

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

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и облачные технологии**

Направление подготовки (специальность) **09.03.03 Прикладная информатика**

## **КУРСОВОЙ ПРОЕКТ**

**по дисциплине «Инфокоммуникационные системы и технологии»**

на тему:

Разработка мобильного приложения для микрообучения студентов по различным образовательным дисциплинам

Обучающийся Мульдияров Арсений Алексеевич, группа К3141

**Работа сдана**  
Дата 04.01.2025

Санкт-Петербург 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Описание проекта.....	5
2 Работа над проектом .....	6
2.1 Задачи команды .....	6
2.2 Задачи, поставленные передо мной.....	6
2.3 Ход работы .....	6
3 Коммуникация в команде .....	15
ЗАКЛЮЧЕНИЕ .....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	18
ПРИЛОЖЕНИЕ .....	19

## **ВВЕДЕНИЕ**

Современное образование стремительно адаптируется к условиям цифровизации и изменениям в потребностях студентов. Одним из таких изменений является возрастающая популярность мобильных приложений, которые предоставляют доступ к образовательным материалам в удобной и интерактивной форме. Формат микрообучения, предполагающий предоставление знаний малыми порциями и в краткие временные промежутки, становится все более востребованным, особенно среди студентов вузов, таких как ИТМО. Это связано с необходимостью совмещения учебной деятельности с другими важными аспектами жизни: стажировками, проектной работой и саморазвитием.

Для студентов ИТМО, обучающихся в интенсивном академическом ритме, важно иметь возможность гибко подходить к организации учебного процесса. Однако традиционные методы подготовки к экзаменам или повышения квалификации часто требуют значительных временных затрат. В связи с этим становится актуальной разработка мобильных приложений, которые не только обеспечат доступ к необходимым материалам, но и помогут повысить эффективность обучения за счет интерактивности и персонализированного подхода.

Целью проекта является разработка мобильного приложения для микрообучения, которое позволит студентам ИТМО эффективно готовиться к экзаменам, осваивать новые навыки и повышать квалификацию в рамках своей профессиональной деятельности. Приложение ориентировано на создание комфортной образовательной среды, где знания предоставляются в компактной, легко усваиваемой форме.

Для достижения указанной цели были поставлены следующие задачи:

- .....  
роанализировать предметную область, выявить конкурентов и понять, чем они хуже или лучше, прописать стек технологий, который используется в подобных приложениях, собрать пулл наилучших практик по реализации дизайна.
- .....  
оздать прототип дизайна мобильного приложения.
- .....  
о результатам анализа функциональных требований и предметной области подготовить постановку на необходимые витрины данных.
- .....  
о постановке аналитика создать скрипты для витрин баз данных.
- .....  
азвернуть базы данных, докер, подготовить инфраструктуру для работы команды.
- .....  
азработать все необходимые процедуры для бэк-части мобильного приложения.
- .....  
о макетам аналитика сверстать фронт-часть мобильного приложения.
- .....  
бъединить фронт и бэк части мобильного приложения.
- .....  
ходе работы проводить поэтапное тестирование продукта.
- .....  
одготовить презентацию для защиты проекта, собрать основные результаты, отобразить их.

## 1 Описание проекта

Мобильное приложения для микрообучения представляет собой сервис для практически неограниченного доступа к курсам, целью которых является подготовка студентов к экзаменам, зачетам и другим ключевым точкам во время образовательного процесса, а также повышение квалификации и профессиональных навыков для более комфортного вхождения в IT-индустрию.

Приложение было разработано с использованием современных технологий, обеспечивающих user-friendly интерфейс, который позволяет комфортно обучаться новым дисциплинам прямо с экрана смартфона. Более подробное описание используемых технологий:

- Kotlin поддерживает функциональное и объектно-ориентированное программирование, что делает его универсальным инструментом для разработки.
- Android Studio базируется на платформе IntelliJ IDEA и предоставляет все необходимые инструменты для эффективной разработки Android-приложений
- .NET — это кроссплатформенная платформа разработки, которая используется для создания различных приложений, включая веб-сервисы, десктопные программы, мобильные приложения и облачные решения.
- PostgreSQL — это мощная объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом и поддержкой множества типов данных, которая отлично подходит для прототипирования.
- Figma — это облачная платформа для дизайна интерфейсов, прототипирования и совместной работы над проектами.

В совокупности эти инструменты обеспечивают разработку производительного и удобного мобильного приложения для микрообучения.

## **2 Работа над проектом**

### **2.1 Задачи команды**

В начале работы над проектом-приложением командой был составлен план разработки:

- 1) в соответствии с распределенными ролями в команде, разделить обязанности и установить сроки выполнения задач
- 2) выполнение индивидуальных задач
- 3) подготовка к защите проекта-приложения
- 4) командная защита проекта-приложения
- 5) написание индивидуального отчета по проделанной работе

### **2.2 Задачи, поставленные передо мной**

Задача, которая стояла передо мной в проекте - системная аналитика разрабатываемого приложения. Если говорить конкретней, то мне было необходимо проанализировать предметную область, а именно стек технологий, который использовался при разработке подобных приложения по микрообучению, подготовить постановку на витрины данных, создать ER-диаграмму для наглядного представления, как должны взаимодействовать между собой базы данных в мобильном приложении, прописать тестовые данные для заполнения баз данных, создать презентацию для итоговой защиты проекта-приложения.

### **2.3 Ход работы**

Перед началом работы над курсовой работой мне нужно было изучить базовую теорию баз данных, как они работают и для чего они нужны. Этот материал мне удалось найти в статье на сайте Хабр[1]. После этого я принялся изучать конкурентов, а именно: какие технологии использовались при разработке, с какими трудностями столкнулась команда разработки, какая

функциональность у аналогов, какие фишки и особенности можно позаимствовать для нашего проекта. В этом мне помогли данные с таких сайтов, как: bsense[2], Хабр[3], IEEE Spectrum[4]. Оказалось, что на данный момент самыми популярными сервисами по микрообучению являются Duolingo, Stepik, iSpring learning. В основном при разработке данных приложений использовались примерно одни и те же технологии, код писали на похожих или взаимозаменяемых языках. Из ключевых особенностей конкурентов можно выделить помощь в улучшении и дообучении нейронных сетей при выполнении заданий в Duolingo, а также возможность создавать свои собственные курсы и даже монетизировать их в Stepik. Благодаря информации из статей и сервисов я создал файл с подробной и полезной информацией для команды разработки.

После выполнения этой задачи, я приступил к подготовке постановки на витрины данных. В том, как именно это делается мне помог наш project manager - Ганичева Лидия. Лидия уже не первый год работает в IT-сфере на позиции аналитика, поэтому она с радостью поделилась своими знаниями, рассказала про свой рабочий день, что именно входит в её обязанности и в чём заключается её работа. Для удобства она загрузила файл с полезной информацией об аналитике в IT на Google диск нашей команды. После нескольких консультаций с Лидией, я принялся за постановку на витрины данных.

Витрина данных (или Data Mart) — это специализированное хранилище данных, которое предназначено для поддержки анализа и отчетности в определенной области бизнеса или функциональной области. В отличие от более крупного хранилища данных (Data Warehouse), которое может содержать данные из всей организации, витрина данных фокусируется на конкретных аспектах, таких как продажи, маркетинг или финансы. В моём случае мне нужно было продумать логику взаимодействия баз данных между собой. То есть если говорить простым языком, нужно было придумать по

каким ключам будут связаны между собой данные в серверной части мобильного приложения. Оказалось, что это не так уж и сложно.

В первой таблице (базе данных) у нас хранится вся информация о пользователях, а именно: имя пользователя, ID пользователя, почта, на которую зарегистрирован аккаунт пользователя, пароль от аккаунта, ITMO ID и пароль от ITMO аккаунта, если пользователь решил зарегистрироваться по данным ВУЗа. В системе в эти данные относятся к одному из типов данных: integer или varchar. Integer - целочисленный тип данных. Varchar (variable character) — это тип данных, используемый в реляционных базах данных для хранения строковых значений переменной длины. Интересно, что пароль в системе хранится, как хеш-функция. А, в свою очередь, хеш-функция - это функция, которая преобразует входные данные произвольной длины (например, текст, файл или сообщение) в фиксированное значение фиксированной длины, называемое хеш-значением или хешем.

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, представляющая то, что будет храниться в базе о пользователях

Зеленым в таблице помечены ячейки, которые в системе числятся, как primary key (первичный ключ). Первичный ключ - это уникальный идентификатор для каждой записи в таблице реляционной базы данных. Он



обеспечивает уникальность строк и позволяет эффективно управлять данными. Синие ячейки - просто название таблицы (помечены для удобства).

Следующая таблица представляет информацию о курсах, которые будут проходить пользователи в мобильном приложении. У каждого курса есть следующие пункты: ID курса, название курса, примерное время прохождения курса, описание того, что вы узнаете, пройдя курс от начала и до конца, ссылка на курс, если он находится на сторонней платформе. В этой таблице (базе) используются все те же типы данных (integer и varchar), но добавляется ещё один - time. Time - тип данных, обозначающий время.

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, представляющая то, что будет храниться в базе о курсах

Здесь ячейки зеленого и синего цветов обозначают то же самое, что и в таблице, описанной выше.

Завершающей таблицей для данных в системе, стала таблица, связывающая пользователей и курсы. В этой таблице должны храниться следующие данные: ID пользователя, ID курса, прогресс пользователя и маркер, показывающий актуальность записи. В основном все данные в этой таблице (базе) представлены в целочисленном типе данных (integer), кроме маркера актуальности. Маркер относится к бинарному (binary) типу данных, то есть может иметь всего два значения: 0 или 1.

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, представляющая, как будут связаны между собой пользователь и курсы в приложении

В этой таблице красным помечены ячейки, обозначающие foreign key (внешний ключ). Внешний ключ - это поле или набор полей в таблице реляционной базы данных, которое используется для создания связи между данными в двух таблицах. То есть информация о пользователе поступает в данную таблицу (базу) по ID пользователя, а информация о курсе - по ID курса.

Прогресс прохождения курса мы с командой решили вычислять следующим способом:

	предложение по расчету					
PROGRESS	Триггер на запуск видео/теста	Триггер на окончание видео	Каунт1 модулей (видео) всего	Каунт2 выполненных модулей (просмотренных видео)	$W=1/2*100$	Инсерты при каждом обновлении прогресса с условием что последняя запись cnr_f=1, а предыдущие неактуальные =0

Рисунок 4 - представление о том, как будет просчитываться прогресс прохождения курса

После этого я решил продумать несколько дополнительных функций, которые можно будет реализовать в последствии, если проект продолжит развиваться вне курсовой работы.

Возможный функционал				
Подсчет количества допущенных ошибок за время прохождения курса	Подсчет времени прохождения курса (узнав среднее значение времени, можно более точно указать, сколько часов уйдет на прохождение курса => преподаватель может включать прохождение курса в учебный план)	После прохождения курса периодически предлагать выполнить тест для закрепления/запоминания пройденного материала	При регистрации уточнять интересы пользователя, дабы точнее составлять рекомендации по курсам	Факт дня (каждый день при первом заходе в приложение выдавать интересный факт по интересующим пользователя темам)

Рисунок 5 - список дополнительных функций для мобильного приложения (часть 1)

Система достижений (прошел свой первый курс на платформе; прошел пять/десять курсов; прошел курс быстрее указанного времени и т.д.)	В описании к курсу указывать рекомендации (каким уровнем знаний нужно обладать)	Серия ежедневных заходов в приложение	Пуш-уведомления о не пройденном курсе/обнуление стрика заходов в приложение
---	---	---------------------------------------	---

Рисунок 6 - список дополнительных функций для мобильного приложения (часть 2)

Затем Лидия попросила меня составить ER-диаграмму, чтобы команде разработки было наглядно видно, как взаимодействуют между собой базы данных и какие между ними связи. В том, как именно составить такую диаграмму мне помогла статья на блоге Яндекс Практикума[5]. Из интересного могу отметить, что сам процесс составления таких диаграмм - дело очень затягивающее. Оказывается есть несколько типов связи, такие как:

один, много, один и только один, ноль или один, один или много, ноль или много.

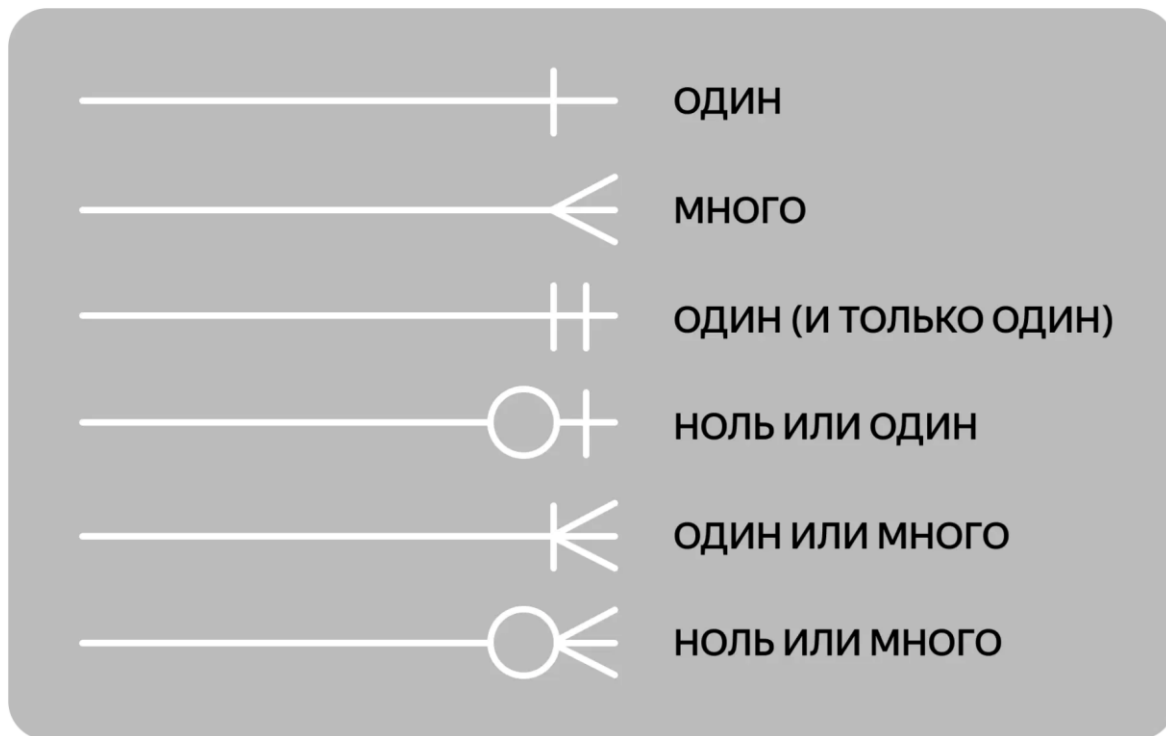


Рисунок 7 - виды связи между сущностями в ER-диаграммах

В нашем случае диаграмма выглядела следующим способом:

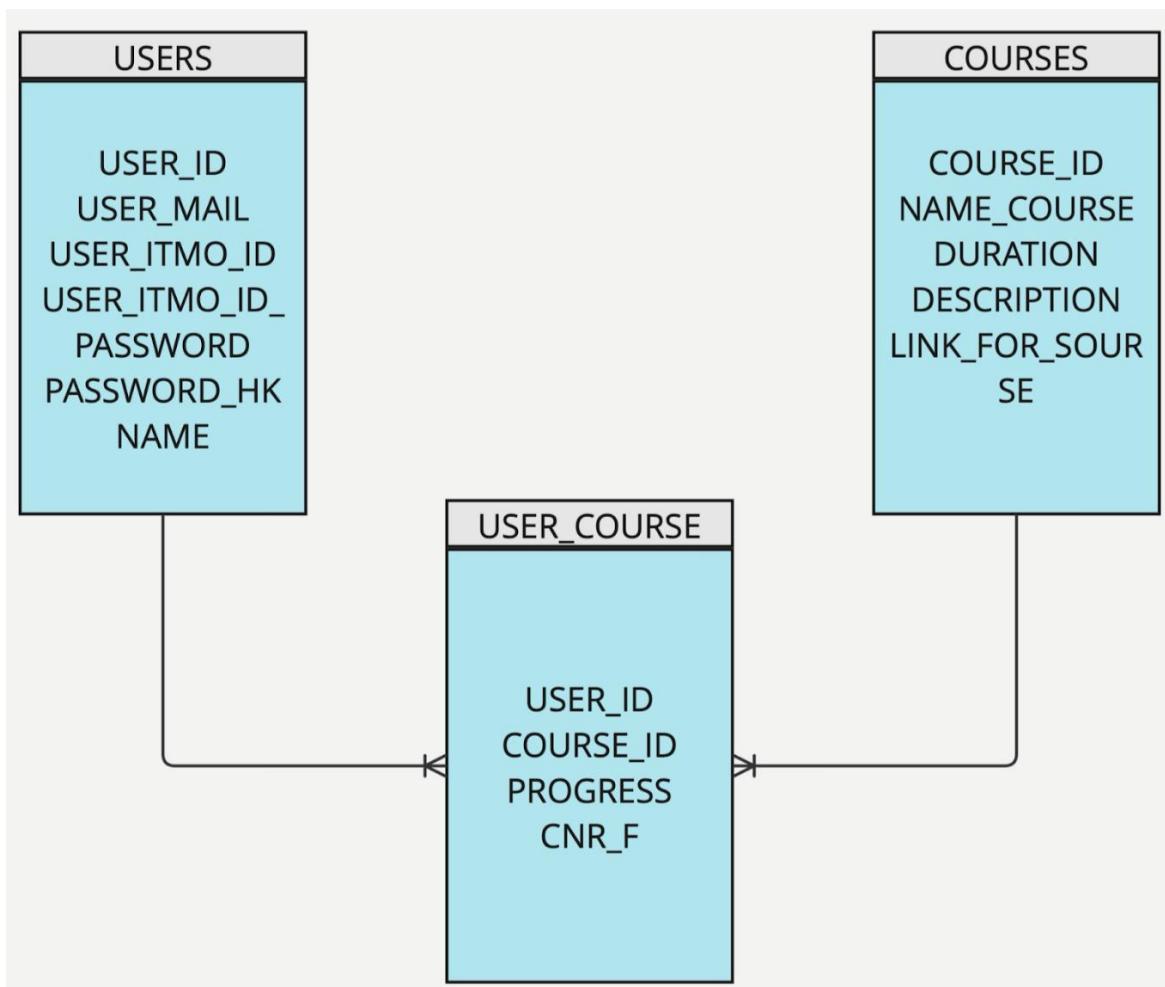


Рисунок 8 - ER-диаграмма мобильного приложения по микрообучению

После этого я решил сделать классическую диаграмму, чтобы каждому из команды было понятно, что нужно сделать в серверной части приложения.

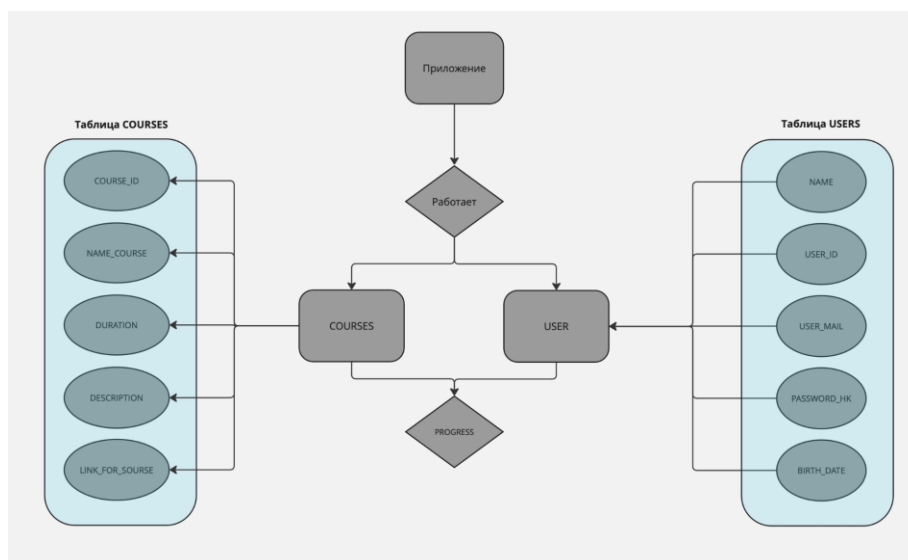


Рисунок 9 - диаграмма, разрабатываемого приложения

Завершив постановку на витрины данных и проработку диаграмм, я принялся за написание тестовых данных для наполнения баз данных. Достаточно скучное занятие, которое можно ускорить с помощью нейросетей, но я решил заняться этим вручную. Все данные я брал из головы и затем сгруппировал их по таблицам в соответствии с постановкой. Далее файл с этими данными я передал команде разработчиков.

Завершающим этапом моей работы была презентация для итоговой защиты. Для её создания я использовал известный конструктор презентаций Canva. К сожалению, нашей стране ограничили доступ к этой платформе, поэтому мне пришлось воспользоваться VPN-сервисом. При создании презентации я опирался на тематику нашей курсовой работы, на стилистику нашего ВУЗа и современные тренды в создании подобного контента. Конечным результатом были довольны как члены команды, так и преподаватели из комиссии, которые слушали нашу защиту курсовой работы.

### **3 Коммуникация в команде**

В процессе работы над курсовой работой мне приходилось консультироваться с нашим руководителем, Ганичевой Лидией, и с ребятами, которые занимались разработкой технической части мобильного приложения. Один раз мы самостоятельно организовали созвон, так как Лидия была занята. Именно в этот момент я почувствовал себя руководителем, примерил эту роль на себя, чему безумно рад, ибо сам хочу добиться высот в работе и стать CEO или даже управляющим компании.

Для общих собраний мы использовали известную платформу Zoom, для быстрых диалогов предпочитали чат в Telegram, а если были какие-либо замечания по проделанной работе, Лидия оставляла комментарии прямо в файлах на Google диске команды.

Благодаря Лидии команда сплотилась, образовались хорошие взаимоотношения и дружеская атмосфера. Она делилась опытом аналитика и в целом опытом работы в IT-сфере, что очень ценно услышать и узнать на первых порах обучения в ИТМО. Искренне считаю, что наш руководитель достоин высокой оценки!

## **ЗАКЛЮЧЕНИЕ**

Наша команда успешно справилась с поставленной задачей. Мы смогли наладить процесс разработки так, что впоследствии реализовали мобильное Android приложение с функциями регистрации нового пользователя, входа в уже существующий аккаунт, просмотр ленты всех курсов в приложении, просмотр информации о курсе и просмотр видео-контента прямо в приложении.

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. И самой желанной функцией, которую мы хотели добавить в наше приложение, но не смогли - это возможность регистрироваться и входить в профиль по ITMO ID. Хотя эта функция не так уж и сложна в реализации, но данные о студентах ИТМО хранятся в строгой конфиденциальности и без согласования с управлением ВУЗа не распространяются. Но несмотря на то, что мы не всё успели реализовать, приложение получилось очень удобным, стильным, хорошо вписывающимся в идеологию и стилистику ИТМО.

Я считаю, что моя работа была достаточно полезной для остальной команды разработки, ведь благодаря постановке, которую я сделал в начале курсовой работы, ребята, занимающиеся серверной частью приложения, смогли легко реализовать базы данных и связь между ними. Бэк-энд разработчикам не пришлось продумывать логику взаимоотношения данных, ведь я сделал это за них. Также разработчикам не пришлось тратить время на прописывание тестовых данных, ведь этим аспектом я тоже занялся. Ну и наконец, горжусь, получившейся, презентацией. Она вышла и вправду стильной и симпатичной.



За время работы над мобильным приложением я познакомился с профессией системного аналитика, разобрался в теории баз данных, понял, как делать постановки на витрины данных и составлять ER-диаграммы. А самым ценным оказался, полученный, опыт работы в команде. До этого если и работал в команде, то совсем в короткие сроки и над маленькими проектами. Спасибо за возможность почувствовать себя настоящим разработчиком

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Статья на Хабр “Что такое База Данных (БД)” - <https://habr.com/ru/articles/555760/>
- 2 Duolingo - Company Information - <https://6sense.com/company/duolingo/5c3b0113d55ae49f1b7757e1#>
- 3 Вопрос на Хабр “Хочу понять каким способом пишутся элементы геймификации на сайте?” - <https://qna.habr.com/q/445392>
- 4 Статья на IEEE Spectrum “Как искусственный интеллект Duolingo понимает, что вам нужно выучить” - [https://spectrum-ieee-org.translate.goog/duolingo? x\\_tr\\_sl=en& x\\_tr\\_tl=ru& x\\_tr\\_hl=ru& x\\_tr\\_pto=rq #:~:text=Duolingo%20uses%20machine%20learning%20and,more%20than%2040%20different%20languages.](https://spectrum-ieee-org.translate.goog/duolingo? x_tr_sl=en& x_tr_tl=ru& x_tr_hl=ru& x_tr_pto=rq #:~:text=Duolingo%20uses%20machine%20learning%20and,more%20than%2040%20different%20languages.)
- 5 Статья на Блог Яндекс Практикума “Сущности и связи: как и для чего системные аналитики создают ER-диаграммы” - <https://practicum.yandex.ru/blog/что-такое-er-diagramma/>

## ПРИЛОЖЕНИЕ

### 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 базам 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 ресеч 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, заинсертировать 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:

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