

**Київський національний торговельно-економічний університет**  
**Кафедра комп'ютерних наук та інформаційних систем**

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**

на тему:

**«Розробка мобільного додатку для відвідування подій  
“Blink” за допомогою мови програмування Swift»**

Студентки 4 курсу, 10 групи,  
спеціальності  
122 «Комп'ютерні науки»

Медяник  
Вероніки  
Вадимівни

*підпис студента*

Науковий керівник  
кандидат фізико-математичних наук,  
доцент

Самойленко Ганна  
Тимофіївна

*підпис керівника*

Гарант освітньої програми  
кандидат технічних наук, доцент

Демідов Павло  
Георгійович

*підпис керівника*

**Київ 2021**

# Київський національний торговельно-економічний університет

Факультет інформаційних технологій  
Кафедра комп'ютерних наук та систем  
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри \_\_\_\_\_

**Затверджую**  
Пурський О.І.  
« » грудня 2020р.

## **Завдання на випускну кваліфікаційну роботу студенту**

**Медяник Вероніці Вадимівні**  
(прізвище, ім'я, по батькові)

### 1. Тема випускної кваліфікаційної роботи (проекту)

*«Розробка мобільного додатку для відвідування подій “Blink” за допомогою мови програмування Swift»*

Затверджена наказом ректора від «04» грудня 2020 р. № 4111

### 2. Строк здачі студентом закінченої роботи 29 травня 2021 року

### 3. Цільова установка та вихідні дані до роботи

Мета роботи: обґрунтування та розробка мобільного додатку для відвідування подій з урахуванням сучасних тенденцій застосування мов програмування в розробці мобільних додатків.

Об'єкт дослідження: процес проектування мобільного додатку для відвідування подій

Предмет дослідження: засоби створення мобільного додатку.

### 4. Перелік графічного матеріалу \_\_\_\_\_

---

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Самойленко Г.Т.	15.12.2020 р.	15.12.2020 р.
2	Самойленко Г.Т.	15.12.2020 р.	15.12.2020 р.
3	Самойленко Г.Т.	15.12.2020 р.	15.12.2020 р.

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

## ВСТУП

### РОЗДІЛ 1. Сучасні технології в розробці мобільних додатків

#### 1.1 Сутність поняття мобільних технологій

#### 1.2 Мови створення мобільних додатків

#### 1.3 Різновиди мобільних додатків та їх застосування

#### Висновки до розділу

### РОЗДІЛ 2. Організація розробки мобільного додатку для відвідування подій

#### 2.1. Огляд існуючих технологій проектування

#### 2.2 Особливості мови програмування Swift

#### 2.3 Розробка вимог до додатку

#### Висновки до розділу

### РОЗДІЛ 3. Програмна реалізація мобільного додатку для відвідування подій

#### “Blink”

#### 3.1. Розробка структури мобільного додатку для відвідування подій “Blink”

#### 3.2. Розробка моделі функціонування мобільного додатку для відвідування подій

#### “Blink”

#### 3.3. Програмна реалізація мобільного додатку для відвідування подій “Blink”

#### 3.4 Висновки до розділу

### РОЗДІЛ 4. Способи взаємодії з користувачем за межами додатку

## ВИСНОВКИ

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

## ДОДАТКИ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4

1	<i>Вибір теми випускної кваліфікаційної роботи</i>	<i>01.10.2020</i>	<i>01.10.2020</i>
2	<i>Розробка та затвердження завдання на випускні кваліфікаційну роботу</i>	<i>15.12.2020</i>	<i>15.12.2020</i>
3	<i>Вступ</i>	<i>03.02.2021</i>	<i>03.02.2021</i>
4	<i>РОЗДІЛ 1_Сучасні технології в розробці мобільних додатків</i>	<i>28.02.2021</i>	<i>26.02.2021</i>
5	<i>РОЗДІЛ 2_Організація розробки мобільного додатку для відвідування подій</i>	<i>06.04.2021</i>	<i>06.04.2021</i>
6	<i>РОЗДІЛ 3. Програмна реалізація мобільного додатку для відвідування подій "Blink"</i>	<i>12.05.2021</i>	<i>12.05.2021</i>
7	<i>Висновки</i>	<i>15.05.2021</i>	<i>15.05.2021</i>
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	<i>20.05.2021</i>	<i>20.05.2021</i>
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	<i>26.05.2021</i>	<i>03.06.2021</i>
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	<i>27.05.2021</i>	<i>04.06.2021</i>
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедру</i>	<i>29.05.2021</i>	<i>08.06.2021</i>
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	<i>За розкладом роботи ЕК</i>	<i>14.06.2021</i>

8. Дата видачі завдання «15» грудня 2020 р.

9. Керівник випускної кваліфікаційної роботи (проекту)

Самоїленко Г.Т.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Медяник В.В.

(прізвище, ініціали, підпис)



## АНОТАЦІЯ

Текстова частина бакалаврської роботи 51 с., 16 рис., 10 джерел. Основною метою дипломного проекту «Мобільний додаток для відвідування подій Blink, розроблений за допомогою мови програмування Swift» є обґрунтування та розробка мобільного додатку для відвідування подій з урахуванням сучасних тенденцій застосування мов програмування в розробці мобільних додатків.

Для реалізації мети було сформульовано та вирішено наступні завдання:

1. Аналіз існуючих додатків.
2. Вибір середовища розробки та мови програмування.
3. Дослідження мобільного додатку для пошуку та створення подій.
4. Теоретичні основи для розробки програмного засобу.
5. Розробка програмного забезпечення.

Об'єктом дипломної роботи є процес проектування мобільного додатку для відвідування подій.

Предметом дипломної роботи є засоби створення мобільного додатку.

Відповідно до завдання на дипломний проект необхідно було розробити та описати додаток за допомогою мови Swift для операційної системи iOS, з яким користувач має змогу шукати будь-які події у обраному місті України, а також буде мати змогу створювати їх.

Актуальність пошуку, а також створення подій дуже велика: наприклад, організуючи події, користувач може заробляти гроші на реалізації квитків, а шукаючи події, користувач може обрати для себе різноманітні активності для свого розвитку або проведення вільного часу.

**КЛЮЧОВІ СЛОВА:** МОБІЛЬНИЙ ДОДАТОК, ПОДІЇ, IOS, XCODE, SWIFT, UML-ДІАГРАМИ, UIKIT, APP STORE.

## ANNOTATION

Text part of the bachelor's thesis 51 pp., 16 figs., 10 sources. The main goal of the diploma project "Mobile application for attending events Blink, developed using the programming language Swift" is to substantiate and develop a mobile application for attending events, taking into account current trends in the use of programming languages in the development of mobile applications.

To achieve this goal, the following tasks were formulated and solved:

1. Analysis of existing applications.
2. Choice of development environment and programming language.
3. Research a mobile application for finding and creating events.
4. Theoretical foundations for software development.
5. Software development.

The object of the thesis is the process of designing a mobile application for attending events.

The subject of the thesis is the means of creating a mobile application.

According to the task for the diploma project, it was necessary to develop and describe an application using the Swift language for the iOS operating system, with which the user can search for any events in the selected city of Ukraine, as well as be able to create them.

The relevance of the search, as well as the creation of events is very high: for example, organizing events, the user can earn money by selling tickets, and looking for events, the user can choose a variety of activities for their development or leisure.

**KEY WORDS:** MOBILE APP, EVENTS, IOS, XCODE, SWIFT, UML DIAGRAMS, UIKIT, APP STORE.



## ЗМІСТ

<u>ВСТУП.....</u>	<u>11</u>
<u>РОЗДІЛ 1. СУЧАСНІ ТЕХНОЛОГІЇ В РОЗРОБЦІ МОБІЛЬНИХ ДОДАТКІВ....</u>	<u>14</u>
1.1 СУТНІСТЬ ПОНЯТТЯ МОБІЛЬНИХ ТЕХНОЛОГІЙ .....	14
1.2 МОВИ СТВОРЕННЯ МОБІЛЬНИХ ДОДАТКІВ .....	16
1.2.1 ОБ'ЄКТИВНЕ-С .....	16
1.2.2 SWIFT.....	17
1.3 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	19
ВИСНОВКИ ДО РОЗДІЛУ .....	23
<u>РОЗДІЛ 2. ОРГАНІЗАЦІЯ РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ “BLINK” .....</u>	<u>24</u>
2.1 ОГЛЯД ІСНУЮЧИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ.....	24
2.2 ОСОБЛИВОСТІ МОВИ ПРОГРАМУВАННЯ SWIFT .....	24
2.3 РОЗРОБКА ВИМОГ ДО ДОДАТКУ .....	26
2.3.1 ВИМОГИ ДО ГРАФІЧНОГО ІНТЕРФЕЙСУ.....	26
2.3.2 ВИМОГИ ДО НАДІЙНОСТІ ПРОГРАМИ.....	35
2.3.3 ВИМОГИ ДО ВИХІДНОГО КОДУ І МОВ ПРОГРАМУВАННЯ .....	35
2.3.4 ВИМОГИ ДО ПРОГРАМНИХ ЗАСОБІВ, ЯКІ ВИКОРИСТОВУЮТЬСЯ ПРОГРАМОЮ .....	35
ВИСНОВКИ ДО РОЗДІЛУ .....	35
<u>РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВІДВІДУВАННЯ ПОДІЙ “BLINK”.....</u>	<u>37</u>
3.1 РОЗРОБКА СТРУКТУРИ МОБІЛЬНОГО ДОДАТКУ “BLINK” .....	37
3.2 РОЗРОБКА МОДЕЛІ ФУНКЦІОНУВАННЯ МОБІЛЬНОГО ДОДАТКУ .....	38
3.3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ “BLINK” .....	38

3.3.1 ВІДОБРАЖЕННЯ ДАНИХ .....	39
3.3.2 ВЗАЄМОДІЯ З СЕРВЕРОМ.....	41
3.3.3 РОБОТА З КАРТАМИ.....	42
3.3.4 СПОСОБИ ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ ЗА МЕЖАМИ ДОДАТКУ ....	43
ВИСНОВКИ ДО РОЗДІЛУ .....	48
<u>ВИСНОВКИ ТА РЕЗУЛЬТАТИ.....</u>	<u>50</u>
<u>ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</u>	<u>52</u>
<u>ДОДАТОК А .....</u>	<u>53</u>
<u>ДОДАТОК Б.....</u>	<u>54</u>
<u>ДОДАТОК В.....</u>	<u>55</u>

## Перелік основних позначень і скорочень

iOS - операційна система для смартфонів

XCode – інтегроване середовище розробки

FB - «Facebook» — соціальна мережа

GPS - Global Positioning System — система глобального позиціонування

GUI - Graphical User Interface - графічний користувальницький інтерфейс

HTTP - HyperText Transfer Protocol — «Протокол передачі гіпертексту»

MySQL - вільна реляційна система управління базами даних

App Store — онлайн магазин додатків для пристроїв з системою iOS

SQL – StructureQueryLanguage - структурована мова запитів

UML – Universal Modeling Language - універсальна мова моделювання

БД – база даних

ПЕОМ – персональна електронно-обчислювальна машина

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

СУБД – система управління базами даних

## ВСТУП

Тема проектної задачі «Мобільний додаток для відвідування подій Blink, розроблений за допомогою мови програмування Swift».

Мета: за допомогою мови програмування Swift розробити мобільний додаток, який буде давати користувачу можливість шукати та створювати події у реальному часі.

Обґрунтування вибору теми та її актуальність: мобільні додатки мають відчутні переваги, як технічні, так, безумовно, і в сфері маркетингу та користуються величезним попитом по всьому світу.

Додаток досить лише один раз завантажити з маркету на мобільний пристрій, і можна користуватися ним в будь-який час без підключення до інтернету або повторного скачування.

У додатку є дуже багато можливостей для доступу до пристрою. Таким чином через мобільний додаток можна отримати місце розташування користувача, доступ до каталогу файлів, вихід в Інтернет, доступ до камери, мікрофона та динаміків, так само можна отримати доступ до всіх сенсорів телефону. Таким чином мобільний додаток надає дуже широкий спектр можливостей, якими можна скористатися при розробці.

Користуючись мобільним телефоном людина найбільше часу проводить у соціальних мережах.

У кожної соціальної мережі є свій окремий додаток. Найчастіше додатки використовуються більш широко, тому що вважаються зручнішим видом взаємодії користувача з контентом.

Отже можна сказати, що створення нового програмного продукту, додатку для відвідування подій, - дуже актуально, та він на сучасному ринку буде конкурентоспроможним.

Ступінь вивчення проблеми: Існує ряд програмних продуктів, що дають змогу користувачу шукати та створювати події.

Такими програмними продуктами є Eventbrite, Evenssion та інші. Вони мають ряд особливостей та переваг але їх об'єднує перелік таких проблем, як, наприклад, відсутність картографії, неорієнтованість на український ринок, застарілість інтерфейсів, та інші недоліки, які були описані в аналізі існуючих рішень.

Об'єктом дипломної роботи є процес пошуку та створення подій у будь-якій частині України.

Предметом роботи є додаток для пошуку та створення подій, розроблений за допомогою мови Swift.

Метою роботи є спрощення процесу пошуку та створення подій у будь-якій частині України.

Завданням роботи є розробка за допомогою мови програмування Swift мобільного додатку, який буде давати користувачу можливість фіксувати, переглядати та додавати події у реальному часі.

В процесі дослідження вирішувалися наступні завдання:

- досліджено процес пошуку та створення подій;
- виявлено основні проблеми, які виникають у користувачів під час досліджених процесів;
- розроблено зрозумілий інтерфейс користувача;
- проведено маркетингове дослідження існуючих рішень проблеми та виявлено основні недоліки;
- розроблено додаток для платформи iOS за допомогою мови Swift.

Наукова новизна справжнього дослідження полягає в розробці не тільки iOS-частини системи, а ще й в розробці універсального технічного рішення, яке можна буде легко масштабувати з напливом нових користувачів та ускладненням платформи цілком, а також можна буде використовувати для розробки сайту, або Android-додатку.

Практична значущість результатів дослідження полягає в рішенні проблеми пошуку та створення подій для користувачів, а також в проведенні подальших досліджень цієї проблеми.

# РОЗДІЛ 1. СУЧАСНІ ТЕХНОЛОГІЇ В РОЗРОБЦІ МОБІЛЬНИХ ДОДАТКІВ

## 1.1 Сутність поняття мобільних технологій

Тема проектної задачі «Мобільний додаток для відвідування подій blink розроблений за допомогою мови програмування Swift».

Мета: за допомогою мови програмування Swift розробити мобільний додаток, який буде давати користувачу можливість шукати, переглядати та додавати події у реальному часі.

Мобільний додаток — це спеціально розроблений додаток для конкретної мобільної платформи (iOS, Android). Зазвичай додаток розробляється на мові високого рівня і компілюється в нативний код ОС, що дає максимальну продуктивність.

У додатку дуже багато можливостей для доступу до пристрою. Таким чином через мобільний додаток можна отримати місце розташування користувача, отримати доступ до каталогу файлів, вихід в Інтернет, доступ до камери, мікрофона і динаміків, так само можна отримати доступ до всіх сенсорів телефону. Таким чином мобільний додаток надає дуже широкий спектр можливостей, якими можна скористатися при розробці.

Мобільні телефони стали невід'ємною частиною нашого життя і використовуються потенційними користувачами постійно протягом всього дня, незалежно від місцезнаходження користувача.

Користуючись мобільним телефоном користувач найбільше часу проводить у соціальних мережах.

У кожній соціальній мережі є свій окремий додаток. Найчастіше додатки використовуються більш широко, тому що вважаються зручнішим видом взаємодії користувача з контентом.

Соціальна мережа — це он-лайн сервіс, призначений для відображення і організації соціальних взаємовідносин в Інтернеті. Основне завдання соціальних мереж — це допомога користувачеві поділитися з друзями його публікацією.

Соціальні мережі класифікуються по типу контенту. Є багато різних проектів з різним типом контенту і для різних цілей, кожен з яких зайняв свою нішу. Якщо проаналізувати цю класифікацію, не складно зробити висновок, що зараз соціалізовані вже всі основні тематики в мережі, незалежно від типу контенту; бурхливий розвиток триває тільки в окремих нішах, в яких активно з'являються тематичні соціальні мережі. Але є тип контенту, який на даний момент не достатньо соціалізований. Кожен день в світі відбувається дуже багато подій, за якими цікаво спостерігати, і на які цікаво потрапити. Але окремого порталу або зручного додатку для пошуку, додавання і збереження подій не знайдено. Тому було прийнято рішення розробити додаток для мобільного пристрою, що надає широкий спектр можливостей.

З цим додатком користувач зможе:

- отримувати інформацію про найбільш популярні події у світі;
- шукати найближчі події на мапі, а також користуватися нею для орієнтування на місцевості навколо події, яка зацікавила користувача;
- дивитися якими подіями зацікавлені друзі з «Facebook»;
- ділитися с друзями цікавими подіями;
- додавати свої події;
- зберігати події, які зацікавили користувача;
- спостерігати за запланованими або минулими подіями.

При виборі події користувач буде отримувати всю необхідну інформацію: назва події, її опис, час та місце проведення, а також фрагмент мапи, на якому користувач отримає більш детальну інформацію про місце розташування події. Також якщо подія відбуватиметься в даний момент часу, користувач дізнається про це по напису «live».

Зайти в додаток можна буде двома способами. Перший спосіб – вхід після реєстрації. В такому випадку необхідно буде підтвердити адресу електронної пошти. Другий спосіб – вхід через соціальну мережу «Facebook». Потім в прикладці можна буде додати іншу соціальну мережу до свого додатку.

Вхід через соціальну мережу додає декілька можливостей:

- доступ до друзів користувача та їх інформації с цієї мережі;
- доступ до профілю користувача (фото, інформація).

## 1.2 Мови створення мобільних додатків

Операційну систему iOS випустила компанія Apple у 2007 році. Це не обмежується iPhone та iPad, але я буду говорити лише про створення додатків для цих пристроїв. Окрім власного програмування, існує також міжплатформене програмування, де додаток пишеться відразу для iOS та Android іншою мовою, ніж рідна.

Процес створення власного додатка полягає у написанні програми однією з основних мов програмування iOS: Objective-C або Swift. Цей шлях був створений корпорацією Apple і передбачає слідування всім її ідеям. Apple надає розробникам найновіший комплект для розробки програмного забезпечення Xcode (SDK), документацію та середовище розробки.

### 1.2.1 Objective-C

Мова програмування для додатків iOS, розроблена на початку 1980-х років шляхом перетину C (C) з популярним на той час Smalltalk (спілкування з об'єктами за допомогою повідомлень). Спочатку Objective-C розглядався як просте доповнення до мови C, модифікуючи деякі його синтаксичні конструкції, але після ліцензування він спочатку взяв наступний крок, а потім, як наступник Apple, Objective-C став однією з найпопулярніших мов для розробки додатків для iPhone та iPad. Тому багато типів даних у Objective-C успадкували префікс NS (наступний крок). Це основна мова, якою користується Apple, і її знання дозволяє писати на будь-якій платформі Apple, включаючи macOS.

Плюси:

- високий ступінь обслуговування коду: з кожним оновленням зміни в Objective-C є мінімальними;



- велика кількість документації, технічної літератури та величезна громада. Apple забезпечує та регулярно оновлює свої офіційні книги та ресурси. Ви можете точно дізнатись, які знання потрібні розробнику iOS з нашої статті;
- швидкий перехід з однієї з мов C. Objective-C є розширенням мови C. Це означає, що будь-який код на C також є дійсним кодом для Objective-C, вам просто потрібно звикнути до синтаксису;
- Взаємодія Objective-C у проектах Swift дозволяє одночасно використовувати дві мови.

#### Недоліки:

- якщо ви ніколи не досвідчували програмування і вирішили негайно розпочати роботу з Objective-C, може бути важко зрозуміти правила ООП та стек синтаксису. Труднощі зникнуть у процесі навчання;
- низька читабельність коду: на початку тесту синтаксис здаватиметься складним;
- система динамічного типу даних, що також є плюсом, передбачає можливість помилок навіть під час компіляції. Помилки, зокрема, можуть затримати процес;
- низька продуктивність у порівнянні зі Swift;
- взаємодія з файлами Swift здійснюється за допомогою "моста" (умовного адаптера, який переводить код Swift у Objective-C), що значно уповільнює процес компіляції.

### 1.2.2 Swift

Молода, потужна та універсальна мова програмування з відкритим кодом. Офіційно від Apple 2 червня 2014 р. Вона поєднує в собі все, від C та Objective-C, але не має найкращого вибору останнього через його сумісність з C. Swift використовує потужне друкування заріє. Швидко додає такі функції, як загальні засоби, вимкнення, багаторазова повернення цінності та багато іншого, роблячи розробку додатків гнучкою та цікавою. Нова мова програмування базується на початковому компіляторі, налагоджувачі та фреймворках.

Swift буде знайомий розробникам Objective-C. Він поєднує в собі читабельність названих параметрів з потужністю динамічної об'єктної моделі Objective-C.

Плюси:

- швидкість. Зараз мова майже нарівні з C++, і Купертіно обіцяє, що це не межа;
- спрощена навігація по файлах проекту. У рішенні Objective-C, яке створює два файли для декларування та реалізації, Swift має справу лише з одним. Крім того, назви методів та коментарі між шинами автоматично синхронізуються;
- легко, тому що мова не побудована на C. Наприклад, вам не потрібно ставити крапку з комою в кінці рядка та писати дужки, щоб обернути вашу роботу всередині if / else. Без квадратних дужок Swift схожий на звичайну англійську мову, набагато чистіший та спрощений синтаксис;
- стислість. Кількість коду в Swift менше. Наприклад, за допомогою оператора + можна додати два рядки;
- чудові можливості порівняно з Objective-C. Наприклад, дженерики (дженерики). Універсальний код дозволяє писати гнучкі та загальні функції, які можуть працювати з будь-якими іншими типами. Ви можете написати код, який не повторюється, і виглядати абстрактно;
- повна взаємодія з кодом, написаним на Objective-C, дозвіл на прийняття з двох мов;
- підвищений рівень безпеки. Swift у рішенні Objective-C набирається сильно, що означає, що при оголошенні іменованих параметрів потрібно чітко вказати тип даних, інакше компілятор не вдасться. Це полегшить процес виправлення помилок, оскільки ви можете правильно запустити проблему далеко;
- підтримка динамічних бібліотек. Однією із суттєвих змін Swift є перехід від статичної до динамічної бібліотеки, яка по суті є незахищеними фрагментами коду. Вони приєднуються до програми і "прив'язуються" до мовних версій, що забезпечує стабільну роботу програми.

Недоліки:

- швидко зростає і змінюється. Наприклад, виклик методу може змінитися після оновлення. На щастя, Apple побудувала цей процес таким чином, що написаний код, попередні версії не будуть порушені. Ви побачите лише попередження про те, що ваш код є старою версією, і редактор запропонує оновлення та допоможе вам завершити процес, рядок за рядком;
- взаємодія з режимом Objective-C здійснюється через "міст", що значно уповільнює процес компіляції.

### 1.3 Аналіз існуючих рішень

На сьогоднішній день український ринок подій відчуває явний недолік спеціалізованих додатків безпосередньо для роботи з подіями. Рішень, які дають змогу використовувати один додаток не тільки для відвідування, а також і для організації подій взагалі не має. Але додатки, не призначені безпосередньо для локального ринку існують. Розглянемо характеристики кожного з них в окремо та проведемо порівняльну характеристику.

«Eventbrite». Додаток знаходиться за посиланням:

<https://apps.apple.com/ru/app/eventbrite/id487922291>

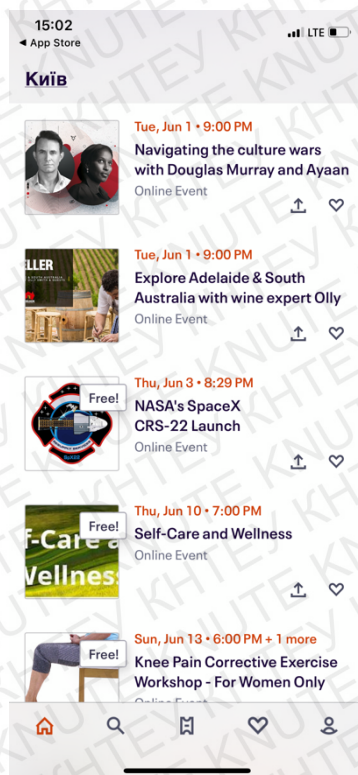


Рисунок 1.1 – Графічний інтерфейс програми «Eventbrite»

Додатком «Eventbrite» користується кілька мільйонів людей по всьому світу і сама програма поєднує в собі зручність, простоту і красу в інтерфейсі але є і ряд мінусів.

Евентбрайт не користується популярністю як в Україні, так і в країнах СНД, відповідно кількість створених подій в місті Києві, наприклад, становить 10 штук по всіх категоріях, що є дуже низьким показником.

Що стосується самих категорій, то їх явно недостатньо для різнобічних користувачів і має бути як мінімум удвічі більше, ніж 7 штук. Так само в додатку відсутні елементи карти, які в свою чергу дуже сильно спрощують пошук подій та так само розуміння того, де саме ця подія проходить. Так само в мінус можна виділити поділ системи на додаток для організаторів і для звичайних користувачів.

Також у додатку не має ніякої інтерактивної частини, тобто не має змоги дивитися якими подіями інтересуються друзі, та пропонувати їм відвідати ту чи іншу подію разом.

Але в «Eventbrite» є також і ряд переваг, наприклад згодом події починають сортуватися з урахуванням інтересів користувача і таким чином система постійно адаптується під поточного користувача і його потреби.

Основною перевагою «Eventbrite» є наявність сайту. За допомогою нього набагато простіше просувати додатки для користувачів, так як сайт одним з перших відображається в пошуковій видачі Google.

У цієї системи дуже багато переваг для користувачів, які викладають події. Такі користувачі мають змогу перевірити популярність своєї події за допомогою зібраної статистики, можуть продавати білети налаштувавши свої банківські дані не виходячи із системи, але за продаж білетів «Eventbrite» утримує процент з прибутку, який становить 10% від ціни одного білета. Це дуже високий відсоток сервісного збору і в майбутньому Blink зможе давати організаторам на українському ринку таку можливість набагато дешевше за рахунок використання місцевих платіжних операторів, наприклад LiqPay або WayForPay.

«Eventssion». Додаток знаходиться за посиланням:

<https://apps.apple.com/ua/app/eventssion-афиша-событий>

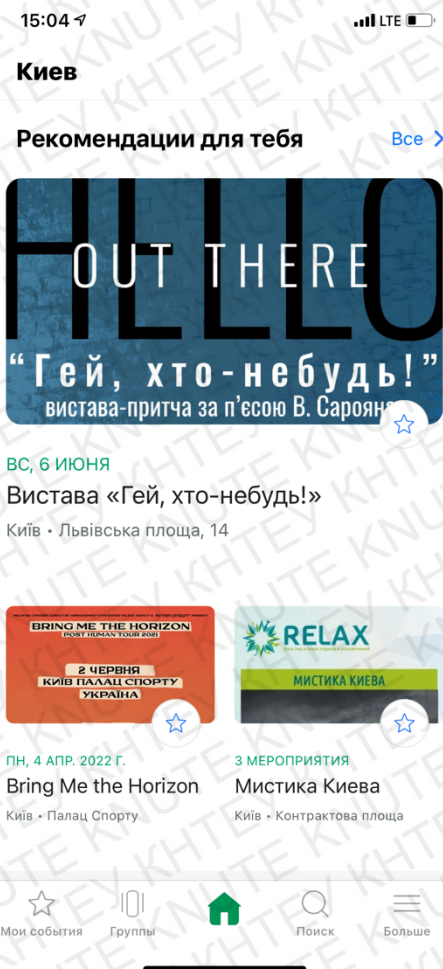


Рисунок 1.2 – Графічний інтерфейс програми «Eventssion»

Додаток було спочатку розроблено для українського ринку, але в підсумку було перейнято занадто багато елементів застарілого дизайну у «Facebook», які псують загальне почуття роботи з програмою, вона відчувається застарілою і дуже повільною. Так само кількість створюваних подій недостатньо велика, щоб тримати користувача постійно в системі.

Також відсутня картографія та елементарна можливість відкрити адресу проведення події в додатку з картою, щоб подивитися як можна дістатися до місця призначення.

Можливість створення подій реалізована тільки через ручне підтвердження користувача, що в свою чергу змушує потенційних організаторів чекати дозволу,

замість того, щоб викладати нові події. Відсутня можливість подивитися хто йде на подію, а також не можна створити або подивитися його план.

### **Висновки до розділу**

Підводячи підсумки можна вивести кілька основних мінусів, узагальнюючих наведені вище програми:

1. Відсутність картографії для зручності просування до місця проведення події користувача.
2. Застарілі рішення для інтерфейсів.
3. Неорієнтованість на український ринок.
4. Відсутність елементів взаємодії зі своїми друзями (можливість поділитися подією, можливість подивитися куди ходять друзі).
5. Незручна робота з організацією подій, процес викладення події займає багато часу та більшість запрошеної інформації у цьому процесі далеко не завжди потрібна та використовується.
6. Відсутність інтерактивності, додатками хочеться користуватися тільки тоді, коли потрібно знайти якусь подію, на самій події немає тяги скористатися додатком.
7. Відсутність можливості подивитися план події або ж створити його, якщо користувач є організатором.
8. Висока ціна сервісного збору для організаторів.
9. Бюрократизований процес вводу нового організатора на платформу.
10. Для роботи у якості організатора потрібно встановлювати додатковий додаток, таким чином людина може не мати уявлення як його розміщені події виглядають для кінцевих користувачів.
11. Відсутня можливість подивитися чи є організатор перевіреним
12. За замовчуванням недоступні засоби зв'язку з організатором.

## РОЗДІЛ 2. ОРГАНІЗАЦІЯ РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ “BLINK”

### 2.1 Огляд існуючих технологій проектування

Процес створення власного додатка полягає у написанні програми однією з основних мов програмування iOS: Objective-C або Swift. Цей шлях був створений корпорацією Apple і передбачає слідування всім її ідеям. Apple надає розробникам найновіший комплект для розробки програмного забезпечення Xcode (SDK), документацію та середовище розробки.

Обидві мови розробки додатків iOS пов'язані з об'єктно-орієнтованим програмуванням (ООП) і успішно слідують його основним парадигмам: успадкування, поліморфізм, інкапсуляція та абстракція. Простіше кажучи, ООП - це стиль кодування, який дозволяє розробникам групувати подібні завдання за класами. Код дотримується принципу DRY (don't repeat yourself — не повторюйся) і стає простим в обслуговуванні.

### 2.2 Особливості мови програмування Swift

Objective-C поступово поступається дорогу молодому Swift, який Apple всіляко популяризує. У минулому 2018 році в рейтингу Tiobe Objective-C займав нижчу позицію по відношенню до Swift.

Враховуючи той факт, що Swift був розроблений не так давно, він враховує безліч недоліків, які були властиві більш древнім мовам програмування. Наприклад, Swift добре підходить для початку роботи та швидкого зростання.

Створюючи цю мову програмування, команда націлювалася на такі цілі, як простота навчання та прискорення циклу розвитку. Окрім того, Swift має простіший та стисліший синтаксис, ніж Objective-C, що економить ваш час, писаючи менше коду.

Ще однією важливою особливістю є те, що Swift має високу продуктивність. Хоча це мова високого рівня, призначена для швидкого навчання, вона досить швидка. За словами його творців, Swift майже втричі швидший за Objective-C та



майже у 8 разів швидший за Python. Триває робота, щоб випереджати C ++ за швидкістю.

Окрім швидкості, Swift має ще одну важливу перевагу - наявність сучасних бібліотек та фреймворків для написання функціонального коду.

Swift, як і більшість інших мов програмування, є повністю відкритим і безкоштовним. Більше того, це програмне забезпечення з відкритим кодом. Незважаючи на те, що в сучасних реаліях багато мов програмування зазвичай пропонують відкритий код, для Apple це досить дивно, оскільки ця компанія зазвичай не робить таких кроків. Однак така поведінка розробника цілком виправдана. Майже відразу навколо мови сформувалося велике співтовариство, де розробники пропонують рішення проблем, виправляють помилки та роблять корисні виправлення.

Ще однією характеристикою Свіфта є безпека. Завдяки багатьом нововведенням, а також вдосконаленням синтаксису, Swift став набагато безпечнішим, ніж об'єктно-орієнтований C. Наприклад, завдяки вдосконаленням роботи з пам'яттю, розробники практично виключили можливість несанкціонованого доступу до даних. Крім того, кількість критичних сценаріїв зменшилась завдяки ефективнішій обробці сценаріїв.

Важливим моментом, який слід зазначити, є здатність візуалізувати результат. Swift має спеціальну пісочницю - Playground, яка дозволяє продемонструвати роботу програми. Іноді такий підхід дуже корисний, оскільки дозволяє швидко знаходити проблеми у коді та скорочувати час розробки.

Де використовується Свіфт? Як зазначалося вище, мова призначена для роботи з платформами iOS та macOS. Це означає, що він не є універсальним. Однак це не означає, що сфера дії цієї PL дуже обмежена, а попит на розробників низький.

За допомогою Swift ви можете розробити будь-який додаток для свого мобільного пристрою iOS. Деякий час ходили чутки, що існував план перенести Android на Swift. Однак поки що ця інформація нічим не підтверджена. Крім того, розробники Android мають власні мови програмування, і такий глобальний трансфер навряд чи сприятиме популяризації платформи.

Які мобільні програми можна створювати за допомогою Swift? Якщо ви відкриєте App Store і ретельно дослідите вміст, ви можете виявити, що розробники мають безмежні можливості для фантазії. Тут може бути що завгодно - від найпростіших віджетів для мобільних платформ на базі iOS до найскладніших програм.

Чи є у цієї мови програмування недоліки? Так, і одна з них полягає в тому, що для взаємодії з Objective-C використовується спеціальний «міст», який значно уповільнює процес побудови будь-якого додатка. Також до недоліків цієї мови можна віднести те, що вона постійно змінюється. З одного боку, це добре, з іншого боку, якщо Apple не будувала процес певним чином, старий код може не працювати.

### **2.3 Розробка вимог до додатку**

У цьому розділі будуть описані вимоги до додатку, що розробляється. Основним моментом, що враховувався в ході розробки вимог, було наступне: максимально зручний у використанні мобільний додаток, який повинен поєднувати всі перераховані нижче аспекти, і при розробці програми необхідно усунути якомога більше недоліків, виявлених в ході аналізу аналогічних програмних продуктів.

#### **2.3.1 Вимоги до графічного інтерфейсу**

Взаємодія користувача з прикладним програмним забезпеченням, який входить до складу системи повинна здійснюватися засобами візуального графічного інтерфейсу (GUI). Інтерфейс системи повинен бути зрозумілим і зручним, не повинен бути перевантаженим графічними елементами і повинен забезпечити швидке відображення екранних форм.

Інтерфейс повинен бути орієнтований на використання мобільного або планшетного пристрою.

Всі екранні форми інтерфейсу повинні бути виконані в одному графічному дизайні, з однаковим розміщенням основних елементів управління і навігації.

Додаток розрахован не тільки для країн СНД, але і для усіх країн світу, тому інтерфейс буде реалізований на англійській мові.

При запуску додатку з'являється активність авторизації, яка показана на рисунку 2.1.

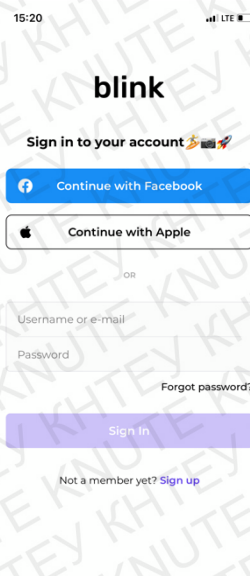


Рисунок 2.1 – Активність авторизації

Після авторизації користувача з'являється основна активність додатку, відкрита на вкладці «Explore» (Пошук), яка показана на рисунку 2.2.

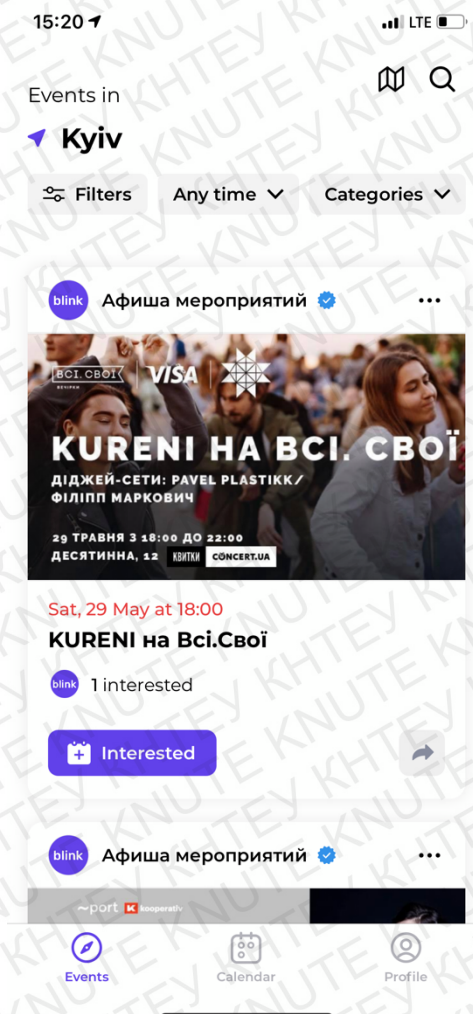


Рисунок 2.2 – Вкладка «Popular» (Популярне)

Обмеження на даній вкладці:

- «Час проведення» – символічний рядок з кількістю символів від 0 до 30;
- «Назва події» – символічний рядок з кількістю символів від 0 до 60;
- «Місце розташування» – символічний рядок з кількістю символів від 0 до 100;
- «Категорія» – символічний рядок з кількістю символів від 0 до 30;
- «Автор» – символічний рядок з кількістю символів від 0 до 30;
- «Кількість зацікавлених» – ціле число;

– «Фото події» – файл з розширенням jpg або png розміром до 10 мегабайтів.

При натисканні на кнопку з іконкою календаря, подія додається у вкладку «Calendar» (Календар) рис. 2.6.

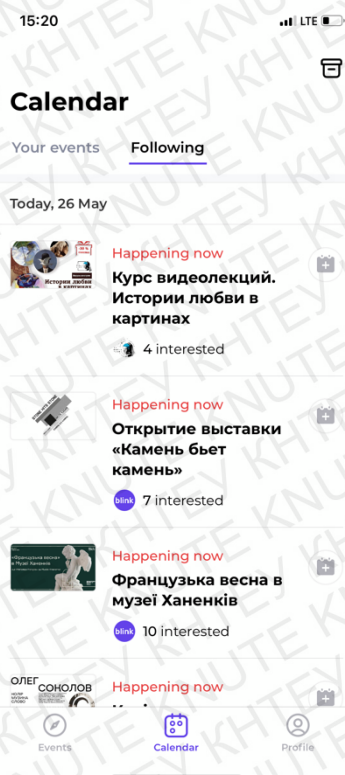


Рисунок 2.3 – Вкладка «Friends» (Друзі)

Обмеження на даній вкладці:

- «Назва події» – символічний рядок з кількістю символів від 0 до 60;
- «Ім'я друга» – символічний рядок з кількістю символів від 0 до 60;
- «Фото події» – файл з розширенням jpg або png розміром до 10 мегабайтів;
- «Фото профілю друга» – файл з розширенням jpg або png розміром до 10 мегабайтів;
- «Час проведення» – символічний рядок з кількістю символів від 0 до 30.

Також у користувача є можливість обрати потрібну йому вкладку з представлених в додатку: «Calendar» (Календар) (рисунок 2.6), «Profile» (Профіль) (рисунок 2.7).

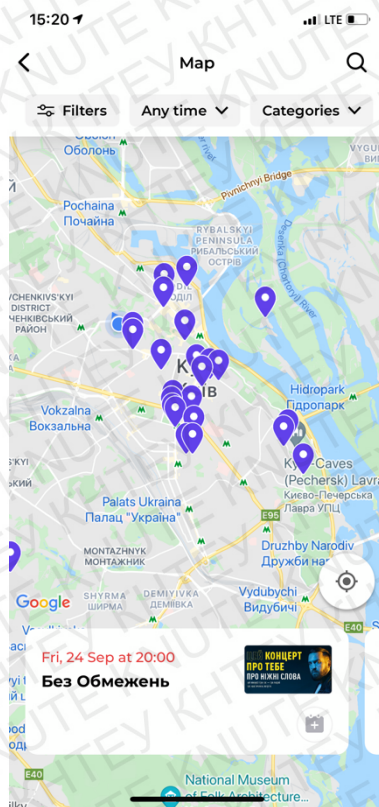


Рисунок 2.4 – Вкладка «Мар»(Мапа)

Обмеження на даній вкладці:

- «Час проведення» – символічний рядок з кількістю символів від 0 до 30;
- «Назва події» – символічний рядок з кількістю символів від 0 до 60;
- «Фото події» – файл з розширенням jpg або png розміром до 10 мегабайтів.

Для відображення мапи у додатку використовується бібліотека «Google maps».

При натисканні на кнопку в верхній частині активності користувачу дається можливість обрати критерії пошуку.

При «свайпі» події у області під мапою, з'являється нова подія, а мапа переходить до її місцезнаходження.

При натисканні на будь-яку подію відкривається активність з детальною інформацією «Info»(Інформація), яка показана на рисунку 2.8.

Рисунок 2.5 – Вкладка «New event»(Нова подія)

Обмеження на даній вкладці:

- «Event name»(назва події) – символний рядок з кількістю символів від 0 до 100;
- «Description»(опис) – символний рядок з кількістю символів від 0 до 255;
- «Location»(місце розташування) – символний рядок з кількістю символів від 0 до 100;
- «Формат дати» – день неділі, Місяць число місяця рік;
- «Формат часу» – година:хвилина.

При натисканні на область «Add photo»(Додати фото) відчиняється галерея з пристосування користувача.

При натисканні на поле під надписом «From»(З) відчиняється діалог з користувачем, де він зможе обрати дату та час початку події.

При натисканні на поле під надписом «To»(До) відчиняється діалог з користувачем, де він зможе обрати дату та час закінчення події.

При натисканні на кнопку з іконкою галочки, користувач додає подію до бази даних додатку. Кнопка переходить до активного стану, коли усі поля вкладки заповнені. Якщо є незаповнене поле, користувач не зможе натиснути на цю кнопку.

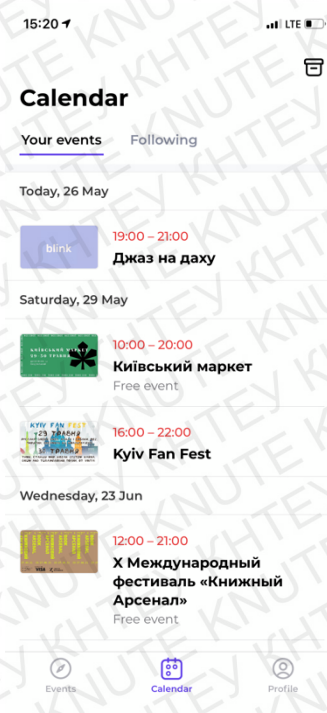


Рисунок 2.6 – Вкладка «Calendar»(Календар)

Обмеження на даній вкладці:

- «Час проведення» – символічний рядок з кількістю символів від 0 до 30;
- «Назва події» – символічний рядок з кількістю символів від 0 до 60;
- «Фото події» – файл з розширенням jpg або png розміром до 10 мегабайтів.

При натисканні на будь-яку подію відкривається активність з детальною інформацією «Info»(Інформація), яка показана на рисунку 2.8.



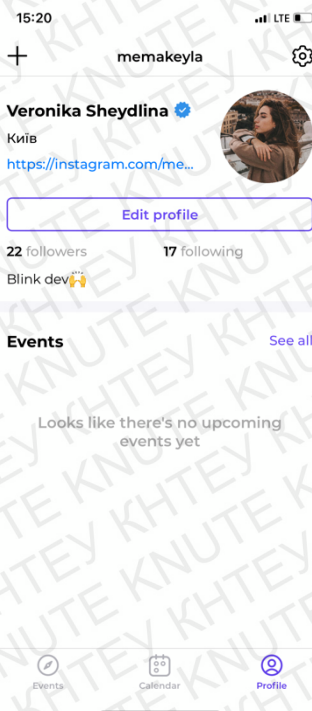


Рисунок 2.7 – Вкладка «Profile»(Профіль)

Обмеження на даній вкладці:

- «Повне ім'я» – символний рядок з кількістю символів від 0 до 60;
- «Місце проживання» – символний рядок з кількістю символів від 0 до 100;
- «Інформація» – символний рядок з кількістю символів від 0 до 100;
- «Фото профілю» – файл з розширенням jpg або png розміром до 10 мегабайтів;
- «Кількість подій» – ціле число;
- «Кількість друзів» – ціле число.

При натисканні на фото користувача відчиняється галерея користувача, де він може обрати нове фото для профілю.

При натисканні на кнопку з іконкою шестерінки відкривається активність з настройками додатку.

При натисканні на будь-яку подію відкривається активність з детальною інформацією «Info», яка показана на рисунку 2.8.

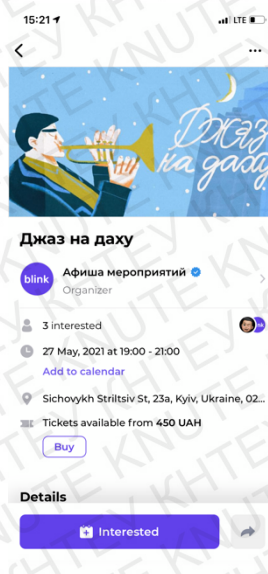


Рисунок 2.8 – Вкладка «Info»(Інформація)

Обмеження на даній вкладці:

- «Час проведення» – символічний рядок з кількістю символів від 0 до 30;
- «Назва події» – символічний рядок з кількістю символів від 0 до 60;
- «Місце розташування» – символічний рядок з кількістю символів від 0 до 100;
- «Категорія» – символічний рядок з кількістю символів від 0 до 30;
- «Автор» – символічний рядок з кількістю символів від 0 до 30;
- «Кількість зацікавлених» – ціле число;
- «Фото події» – файл з розширенням jpg або png розміром до 10 мегабайтів;

При натисканні на кнопку з іконкою ока можна додати обрану подію до вкладки «Події», чи видалити її.

### **2.3.2 Вимоги до надійності програми**

Надійне (стійке) функціонування програми має бути забезпечене виконанням користувачем сукупності організаційно-технічних заходів, перелік яких наведено нижче:

- організацією безперебійного живлення технічних засобів;
- використанням ліцензійного програмного забезпечення;
- регулярним виконанням вимог щодо захисту інформації.

### **2.3.3 Вимоги до вихідного коду і мов програмування**

Розробляема система повинна передбачати можливість її подальшого розвитку, модифікації і включення нових функцій в систему, поліпшення коду.

Розробляема система повинна являти собою мобільний додаток, доступний в App Store (онлайн-магазин додатків для ОС iOS).

Структура програми повинна передбачати можливість її розвитку за рахунок розробки і включення в неї нових модулів.

В якості мови програмування необхідно обрати об'єктно-орієнтовану мову програмування. Наприклад: Swift, Objective-C. Вибір інших мов недоцільний.

### **2.3.4 Вимоги до програмних засобів, які використовуються програмою**

Системні програмні засоби, що використовуються програмою, повинні бути представлені ліцензійною версією операційної системи iOS (12.0 і вище), а також ліцензійною версією програми «Blink».

### **Висновки до розділу**

На даний момент можна знайти досить багато оголошень для набору розробників, які вміють писати код на Swift. Це пов'язано з тим, що кількість програм та можливостей мобільної платформи iOS, а також кількість користувачів macOS постійно зростають.

Відповідно зростає й попит на написання певних програм для мобільних та стаціонарних пристроїв. Тому розробнику Swift сьогодні не важко влаштуватися на роботу. В Україні не так мало вакансій, які потребують саме таких фахівців.

Що стосується перспектив розвитку мови, Apple навряд чи відмовиться від нього в найближчі 5-10 років. Він не був розроблений для цього, і на це було витрачено багато грошей. Швидше за все, Swift буде продовжувати рости, і враховуючи зростаючу спільноту та відкритість коду Java, процес просування вперед буде помітно швидшим.

## РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВІДВІДУВАННЯ ПОДІЙ “BLINK”

### 3.1 Розробка структури мобільного додатку “blink”

Для того, щоб визначити структуру розроблюваної системи ми побудуємо діаграму компонентів, яка описує особливості фізичного представлення системи. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними.

Діаграма компонентів розробляється для наступних цілей:

- візуалізації загальної структури вихідного коду програмної системи;
- специфікації виконуваних варіантів програмної системи;
- забезпечення багаторазового використання окремих фрагментів програмного коду;
- представлення концептуальної і фізичної схем баз даних.

На додатку Б зображена діаграма компонентів розгортання.

Для уявлення фізичних сутностей застосовується спеціальний термін - компонент. Компонент реалізує деякий набір інтерфейсів і служить для загального позначення елементів фізичного представлення моделі. Для графічного представлення елемента служить спеціальний символ - прямокутник зі вставленими зліва двома і більше дрібними прямокутниками. Усередині великого прямокутника записується ім'я компонента і, при необхідності, деяка додаткова інформація. Зображення цього символу може незначно змінитися в залежності від характеру асоційованої з компонентом інформації.

Діаграма компонентів показує розподіл програмної системи на структурні компоненти і зв'язки між ними.

Дана діаграма містить 2 основних компоненти: «Пристрій користувача», «Сервер».

У компоненті «Пристрій користувача» знаходиться «Додаток Blink», «Сервер» містить «БД Blink». Обмін даними між основними компонентами відбувається по набору мережевих протоколів TCP / IP, що включають в себе протокол прикладного рівня HTTP (HTTPS).

### **3.2 Розробка моделі функціонування мобільного додатку**

На додатку А зображена діаграма використання проектної задачі. На діаграмі зображено два актори: авторизований і неавторизований користувачі. У авторизованого актора є можливість виконувати в додатку наступні дії: додавання подій; перегляд подій; перегляд профілів; перегляд подій, якими зацікавлені друзі; перегляд подій на мапі; збереження подій. Неавторизований користувач має доступ до всіх функцій окрім перегляду подій, якими зацікавлені друзі, також він не має змоги додавати події. Можливість переглядати події у якості неавторизованого актора появиться згодом.

Діаграма варіантів використання зображена на додатку А.

### **3.3 Програмна реалізація мобільного додатку “blink”**

У цьому розділі будуть обговорені ключові моменти реалізації функціональних можливостей програми: описані інструменти, за допомогою яких відбувалася розробка, та наведені розрахунки вихідного коду програми, що відповідають за реалізацію базової функціональності програми.

Мобільний додаток для iOS складається з набору активностей, кожен з яких відповідає вікну керування. Кожна активність представлена в проекті класом, реалізованим на мові Swift, що зберігається в однойменному файлі з розширенням .swift.

Додаток також містить різні ресурси, такі як графіка, текст, аудіофайли, а також файли макета та макети меню, що зберігаються серед папки «Project». Такі файли, як бази даних та файли шрифтів, знаходяться в каталозі ресурсів "Рисунок 5".

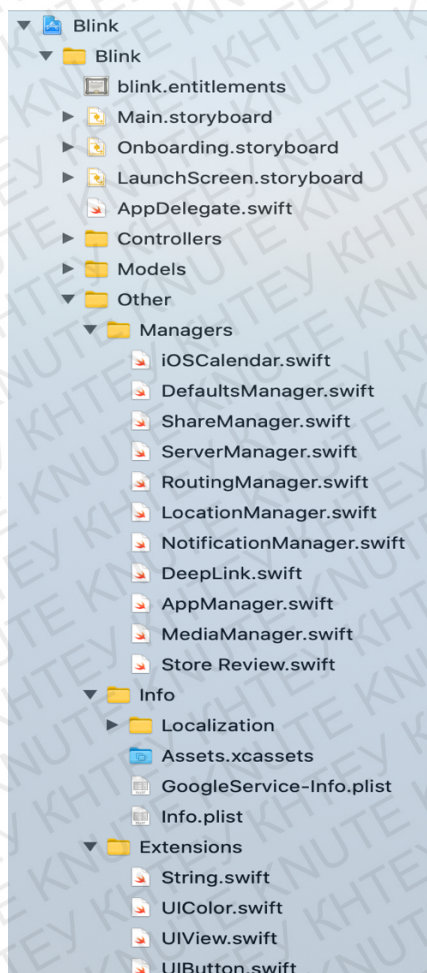


Рисунок 3.1 – Ресурси проекту

### 3.3.1 Відображення даних

Презентація даних, тобто користувальницький інтерфейс, відіграє важливу роль у розробці програмного забезпечення загалом та мобільних додатків зокрема. Інтерфейс користувача - це перше, що бачить користувач під час запуску програми, і перше, що служить імпульсом, щоб скласти враження про програму. Xcode надає розробникам зручний інструмент побудови інтерфейсу під назвою Interface Builder (приклад можна побачити на рис. 3.2).

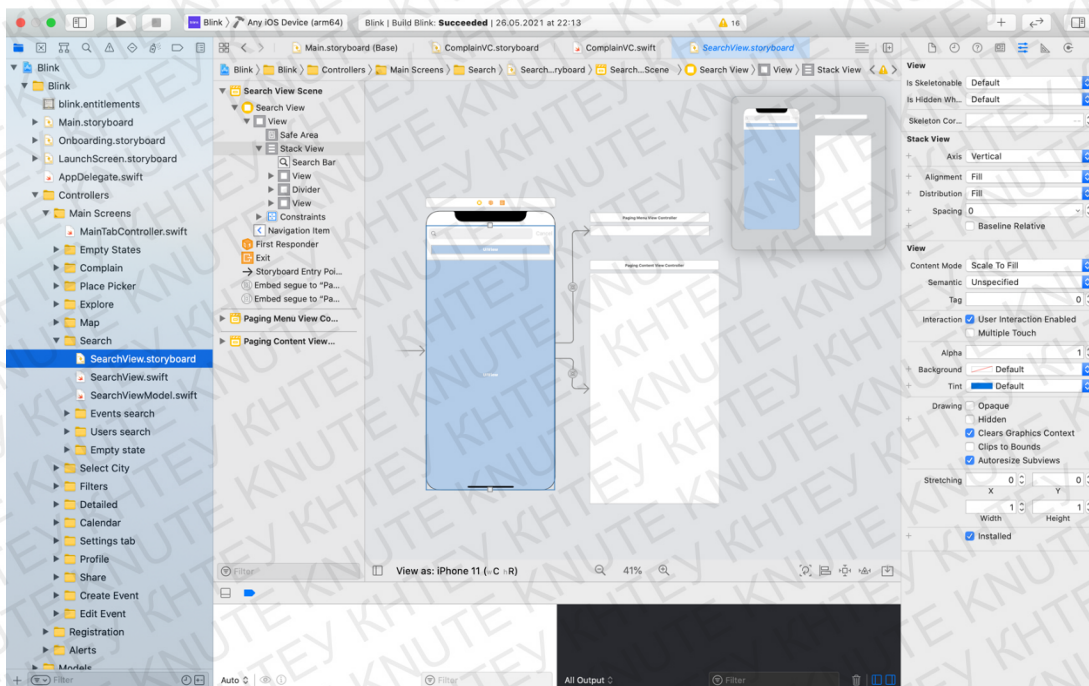


Рисунок 3.2 – Interface Builder

Interface Builder дає можливість розробляти користувальницький інтерфейс без написання коду. Interface Builder дозволяє перенести вікна, кнопки, текстові поля та інші об'єкти бібліотеки у візуальне представлення інтерфейсу Storyboard, налаштувати їх властивості та об'єднати їх для створення функціонального інтерфейсу користувача. Оскільки структури Cocoa Touch будуються з використанням концепції MVC, інтерфейси створюються незалежно від їх змісту, а інструменти Interface Builder прив'язані до відповідних об'єктів поведінки. Крім того, кожен об'єкт в Storyboard, включаючи саму Storyboard, має ідентифікатор, який можна використовувати для взаємодії з елементами інтерфейсу через код (рис 3.3).



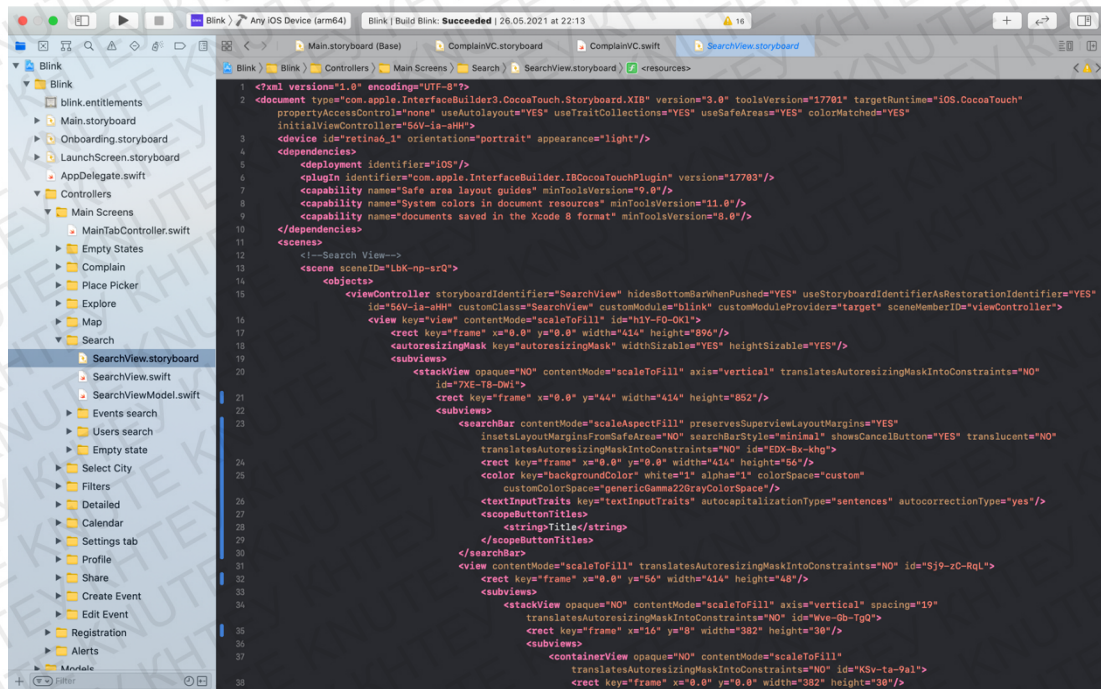


Рисунок 3.3 – Interface Builder source code

### 3.3.2 Взаємодія з сервером

Щоб реалізувати цілі роботи, додаток «blink» повинен мати можливість завантажувати дані на сервер й отримувати їх з сервера. Метод POST використовується для надсилання інформації на сервер, а для їх отримання використовується тип запису GET HTTP. Передача персональних даних користувача, таких як логін та пароль, відбувається до системи за допомогою захищеного протоколу HTTPS. Надсилання інформації в Swift за допомогою iOS SDK реалізується наступним чином.

Ми створюємо запит з параметрами POST:

```

924 let domain = "https://api.blink.so/events/create"
925 let urlComponent = URLComponents(string: domain)
926 var request = URLRequest(url: (urlComponent?.url!))
927
928 request.httpMethod = "POST"
929 request.setValue("application/json", forHTTPHeaderField: "Content-Type")
930 request.httpBody = try! JSONEncoder().encode(event)
931 request.headers = getHeaders()

```

Надсилаємо запит на сервер:

```
933 AF.request(request).validate().responseJSON {
934     [weak self]
935     response in
```

В результаті виконання цих методів ми отримуємо файл з усіма даними, які були отримані в результаті виконання запиту:

```
937     switch response.result {
938     case .success:
939         if response.response?.statusCode == 200 {
940             do {
941                 let jsonObject = try JSONSerialization.jsonObject(with: response.data ?? Data(), options: .mutableContainers) as? [String:
942                     AnyObject]
943                 let decoder = JSONDecoder()
944                 decoder.keyDecodingStrategy = .convertFromSnakeCase
945                 let newEvent = try decoder.decode(Event.self, from: JSONSerialization.data(withJSONObject: jsonObject ?? {}, options: []))
946                 onSuccess(newEvent)
947             } catch _ as NSError {
948                 onFailure(.generalError)
949             }
950         }
951     }
952 }
```

Код об'єкту «Event» представлений у додатку В.

### 3.3.3 Робота з картами

Навігація серед найближчих до користувача подій є однією з ключових цілей проекту й передбачає використання карт в додатку. IOS SDK включає платформу Мар Kit, яка забезпечує інтерфейс для вбудови карти в додаток. Крім того, набір карт дозволяє додавати маркери та анотації на карту, а також геокодування для присвоєння географічних координат об'єктам карти та зворотне геокодування для отримання координат об'єкта.

Однак рамка Мар Kit відображає стандартні карти Apple, що надаються серед інших інструментів розробника iOS, що мають один суттєвий недолік – обмежена деталізація, яка не дозволяє відображати необхідні будівлі на карті. Деталізація є однією з основних особливостей карти, необхідної у коректній роботі додатку для пошука подій, тому було вирішено відмовитись від вбудованого фреймворку Мар Kit та використовувати Google Maps SDK у якості інструменту роботи з картами.

Додавання карти на екран, визначення центру карти та рівень масштабування відбувається таким чином:

```

118 // MARK: - Private methods
119
120 private func setupMap() {
121     mapView.delegate = self
122     mapView.isMyLocationEnabled = true
123     mapView.isUserInteractionEnabled = false
124     mapView.settings.myLocationButton = true
125
126     let mapBottomPadding = cardView.frame.size.height + view.safeAreaInsets.bottom + 16
127     mapView.padding = UIEdgeInsets(top: 0, left: 0, bottom: mapBottomPadding, right: 0)
128
129     let location = LocationManager.kyivLocation
130     let camera = GMSCameraPosition(
131         latitude: location.coordinate.latitude,
132         longitude: location.coordinate.longitude,
133         zoom: 15)
134
135     mapView.camera = camera
136 }

```

```

477 extension MapView: GMSMapViewDelegate {
478
479     // Enable map when it's finished loading
480     func mapViewSnapshotReady(_ mapView: GMSMapView) {
481         mapView.isUserInteractionEnabled = true
482     }
483
484     func mapView(_ mapView: GMSMapView, markerInfoWindow marker: GMSMarker) -> UIWindow? {
485         guard
486             let id = marker.title,
487             let selectedEvent = viewModel.getEvent(with: id),
488             let eventIndexPath = viewModel.getEventIndex(for: id)
489             else { return nil }
490
491         let infoWindow = MarkerInfoWindow.instantiateFromNib()
492         infoWindow.set(title: selectedEvent.title, titleWidth: view.frame.size.width - 48)
493
494         eventsCollectionView.scrollToItem(at: IndexPath(row: eventIndexPath, section: 0), at: .centeredHorizontally, animated: true)
495
496         return infoWindow
497     }
498
499     func mapView(_ mapView: GMSMapView, didTapInfoWindowOf marker: GMSMarker) {
500         guard
501             let eventId = marker.title,
502             let index = viewModel.getEventIndex(for: eventId)
503             else { return }
504
505         viewModel.didSelectEvent(at: index)
506     }
507
508     // Map did end moving
509     func mapView(_ mapView: GMSMapView, idleAt position: GMSCameraPosition) {
510         if mapView.camera.zoom < maxMapZoom {
511             makeSearchButton(visible: false)
512         } else {
513             checkForEvents(currentMapView: mapView)
514         }
515     }
516 }

```

SDK Карти Google надає широкий спектр можливостей роботи з картами та забезпечує повне охоплення всіх навігаційних завдань додатку для пошуку подій.

### 3.3.4 Способи взаємодії з користувачем за межами додатку

За допомогою програми тримати користувача проінформованим про останні зміни досить складно, так як при зачиненні додатку в системі з'єднання з користувачем втрачається і потрібно шукати інші можливості, з допомогою яких можна повернути користувача в додаток де можуть бути, наприклад додані нові події.

Взаємодія з користувачем поза додатку може бути організована за допомогою двох видів повідомлень

- пуш повідомлення;
- імейл повідомлення.

Обидва способи використовуються в Blink, але найбільший вплив на користувача мають пуш повідомлення, так як їх користувач може побачити відразу ж, як тільки візьме мобільний пристрій з включеним інтернетом до рук.

Для роботи з пуш повідомленнями використовується сервіс Firebase Cloud Messaging.

Firebase Cloud Messaging (FCM) - це рішення для обміну повідомленнями між платформами, яке дозволяє надійно надсилати повідомлення без будь-яких витрат.

Використовуючи FCM, можна повідомити клієнтську програму про те, що нові події чи інші дані доступні для синхронізації. Можна надсилати повідомлення-сповіщення для повторного залучення та утримання користувачів. У випадках використання, таких як обмін миттєвими повідомленнями, повідомлення може перенести корисні дані до 4 КБ на клієнтську програму.

При встановленні додатку Blink кожному девайсу користувача додається унікальний токен, який відправляється на сервер під час авторизації користувача для подальшого використання.

Система має 9 різних стандартних повідомлень для користувачів:

- коли хтось з підписок користувача додає нову поточний користувач отримає повідомлення;
- коли подію, на яку збирається піти користувач відмінюють, він отримає повідомлення;

- за час до початку події, в якій заінтересован поточний користувач, система оповістить його повідомленням;
- коли подія, на яку збирався піти користувач, змінює свою інформацію, він отримає повідомлення;
- коли план події змінюється, заінтересовані користувачі отримають повідомлення;
- якщо в користувача є заплановані події на поточний день, зранку йому прийде повідомлення.
- коли новий користувач заінтересувався подією, організатору події прийде повідомлення;
- коли хтось підписується на користувача, йому прийде повідомлення;
- якщо у платформи Blink є якісь нововведення, усім користувачам прийде маркетингове повідомлення.

Усі повідомлення мають одну з трьох мов, які підтримуються платформою Blink:

- українська;
- російська;
- англійська.

Данні про поточну мову, встановлену на девайсі користувача передаються вперше під час авторизації, а також кожного разу, коли користувач відкриває додаток. На рисунку 5.1 можна побачити приклад повідомлення на платформі Blink.

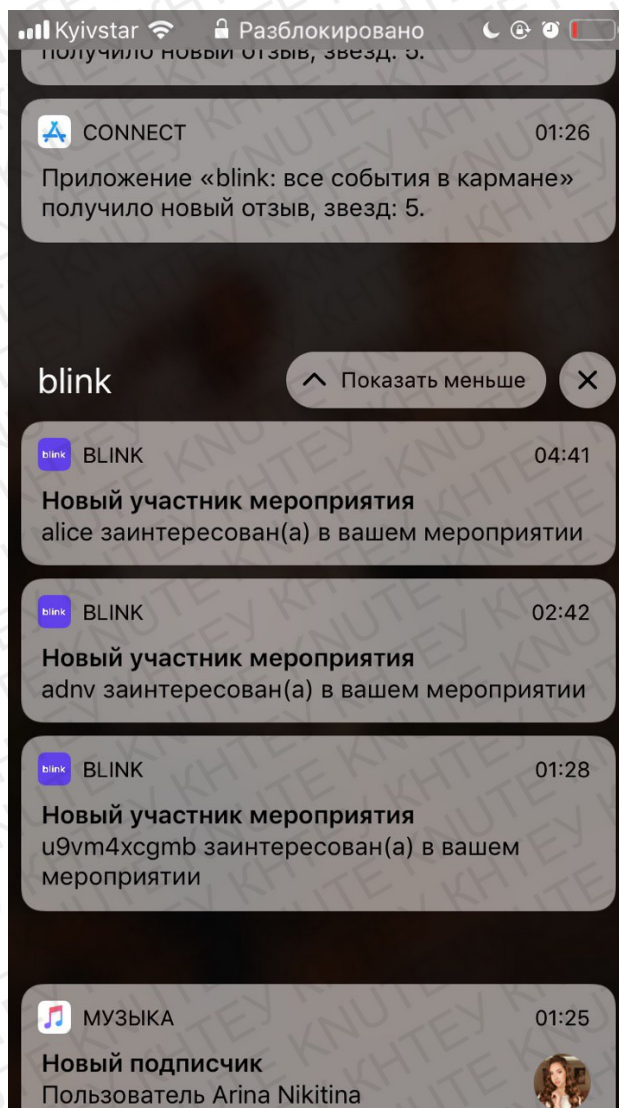


Рисунок 3.4 – Пуш повідомлення

Таким чином користувач завжди залишається повідомленим про усі нововведення на платформі.

Другим типом повідомлень є письма на електронну адресу користувача. Завдяки цьому типу можна розповісти користувачу за нововведення в додатку, також можна переконатись чи є електронна адреса дійсною при реєстрації.

Типи імейл листів на платформі Blink:

– повідомлення під час реєстрації, яке містить посилання для підтвердження адреси користувача. Без переходу по цьому посиланню користувач

не буде мати змоги користуватися додатком, таким чином здійснюється перевірка на дійсність адреси;

- повідомлення для зміни адреси користувача, яке містить посилання для підтвердження цієї дії, без переходу по якому зміна адреса неможлива;
- повідомлення про скидання пароля користувача, яке містить посилання на сайт, де можна встановити новий пароль, цієї дії, без переходу по якому зміна паролю неможлива;
- повідомлення про нововведення у додатку, які містять усю нову інформацію про введення нового функціоналу, та інше;
- що недільні повідомлення з подіями, які збираються з урахуванням інтересів користувача.

На рисунку 5.2 можна подивитися приклади імейл листа для підтвердження реєстрації.

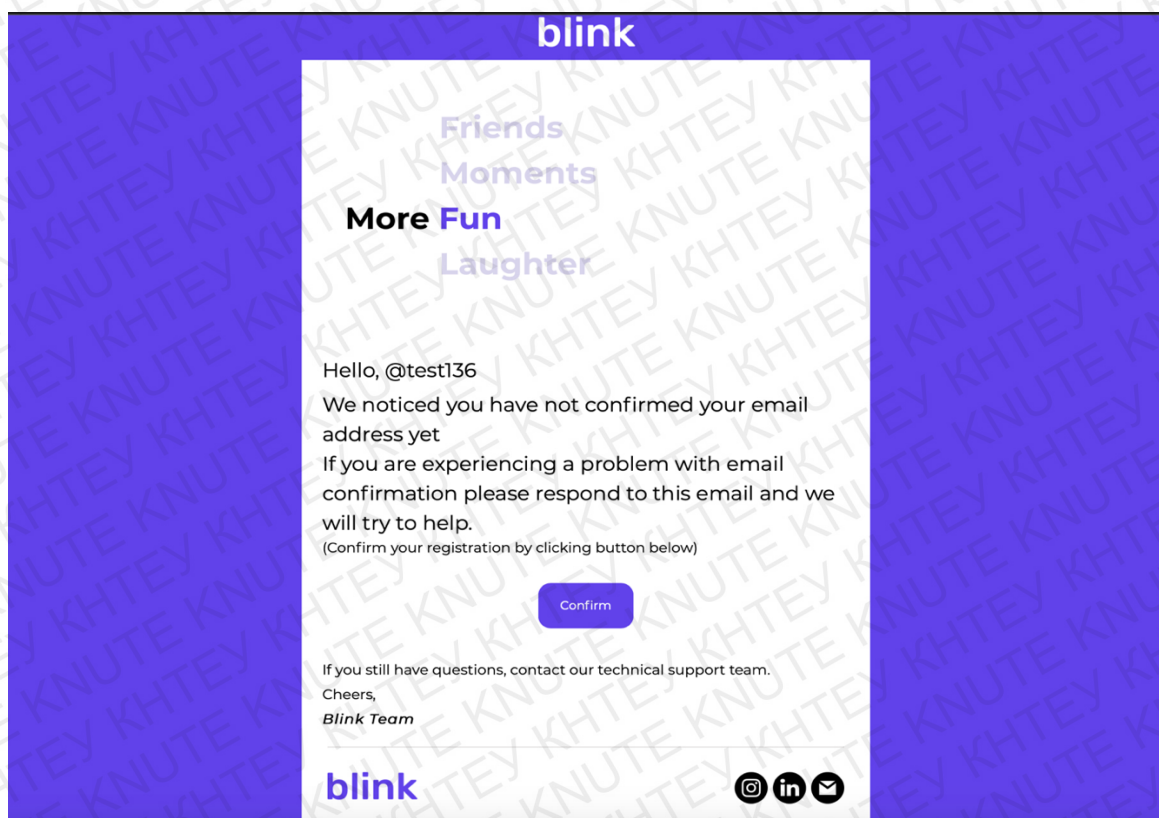


Рисунок 3.5 – Імейл повідомлення для реєстрації користувача

Також на рисунку 5.3 можна подивитися приклад що недільного імейл листа с підібраними для користувача подіями.



Рисунок 3.6 – Імейл повідомлення з підібраними для користувача подіями

Таким чином використовуючи імейл повідомлення можна не тільки перевірити чи є адреса користувача дійсною, але й мати змогу тримати з користувачами постійний зв'язок.

### Висновки до розділу

Відповідно до вимог була спроектована та розроблена архітектура додатку «blink». У процесі розробки додатку також відбувалось поетапне тестування з метою виявлення програмних помилок та невідповідність ТЗ (технічному завдання). Для



цього був створений емулятор смартфона з різними діагональними екранами для різних версій iOS. Додаток послідовно запускався на цих емуляторах, його поведінка аналізувалось і за необхідності за результатами аналізу вносились зміни в код.

Для тестування окремих модулів роботи з базовими даними в текстовій програмі були введені спеціальні функції, що дозволяють аналізувати базу даних та, при подіях на помилку, виводити повідомлення в системному журналі. Вони також відомі як юніт-тести. Наприклад, при змінах у базі даних проводиться перевірка цілісності баз даних (перевірка відповідності ключів – індексам) після чого при необхідності виводиться повідомлення в системний журнал.

## ВИСНОВКИ ТА РЕЗУЛЬТАТИ

Метою написання даної випускної роботи була розробка мобільного додатку для відвідування подій «blink», розробленого за допомогою мови програмування Swift.

Поставлена мета досягнута в повному обсязі:

- розроблено архітектуру та інтерфейс мобільного додатка;
- була зібрана та проаналізована інформація про аналогічні додатки та продуктові рішення;
- розроблено та реалізовано взаємозв'язок із серверною частиною програми;
- розроблена модель використання карти для навігації в додатку та реалізована функціональність роботи з даними геоконтексту;
- програмне забезпечення «blink» для платформи iOS розроблено та опубліковано в App Store.

Використання цього додатка значно полегшує пошук подій, їх виявлення в найближчому для користувача радіусі, додавання власних подій для подальшого просування і збереження подій.

Експлуатація програми не вимагає спеціальної кваліфікації та розрахована на постійне використання користувачем, знайомим з основними принципами взаємодії з інтерфейсом цільової операційної системи (iOS 12.0 і вище).

Користувачами мобільного додатка є користувачі смартфонів (з iOS 12.0 і вище), які зареєстровані в соціальних мережах (Facebook), або просто мають вихід в інтернет.

Методами вирішення поставленого завдання були:

- технології розробки програмних продуктів на основі візуального об'єктно-орієнтованого програмування за допомогою мови Swift;
- інструменти середовища програмування XCode;
- комплект засобів розробки, який дозволяє здійснити спрощений обмін даними з цільової технологією API Blink.

Мобільний додаток було виконано відповідно до поставленого завдання.

У випускній роботі виконані наступні розділи: основна частина. У основній частині проведений аналіз предметної області, різних інформаційних джерел, наведена постановка задачі та проаналізовано аналогічні рішення. Розробка вимог до додатку, де було проаналізовано інтерфейсні рішення. Моделювання проєктованого завдання, де було наведено діаграми варіантів використання, діяльності та компонентів розгортання, а також описано інфологічну модель. Програмне забезпечення - в цьому розділі було обґрунтовано вибір мови програмування та системи управління базами даних та розділ способів взаємодії з користувачем за межах додатку де було наведено приклади роботи системи за межами iOS додатку.

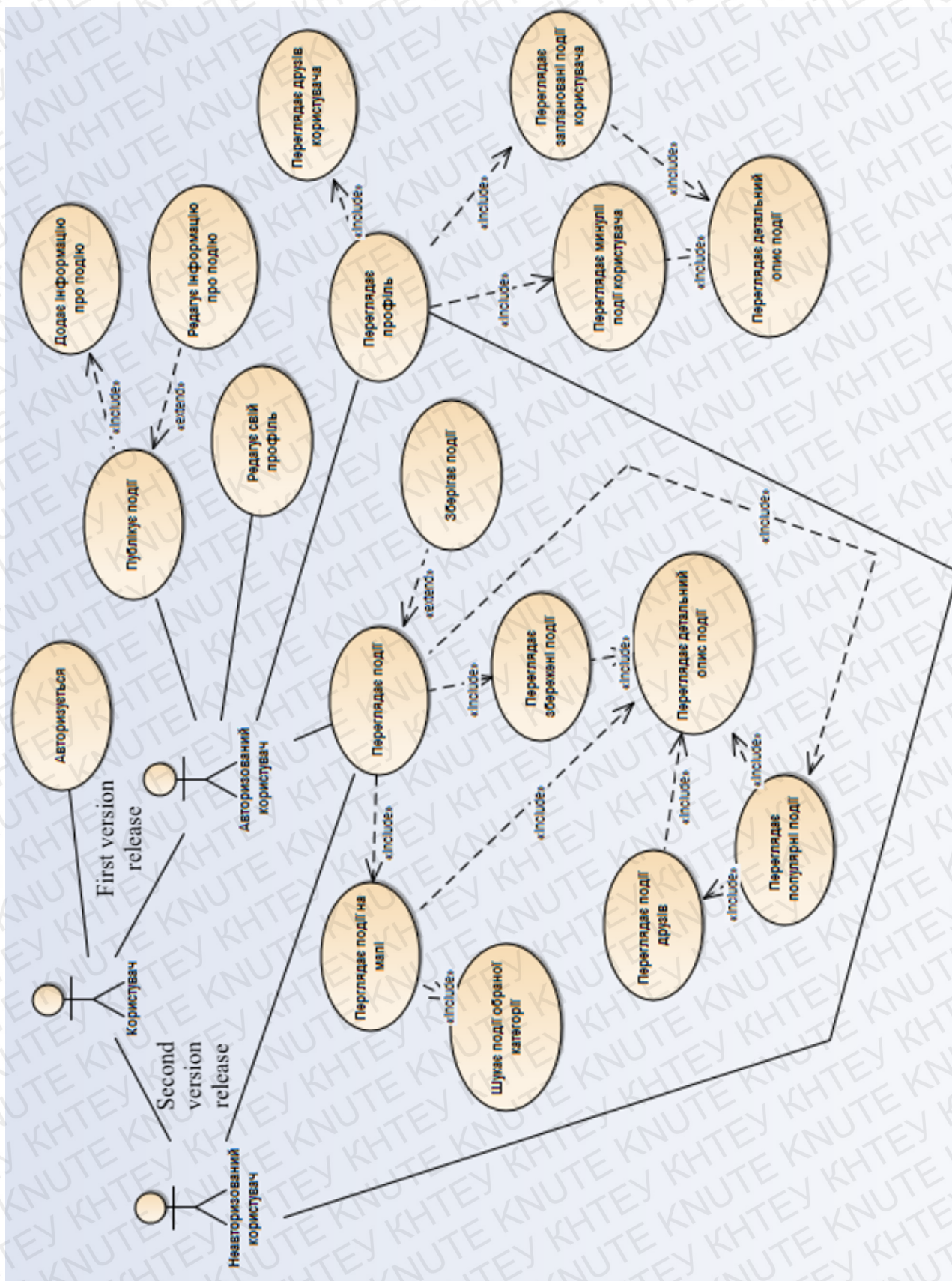
Отже, підбиваючи підсумок випускної роботи можна сказати, що поставлені цілі були досягнуті. Перевагою розробленого мобільного додатку є те, що він дозволяє в режимі реального часу шукати, переглядати та додавати події в будь-якій частині України.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

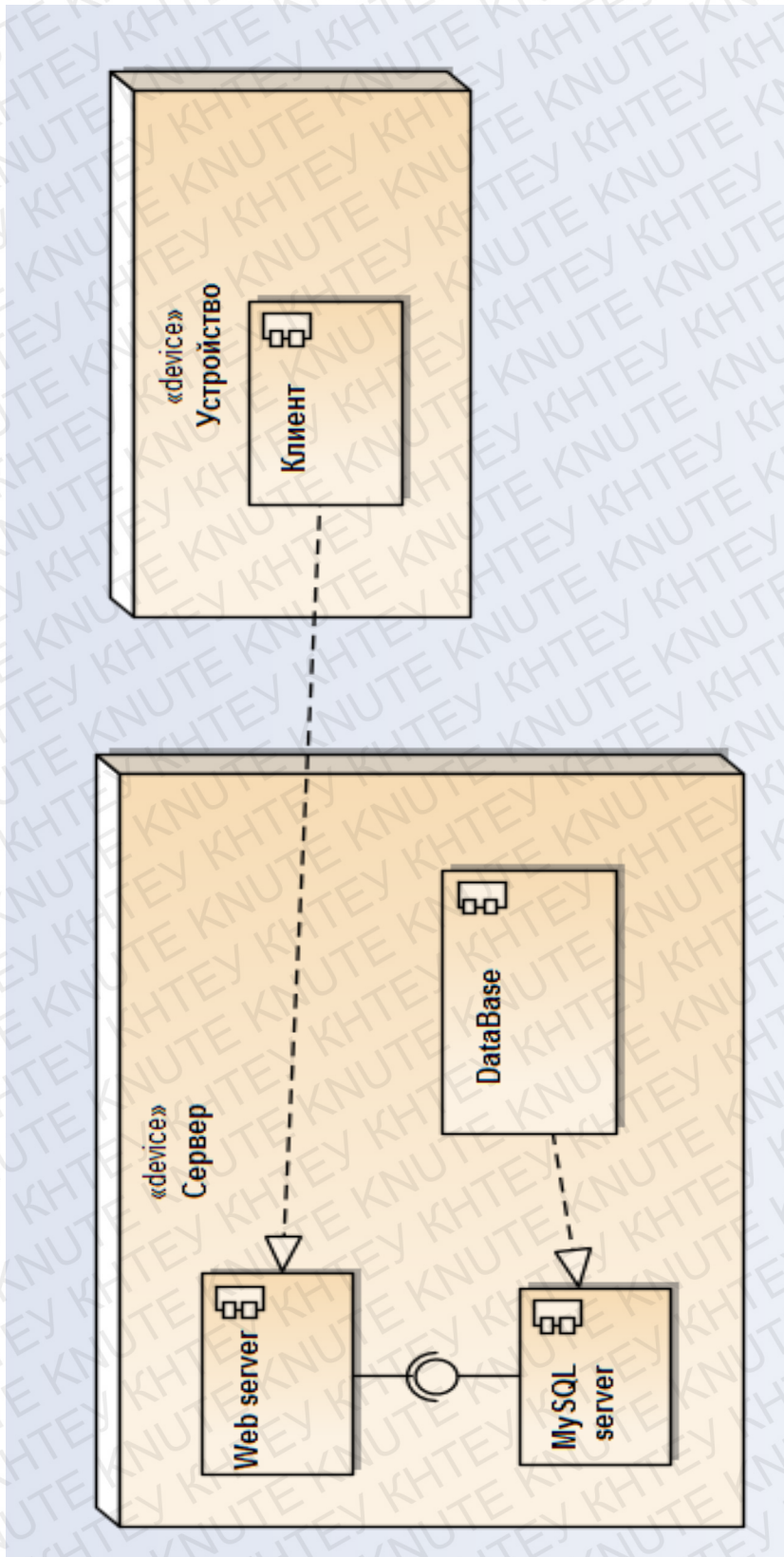
1. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: Держстандарт України, 1995. – 38 с.
2. Systems and software engineering — Life cycle processes — Requirements engineering [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: <https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:29148:ed-1:v1:en>
3. Material Design Guidelines [Електронний ресурс] – Режим доступу до ресурсу: <https://material.io/design/guidelines-overview>
4. Фаулер М., Скотт К., UML. Основи. Пер.с англ. – СП.: Символ-плюс – 2002. – 192 с., ил.
5. Building Mobile Apps at Scale: 39 Engineering Challenges. Gergely Orosz – 2018. – Режим доступу до ресурсу: <https://www.amazon.com/Building-Mobile-Apps-Scale-Engineering-ebook-dp-B091XSR1R7/dp/B091XSR1R7>
6. Apple Worldwide Developers Conference [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.apple.com/wwdc21/>
7. iOS Programming: The Big Nerd Ranch Guide – 2020. – Режим доступу до ресурсу: <https://www.amazon.com/iOS-Programming-Nerd-Ranch-Guide/dp/0135264022>
8. Канал розробки ios-додатків [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hackingwithswift.com>
9. Офіційний сайт Apple Developer [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.apple.com>
10. Think Data Structures. Allen B. Downey – 2017. – Режим доступу до ресурсу: <https://www.ebooks.com/en-ua/book/95800942/think-data-structures/allen-b-downey/>

## Додаток А

## Діаграма варіантів використання



Додаток Б  
Діаграма компонентів розгортання



## Додаток В

### Об'єкт «Event» у форматі JSON.

Код об'єкту «Event» у форматі JSON:

```
class Event: Codable, Equatable {
  static func == (lhs: Event, rhs: Event) -> Bool {
    return lhs.eventID == rhs.eventID &&
      lhs.interestsCount == rhs.interestsCount &&
      lhs.userInterested == rhs.userInterested &&
      lhs.interested == rhs.interested &&
      lhs.image == rhs.image &&
      lhs.title == rhs.title &&
      lhs.description == rhs.description &&
      lhs.timeZone == rhs.timeZone &&
      lhs.startTime == rhs.startTime &&
      lhs.endTime == rhs.endTime &&
      lhs.addTime == rhs.addTime &&
      lhs.position == rhs.position &&
      lhs.price == rhs.price &&
      lhs.user == rhs.user &&
      lhs.categories == rhs.categories &&
      lhs.hasSchedule == rhs.hasSchedule &&
      lhs.cancelled == rhs.cancelled &&
      lhs.online == rhs.online
  }
}
```

```
let eventID: String
```

```
var interestsCount: Int
```

```
var userInterested: Bool
```

```
var interested: [User]
```

```
var image: String
```

```
var title, description, timeZone: String
```

```
var startTime, endTime, addTime: Int
```

var position: Position?

var price: Price?

var user: User

var categories: [Category]

var hasSchedule: Bool?

var cancelled: Bool?

var online: Bool?

```
init(eventID: String, interestsCount: Int, userInterested: Bool, interested: [User], image: String, title: String, description: String, timeZone: String, startTime: Int, endTime: Int, addTime: Int, position: Position?, price: Price, user: User, categories: [Category], hasSchedule: Bool, cancelled: Bool, online: Bool?) {
```

```
    self.eventID = eventID
```

```
    self.interestsCount = interestsCount
```

```
    self.userInterested = userInterested
```

```
    self.interested = interested
```

```
    self.image = image
```

```
    self.title = title
```

```
    self.description = description
```

```
    self.timeZone = timeZone
```

```
    self.startTime = startTime
```

```
    self.endTime = endTime
```

```
    self.addTime = addTime
```

```
    self.position = position
```

```
    self.price = price
```

```
    self.user = user
```

```
    self.categories = categories
```

```
    self.hasSchedule = hasSchedule
```

```
    self.cancelled = cancelled
```

```
        self.online = onli
```





