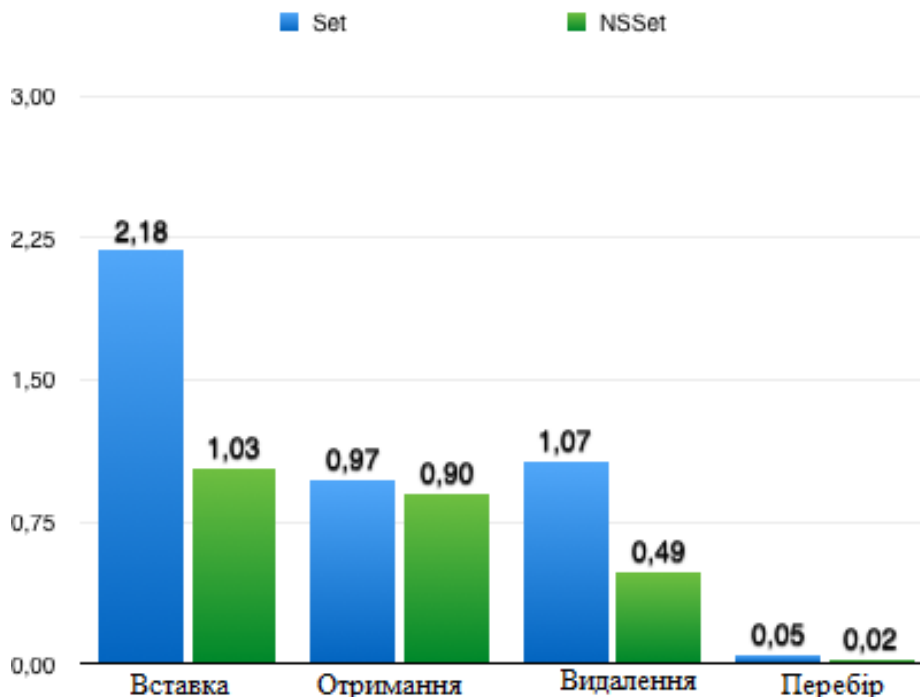


Рисунок 3.3 – Порівняння Set та NSMutableSet

На основі даних, приведених у таблиці 3.1, можна зробити такі висновки:

1. Swift має більш гнучкий та сучасний синтаксис.
2. Swift більш актуальний та сучасний ніж Objective-C.
3. На двох з трьох основних колекцій Swift виявився значно швидкішим ніж Objective-C.

Саме тому, у якості мови розробки для написання мобільного додатку обираємо Swift.

3.2 Вибір архітектурного паттерну

3.2.1 MVC

Паттерн MVC (Model View Controller) має дві варіації: традиційну та розроблену Apple (Рисунок 3.4). Через те, що обирається архітектурний патерн для розробки мобільного додатку під платформу iOS, розглянемо другий варіант.

Дана архітектура складається з трьох рівнів:

1. Models – рівень, який маніпулює даними;

2. Views – рівень, відповідальний за рівень представлення (UI); у випадку розробки під iOS, це все, що починається з префікса UI;

3. Controller / Presenter / ViewModel – рівень, що виступає у ролі посередника між Model і View; вносить зміни у Model, як реакцію на дії користувача, виконані на View, чи навпаки оновлює View через зміни у Model.

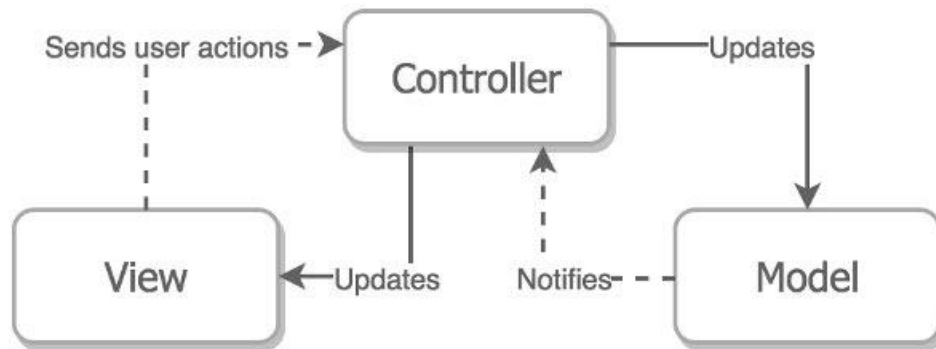


Рисунок 3.4 – MVC згідно з Apple

Проте у данній архітектурі є істотний недолік, через те, що вся відповідальність за бізнес-логіку та навігацію знаходиться у Controller, він стає занадто великим. А це у свою чергу призводить до того, що проект стає важко підтримувати та тестувати.

3.2.2 Координатор

Ще один досить розповсюджений патерн – координатор. У основі цього патерну лежить концепція UserStories, яка полягає у тому, що додаток ділиться на логічні одиниці «історії». Далі кожна історія розподіляється на певні рівні:

1. Координатор (Coordinator) – рівень, що відповідає за навігацію у середині історії, а також між різними історіями.
2. Роутер (Router) – рівень, що відповідає за анімацію появи нової історії.
3. ViewModel – потрібна для зв'язку між координатором та представленням.

4. View – рівень, відповідальний за рівень представлення (UI); у випадку розробки під iOS, це все, що починається з префікса UI.
5. Models – рівень, який маніпулює даними.

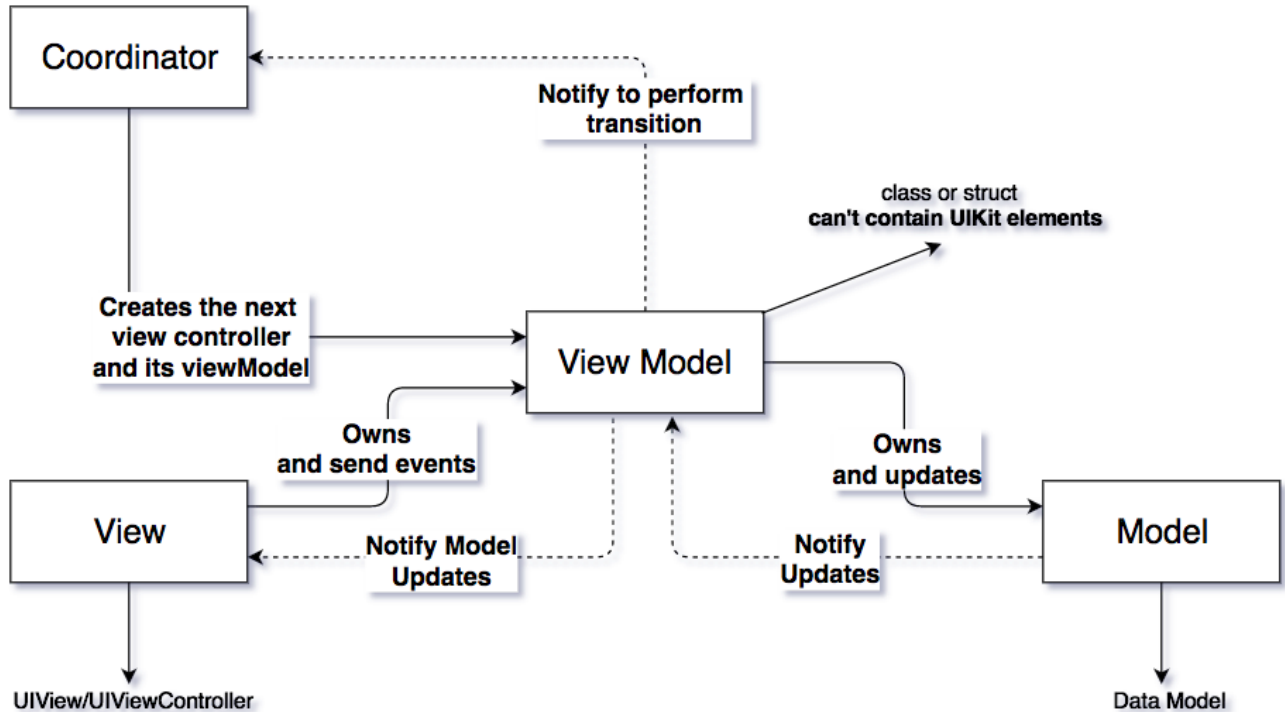


Рисунок 3.5 – Патерн Coordinator

Координатор знімає с контролеру більше частину відповідальності, залишаючи лише відображення UI, завдяки чому код легше відтримувати та тестувати.

3.3 Core Data

Core Data – це потужний і гнучкий фреймворк від компанії Apple для зберігання і управління об'єктним графом моделі даних, вбудований в операційну систему iOS, MacOS.

Використовуючи телефони для виконання своїх завдань, люди розраховують, що внесені ними зміни буде збережено. Збереження змін грає важливу роль в офісних програмних пакетах, текстових редакторах, іграх, браузерях і тд.

Більшість програмного забезпечення потребує можливості зберігати введені користувачем дані для подальшого відновлення стану роботи. Розуміння того, яким чином можна зберігати дані на iDevice, є критично важливим при розробці просунутих додатків.

Apple надає гнучкий фреймворк для роботи з збереженими на пристрої даними - Core Data. Більшість деталей по роботі зі сховищем даних Core Data приховує, дозволяючи вам сконцентруватися на тому, що дійсно робить додаток веселим, унікальним і зручним у використанні. Також через прихованість реалізації дана база є більш безпечною.

Core Data може зберігати дані в реляційній базі даних на зразок SQLite, проте вона не є СУБД. По-правді Core Data в якості сховища може взагалі не використовувати реляційні бази даних. Core Data є оболонкою / фреймворком для роботи з даними, яка дозволяє працювати з сутностями і їх зв'язками (відносинами до інших об'єктами), атрибутами, в тому вигляді, який нагадує роботи з об'єктним графом в звичайному об'єктно-орієнтованому програмуванні.

3.4 Keychain

Keychain - це зашифрована база даних для зберігання метаданих та конфіденційної інформації від Apple.

Взаємодія додатки безпосередньо з Keychain ускладнюється, особливо у Swift. Потрібно використовувати Apple Security Framework, який переважно написаний на мові C. Проте можна уникнути використання низькорівневого API, запозичуючи Swift оболонку GenericKeychain від Apple.

Відповідальність мобільного розробника за безпеку даних в тому, щоб всі потрібні дані зберігалися в Keychain.

3.5 Додаткові методи захисту

Окрім надійного сховища даних потрібно також додати інтерфейсний захист:

1. Захист даних та доступу до додатку за допомогою пін-коду.
2. Створення «фейкових» несправжніх даних: на той випадок, якщо все ж таки зловмисники змусять користувача ввести пін-код можна зробити захисний код, який також буде розпізнаватися додатком як справжній, проте буде відображати «фейкові» дані.
3. Скривання останнього екрану додатку при згортанні.

3.6 Архітектура додатку

3.6.1. Рівень бази даних

Для роботи з Core Data було створено обгортку CoreDataStorage, інтерфейс якого винесен у протокол StorageProtocol (рисунок 3.6). Даний клас необхіден для зберігання моделей візитних карток у базі даних, а також їх читання та видалення.

```
protocol StorageProtocol {
    func getAllData() -> [CardModel]
    func setData(_ data: CardModel) -> Bool
    func clearAll()
}
```

Рисунок 3.6 – StorageProtocol

3.6.2 Рівень розпізнання телефонних номерів

Для розпізнавання телефонних номерів було створено клас TextDetector, його інтерфейс винесено до протоколу TextDetectorProtocol. Цей об'єкт є обгорткою для методів та налаштування Firebase ML Kit.


```
import UIKit

 typealias TextDetectorSuccessBlock = ([String]) -> ()
 typealias TextDetectorFailureBlock = (String) -> ()

 protocol TextDetectorProtocol {

     func detectText(fromImage image: UIImage,
                    successBlock: @escaping TextDetectorSuccessBlock,
                    failureBlock: @escaping TextDetectorFailureBlock)
 }

```

Рисунок 3.7 – TextDetectorProtocol

Для пошуку телефонних номерів у розпізнанному тексті було створено клас PhoneNumberSearcher (Рисунок 3.8).

```
import Foundation

 internal class PhoneNumberSearcher {

     internal static func searchPhoneNumbers(inLines lines: [String]) -> [String]
 }

```

Рисунок 3.8 – PhoneNumberSearcher

3.6.3 Рівень презентації

Тому, що було прийнято рішення розробляти додаток на основі паттерну Координатор, рівень презентації було розділено на логічні частини - UserStory (Рисунок 3.9).

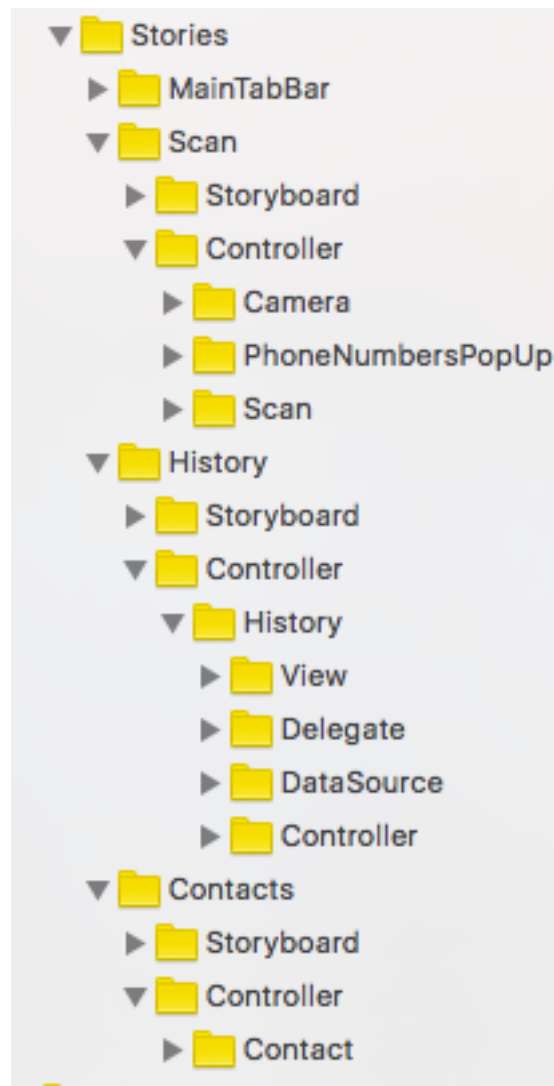


Рисунок 3.9 – Рівень презентації

3.6.4 Додаткова логіка

Уся інша додаткова логіка (Рисунок 3.10) для навігації, а також створення об'єктів, була реалізована згідно з правилами патерну Координатор. Додатково було використано патерн Фабріка, що необхіден для створенням усіх об'єктів, у тому числі координаторів.

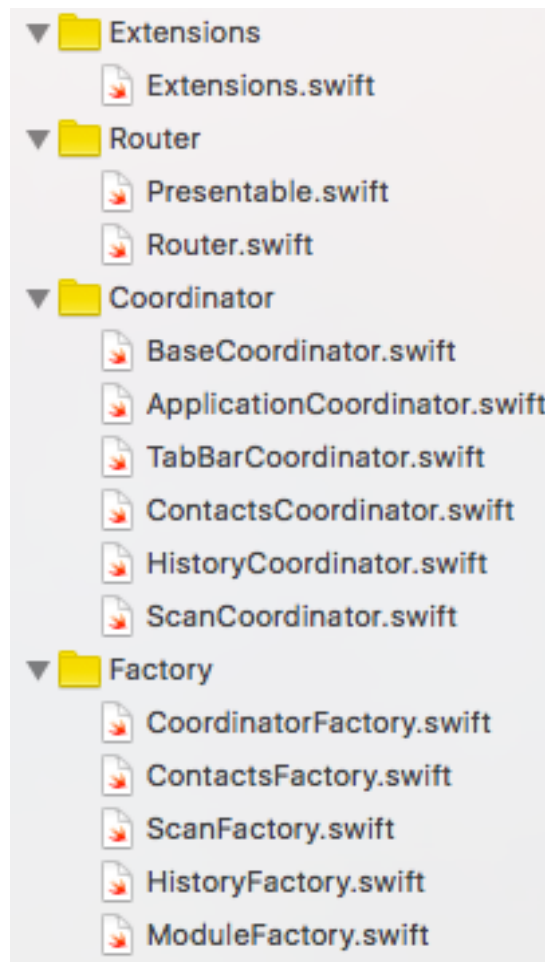


Рисунок 3.10 – Додаткова логіка

3.7 Результат розробки

Результатом проектування і розробки програмного забезпечення є програма, яка має наступний функціонал:

1. Екран налаштувань без встановленого паролю (Рисунок 3.11).
2. Екран налаштувань після встановлення паролю (Рисунок 3.12).
3. Екран налаштувань після встановлення фейкового паролю (Рисунок 3.13).
4. Екран налаштувань встановлення фейкового паролю (Рисунок 3.14).
5. Екран введення паролю (Рисунок 3.15).
6. Розмилення додатку при згортанні (Рисунок 3.16).



Рисунок 3.11 – Экран налаштувань без встановленню паролю

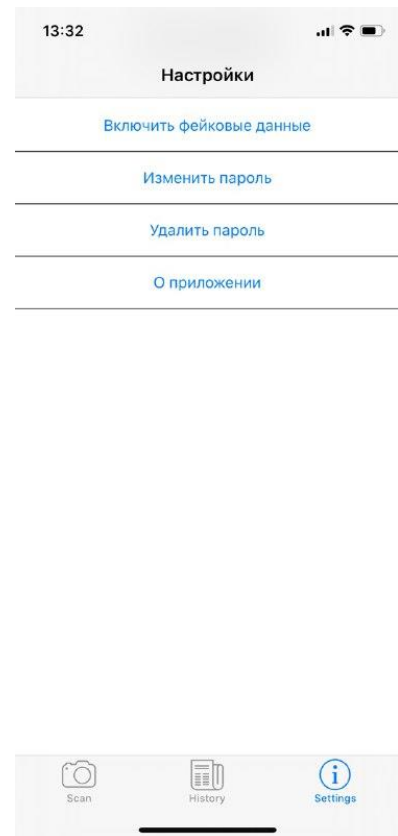


Рисунок 3.12 – Экран налаштувань після встановлення паролю

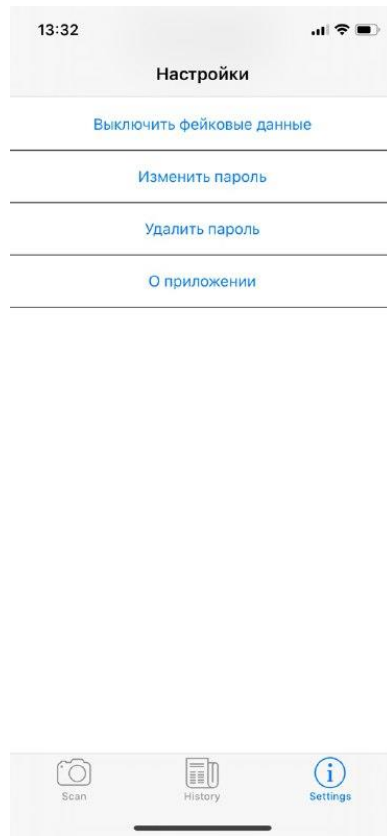


Рисунок 3.13 – Экран налаштувань після встановлення фейкового паролю

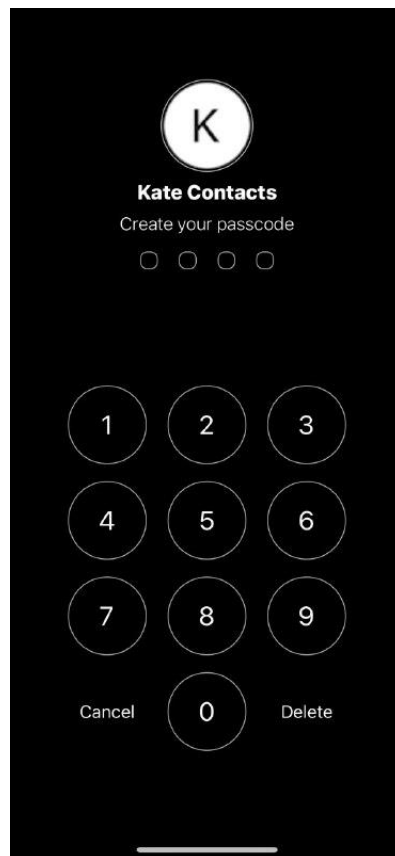


Рисунок 3.14 – Экран встановлення паролю

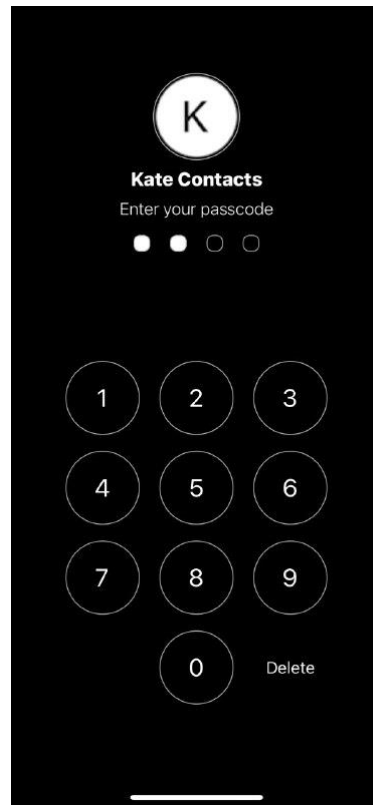


Рисунок 3.15 – Екран введення паролю

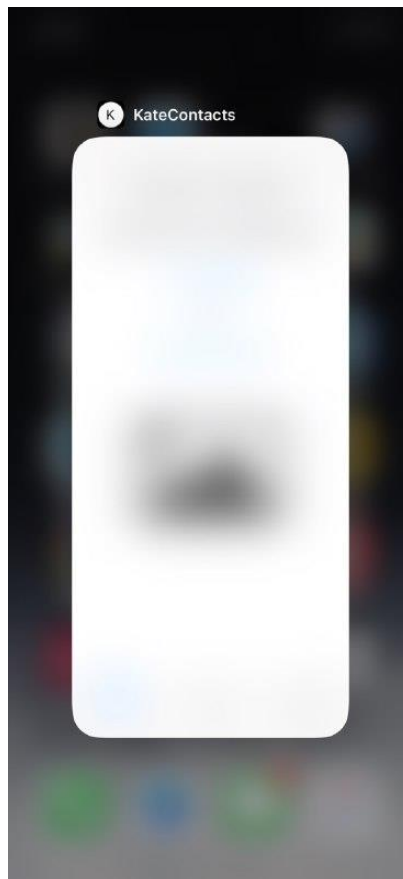


Рисунок 3.16 – Розмилення додатку при згортанні

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Опис програмного продукту

У даному розділі проведена кількісна оцінка матеріальних витрат на розробку представленого в дипломній роботі програмного продукту, призначеного для безпечного зберігання контактних даних.

Під програмним продуктом (ПП) розуміється програмне забезпечення (ПЗ) як результат людської діяльності, виставлений на ринку масового покупця в якості товару і має ненульову споживчу вартість.

Продукт розроблений в програмному середовищі Xcode.

4.2 Аналіз ринків збуту

Проведемо дослідження і виявимо потенційних покупців серед організацій, що потребують послуг ПП, наведених вище (таблиця 4.1).

Таблиця 4.1 – Потенційні покупці

Найменування організації	Кількість екземплярів ПП
Apple	1
Amazon	1
AT&T	1
Baidu	1

4.3 Визначення трудовитрат робіт

Метою даного розділу є розрахунок собівартості і вартості програмного продукту.

Для оцінки вартості розробленого програмного продукту необхідно:

- скласти перелік робіт, які слід виконати, потім розрахувати трудовитрати на

їх виконання;

- розрахувати заробітну плату розробників;
- розрахувати витрати на матеріали, комплектуючі та машинний час;
- оренда офісу;
- відрахування на соціальні заходи.

До переліку етапів робіт, які необхідно виконати входить:

- формулювання постановки задачі;
- проектування програмного продукту;
- розробка програмного продукту;
- тестування і впровадження продукту.

Тривалість кожного етапу визначається за формулою:

$$T = \frac{t}{n},$$

де T – тривалість етапу в робочих днях; t – трудовитрати етапу; n – кількість виконавців, одночасно зайнятих на певному етапі роботи.

4.4 Перелік робіт для створення програмного продукту

Для керування ходом робіт і ведення всього проекту в цілому необхідна посада керівника. Для розробки програмного продукту та його подальшого налагодження необхідна участь програміста. Для проведення тестування необхідна посада тестувальника.

Розрахуємо тривалість розробки продукту за видами робіт. Результати розрахунків містяться в таблиці 4.2.

Таблиця 4.2 – Перелік робіт

№ э т а п у	Найменування етапу	Тривалість, дн.			Трудовит- рати, лд.-дн.
		Керівник	Програміст	Тестуваль- ник	
Розробка технічного завдання					
1	Постановка зав- дання, виявлення ключових ці-лей проекту	2			2
2	Розробка технічного завдання	1			1
Разом:		3			3
Постановка задачі					
1	Розробка макетів	2			2
2	Проектування бізнес логіки до- датку	2			2
3	Розробка опису зав- дання і технічного завдання	1			1
Разом:		5			5
Розробка програмного продукту					

1	Розробка мобільного додатку		5		5
2	Розробка документації			1	1
3	Тестування			1	1
Разом:			5	2	7
Всього:		8	5	2	15

В кінцевому підсумку, ми отримали, що термін створення програмного продукту – 15 днів.

4.5 Склад виконавців роботи та розрахунок заробітної плати

Дані про посадові оклади і склад виконавців роботи занесені в таблицю 4.3. Тривалість робочого місяця будемо вважати 21 день. Робочий день – восьмигодинний.

Таблиця 4.3 – Склад виконавців роботи

Посада	Посадовий оклад, грн.	
	Місячний	Добовий
Керівник	16 000	761,9
Програміст	10 000	476,2
Тестувальник	6 000	285,7

Заробітна плата – винагорода за працю залежно від кваліфікації працівника, складності, кількості, якості та умов виконуваної роботи, а також компенсаційні виплати і стимулюючі виплати.

До витрат на заробітну плату праці відносяться основна і додаткова заробітна плата персоналу, зайнятого безпосередньо при виконанні конкретної теми: науковці, науково-технічний, науково-допоміжний персонал і виробничі робітники.

Основна заробітна плата ($ЗП_{осн}$) складається з суми середньої добової заробітної плати керівника та програміста, помноженої на тривалість їх праці відповідно і розраховується за формулою:

$$ЗП_{осн} = ЗП_{добкер} * T_{кер} + ЗП_{добпрог} * T_{прог} + ЗП_{добтест} * T_{тест},$$

де $ЗП_{добкер}$, $ЗП_{добпрог}$, $ЗП_{добтест}$ – добова заробітна плата керівника, програміста і тестувальника відповідно (таблиця 4.3); $T_{кер}$, $T_{прог}$, $T_{тест}$ — тривалість праці керівника, програміста і тестувальника відповідно (таблиця 4.2).

Таким чином, основна заробітна плата дорівнює:

$$\begin{aligned} ЗП_{осн} &= 761,9 * 8 + 476,2 * 5 + 285,7 * 2 = 6095,2 + 2381 + 571,4 \\ &= 9047,6(\text{грн}). \end{aligned}$$

Далі проводиться розрахунок додаткової заробітної плати ($ЗП_{дод}$), яка становить 17% від основної заробітної плати і розраховується:

$$ЗП_{дод} = ЗП_{осн} * N_{дод}, \quad (4.1)$$

де $N_{дод}$ – коефіцієнт додаткової зарплати, рівний 25%.

Отже, за формулою (4.1) додаткова заробітна плата дорівнює:

$$ЗП_{\text{дод}} = 9047,6 * 0,25 = 2261,9(\text{грн}).$$

Разом, загальний фонд заробітної плати становить:

$$ЗП = ЗП_{\text{осн}} + ЗП_{\text{дод}} = 9047,6 + 2261,9 = 11309,5(\text{грн}).$$

Нарахування на заробітну плату (єдиний соціальний внесок – $ЗП_{\text{соц}}$) складають 22% і розраховується за формулою:

$$ЗП_{\text{соц}} = ЗП * Н_{\text{соц}}, \quad (4.2)$$

де $Н_{\text{соц}}$ – коефіцієнт єдиного соціального внеску, рівний 22%.

Отже, за формулою (4.2) нарахування на заробітну плату становлять:

$$ЗП_{\text{соц}} = 11309,5 * 0,22 = 2488,09 (\text{грн}).$$

4.6 Розрахунок витрат на матеріал, комплектуючі, оренду офісу та електроенергію

Розрахуємо витрати на матеріали і комплектуючі (V), необхідні для створення додатку. Результати представлені в таблиці 4.4.

Таблиця 4.4 – Вартість матеріалів і комплектуючих

№ п/п	Найменування Матеріалу	Призначення	Кількість,шт.	Ціна за одиницю, грн.
1	Папір	Документація, прототи-типи	400 аркушів	0.5
2	Канцелярські принаді	Малювання схем	10	15
Разом:				350

Амортизацію не розраховуємо, тому что орендуємо приміщення з усім необхідним обладнанням. Оренда офісу на два 15 днів коштує 5357 грн (А).

Вартість технологічної електроенергії S_E обчислюється за формулою:

$$S_E = T_{ap} * T * W, \quad (4.3)$$

де T_{ap} – тариф електроенергії за один кВт, рівний 2,01 грн.;

T – кількість годин роботи;

W – споживана технологічна потужність ($W = 0,032$ кВт).

Отже, за формулою (4.3) вартість технологічної електроенергії:

$$S_E = 2,01 * 120 * 0,032 = 7,7(\text{грн}).$$

Вартість освітлювальної електроенергії S розраховується за формулою:

$$S = T_{ap} * T * W, \quad (4.4)$$

де W – споживана потужність освітлювальним прибором ($W = 0,1$ кВт).

Отже, за формулою (4.4) вартість освітлювальної електроенергії:

$$S = 2,01 * 120 * 0,1 = 24,12(\text{грн}).$$

4.7 Розрахунок собівартості та повної вартості програмного продукту

Собівартість – це вартісна оцінка використаних в процесі виробництва продукції (робіт, послуг) природних ресурсів, сировини, матеріалів, палива, енергії, основних фондів, трудових ресурсів та інших витрат на її виробництво і реалізацію. Собівартість дорівнює сумі всіх витрат на розробку проекту і розраховується за формулою:

$$C = 3П + 3П_{\text{соц}} + S_E + S + A + V \quad (4.5)$$

Отже, з формули (4.5) собівартість становить:

$$\begin{aligned} C &= 11309,5 + 2488,09 + 516,32 + 7,7 + 24,12 + 5357 + 350 \\ &= 20052,73(\text{грн}). \end{aligned}$$

Прибуток від розробленого програмного продукту (П) дорівнює:

$$П = C * H_{\text{приб}}, \quad (4.6)$$

де $H_{\text{приб}} = 0,15$, так як під час вивчення ринку мобільних програм аналогів даного продукту не було знайдено.

Отже, з формули (4.6) прибуток становить:

$$П = 20052,73 * 0,15 = 3007,9(\text{грн}).$$

Повна вартість розробленого програмного продукту без ПДВ розраховується за наступною формулою:

$$ПВ_{\text{безПДВ}} = (C + П), \quad (4.7)$$

де $ПВ_{\text{безПДВ}}$ – повна вартість розробленого програмного продукту без ПДВ.

Отже, за формулою (4.7) повна вартість розробленого програмного продукту без ПДВ:

$$ПВ_{\text{безПДВ}} = 20052,73 + 3007,9 = 23060,64(\text{грн}).$$

Повна вартість розробленого програмного продукту з ПДВ розраховується за наступною формулою:

$$ПВ_{\text{зПДВ}} = ПВ_{\text{безПДВ}} + Н_{\text{ПДВ}} * ПВ_{\text{безПДВ}}, \quad (4.8)$$

де $ПВ_{\text{зПДВ}}$ – повна вартість розробленого програмного продукту з ПДВ;

$Н_{\text{ПДВ}}$ – податок на додану вартість, рівний 20%.

Отже, за формулою (4.8) повна вартість розробленого програмного продукту з ПДВ:

$$ПВ_{\text{зПДВ}} = 23060,64 + 23060,64 * 0,2 = 27672,77(\text{грн}).$$

Калькуляційні розрахунки на розробку програмного забезпечення представлені в таблиці 4.5.

Таблиця 4.5 – Статті калькуляції на розробку програмного продукту

№	Стаття калькуляції	Витрати, грн.
1	Основна заробітна плата	9047,6
2	Додаткова заробітна плата	2261,9
3	Загальна заробітна плата	11309,5
4	Єдиний соціальний внесок	2488,09
5	Оренда офісу	5357
6	Матеріали і комплектуючі	350
7	Вартість технологічної електроенергії	7,7
8	Вартість освітлювальної електроенергії	24,12
9	Загальна собівартість розробки	20052,73
10	Прибуток	3007,9
11	Повна вартість розробленого програмного продукту без ПДВ	23060,64
12	Повна вартість розробленого програмного продукту з ПДВ	27672,77

4.8 Альтернативний процес розробки програмного продукту

В цілому для розробки гарного продукту необхідна велика команда з чітко розподіленими ролями, проте у нашому випадку проект досить маленький, тому одна й та ж сама людина може виконувати декілько ролей. Тому складемо команду с двох кваліфікованих програмістів.

Розрахуємо тривалість розробки продукту за видами робіт. Результати розрахунків містяться в таблиці 4.6.

Таблиця 4.6 – Перелік робіт

№ етапу	Найменування етапу	Тривалість, дн.		Трудовитрати, лд.-дн.
		Програміст 1	Програміст 2	
1	Постановка завдання, виявлення ключових цілей проекту	1	1	2
2	Розробка технічного завдання	0.5	0.5	1
Разом:		1.5	1.5	3
1	Розробка макетів	1	1	2
2	Проектування бізнес логіки додатку	1	1	2
3	Розробка опису завдання і технічного завдання	0.5	0.5	1
Разом:		2.5	2.5	5
1	Розробка мобільного додатку	2	2	4
2	Розробка документації	0.5	0.5	1

3	Тестування	0.5	0.5	1
Разом:		3	3	6
Всього:		7	7	14

В кінцевому підсумку, ми отримали, що термін створення програмного продукту – 14 днів.

4.9 Склад виконавців роботи та розрахунок заробітної плати

Дані про посадові оклади і склад виконавців роботи занесені в таблицю 4.7. Тривалість робочого місяця будемо вважати 21 день. Робочий день – восьмигодинний.

Таблиця 4.7 – Склад виконавців роботи

Посада	Посадовий оклад, грн.	
	Місячний	Добовий
Програміст	10 000	476,2
Програміст	10 000	476,2

Основна заробітна плата ($ЗП_{осн}$) складається з суми середньої добової заробітної плати програміста1 та програміста2, помноженої на тривалість їх праці відповідно і розраховується за формулою:

$$ЗП_{осн} = ЗП_{добпрог1} * T_{прог1} + ЗП_{добпрог2} * T_{прог2},$$

де $ЗП_{добпрог1}$, $ЗП_{добпрог2}$ – добова заробітна плата програміста1 і програміста2

відповідно (таблиця 4.7); $T_{\text{прог1}}$, $T_{\text{прог2}}$ – тривалість праці програміст1 і програміста2 відповідно (таблиця 4.6).

Таким чином, основна заробітна плата дорівнює:

$$ЗП_{\text{осн}} = 476,2 * 7 + 476,2 * 7 = 2857,1 + 2857,1 = 5714,2(\text{грн}).$$

Далі проводиться розрахунок додаткової заробітної плати ($ЗП_{\text{дод}}$), яка становить 17% від основної заробітної плати і розраховується.

Отже, за формулою (4.1) додаткова заробітна плата дорівнює:

$$ЗП_{\text{дод}} = 5714,2 * 0,25 = 1428,55(\text{грн}).$$

Разом, загальний фонд заробітної плати становить:

$$ЗП = ЗП_{\text{осн}} + ЗП_{\text{дод}} = 5714,2 + 1428,55 = 7142,75(\text{грн}).$$

Нарахування на заробітну плату (єдиний соціальний внесок – $ЗП_{\text{соц}}$) складають 22.

Отже, за формулою (4.2) нарахування на заробітну плату становлять:

$$ЗП_{\text{соц}} = 7142,75 * 0,22 = 1571,4(\text{грн}).$$

4.10 Розрахунок витрат на матеріал, комплектуючі, оренду офісу та електроенергію

Розрахуємо витрати на матеріали і комплектуючі (V), необхідні для створення додатку. Результати представлені в таблиці 4.4.

Таблиця 5.4 – Вартість матеріалів і комплектуючих

№ п/п	Найменування Матеріалу	Призначення	Кількість,шт.	Ціна за одиницю, грн.
1	Папір	Документація, прототи-типи	400 аркушів	0.5
2	Канцелярські принаді	Малювання схем	10	15
Разом:				350

Амортизацію не розраховуємо, тому що орендуємо приміщення з усім необхідним обладнанням. Оренда офісу на два 14 днів коштує 5000 грн (А).

За формулою (4.3) вартість технологічної електроенергії:

$$S_E = 2,01 * 112 * 0,032 = 7,2(\text{грн}).$$

За формулою (4.4) вартість освітлювальної електроенергії:

$$S = 2,01 * 112 * 0,1 = 22,5(\text{грн}).$$

4.11 Розрахунок собівартості та повної вартості програмного продукту

Собівартість – це вартісна оцінка використовуваних в процесі виробництва продукції (робіт, послуг) природних ресурсів, сировини, матеріалів, палива, енергії, основних фондів, трудових ресурсів та інших втрат на її виробництво і реалізацію.

З формули (4.5) собівартість становить:

$$C = 7142,75 + 1571,4 + 7,2 + 22,5 + 5000 + 350 = 14093,85(\text{грн}).$$

З формули (4.6) прибуток становить:

$$П = 14093,85 * 0,15 = 2114,1(\text{грн}).$$

За формулою (4.7) повна вартість розробленого програмного продукту без ПДВ:

$$ПВ_{\text{безПДВ}} = 14093,85 + 2114,1 = 16207,93(\text{грн}).$$

За формулою (4.8) повна вартість розробленого програмного продукту з ПДВ:

$$ПВ_{\text{зПДВ}} = 16207,93 + 16207,93 * 0,2 = 19445,5(\text{грн}).$$

Калькуляційні розрахунки на розробку програмного забезпечення представлені в таблиці 4.9.

Таблиця 4.9 – Статті калькуляції на розробку програмного продукту

№	Стаття калькуляції	Витрати, грн.
1	Основна заробітна плата	5714,2
2	Додаткова заробітна плата	1428,55
3	Загальна заробітна плата	7142,75
4	Єдиний соціальний внесок	1571,4
5	Оренда офісу	5000
	Матеріали і комплектуючі	350

6	Вартість технологічної електроенергії	7,2
7	Вартість освітлювальної електроенергії	22,5
8	Загальна собівартість розробки	14093,85
9	Прибуток	2114,1
1 0	Повна вартість розробленого програмного продукту без ПДВ	16207,93
1 1	Повна вартість розробленого програмного продукту з ПДВ	19445,5

Висновок:

На сьогоднішній день в умовах високої конкуренції на IT-ринку дуже важливо зробити продукт для прийнятної для клієнта ціною. Для цього до початку процесу розробки необхідно знати собівартість і ціну готового продукту. Також потрібно визначити складність створюємого продукту та оптимальну команду.

Внаслідок зазначених вище причин в даному розділі було проведено порівняльний аналіз ефективності роботи двох видів команди (розподільної та однорідної):

- загальна собівартість розробки розподільною командою склала 20052,73 грн, а однорідною 14093,85 грн;
- повна вартість розробленого програмного продукту з ПДВ розподільною командою склала 27672,77 грн, а однорідною 19445,5.

Так як проект невеликий, а розробка однорідною командою виявилась значно дешевшею та швидшою, робимо висновок, що у даному випадку краще використовувати однорідну команду.

ВИСНОВКИ

В дипломному проекті було:

- Розглянуто основні способи покращення інформаційної безпеки у мобільних додатках. Розглянена і обґрунтована актуальність необхідності інформаційної безпеки.

- Розглянуто мови програмування, наведене їх порівняння.
- Розглянуто різні архітектурні підходи до розробки мобільних додатків.

На підставі проведенного дослідження було:

- Прийнято рішення для зберігання персональних даних користувача використати Keuchain.

- Додати захист за допомогою пін-коду.
- Створення «фейкових» несправжніх даних: на той випадок, якщо все ж таки зловмисники змусять користувача ввести пін-код можна зробити захисний код, який також буде розпізнаватися додатком як справжній, проте буде відображати «фейкові» дані.

- Скривання останнього екрану додатку при згортанні.
- У якості мови програмування було обрано Swift, т.я. його перевагою є швидкість та сучасність.

- У якості архітектури додатку було обрано Координатор, т.я. він робить підтримку кода легшою.

Результатом виконання дипломної роботи є розроблене програмне забезпечення, складене на підставі технічне завдання, що знаходиться в додатку А.

Надалі має сенс проводити роботу в наступних напрямках:

- розширення мобільного додатку додатковими функціями;
- розробка додатку для інших платформ;
- покращення інтерфейсу.

ГЛОСАРІЙ

СУБД – комплекс програмного забезпечення, що надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

SDK – набір із засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи (програмної або програмно-апаратної).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. M. Bartlett, J. Hager, P. Ekman, and T. Sejnowski. Measuring facial expressions by computer image analysis. *Psychophysiology*, 36, 1999. – p. 213.
2. Симанков В.С. Адаптивное управление сложными системами на основе теории распознавания образов / В. С. Симанков, Е. В. Луценк – Краснодар: Техн. ун-т Кубан. гос. технол. ун-та, 1999. – 318 с.
3. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – Москва : Техносфера, 2005. – 1072 с.
4. Лепский А.Е., Броневиц А.Г. Математические методы распознавания образов: Курс лекций./ А.Е. Лепский, А.Г. Броневиц – Таганрог: Изд-во ТТИ ЮФУ, 2009. – 155 с.
5. Шапиро Л. Компьютерное зрение / Шапиро Л., Стокман Дж. – Бинном, 2013. – 752 с.
6. Duda, R. O., and P. E. Hart. 1973. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York . – p. 213.
7. Duda, R. O., D. Stork, and P. Hart. 2000. *Pattern Classification*. John Wiley & Sons, New York. – p. 43.
8. Feller, W. 1957. *An Introduction to Probability Theory and Its Applications*, vols. I and II. John Wiley & Sons, New York. – p. 21 – 25.
9. Haykin, S. 1994. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing, New York. – p. 34 – 43.
10. Hertz, J., A. Krogh, and R. Palmer. 1991. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Reading, MA. – p. 125.
11. Hogg, R., and A. Craig. 1970. *Introduction to Mathematical Statistics*. – p. 78.
12. Jain, A. K., J. Mao, and K. M. Mohiuddin. 1996. Artificial neural networks: A tutorial. *IEEE Comput.* 29(3). – p. 65.
13. Jain, R. Duin, and J. Mao. 2000. Statistical pattern recognition, a review. *IEEE-TPAMI*. 22(1): 4–37.

14. Fukunaga, K. 1990. Introduction to Statistical Pattern Recognition, 2nd ed. Academic Press, New York. – p. 213.
15. Kulkarni, A. 1994. Artificial Neural Networks for Image Understanding. Van Nostrand-Reinhold, New York. – p. 342.
16. Y. LeCun, O.Matan, B.Boser, J.S.Denker, D.Henderson, R.E.Howard, W.Hubbard, L.D.Jackel, H.S.Baird. “Handwritten Zipcode Recognition With Multilayer Networks,” Proc. of International Conference on Pattern Recognition, Atlantic City, 1990. – p. 24 – 25.
17. Krzyzak, W. Dai, C.Y.Suen. “Unconstrained Handwritten Character Classification Using Modified Backpropagation Model,” Proc. 1st Int. Workshop on Frontiers in Handwriting Recognition, Montreal, Canada, Стр.155-166, 1990.
18. Seong-Wang Lee, Young Joon Kim. “Off-line Recognition of Totally Unconstrained Handwritten Numerals Using Multilayer Cluster Neural Network.” Proc. Of the 12th IAPR International Conference on Pattern Recognition. Jerusalem, Israel. 1994. Стр. 507-509.
19. R. Schapire, “The Strength of Weak Learnability,” Machine Learning. 5 197-227 (1990).
20. H.Drucker, R.Schapire, P.Simard. “Boosting Performance in Neural Networks.” International Journal of Pattern Recognition and Artificial Intelligence. 7 705-720 (1993).
21. Fukushima, K. (1988). Neocognition: a hierarchical neural network capable of visual pattern recognition. Neural Networks 1 (2), Стр. 119-130.
22. Knerr, S., Personnaz, L. & Dreyfus, G. (1992) Handwritten digit recognition by neural networks with single-layer training. IEEE Transactions on Neural Networks 3, 962-968.
23. Lee, Y. (1991) Handwritten digit recognition using K nearest-neighbor, radial-basis function, and back-propagation neural networks. Neural Computation 3, 440-449.
24. Martin, G.L. & Pitman, J. A. (1991) Recognizing hand-printed letters and digits using backpropagation learning. Neural Computation 3, 258-267.

25. J'son & Partners Consulting (2017) Искусственный интеллект (ИИ) / Artificial Intelligence (AI) как ключевой фактор цифровизации глобальной экономики, – 130 стр.
26. Kutter M., Voloshynovskiy S., Herrigel A. The Watermark Copy Attack // Proceedings of SPIE: Security and Watermarking of Multimedia Content II. 2000. Vol. 3971. – p. 21 – 25.
27. Lin C. Watermarking and Digital Signature Techniques for Multimedia Authentication and Copyright Protection. PhD Thesis, Columbia University, 2000. – p. 321.
28. Wu M. Multimedia Data Hiding. PhD Thesis, Princeton University, 2001. – p. 67.
29. Su J.K., Eggers J.J., Girod B. Analysis of Digital Watermarks Subjected to Optimum Linear Filtering and Additive Noise // Signal Processing. Special Issue on Information Theoretic Issues in Digital Watermarking. 2001. №6. P.1141-1175.
30. Marvel L. Image Steganography for Hidden Communication. PhD Thesis. University of Delaware, 1999. 155 p.
31. O. Trier, T. Taxt, and A. Jain. Data capture from maps based on gray scale topographic analysis. In The Third International Conference on Document Analysis and Recognition, Montreal, Canada, 1995. – p. 54.
32. L. T. Watson, T. J. Laffey, and R. M. Haralick. Topographic classification of digital image intensity surfaces using generalized splines and the discrete cosine transform. Computer Vision, Graphics and Image Processing, 29:143–167, 1985.
33. Joe Conway, Aaron Hillegass. iOS Programming: The Big Nerd Ranch Guide, 3rd Edition – 2013. – 121 с.
34. Matthijs Hollemans. iOS Apprentice. 2015. – 58 с.
35. Anthony Gray. Swift Pocket Reference: Programming for iOS and OS X. – 2014. – 341 с.
36. Paul Hudson. Pro Swift.– 2016. – 211 с.

37. Erica Sadun. The Core iOS Developers Cookbook. – 2017. – 121 с.
38. Eric Buck, Donald Yactman. Cocoa Design Patterns. – 2009. – 218 с.
39. Craig Hockenberry. iPhone App Development: The Missing Manual.– 2011. – 123 с.
40. Ханг Во. Оптимизация производительности приложений для iOS. Для профессионалов. – 2012. – 312 с.
41. John Gallagher, Swift Programming: The Big Nerd Ranch Guide, 2015. – 231 с.
42. .Гэлловей Мэтт Сила Objective-C 2.0. Эффективное программирование для iOS и OS. – СПб: Питер, 2014. – 304 с.
43. Мэтт Нойбург. Программирование для iOS 7. Основы Objective-C, Xcode и Сосоа. — Вильямс, 2014. — 384 с.
44. David Mark, Jack Nutting, Jeff LaMarche, Fredrik Olsson. Beginning iOS 6 Development – 2012. – 412 с.

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ БЕЗПЕЧНОГО ЗБЕРІГАННЯ КОНТАКТНОЇ ІНФОРМАЦІЇ

А.1. ВСТУП

А.1.1 Найменування програми

Найменування програми: «KateContacts».

А.1.2 Призначення та область застосування

Програма призначена для безпечного зберігання контактних даних.

А.2. ВИМОГИ ДО ПРОГРАМИ

А.2.1 Вимоги до функціональних характеристик

Програма повинна забезпечувати можливість виконання перерахованих нижче функцій:

- Розпізнавання телефонних номерів із цифрових зображень.
- Можливість загрузити фотографію з галереї.
- Можливість зробити фотографію через додаток.
- Захищене збереження розпізнаних номерів разом із зображенням.
- Збереження телефонного номеру у “Контакти”.
- Зателефонувати на розпізнаний номер.
- Встановлення паролю для доступу до додатку.
- Замилення екрану при згортанні додатку.
- Доступ до додатку під фейковим паролем.

A.2.2 Вимоги до нефункціональних характеристик

Програма повинна забезпечувати можливість виконання перерахованих нижче функцій:

- програма повинна робити розпізнавання за відсутності інтернету;
- програма повинна бути зроблена для операційної системи iOS, і підтримуватися, починаючи з версії iOS 12;
- використання надійного сховища даних.

A.3. УМОВИ ЕКСПЛУАТАЦІЇ

A.3.1 Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти вимогам, що пред'являються до технічних засобів в частині умов їх експлуатації.

A.3.2 Вимоги до складу та параметрів технічних засобів

Для повноцінного користування мобільним додатком, він повинен мати справну фотокамеру.

A.3.3 Вимоги до вихідного коду та мов програмування

Для забезпечення подальшого розвитку та зручності супроводу програмного забезпечення, програма повинна бути написана на мові програмування Swift.

А.4. ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

А.4.1 Попередній склад програмної документації

Склад програмної документації повинен включати в себе:

- технічне завдання;
- вихідний код програми;
- керівництво користувача.

А.5. ТЕХНІКО-ЕКОНОМІЧНІ ПОКАЗНИКИ

А.5.1 Економічні переваги розробки

Орієнтовна економічна ефективність на етапі складання технічного завдання не розраховується. Аналогія не проводиться зважаючи на унікальності пропонованих вимог до розробки.

А.6. СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

А.6.1 Стадії розробки

Розробка повинна бути проведена в три стадії:

- розробка технічного завдання;
- робоче проектування;
- впровадження.

А.6.2 Етапи розробки

На стадії розробки технічного завдання повинен бути виконаний етап розробки, узгодження та затвердження цього технічного завдання.

На стадії робочого проектування повинні бути виконані перераховані нижче етапи робіт:

- розробка програмного забезпечення;
- розробка документації програмного забезпечення;
- контроль якості програмного забезпечення.

На стадії впровадження повинен бути виконаний етап розробки та передача програмного забезпечення.