

**Министерство науки и высшего образования Российской
Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»
(Университет ИТМО)**

Факультет Прикладной информатики

**Направление подготовки 09.03.03 Прикладная информатика
Образовательная программа Мобильные и сетевые технологии**

КУРСОВОЙ ПРОЕКТ

Тема: «Мобильное приложение для организации мероприятий»

Обучающийся: Андреюк Николай Ростиславович, К3141

Санкт-Петербург 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ОСНОВНАЯ ЧАСТЬ.....	5
1 Суть проекта	5
2 Разработка.....	5
3 Задача поставленная передо мной.....	6
4 Анализ собственной работы	8
5 Взаимодействие с командой	8
6 Оценка работы руководителя проекта	9
ЗАКЛЮЧЕНИЕ	10
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	11
ПРИЛОЖЕНИЕ 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ	13

ВВЕДЕНИЕ

В современном мире, где технологии стремительно развиваются, а общественные связи становятся все более важными, организация мероприятий и участие в них приобретают особое значение. С каждым годом количество различных событий — от культурных и образовательных до спортивных и деловых — растет, что создает потребность в удобных и эффективных инструментах для их организации и управления.

В этой связи разработка мобильного приложения для платформ Android и iOS (имея в виду распространенность обеих операционных систем), которое позволит пользователям создавать, редактировать и публиковать мероприятия, а также подавать заявки на участие в них, становится актуальной задачей.

Одной из ключевых особенностей такого приложения должна стать его многофункциональность. Пользователи должны иметь возможность не только организовывать собственные мероприятия, но и находить интересные события в своем регионе или за его пределами.

Это создаст платформу для обмена информацией и идеями, а также позволит людям легче находить единомышленников и участвовать в мероприятиях, которые соответствуют их интересам.

Важно отметить, что приложению следует быть ориентированным как на организаторов, так и на участников, что сделает его универсальным инструментом для всех заинтересованных сторон.

Разработка такого программного продукта - комплексный процесс, требующий вдумчивого поэтапного проектирования, разумного выбора технологий, инструментов разработки для того, чтобы обеспечить легкую расширяемость и возможность интеграции сторонних сервисов (например, сервисов карт - Яндекс. Карты, Google Карты, 2ГИС и прочих или сервисов прогноза погоды), и самое главное - времени.

Последнее, как известно, всегда в дефиците, поэтому реализовать по-настоящему большую систему в сжатые сроки - задача крайне амбициозная и обреченная на провал.

Понимая это обстоятельство, мы решили разработать приложение, содержащее лишь часть возможного функционала.

ОСНОВНАЯ ЧАСТЬ

1 Суть проекта

Данный проект решает задачу как организации мероприятий, так и систематизации данных о планируемых мероприятиях. Приложение предоставляет функционал как для обычных пользователей – участников, волонтеров и персонала мероприятия, так и для организаторов.

Из технических моментов на момент сдачи курсовой работы в проекте реализованы следующие элементы:

- Экраны входа и регистрации,
- Экран списка событий,
- Экран с детализированной информацией о событии,
- Экран создания события,
- Экран профиля пользователя,
- Экран редактирования профиля.

К сожалению, нам не хватило времени для завершения этого проекта и в наших планах остались задачи, которые мы хотели бы выполнить и сохранить проект в своем портфолио. Среди них – реализация взаимодействия с серверной частью (хранение аутентификационных данных, данных о мероприятиях, данных пользователей).

2 Разработка

После формирования команды, мы обсудили этапы разработки проекта и спланировали свою работу.

Таблица 1 – Этапы разработки приложения

Этап	Сроки выполнения
Анализ требований	1.11.2024 – 10.11.2024
Изучение литературы	1.11.2024 – 24.11.2024

Проектирование UI	25.11.2024 – 8.12.2024
Разработка приложения	25.11.2024 – 16.12.2024
Тестирование	16.12.2024 – 18.12.2024

3 Задача поставленная передо мной

Руководитель проекта передо мной поставил задачу: разработать экран профиля пользователя по подобию профиля из другого приложения. Также должна быть реализована возможность редактировать данные собственного профиля.

Чтобы успешно выполнить поставленную задачу, мне потребовалась дополнительная литература. К сожалению, наш руководитель, специализирующийся на разработке для iOS, не мог предоставить мне всей необходимой информации. Однако в нашей команде нашелся человек, имеющий опыт в разработке для Android. Он помог мне найти нужные источники, в которых было описано, как разрабатываются подобные проекты и как я могу помочь команде в разработке.

В итоге, мы решили использовать следующий набор технологий:

- Jetpack Compose – для верстки UI
- Decompose – для навигации между экранами приложения
- Koin – для реализации инъекции зависимостей
- Kotlin serialization – для преобразования данных в программные объекты и наоборот
- Coil – для загрузки и отображения изображений
- Google Firebase – для взаимодействия с облачными сервисами Firebase от Google

После изучения литературы, я приступил к выполнению своей задачи. К этому моменту уже был разработан дизайн-макет приложения и описаны основные классы и структуры данных в нашем проекте, и я беспрепятственно приступил к работе. Я описал классы, в которых задается логика взаимодействия между экранами смежными с экраном просмотра профиля.

Также, я написал класс, в котором описываются элементы UI.

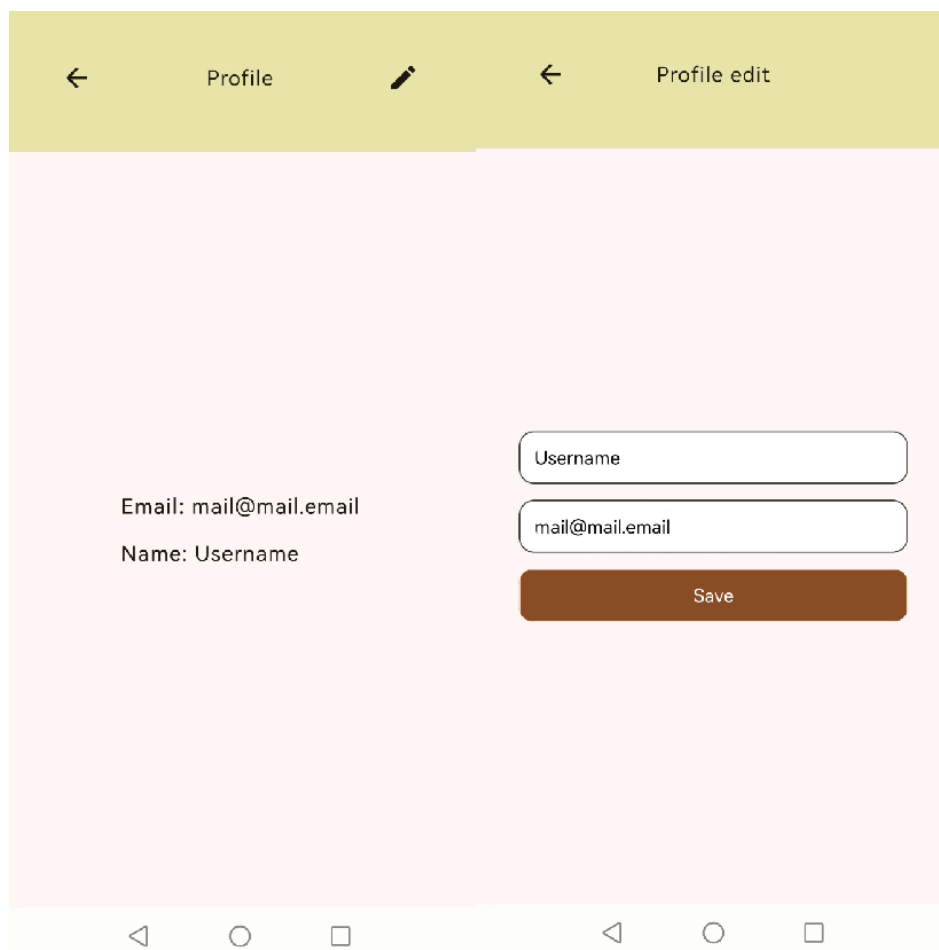


Рисунок 1 – Демонстрационный вариант оформления профиля

Также, была реализована проверка валидности вводимого адреса почты при изменении ее через редактирование профиля.

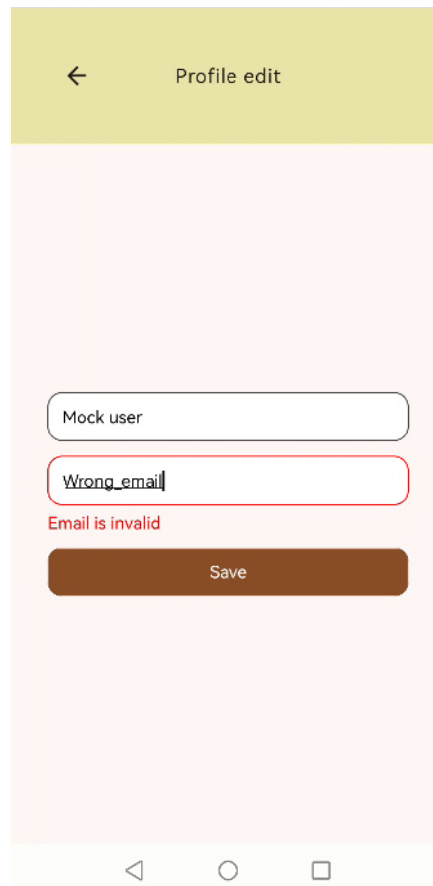


Рисунок 2 – Ошибка ввода электронной почты

4 Анализ собственной работы

Основные трудности при работе над проектом вызвал недостаток времени для комплексного изучения всех технологий и опыта их реализации.

Мы относительно равномерно двигались по поставленному плану, но из-за того, что мы его составили недостаточно подробно, мы не успели разработать всю функциональность.

За время выполнения курсового проекта я научился быстро изучать новые технологии. Кроме того, я обнаружил множество источников информации, которые могут быть полезны для повышения уровня моих навыков в области разработки.

5 Взаимодействие с командой

Взаимодействие с командой происходило посредством мессенджера, а также мы периодически устраивали небольшие созвоны, чтобы обсудить некоторые важные рабочие моменты.

Так как мы разделились на две подкоманды - iOS и Android разработки, я взаимодействовал больше с Android разработчиками, в частности с Ярославом, так как он знал больше и задавал вектор в котором нам нужно было действовать.

Мы были на связи с нашим руководителем и периодически сообщали ему о статусе выполнения наших задач.

6 Оценка работы руководителя проекта

Руководитель нашего проекта больше специализируется на iOS разработке, поэтому оценить его педагогические способности (его помощь в изучении Android) не представляется возможным. Однако, он хорошо понимал, что из себя представляет разработка приложений под эту ОС и задал хорошую задачу для такого начинающего специалиста, как я.

В общем, я оцениваю работу нашего руководителя достаточно хорошо, так как он своевременно отвечал на интересующие нас организационные вопросы и посильно помогал в работе.

ЗАКЛЮЧЕНИЕ

Проект мобильного приложения для организации мероприятий был не завершен, однако большая часть поставленных перед нами задач была выполнена успешно. К сожалению, мы не успели реализовать функционал взаимодействия с базой данных Firebase, так как нам попросту не хватило времени.

Работая над проектом каждый из участников получил хороший набор знаний и навыков для будущего их развития. Лично я познакомился с языком Kotlin, а также с большим количеством неизвестных мне ранее библиотек и технологий для работы над приложением, научился создавать простой дизайн элементов приложения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лекция по многопоточности в Java -
<https://www.youtube.com/watch?v=ShzQJUFzq58>
2. Продолжение лекции по многопоточности в Java -
<https://www.youtube.com/watch?v=5YLA29EybMo&list=PLlb7e2G7aSpTCB2OxGlezpgOXwq4xer7Z&index=14>
3. Лекция по корутинам в языке программирования Kotlin -
https://www.youtube.com/watch?v=ITLe4FIrrTg&list=PLlh9yLdjK2YeRLnD-gJyVWIq_w-7OMv8r&index=7
4. Лекции от Яндекса по Android-разработке -
https://www.youtube.com/watch?v=Yz2K1p8OJ1o&list=PLlh9yLdjK2YeRLnD-gJyVWIq_w-7OMv8r
5. Курс по корутинам в языке программирования Kotlin от Кирилла Розова -
<https://www.youtube.com/watch?v=mD1r9zIwHbs&list=PL0SwNXXKJbuNmsKW9mtTSxNn00oJIYOLA>
6. Курс по Android-разработке от Андрея Сумина -
<https://stepik.org/course/121507/promo>
7. Курс по профессиональной Android-разработке от Андрея Сумина -
<https://stepik.org/course/117314/promo>
8. Курс по Jetpack Compose от Андрея Сумина -
<https://stepik.org/course/125685/promo>
9. Курс от Google по Jetpack Compose и Android -
<https://developer.android.com/courses/android-basics-compose/course>

10. Документация языка программирования Kotlin -
<https://kotlinlang.org/docs/home.html>

10. Документация языка программирования Kotlin на русском языке -
<https://kotlinlang.ru/docs/kotlin-doc.html>

11. Статья про Clean Architecture -
<https://habr.com/ru/companies/mobileup/articles/335382/>

12. Статья про MV* паттерны -
<https://habr.com/ru/companies/mobileup/articles/313538/>

13. Документация библиотеки для инъекции зависимостей Koin -
<https://insert-koin.io/>

14. Документация библиотеки Decompose -
<https://arkivanov.github.io/Decompose/>

15. Гитхаб библиотеки Decompose (установка и версии) -
<https://github.com/arkivanov/Decompose>

16. Гитхаб Kotlin serialization (установка и версии) -
<https://github.com/Kotlin/kotlinx.serialization>

17. Документаця Kotlin serialization -
<https://github.com/Kotlin/kotlinx.serialization/blob/master/docs/serialization-guide.md>

18. Документация мультиплатформенной библиотеки для загрузки изображений Coil -
<https://coil-kt.github.io/coil/>

19. Гитхаб мультиплатформенной библиотеки для загрузки изображений Coil - <https://github.com/coil-kt/coil>

ПРИЛОЖЕНИЕ 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1. Наименование проекта: Мобильное приложение для организации мероприятий

2. Цель: Создать прототип мобильного iOS/Android приложения для организации мероприятий и участия в них.

3. Сроки выполнения: 1.11.2024 – 18.12.2024

4. Руководитель проекта: Бацанов Артем Александрович

5. Термины и сокращения:

«ОС» - Операционная система;

«UI» - пользовательский интерфейс;

«Android» – ОС для смартфонов и планшетов от Google;

«iOS» – ОС для смартфонов и планшетов от Apple;

«Firebase» – платформа, набор инструментов и сервисов для разработки аутентификационной компоненты приложения;

«Мероприятие» – совокупность действий, нацеленных на выполнение единой задачи. Дополнительно может подразумевать наличие списка участников и некоторого плана мероприятия;

6. Технические требования

Таблица 1 – Общие требования к разрабатываемой системе

Требование	Способ реализации	Используемые инструменты	Потребитель
Кроссплатформенность	Разработка двух приложений под Android и iOS	Kotlin / Swift	Все пользователи

Возможно сть входа по электронной почте	Реализаци я способа взаимодействия с базой данных	Firebase	Разработч ики и пользователи
Планиров ание и просмотр мероприятий: описание, дата, время, место	Создание соответствующ их экранов в приложении	Kotlin / Swift	Все пользователи
Редактиро вание и просмотр профиля, просмотр списка созданных и посещенных мероприятий	Создание соответствующ их экранов в приложении	Kotlin / Swift	Все пользователи

Таблица 2 – Функциональные требования к системе

Требован ие	Способ реализации	Использу емые инструменты	Потребит ель
Если пользователь авторизован, то	Создание зависимостей компонентов	Kotlin / Swift	Все пользователи

отображается главный экран со списком всех мероприятий и навигационной панелью			
Экран создания встречи имеет форму с обязательными полями информации о мероприятии	Создание соответствующ их элементов дизайна	Figma	Все пользователи
При клике на мероприятие отображается подробная информация о встрече, список участников	Создание зависимостей компонентов	Figma; Kotlin / Swift;	Все пользователи

7. Содержание работы

Таблица 3 – Этапы разработки

Этап	Сроки выполнения
------	------------------

Анализ требований	1.11.2024 – 10.11.2024
Изучение литературы	1.11.2024 – 24.11.2024
Проектирование UI	25.11.2024 – 8.12.2024
Разработка приложения	25.11.2024 – 16.12.2024
Тестирование	16.12.2024 – 18.12.2024

8. Основные результаты работы и формы их представления

Таблица 4 – список результатов работы и способов их представления

Результат работы	Форма представления
Техническое задание	Файл технического задания оформленный по ГОСТу в формате “docx” или “pdf”
Проект UI	Макет ключевых экранов приложения
Экран входа/регистрации	Набор данных (package) содержащий ключевые файлы обеспечивающие работу этого фрагмента приложения
Экран основного меню	
Экран создания мероприятия	
Экран информации о мероприятии	
Экран профиля	

Экран редактирования профиля	
Презентация	Подготовленный файл презентации в формате “pptx” или подобном с планом выступления