

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем

Розробка мобільного додатку під Android за допомогою фреймворку React Native

**Текстова частина до курсової роботи
за спеціальністю «Прикладна математика»**

Керівник курсової роботи
ст. викладач Борозенний С. О.

(підпис)

“ ____ ” _____ 2022 р.

Виконав студент 3 р. н.

Ярошепта Б.П.

“ ____ ” _____ 2022 р.

Київ – 2022

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем

ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем

доцент, к.ф.-м.н.

О. П. Жежерун

_____2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Ярошепті Богдану Павловичу 3-го курсу факультету
інформатики

ТЕМА Розробка мобільного додатку під Android за допомогою
фреймворку React Native

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

1 Аналіз предметної області

2 Проектування архітектури додатку

3 Огляд фреймворку та знайомство з бібліотеками

4 Розробка додатку

Висновки

Список літератури

Дата видачі „____” _____ 2022 р. Керівник _____

Завдання отримав _____

Тема: Розробка мобільного додатку під Android за допомогою фреймворку React Native

Календарний план виконання роботи:

№ п/п	Назва етапу	Термін виконання	Примітка
1.	Отримання завдання на курсову роботу.	09.11.2021	
2.	Аналіз предметної області.	28.11.2021	
3.	Знайомство фреймворками React Native та Ехро	Грудень 2021	
3.	Проектування архітектури додатку	Лютий 2022	
4.	Розробка додатку	Березень 2022	
5.	Написання текстової частини роботи.	Квітень 2022	
6.	Здача роботи на перевірку на плагіат	07.06.2022	

Студент Ярошепта Б. П.

Керівник Борозенний О. С.

“ _____ ”

Зміст

Анотація	6
Вступ.....	7
1 Аналіз предметної області.....	10
1.1 Характеристика предметної області	10
1.2 Функціональні вимоги	11
2 Вибір фреймворку та його обґрунтування.....	13
2.1 Обґрунтування вибору фреймворку React Native.....	13
2.2 Переваги та недоліки React Native	14
3 Фреймворк Ехро	16
3.1 Опис фреймворку Ехро	16
3.2 Обґрунтування вибору	17
4 Структура додатку.....	18
4. 1 Графічний інтерфейс користувача	18
4. 2. Сервіси додатку.....	26
4. 3 Додаткові бібліотеки	29
Висновки	31
Список літератури	32

Анотація

Курсова робота присвячена створенню мобільного додатку під операційну систему Android за допомогою фреймворку React Native, перша версія якого була випущена 26 березня 2017 року компанією Facebook (нині - Meta). Для розробки була використана версія React Native 0.68.1 (17.03.2022), інтегроване середовище розробки Visual Studio Code, менеджер залежностей Node Package Manager та бібліотеки з кодом у відкритому доступі: React Native Local and Remote Notifications, React Native Modal Dropdown, MomentJS, DateJS, Expo та інші.

Основні функції системи: можливість додати нагадування про день народження людини, вік, посилання на соціальні мережі, номер телефону, e-mail та отримання push-сповіщень в попередньо обраний час з інформацією про людину, в якої день народження.

Ключові слова: JavaScript, React Native, VS Code, Android, Local and Remote Notifications, reminder, birthday, мобільний додаток, фреймворк

Вступ

В сучасному світі життя кожної людини стає все більш динамічним, все менше і менше часу ми готові витратити на читання якоїсь статті, перегляд відео і, навіть, написання повідомлень у месенджерах і соціальних мережах.

Чого тільки вартий додаток Tik-Tok, який був проривом у сфері соціальних мереж та мобільних додатків. Але ж він став таким популярним не просто через якісь «інноваційні рішення в архітектурі додатку» чи «революцію у сфері соціальних мереж», а все банально просто – швидкий потік короткого контенту. Відео максимум по 15 секунд, щоб перейти до іншого – просто свайпни вгору. Сподобався автор? Свайпни вліво – зможеш подивитись всі його відео. Все геніальне – просто!

Але у нашому «новому динамічному житті» є і серйозна проблема – у великому потоці неважливої інформації ми часто губимось, звертаємо увагу не туди куди треба і, що найголовніше, забуваєм важливу інформацію.

Метою курсової роботи є вирішення одного з аспектів цієї проблеми, а саме – створення та дослідження сучасного мобільного додатку під операційну систему Android з використанням фреймворку React Native для нагадувань про дні народження людей та швидка можливість їх привітати у будь-якій соціальній мережі або месенджері. Додаток повинен мати інтуїтивно зрозумілий інтерфейс, бути зручним у використанні та виконувати поставлені перед ним задачі. Основними завданнями даного дослідження є:

- аналіз предметної області та створення функціональних вимог;
- опис вибраних бібліотек та рішень;

- дослідження фреймворку React Native та фреймворку з кодом у вільному доступі Expo;
- створення мобільного застосунку для операційної системи Android.

Об'єктом дослідження є розробка мобільного додатку з використанням фреймворку React Native.

Предметом дослідження є сам мобільний застосунок. Додаток створений згідно функціональних вимог і відповідає критеріям сучасного застосунку, а також використовує функціонал запланованих push-сповіщень та правильні архітектурні рішення під час його проектування.

Робота складається з чотирьох розділів.

У першому розділі буде проведений аналіз предметної області, визначення та опис основних функціональних вимог, згідно яких буде реалізовано всі можливості додатку.

Другий розділ присвячений дослідженню фреймворку React Native, його переваг над нативними способами розробки мобільних додатків.

У третьому розділі обґрунтовується вибір фреймворку з кодом у вільному доступі Expo для розробки мобільного додатку та його переваги над вбудованим функціоналом фреймворку React Native.

В останньому розділі наводиться реалізація графічного інтерфейсу користувача. Окремо будуть описані бібліотеки з кодом у відкритому доступі, які використовувались при розробці застосунку.

У результаті виконання дослідження створено програмний продукт, який призначений для створення, налаштування та отримання push-сповіщень про дні народження знайомих з можливістю побачити вік та

привітати зі святом у будь-якій соціальній мережі, месенджері, за номером телефону або електронної пошти.

1 Аналіз предметної області

1.1 Характеристика предметної області

Обраною темою є створення сучасного мобільного додатку під операційну систему Android за допомогою крос-платформного фреймворку React Native. Додаток складається з двох частин: екранів авторизації та основного додатку. Авторизація та ідентифікація користувача проходить за допомогою 4-значного коду, який користувач отримує на свій номер телефону. Основна частина додатку складається з трьох екранів, на яких реалізований функціонал перегляду подій, відсортованих за часом, людей, день народження яких користувач додав у додаток, та редагування їх інформації з посиланнями на соціальні мережі. Також реалізовано налаштування та отримання push-сповіщень з інформацією про найближчі події та зберігання всієї інформації як в хмарній базі даних, так і локально на девайсу користувача, що дозволяє користуватись додатком без доступу до інтернету.

1.2 Функціональні вимоги

Беручи за основу предметну область і мету розробки проекту, зазначаю функціональні вимоги, поставлені переді мною перед розробкою функціоналу мобільного додатку:

- Можливість додавати, редагувати та видаляти людей з датами народження та посиланнями на соціальні мережі, месенджери та інше.
- Можливість авторизації за допомогою номеру мобільного телефону та його підтвердження, ввівши 4-значний код, отриманий користувачем на введений раніше номер телефону.

- Можливість привітати людину з днем народження просто тапнувши по іконці способу зв'язку.
- Можливі способи зв'язку:
 - Месенджер Telegram
 - Соціальна мережа Instagram
 - Соціальна мережа Facebook
 - Номер телефону
 - Електронна пошта
- Можливість переглянути всі події, відсортовані за датами з інформацією про майбутній вік людини прямо до цього ж дня наступного року.
- Можливість налаштувати час сповіщень, які будуть отримуватись перед днями народження людей, яких додав користувач.
- Можливість роботи з додатком без доступу до інтернету.
- Можливість зберігання всієї інформації між сесіями роботи застосунку.
- Можливість отримати доступ до всієї інформації та налаштувань на іншому девайсі, просто пройшовши авторизацію за номером телефону.
- Адаптація під різні розміри екрану.

2 Вибір фреймворку та його обґрунтування

2.1 Обґрунтування вибору фреймворку React Native

Коли починається планування розробки нового проекту, завжди велику кількість часу займає процес вибору фреймворку та інструментів, за допомогою яких будуть реалізовувати функціональні вимоги. Проводячи це дослідження, я був змушений проводити аналіз та проектування власноруч, не маючи жодного досвіду у цій сфері. Але, тим не менш, кінцевим результатом аналізу я задоволений.

Основною проблемою був вибір фреймворку для розробки мобільного додатку. Варіантів було два: нативні способи (такі, як реалізація за допомогою мов програмування Java та Kotlin) або крос-платформні фреймворки (такі як React Native або NativeScript). Як висновок, я вирішив використовувати фреймворк React Native, розроблений у 2017 році на основі фреймворку ReactJs для розробки веб-додатків компанією Meta (тоді - Facebook), версії 0.68.1, опублікованої 17.03.2022. Основною з причин вибору цього фреймворку є швидка розробка, мій досвід розробки за допомогою ReactJs та й, загалом, моя зацікавленість у розробці крос-платформних мобільних додатків.

2.2 Переваги та недоліки фреймворку React Native

Розробка фреймворку React Native спочатку розпочалася як хакатон у 2013 році. Ідея мобільної альтернативи, яка поєднує переваги розробки мобільних додатків з потужністю та маневреністю фреймворку для веб-розробки ReactJs, який був дуже популярним у спільноті розробників, була достатньо багатообіцяючою. Результатом цього став React Native,

фреймворк, створений Meta (тоді – Facebook), який активно розширюється та постійно набирає популярності у спільноті розробників.

Як наслідок використання середовища, керованого спільнотою, React Native має значні переваги. Для прикладу, наявність численної спільноти ентузіастів JavaScript та нативних розробників, які бажають ділитись своїм досвідом у розробці та підтримці своїм досвідом та знаннями.

Також у фреймворку React Native реалізована така можливість, як “Live-Reload”. Вона являє собою відсутність довгих компіляцій після будь-яких змін у коді. Тобто розробник, тільки зробивши якісь зміни, може зразу побачити результат, що суттєво пришвидшує час розробки застосунку.

Архітектура фреймворку React Native дуже добре налаштована для мобільних девайсів. По цій причині, додатки, створені за допомогою React Native є менш ресурсо-затратними, ніж нативні додатки. І порівнюючи зі всіма фреймворками, опублікованими до React Native, він – найшвидший.

Але React Native має і низку недоліків, які часто стають ключовими, у виборі фреймворку для розробки мобільних додатків. Часто від розробників можна почути, що у React Native дуже важко проводити дебаг, адже весь код React Native компілюється у нативний, а саме мови програмування Java, C/C++, Swift та інші. Тому для більшості мобільних розробників необхідно мати хоча б базові знання з цих мов.

Другою основною проблемою є наслідок крос-платформності. А саме точно невизначений вигляд користувацького інтерфейсу, адже на всіх типах операційних систем додаток може виглядати по різному і іноді це є великою проблемою, яка забирає багато часу на виправлення.

3. Фреймворк Ехро

3.1 Опис фреймворку Ехро

Ехро – це фреймворк та платформа для нативних React додатків. Його можна порівняти з Laravel чи Symfony для PHP-розробників, або Ruby on Rails для Ruby розробників. Ехро виступає надбудовою над вбудованим React Native API, щоб ними було зручніше користуватися та керувати. Він також надає інструменти, які спрощують та пришвидшують ініціалізацію, розробку та тестування React Native додатків. І нарешті він надає готові компоненти інтерфейсу користувача та сервіси, які доступні тільки при додатково встановлених сторонніх нативних компонентів ReactJs. Для прикладу, ехро-notifications, ехро-location, ехро-image-picker та інші. Всі вони та ще понад 90 додаткових бібліотек та компонентів доступні за допомогою Ехро SDK.

Сам фреймворк Ехро було засновано в жовтні 2013 року для покращення мобільного програмного забезпечення та його розробки. Перша версія фреймворку Ехро була опублікована для IOS у травні 2015 року незадовго після першої публікації фреймворку React Native на початку того ж року. Версія фреймворку Ехро для розробки під пристрої на операційній системі Android з'явилась у грудні 2015 року, аналогічно, після пізнішого випуску React Native для Android.

Веб-підтримка Ехро була вперше опублікована в березні 2019 року. Що дозволило розробникам створювати швидкі та сучасні веб-додатки зі своїх проектів, створених за допомогою Ехро, додатково до програм під операційні системи Android та iOS.

3.2 Обґрунтування вибору

Ініціалізація та подальша розробка мобільного додатку за допомогою фреймворку React Native може відбуватись двома способами: використання вбудованих можливостей фреймворку та за допомогою Ехро, який виступає фреймворком-надбудовою над React Native і має в собі великий набір готових рішень та інструментів, які суттєво полегшують розробку.

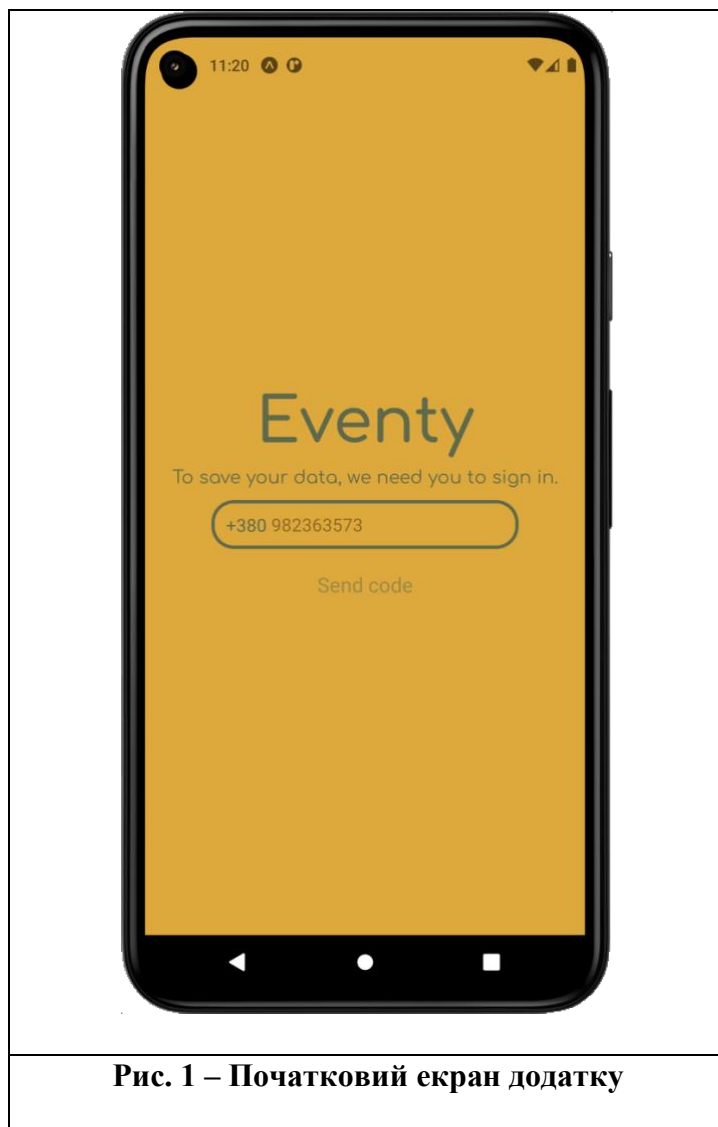
Фреймворк Ехро також вирішує один з головних недоліків React Native – для розробки, використовуючи Ехро, не потрібно ніяких знань нативної мобільної розробки. Також Ехро надає доступ до хмарних серверів, які автоматично та правильно будують як бандл, для завантаження у Google Play (магазин додатків під операційну систему Android) та App Store (магазин додатків під операційну систему IOS), так і готові .apk файли, за допомогою яких можна встановити додаток на будь-який Android пристрій. Додатково, фреймворк Ехро є безкоштовним і, що найголовніше, розробляється суспільством розробників, з чого випливають постійні оновлення і введення саме тих можливостей, які потрібні для звичайних розробників.

Але і під час розробки застосунку Ехро суттєво пришвидшує процес роботи, тестування та дебагу. При білді додатку фреймворк Ехро створює та запускає локальний сервер, на якому компілюється додаток, динамічно оновлюється після будь-якої зміни в коді та дозволяє подивитись результат як на Android пристрої або емуляторі, так і на IOS та веб-браузерах.

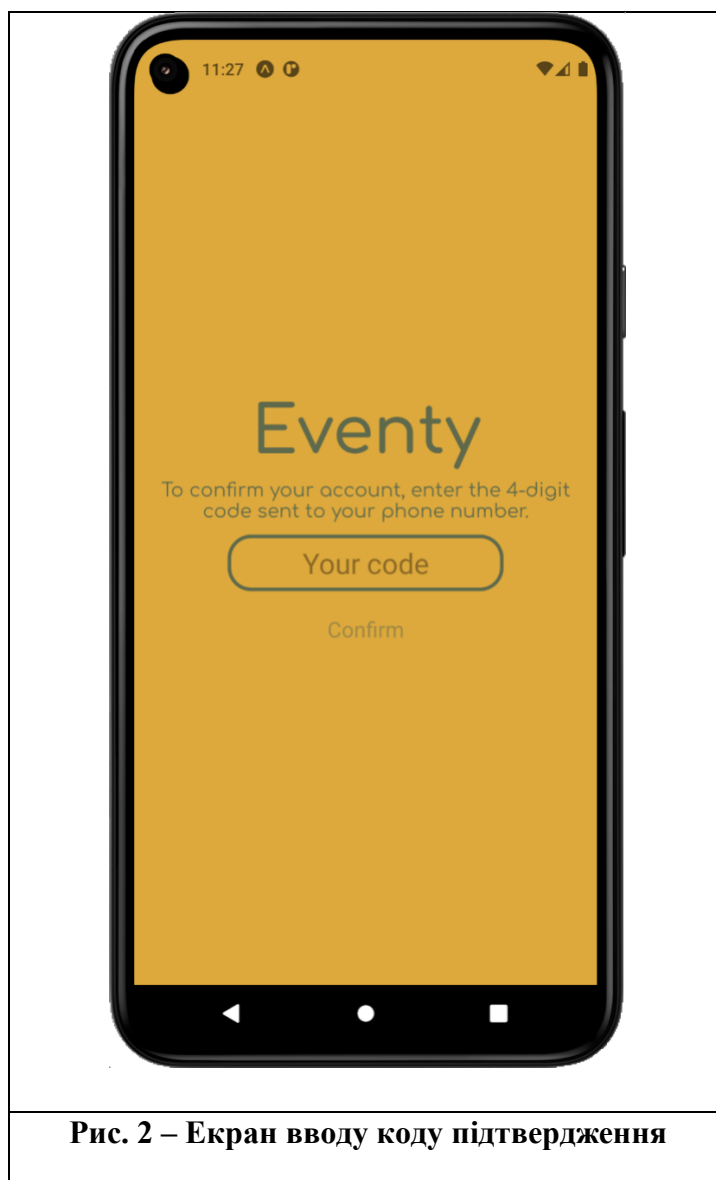
4. Структура додатку

4.1 Графічний інтерфейс користувача

При першому запуску додатку користувач бачить початковий екран авторизації, на якому йому пропонується ввести номер телефону, для авторизації (рис. 1)



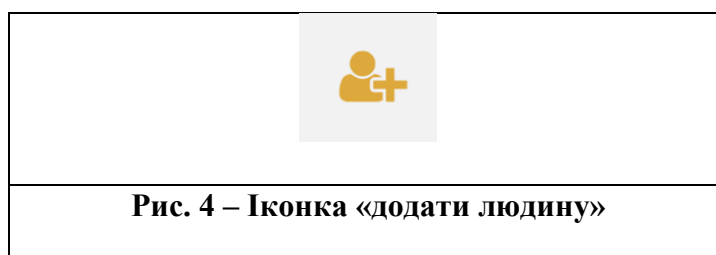
Ввівши номер телефону та натиснувши кнопку “Send Code”, користувач переходить на екран вводу 4-значного коду підтвердження, який було відправлено на його номер телефону, введений на попередньому екрані. (рис. 2)



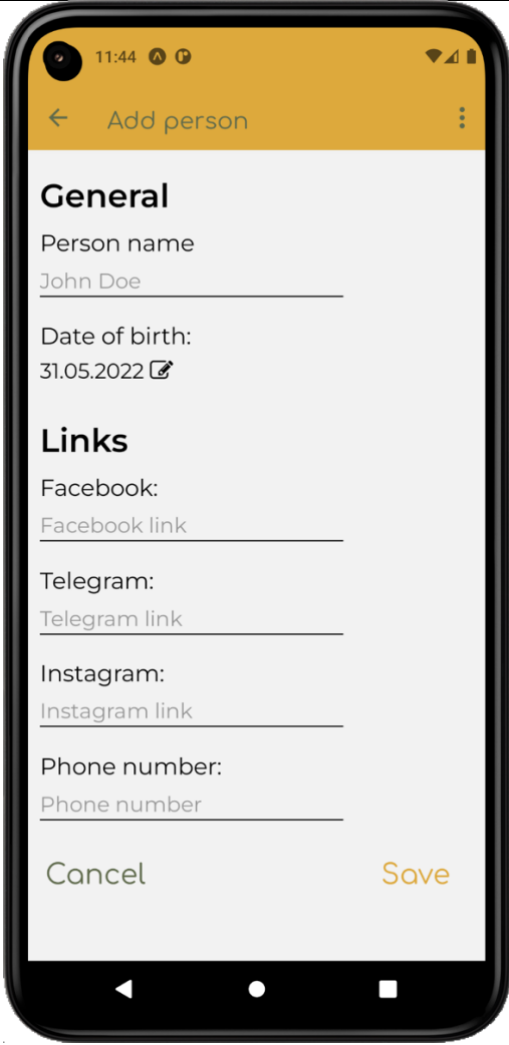
Коли користувач підтверджує номер телефону, ввівши код на попередньому екрані, він переходить на головний екран додатку – “Upcoming Events”. На цьому екрані відображаються всі дні народження, відсортовані за датою аж до цього ж дня наступного року. (рис. 3)



Користувач може додати нову людину з її днем народження, тапнувши по жовтій іконці в правому нижньому кутку екрану (рис. 4)



Після тапу користувача переводить на екран «Add Person», на якому вже можна заповнити всю інформацію про людину, яку користувач бажає додати до застосунку. Тут він має можливість ввести ім'я, дату народження та додати посилання на соціальні мережі, месенджери, номер телефону та інше. Обов'язковими вважаються поля імені та дати народження, всі інші – необов'язкові. (рис. 4)



General

Person name
John Doe

Date of birth:
31.05.2022

Links

Facebook:
Facebook link

Telegram:
Telegram link

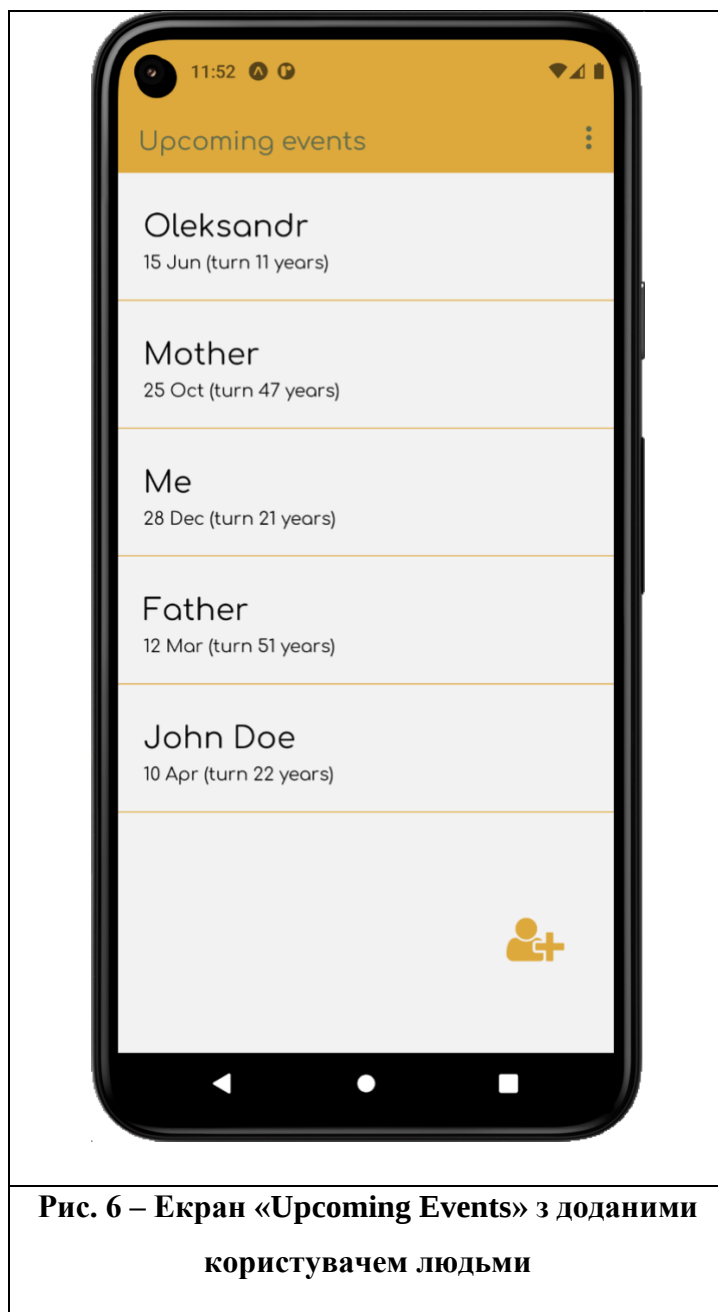
Instagram:
Instagram link

Phone number:
Phone number

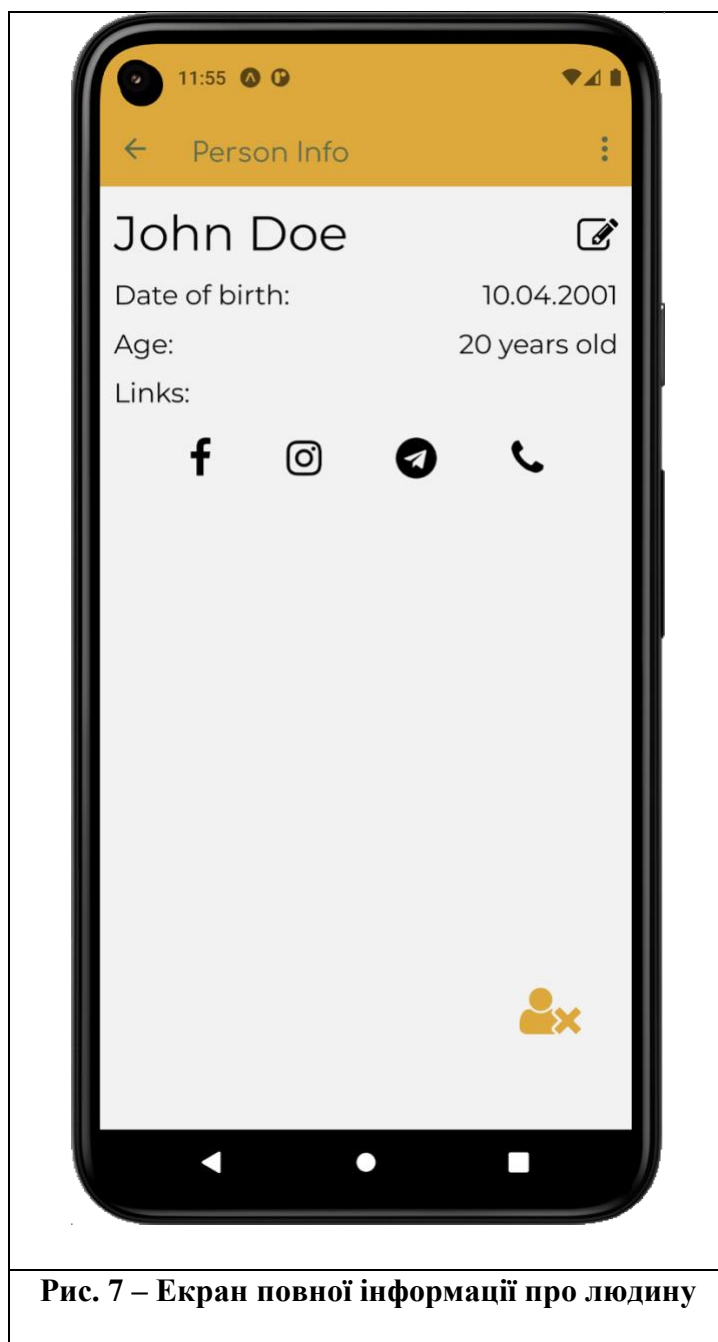
Cancel Save

Рис. 5 – Екран «Add Person» з формою додавання людини

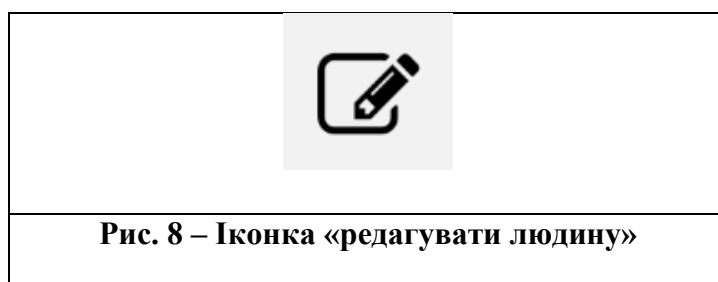
Після того, як користувач додасть кілька людей, головний екран додатку виглядатиме наступним чином: список з усіх доданих людей, з їхніми днями народження та майбутнім віком, який буде в цю дату. (рис. 6)



Якщо користувач тапне по будь-якому елементу списку, він перейде на екран повної інформації про людину, яку він додав. Там він зможе побачити ім'я, дату народження, теперішній вік та кнопки зі всіма посиланнями, які додав користувач цій людині. (рис. 7)



Щоб редагувати людину, користувачеві потрібно тапнути по іконці редагування, яка розташована навпроти імені людини, яку він додав (рис.8)



Після цього – користувач потрапляє на екран редагування людини, який виглядає аналогічно, до екрану додавання, але з заповненою інформацією. (рис. 9)

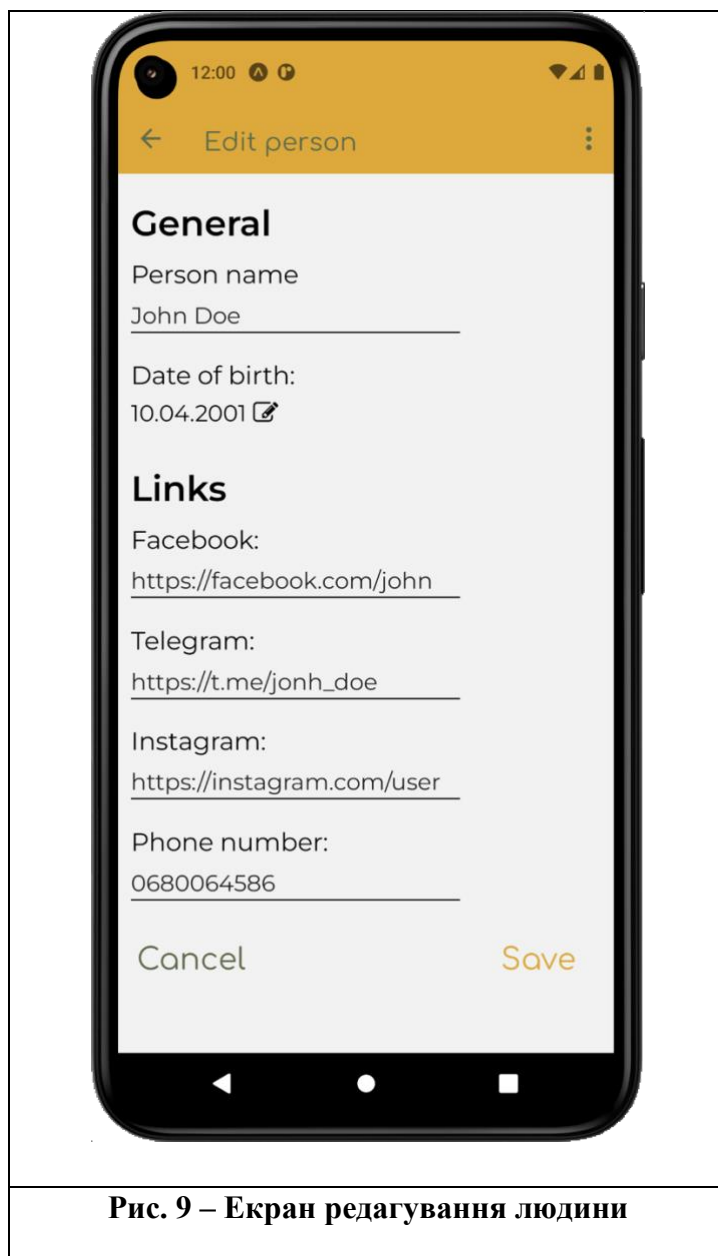
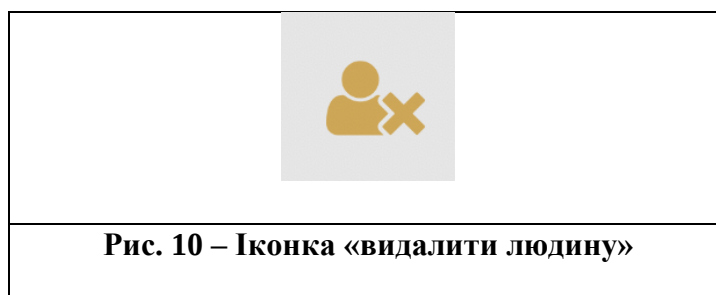


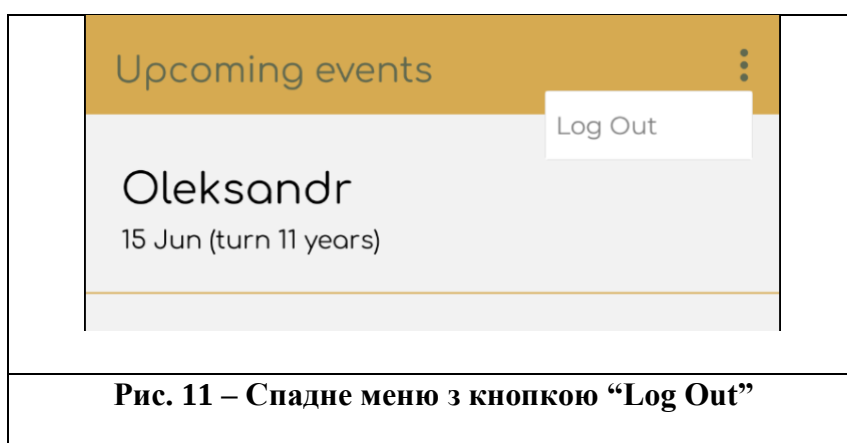
Рис. 9 – Екран редагування людини

Аналогічно, щоб видалити людину, користувачеві достатньо тапнути по іконці видалення, яка розташована на екрані повної інформації про людину у нижньому правому кутку. (рис. 10)



Після видалення користувача переносить на головну сторінку “Upcoming Events”, де людина, яку видалили, вже відсутня.

Щоб вийти з облікового запису, користувачу потрібно тапнути по трьом крапкам у правій частині хедеру додатку та у спадному меню вибрати пункт “Log Out”. (рис. 11)



Після цього, користувач потрапить на початкову сторінку додатку. (рис. 1)

4.2 Сервіси додатку

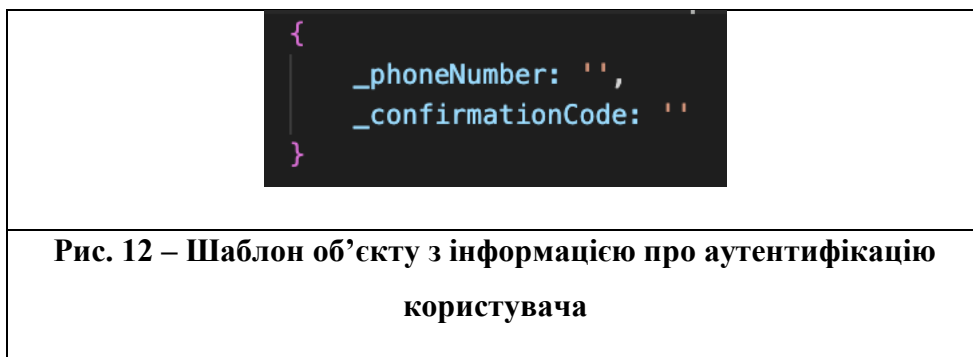
Всього в додатку використано 4 сервіси, кожен з яких, виконує низку функцій. Всі вони написані по шаблону проектування, максимально наближеному до “Singelton”, тобто на весь додаток існує лише один об’єкт класу.

- AsyncStorageService

Сервіс надає функціонал роботи з локальним сховищем девайсу, щоб зберігати інформацію між сесіями роботи застосунку.

Має три методи: `storeData`, `getAuthData` та `getData`.

- `storeData` – асинхронний метод, який приймає на вхід два аргументи: ключ сховища типу стрічки, та значення у вигляді об'єкту, яке буде записано у сховище за цим ключем.
- `getAuthData` – асинхронний метод, який приймає на вхід два аргументи: ключ сховища типу стрічки та функцію колбеку, яка буде викликана асинхронно після того, як сервіс отримає дані з локального сховища. Функція колбеку буде викликана з аргументом, який буде дорівнювати даним, які отримали зі сховища або шаблоном об'єкту з інформацією про аутентифікацію користувача. (рис. 12)



- `getData` – асинхронний метод, який приймає на вхід два аргументи: ключ сховища типу стрічки та функцію колбеку, яка буде викликана асинхронно після того, як сервіс отримає дані з локального сховища. Функція колбеку буде викликана з аргументом, який буде дорівнювати даним, які отримали зі сховища, а саме – масив всіх доданих користувачем людей з днями народження, у випадку, коли у сховищі немає даних, буде повертатись пустий масив.

- AuthService

Сервіс надає функціонал та зберігає всю інформацію про авторизацію користувача. Має в собі п'ять методів: `setOnAuth`, `logOutUser`, `authorizeCode`, `authorizeNumber`, `isAuthorized`.

- `setOnAuth` – метод, який приймає один аргумент, функцію, яка буде викликатись при будь-яких змінах інформації про авторизацію.
- `logOutUser` – метод, що реалізує функціонал виходу з облікового запису.
- `authorizeNumber` – метод, що приймає як аргумент значення номеру телефону, який користувач ввів на початковому екрані авторизації, та записує у локальне сховище за допомогою сервісу `AsyncStorageService`.
- `authorizeCode` – метод, що приймає як аргумент значення коду підтвердження, який користувач отримав на номер телефону і записує у локальне сховище за допомогою сервісу `AsyncStorageService`.
- `isAuthorized` – метод, що повертає булеве значення, в залежності від того, авторизований користувач чи ні.

- PeopleService

Сервіс надає функціонал роботи з списком людей, яких додав користувач. Має шість методів: `getPeople`, `editPerson`, `addPerson`, `removePerson`, `isUniqueName`, `sortPeople`.

- `getPeople` – метод, що приймає аргументом функцію колбеку, отримує масив всіх людей, яких додав

користувач, та викликає колбек, аргументом якого виступає отриманий раніше масив.

- `editPerson` – метод, що приймає аргументами стару інформацію про людину, нову та колбек, перевіряє чи нове ім'я людини є унікальним за допомогою методу `isUniqueName`, записує у локальне сховище та викликає колбек з аргументом “null”, якщо нове ім'я не унікальне, та, в успішному сценарії редагування, викликає колбек, аргументом якого виступає новий масив людей.
- `addPerson` – метод, з реалізацією, аналогічною до `editPerson`, але отримує лише два аргументи: нову людину та колбек.
- `removePerson` – метод, що приймає аргументами два значення: ім'я людини, що видалити та функцію колбеку, яка в результаті викликається з аргументом, який виступає новим масивом людей.
- `isUniqueName` – метод, що приймає аргументом ім'я та перевіряє, чи у масиву немає людини з таким же ім'ям, та, відповідно, повертає булеве значення.
- `sortPeople` – метод, що сортує всіх людей у масиві, в залежності від дати їх найближчого святкування дня народження

- `UtilsService`

Сервіс, що надає допоміжний функціонал, який використовується у кількох місцях застосунку. Має два методи: `getFutureAge` та `getAge`.

- `getFutureAge` – метод, що отримує на вхід аргументом дату, та повертає число, що дорівнює кількості років, яка має бути у людини на найближчому дні народження.
- `getAge` – метод, що отримує на вхід аргументом дату, та повертає число, яке дорівнює кількості поточних років людини, яка народилась у передану аргументом дату.

4.3 Додаткові бібліотеки

Під час розробки додатку для полегшення роботи були використані наступні сторонні бібліотеки, встановлені за допомогою Node Package Manager, який дозволяє встановлювати та керувати всіма бібліотеками та, загалом, додатком:

- Expo Notifications

Бібліотека, вбудована в Expo SDK, яка дозволяє керувати та правильно і легко використовувати нативне Notifications API операційної системи.

- Expo Vector Icons

Бібліотека, вбудована в Expo SDK, яка має в собі великий набір різних векторних спрайтів, які можна легко додавати та використовувати у додатку.

- Expo Async Storage

Бібліотека, вбудована в Expo SDK, яка дозволяє використовувати локальне сховище, для збереження даних між сесіями роботи застосунку.

- Expo Date Time Picker

Компонент, вбудований в Expo SDK, який являє собою модальне вікно у вигляді календаря, для вибору дати.

- React Navigation Stack

Бібліотека, яка дозволяє використовувати у додатку такий тип навігації, як «стек».

- React Native Modal Dropdown

Компонент, який дозволяє використовувати функціонал спадного меню.

- MomentJs та DateJS

Дві бібліотеки, розроблені для мови програмування JavaScript, які суттєво розширюють вбудований у мову функціонал для роботи з датами.

Висновки

Результатом виконання цієї роботи став повністю самостійний, сучасний та актуальний мобільний додаток, який в собі поєднує гарний, зручний та інтуїтивно зрозумілий користувацький інтерфейс, весь необхідний функціонал та виконує всі поставлені перед ним завдання в повному обсязі.

Застосунок буде корисним всім людям, які у швидкому темпі життя часто забувають про такі життєві речі, як день народження знайомих та друзів. Він буде вчасно нагадувати та допомагати витратити мінімально часу на звичайне текстове привітання у будь-якій соціальній мережі, месенджері або, навіть, подзвонивши.

В ході розробки було досліджено фреймворки React Native, Expo, операційна система Android та її особливості, правила побудови правильного та адаптивного користувацького інтерфейсу для мобільних пристроїв.

Варто зауважити, що всі функціональні вимоги, що ставилися переді мною, були виконані в повному обсязі. Весь продукт був виконаний за допомогою мови програмування JavaScript, фреймворків React Native та Expo, інтегрованого середовища розробки Visual Studio Code та вбудованого емулятора Android пристрою у інтегроване середовище розробки Android Studio.

Перспективами розвитку мобільного додатку є розширення функціоналу, додавання готових привітань та створення окремого віджету, який можна додати на головний екран мобільного пристрою.

Список літератури

1. George T. Heineman, William T. Councill (2001). Component-Based Software Engineering: Putting the Pieces Together. Addison-Wesley Professional, Reading 2001. – 818 с.
2. React Native Docs [Електронний ресурс] – режим доступу до ресурсу: <https://reactnative.dev/>.
3. Expo Documentation [Електронний ресурс] – режим доступу до ресурсу: <https://docs.expo.dev/>.
4. Expo Notifications [Електронний ресурс] – режим доступу до ресурсу: <https://docs.expo.dev/versions/v45.0.0/sdk/notifications/>.
5. Expo Vector Icons [Електронний ресурс] – режим доступу до ресурсу: <https://docs.expo.dev/guides/icons/>.
6. Expo Async Storage [Електронний ресурс] – режим доступу до ресурсу: <https://docs.expo.dev/versions/v45.0.0/sdk/async-storage/>.
7. Expo Date Time Picker [Електронний ресурс] – режим доступу до ресурсу: <https://docs.expo.dev/versions/v45.0.0/sdk/date-time-picker/>.
8. React Navigation Stack [Електронний ресурс] – режим доступу до ресурсу: <https://reactnavigation.org/docs/stack-navigator/>.
9. React Native Modal Dropdown [Електронний ресурс] – режим доступу до ресурсу: <https://github.com/siemiatj/react-native-modal-dropdown>.
10. MomentJs [Електронний ресурс] – режим доступу до ресурсу: <https://momentjs.com/docs/>.
11. DateJs [Електронний ресурс] – режим доступу до ресурсу: <https://github.com/abritinthebay/datejs>.
12. React Documentation [Електронний ресурс] – режим доступу до ресурсу: <https://uk.reactjs.org/>.
13. JavaScript | MDN [Електронний ресурс] – режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.