

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 микрообучения»

Обучающийся: Савченко Влад Александрович K3139

Санкт-Петербург 2024

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
ВВЕДЕНИЕ	3
1 Суть проекта	5
2 Процессы работы над всем проектом	6
3 Мои задачи	6
Создание базы данных.....	6
Настройка связей между таблицами	9
Этап интеграции фронтенда и бэкэнда	10
4 Основные трудности	12
1. Изучение нового языка программирования — Kotlin	12
2. Работа с кодом напарников	12
5 Взаимодействие с командой	13
6 Взаимодействие с руководителем	14
ЗАКЛЮЧЕНИЕ.....	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	16
ПРИЛОЖЕНИЕ.....	17

ВВЕДЕНИЕ

На сегодняшний день образовательные технологии переживают бурное развитие. В условиях стремительно меняющегося мира и информационной перегрузки традиционные методы обучения становятся недостаточно эффективными. Люди часто сталкиваются с нехваткой времени на получение новых знаний и освоение навыков, что особенно актуально в условиях высоких требований современного рынка труда. На фоне этих вызовов всё большую популярность приобретает микрообучение, как подход, позволяющий интегрировать учебный процесс в повседневную жизнь без значительных временных затрат.

Микрообучение — это методика, основанная на предоставлении обучающимся небольших, легко усваиваемых порций информации. Такой подход позволяет лучше воспринимать и запоминать материал, а также адаптироваться к индивидуальному темпу обучения. Короткие учебные модули обеспечивают возможность быстрого погружения в материал и возвращения к нему в удобное время, что делает микрообучение особенно востребованным в мобильной среде.

Актуальность разработки мобильного приложения для микрообучения продиктована рядом причин. Во-первых, современные технологии позволяют создавать удобные и функциональные мобильные решения, которые обеспечивают доступ к образовательному контенту в любой момент. Во-вторых, растущий интерес к самообразованию и профессиональному развитию стимулирует спрос на инструменты, упрощающие этот процесс. В-третьих, использование мобильных устройств стало неотъемлемой частью повседневной жизни, что делает мобильные приложения наиболее удобным инструментом для обучения.

Целью данного проекта является создание мобильного приложения, которое будет ориентировано на эффективное и доступное микрообучение. Разработка такого приложения должна учитывать современные

образовательные тенденции, потребности пользователей, а также технические и дизайнерские требования для обеспечения интуитивно понятного интерфейса и удобства использования.

Для достижения поставленной цели в проекте решаются следующие задачи:

1. Проведение анализа существующих решений на рынке микрообучения для выявления их сильных и слабых сторон.
2. Разработка концепции приложения, включающей описание ключевых функций и интерфейсных решений.
3. Создание рабочей версии приложения с базовым набором функций, таких как регистрация и авторизация, доступ к учебным модулям и тестированию.
4. Проведение тестирования приложения с целью выявления возможных недостатков и внесения необходимых улучшений.

1 Суть проекта

Проект "Мобильное приложение для микрообучения" направлен на создание цифрового инструмента, который оптимизирует процесс получения новых знаний и навыков с помощью структурированного подхода к подаче учебных материалов.

Главной задачей приложения является предоставление пользователям удобного способа освоения образовательных материалов небольшими блоками, которые легко воспринимаются и вписываются в повседневный график.

Мобильное приложение создаётся с учётом ключевых принципов микрообучения:

1. Все учебные модули будут разделены на компактные части, каждая из которых рассчитана на 5–7 минут изучения. Это позволяет пользователю сосредоточиться на конкретном аспекте темы и избежать перегрузки информации.
2. Каждый учебный модуль будет сопровождаться практическими заданиями, тестами или короткими опросами. Это способствует лучшему усвоению материала и позволяет пользователю сразу закрепить новые знания.
3. В приложении будет поддерживаться разнообразие форматов: текстовые лекции, графики, видеоуроки. Это обеспечит возможность выбирать наиболее удобный формат для каждого пользователя.

2 Процессы работы над всем проектом

1. Аналитика
2. Создание базы данных
3. Разработка бэкенда
4. Разработка фронтенда
5. Соединение бэкенда и фронтенда
6. Тестирование продукта

3 Мои задачи

Так как в проекте я являлся фуллстак-разработчиком, основными задачами, которые я выполнил, были:

Создание базы данных

Для реализации проекта была спроектирована и создана база данных с использованием реляционной СУБД PostgreSQL. Эта система управления базами данных была выбрана благодаря её надёжности, производительности и поддержке современных функциональных возможностей, таких как работа со сложными запросами и транзакциями.

Проектирование базы данных базировалось на требованиях, предоставленных аналитиками. На основе анализа были выделены три основные сущности, для которых были созданы соответствующие таблицы:

1.1. Таблица пользователей

Эта таблица предназначена для хранения всей информации о пользователях приложения. Она включает данные, необходимые для идентификации пользователей, а также для предоставления персонализированных возможностей. Структура таблицы была разработана следующим образом, изображённом на рисунке 1:

Имя таблицы:	USERS	Уникальная информация о пользователях	
Наименование полей	тип данны	допускается null (y/n)	описание
USER_ID	integer	n	ID-номер
USER_MAIL	varchar (1000	n	Почта
USER_ITMO_ID	integer	n	ITMO ID
USER_ITMO_ID_PASSWORD	integer	n	ITMO ID пароль
PASSWORD_HK	varchar (1000	n	Хешфункция
NAME	varchar (1000	n	Имя пользователя (ФИО)

Рисунок 1 – Таблица с информацией о пользователях

- user_id — уникальный идентификатор пользователя.
- user_mail — адрес электронной почты для связи и восстановления доступа.
- user_itmo_id – уникальный идентификатор студентов из ITMO, для возможности входа через ITMO ID
- user_itmo_id_password — хэш пароля для ITMO ID для студентов из ITMO.
- password_hk – хэш пароль пользователя для обеспечения безопасности
- name — имя пользователя, используемое для авторизации.

1.2. Таблица курсов

Эта таблица хранит данные обо всех учебных курсах, доступных в приложении. Она позволяет организовать учебные материалы в структурированном виде. Поля таблицы изображены на рисунке 2:

Имя таблицы:	COURSES	Уникальная информация о курсах	
Наименование полей	тип данны	допускается null (y/n)	описание
COURSE_ID	integer	n	ID-курса
NAME_COURSE	varchar (1000	n	Название курса
DURATION	time	n	Время прохождения курса
DESCRIPTION	varchar (2000	n	Описание курса
LINK_FOR_SOURCE	varcahr (1000	???	Ссылка на источник

Рисунок 2 – таблица с информацией о курсах

- course_id (PRIMARY KEY) — уникальный идентификатор курса.
- name_course — название курса.
- duration – время необходимое для прохождения курса
- description — описание курса, включающее его цели и содержание.
- link_for_source — ссылка на образовательные материалы.

1.3. Таблица курсы-пользователи

Эта таблица реализует связь между пользователями и курсами, а также хранит данные о прогрессе каждого пользователя. Данная таблица является промежуточной и создаёт связь между пользователями и курсами. Её структура изображена на рисунке 3:

Имя таблицы:	USER_COURSE	Таблица, связывающая данные о пользователе и данные о курсе	
Наименование полей	тип данны	допускается null (y/n)	описание
USER_ID	integer	n	ID-номер пользователя
COURSE_ID	integer	n	ID-курса
PROGRESS	integer	n	расчетная функция
cnr_f	0/1	n	признак актуальности записи

Рисунок 3 – Таблица курсы-пользователи

- user_id - идентификатор пользователя, связанный с курсом.
- course_id - идентификатор курса, с которым взаимодействует пользователь.
- progress - процент выполнения курса пользователем.
- cnr_f – информация хранящая результат выполнения модуля в виде переменной boolean(True или False)

После завершения этапа проектирования структуры базы данных аналитиками были разработаны и выполнены SQL-запросы для создания соответствующих таблиц указанных выше. Вот пример создания таблицы пользователей:

```
CREATE TABLE users (  
    user_id SERIAL PRIMARY KEY,  
    user_mail VARCHAR(100) NOT NULL,  
    user_itmo_id VARCHAR(255) NOT NULL,  
    user_itmo_id_password VARCHAR(255) NOT NULL,  
    password_hk VARCHAR(255) NOT NULL,  
    name VARCHAR(50) NOT NULL,  
);
```

Настройка связей между таблицами

Для обеспечения целостности и логической связности данных в базе данных была проведена тщательная настройка связей между таблицами. На этапе проектирования структуры базы данных были использованы схемы и диаграммы, на которых столбцы некоторых таблиц были выделены цветами – зеленым и красным.

Зелёным цветом обозначались столбцы, которые играют роль первичного ключа (Primary Key). Эти столбцы содержат уникальные значения, используемые для однозначной идентификации каждой записи в таблице.

Красным цветом выделялись столбцы, которые выступают в качестве внешних ключей (Foreign Key). Они обеспечивают связь между таблицами, ссылаясь на первичные ключи других таблиц.

Таким образом, были установлены следующие связи между таблицами: столбец `user_id` из таблицы пользователей был связан с полем `user_id` таблицы "курсы-пользователи", а столбец `course_id` из таблицы курсов — с полем `course_id` таблицы "курсы-пользователи". Эти связи обеспечили корректное взаимодействие данных между таблицами и логическую целостность базы данных.

Этап интеграции фронтенда и бэкэнда

После завершения проектирования и создания базы данных следующим важным шагом стала интеграция фронтенда и бэкэнда. Эта задача была направлена на обеспечение корректного взаимодействия пользовательского интерфейса с серверной частью приложения, что позволило пользователям в реальном времени работать с данными, хранящимися на сервере.

Для этого на сервере была реализована функция, которая обрабатывала входящие HTTP-запросы от фронтенда. Эта функция отвечала за:

1. Приём запросов

Обрабатывались запросы типа GET, POST, PUT или DELETE в зависимости от требуемой операции.

2. Взаимодействие с базой данных

Функция выполняла соответствующий SQL-запрос к базе данных.

3. Возврат ответа

После выполнения операции на сервере данные возвращались фронтенду в формате JSON для дальнейшего отображения или обработки.

Таким образом, интеграция фронтенда и бэкэнда включала в себя:

- Создание API на стороне сервера для обработки запросов.
- Реализацию функций на клиенте, которые отправляют запросы к серверу.
- Настройку связи между действиями пользователя в интерфейсе и функциями фронтенда.

Результатом этого этапа стало взаимодействие между интерфейсом приложения и его серверной частью, однако на данный момент получилось реализовать только регистрацию и авторизацию пользователей. Окна регистрации и авторизации изображены на рисунках 4, 5:

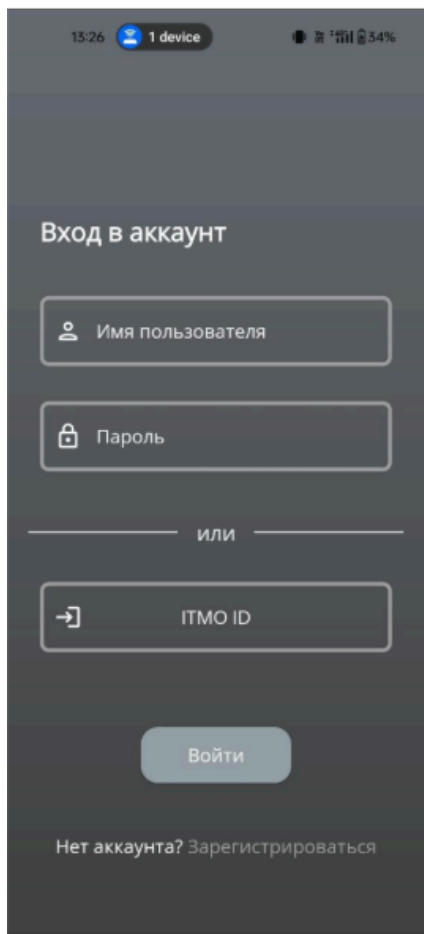


Рисунок 4 – Окно авторизации

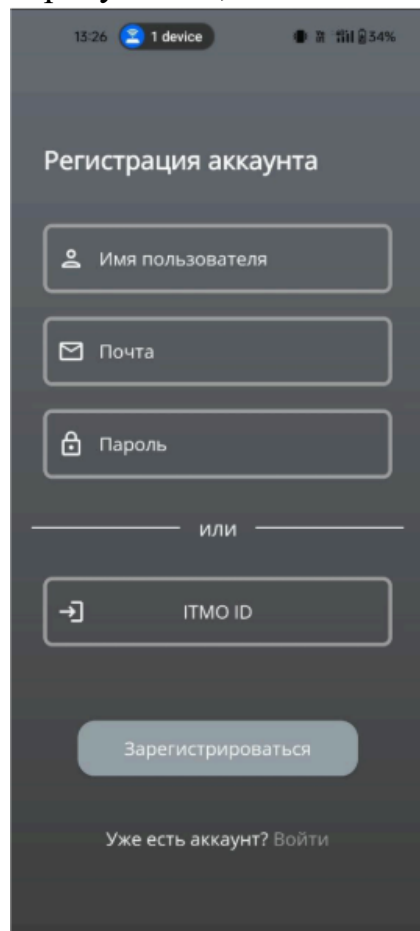


Рисунок 5 – Окно регистрации

4 Основные трудности

В процессе выполнения проекта я столкнулся с рядом сложностей, которые оказали значительное влияние на скорость и организацию моей работы.

1. Изучение нового языка программирования — Kotlin

Одной из главных трудностей стало то, что для разработки проекта требовалось использовать язык программирования Kotlin, с которым я ранее не работал. Несмотря на то, что у меня был опыт программирования на других языках, мне пришлось погрузиться в изучение синтаксиса и особенностей Kotlin.

Эти трудности замедляли процесс работы, так как мне приходилось одновременно изучать новый язык и реализовывать функциональность, но это также дало возможность значительно улучшить свои навыки и освоить современный инструмент разработки.

2. Работа с кодом напарников

Другой важной сложностью стало взаимодействие с кодом, написанным напарниками. После того как они завершали свою часть работы и передавали мне код для интеграции, мне необходимо было детально ознакомиться с их реализацией. Это включало:

- Разбор названий переменных, функций и методов, которые часто отличались от привычных мне принципов именования. Например, некоторые переменные и функции имели названия, не отражающие их реальное назначение, что усложняло их понимание.
- Выявление зависимостей между функциями и их взаимодействия, чтобы корректно интегрировать их с моей частью работы.
- Иногда требовалась оптимизация или изменение частей кода, чтобы обеспечить корректное взаимодействие между компонентами.

Эти трудности удлиняли процесс разработки связей между фронтендом и бэкендом, так как требовалось больше времени на анализ

переданного кода, его тестирование и интеграцию в общую структуру проекта.

Несмотря на указанные трудности, я смог успешно справиться с поставленными задачами. Работа над проектом стала ценным опытом, позволившим мне не только освоить новый язык программирования, но и улучшить навыки анализа и интеграции кода, написанного другими разработчиками. Это также дало понимание важности согласованного подхода к именованию переменных и функций в командной разработке для повышения её эффективности.

5 Взаимодействие с командой

Работа в команде в рамках проекта прошла организованно, эффективно и без задержек. Каждый участник команды выполнял свои задачи своевременно, что позволило поддерживать высокий темп разработки и избежать простоев.

Регулярные обсуждения и согласование этапов работы обеспечили ясность в распределении задач. Все участники проекта ответственно подходили к своей части работы, а при необходимости быстро предоставляли разъяснения или исправления. Благодаря этому процесс интеграции различных компонентов системы, включая базу данных, бэкенд и фронтенд, прошёл без серьёзных затруднений.

Командная работа также отличалась взаимопомощью: при возникновении вопросов или сложностей в реализации, члены команды оперативно предоставляли поддержку и делились своим опытом с своими напарниками. Такой подход позволил решить технические задачи максимально быстро и эффективно.

6 Взаимодействие с руководителем

В процессе работы над проектом важным аспектом была наша взаимодействие с руководителем. Он проявил большую отзывчивость, всегда готов был помочь в решении возникших вопросов и поддержать в трудные моменты. Руководитель грамотно оценивал наши силы и возможности, что позволило правильно распределить задачи и избежать перегрузок в команде. Он устанавливал реалистичные и адекватные сроки для выполнения задач, что способствовало качественному выполнению работы в срок. Такой подход позволил не только эффективно двигаться к цели, но и обеспечил оптимальный рабочий процесс, в котором каждый из нас чувствовал поддержку и понимание.

ЗАКЛЮЧЕНИЕ

В ходе разработки проекта "Мобильное приложение для микрообучения" я значительно улучшил свои навыки работы в команде, что стало важным шагом в профессиональном росте. Работа в коллективе научила меня эффективно взаимодействовать с коллегами, обмениваться идеями и совместно решать задачи.

К сожалению, в силу ограниченности времени и ресурсов, мы не смогли реализовать большое количество курсов внутри приложения. Также не смогли наполнить, созданный, курс информативным контентом по причине нехватки компетентности в данном аспекте. Но несмотря на то, что мы не всё успели реализовать, приложение получилось очень удобным, хорошо вписывающимся в современные нормы.

В процессе разработки я также освоил новый язык программирования — Kotlin, что расширило мои технические возможности и позволило глубже понять особенности разработки под платформу Android. Кроме того, я познакомился с системой управления базой данных PostgreSQL, что значительно расширило мои знания в области работы с базами данных и оптимизации их использования.

Этот опыт был для меня важным этапом в совершенствовании как программиста и специалиста в области мобильной разработки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Статья что такое PostgreSQL - <https://practicum.yandex.ru/blog/cto-takoe-subd-postgresql/>

Освоение PostgreSQL - <https://selectel.ru/blog/tutorials/create-postgresql/>
<https://www.youtube.com/watch?v=HVQNxdI6fqY>

Работа с Android Studio - <https://blog.skillfactory.ru/kak-polzovatsya-android-studio/>

Курс по Kotlin - <https://itproger.com/course/kotlin>

Статья, как работает объединение фронтенда и бэкенда - <https://selectel.ru/blog/tutorials/frontend-and-backend/>

ПРИЛОЖЕНИЕ

Имя таблицы:	USERS	Уникальная информация о пользователях	
Наименование полей	тип данны	допускается null (y/n)	описание
USER_ID	integer	n	ID-номер
USER_MAIL	varchar (1000	n	Почта
USER_ITMO_ID	integer	n	ITMO ID
USER_ITMO_ID_PASSWORD	integer	n	ITMO ID пароль
PASSWORD_HK	varchar (1000	n	Хешфункция
NAME	varchar (1000	n	Имя пользователя (ФИО)

Рисунок 1 – Таблица с информацией о пользователях

Имя таблицы:	COURSES	Уникальная информация о курсах	
Наименование полей	тип данны	допускается null (y/n)	описание
COURSE_ID	integer	n	ID-курса
NAME_COURSE	varchar (1000	n	Название курса
DURATION	time	n	Время прохождения курса
DESCRIPTION	varchar (2000	n	Описание курса
LINK_FOR_SOURCE	varcahr (1000	???	Ссылка на источник

Рисунок 2 – Таблица с информацией о курсах

Имя таблицы:	USER_COURSE	Таблица, связывающая данные о пользователе и данные о курсе	
Наименование полей	тип данны	допускается null (y/n)	описание
USER_ID	integer	n	ID-номер пользователя
COURSE_ID	integer	n	ID-курса
PROGRESS	integer	n	расчетная функция
cnr_f	0/1	n	признак актуальности записи

Рисунок 3 – Таблица курсы-пользователи

Рисунок 4 – Окно авторизации

Рисунок 5 – Окно регистрации

Техническое задание

Название проекта: Мобильное приложение для микрообучения.

Цель: разработать мобильное приложение для микрообучения студентов по различным образовательным дисциплинам.

Сроки выполнения:

Начало 01.11.2024

Окончание 20.12.2024

Исполнитель проекта: Ганичева Лидия Сергеевна.

Термины и сокращения:

МП – Мобильное приложение;

ТД – Тестовые данные;

СУБД – Система управления базами данных;

ТЗ – Техническое задание;

ФТ – Функциональные требования;

Технические требования:

Разработка фронт части МП HTML (для создания базовой структуры страниц и контента), CSS (для стилизации внешнего вида) и JavaScript (для добавления интерактивности), любой другой язык программирования для верстки на усмотрение разработчика и команды.

Разработка бэк части МП Java/PHP/ Ruby/ C#/ любой другой язык программирования на усмотрение разработчика PostgreSQL/MySQL / и т.д.

Функциональные требования:

Безопасность хранения паролей, при обработке паролей от пользователей необходимо использовать хэш функцию для шифрования и хранить в БД зашифрованное поле.

Регистрация и авторизация пользователя в системе.

Перечень курсов и программ отображаются и кликабельны для

пользователей.

Подсчет статистики и прогресса по курсу у каждого пользователя по всем его открытым курсам.

Этапы работы:

Подготовка к разработке МП;

Работы аналитика по реализации ФТ;

Разработка бэка МП;

Разработка фронта МП;

Администрирование инфраструктуры для работы МП;

Тестирование продукта после реализации;

Управление командой разработки;

Задачи проекта:

Проанализировать предметную область (стек). Необходимо на опыте (статьи хабр и т.д.) аналогов выяснить с какими трудностями столкнулись команды при работе с определенным стеком. По окончании анализа совместно с командой обсудить предлагаемый стек, зафиксировать договорённость по стеку;

Проанализировать предметную область (продукт). Необходимо провести ресечь аналогов (в части интерфейса МП), собрать пулл наилучший практик по реализации дизайна МП. Зафиксировать результаты в файле, базе знаний;

Разработать дизайн МП. Необходимо по результатам предыдущей задачи подготовить макеты для экранов МП. Зафиксировать макеты в файле, ссылке на фигму.

Подготовить постановку на витрины данных. По результатам анализа функциональных требований и предметной области подготовить постановку на необходимые витрины данных (атрибутивный состав, типы данных, констрэйны и т.д.). Постановку отобразить в отдельном файле, удобном для

чтения разработчика, реализации витрин.

Создать витрины данных. Необходимо по постановке аналитика создать скрипты для витрин БД. Создать таблицы, подготовить совместно с аналитиком тестовые данные, заинсертить ТД в витрины.

Разработка бэка МП. Необходимо развернуть БД, докер. Подготовить инфраструктуру для работы команды

Разработать бэк. Разработать все необходимые процедуры для бэк части МП;

Сверстать фронт. По макетам аналитика сверстать фронт часть МП;

Объединить фронт и бэк. После верски объединить фронт и бэк приложения.

Тестирование продукта. В ходе работы проводить поэтапное тестирование продукта;

Подготовка презентации на защиту проекта. Подготовить презентацию для защиты проекта, собрать основные результаты, отобразить их, согласовать с командой и руководителем;

Минимальные функциональные требования к приложению:

20

Регистрация и вход в систему;

Перечень курсов и программ;

Подсчет статистики и прогресса по курсу;

Основные результаты работы:

1. Аналитика

Проведен ресечь аналогов, баз знаний, выбран стек технологий для работы команды

Проанализированы ФТ

Подготовлена Постановка на витрины данных

2. Дизайн

Проведен ресечь аналогов, выбраны лучшие примеры реализации

Подготовлены макеты для МП

3. Фронт МП

Сверстан фронт МП

Фронт и бэк МП соединен, обмен информацией работает корректно

4. Бэк МП

Развернута необходимая инфраструктура для работы МП

Созданы витрины данных

Прописан код для работы бэка

5. Тестирование

Проведено тестирование МП, функциональность полностью реализует
ФТ

Устранены дефекты разработки

6. Отчёты

Разработано МП по минимальным ФТ

Подготовлена презентация по результатам работы команды