МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Майкопский государственный гуманитарно-технический колледж

федерального государственного бюджетного образовательного учреждения высшего образования «Адыгейский государственный университет»

КУРСОВОЙ ПРОЕКТ

|  |
| --- |
| 09.02.07. ИС-42. 1091-А. КП |

|  |
| --- |
| *Разработка технического проекта приложения для учета занятости кабинетов в учебном заведении* |
|  |
|  |

|  |  |
| --- | --- |
|  | Работа защищена |
|  | с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | Руководитель проекта: \_\_\_\_\_\_ |
|  | Подпись |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г. |

**Студент** \_\_\_\_\_\_\_\_\_\_/Абдурахманов Т.К./

**Руководитель проекта** \_\_\_\_\_\_\_\_\_\_/Пелипенко Н.А./

2024

**СОДЕРЖАНИЕ**

Введение 4

1 Технический проект 7

1.1 Анализ предметной области 7

1.2 Постановка задачи 8

1.3 Проектирование функциональности web-приложения 8

1.4 Проектирование структуры программного продукта 8

1.5 Проектирование базы данных 8

1.6 Проектирование пользовательского интерфейса 8

1.7 Выбор средств разработки 8

1.8 Проектирование тестовых пакетов 8

1.9 Требования к программному обеспечению и техническим средствам 8

Заключение 32

Приложение А (справочное) Диаграмма вариантов использования 34

Приложение Б (справочное) Логическая модель БД 36

**ВВЕДЕНИЕ**

В современном мире организация образовательного процесса требует применения инновационных технологий и подходов. В колледжах очень часто возникает проблема с выделением определенных кабинетов для занятий. Разработка такого приложения позволит преподавателям оставлять заявку на занятие нужного кабинета на определенную дату и время, а диспетчеры уже одобрят или отклонят эту заявку. Кроме того, в приложении будут расписание всех занятых кабинетов и описание каждой из них. Приложение поможет университетам и колледжам значительно сократить время на организацию занятий и сделать процесс обучения более эффективным и удобным.

На данный момент в сфере управления аудиторными помещениями в образовательных учреждениях существуют несколько значительных противоречий, которые затрудняют эффективное использование ресурсов:

* ручное распределение кабинетов и координация действий преподавателей значительно усложняют процесс планирования и могут приводить к конфликтам в расписании;
* текущие методы управления занятостью кабинетов затрудняют точное отслеживание свободных пространств и их использования, тогда как автоматизация этого процесса через наш сайт значительно упрощает задачу;
* традиционное управление аудиторным фондом требует дополнительных ресурсов, таких как администраторы для обработки заявок и контроля расписаний, что увеличивает затраты и снижает общую эффективность работы учебного заведения.

Создание приложения для управления аудиторными помещениями в образовательных учреждениях является крайне актуальным шагом, так как она упростит и ускорит процесс планирования и распределения учебных помещений, снизит вероятность ошибок при учете и бронировании кабинетов, а также повысит удобство отслеживания занятости как для преподавателей, так и для диспетчеров. Данная система позволит централизовать информацию о свободных и занятых аудиториях, что, в свою очередь, улучшит координацию и повысит общую эффективность работы учебного заведения.

Использование приложения учета занятости кабинетов даст возможность организациям оптимизировать свои процессы благодаря интеграции с внутренними системами расписания и учета. Это обеспечит эффективное отслеживание статусов занятости помещений, а также необходимых ресурсов для проведения занятий. В результате, управление занятостью кабинетов станет более рациональным, что ускорит процессы взаимодействия и повысит качество предоставляемых образовательных услуг. Преподаватели смогут сосредоточиться на учебном процессе, а не на рутинных задачах, что в конечном итоге повысит общую эффективность работы и удовлетворенность как преподавателей, так и студентов.

* при разработке платформы для управления аудиторными помещениями в образовательных учреждениях следует выделить несколько ключевых проблем, которые необходимо учитывать для успешного внедрения и функционирования системы:
* понимание потребностей пользователей: какие функции и возможности преподаватели и администраторы хотели бы видеть в новой платформе? Как можно улучшить процесс бронирования и учета кабинетов?
* разработка интуитивно понятного интерфейса: как создать удобный интерфейс, доступный для пользователей с разным уровнем технической компетенции? какие элементы дизайна и навигации будут наиболее эффективными для упрощения взаимодействия в условиях учебного процесса?
* интеграция с существующими системами управления: как обеспечить бесшовную интеграцию с текущими системами учета и расписания, минимизируя при этом возможные сбои в работе?
* управление данными и безопасность: какие меры безопасности необходимо внедрить для гарантии конфиденциальности и защиты информации о учебных процессах и пользователях платформы?
* тестирование и отладка: какие тесты необходимо выполнить на различных этапах разработки платформы? Какие будут сценарии использования приложения в реальных условиях учебного заведения?
* адаптивность платформы для различных устройств (мобильные телефоны, планшеты, ПК) и масштабируемость для будущих расширений функционала: как заложить такие характеристики в архитектуре приложения, чтобы обеспечить максимальную эффективность и доступность для всех пользователей?

Каждый из этих аспектов требует детального исследования и тщательной проработки для успешного внедрения приложения для учета занятости кабинетов, чтобы обеспечить его эффективность, удобство и безопасность.

Объект исследования – процесс разработки приложения для оптимизации управления занятостью кабинетов в колледже.

Предмет исследования – программное обеспечение учет занятости кабинетов, предназначенный для автоматизации процесса бронирования кабинетов преподавателями.

Цель курсового проекта: разработать функциональный и удобный web-сайт, который позволит преподавателям эффективно управлять занятостью кабинетов и упрощать процесс подачи заявок на бронирование.

Задачи:

* провести исследование для определения потребностей и предпочтений целевой аудитории (как преподавателей, так и диспетчеров);
* спроектировать структуру и функциональность приложения, выбрать технологии и инструменты для разработки;
* реализовать разработанный сайт с интуитивно понятным интерфейсом.

Гипотеза исследования: автоматизация процесса бронирования кабинетов с помощью приложения для учета занятости кабинетов с удобным интерфейсом позволит преподавателям быстро и просто оставлять заявки на занятие кабинетов, сократит время на ожидание одобрения и повысит общую эффективность организации учебного процесса.

Для изучения проблем разработки приложения учета занятости кабинетов будут применяться следующие методы исследования:

* опросы и анкетирование: проведение опросов среди преподавателей и диспетчеров колледжа с целью сбора информации о их потребностях, предпочтениях и текущих проблемах в процессе бронирования кабинетов;
* глубинные интервью с экспертами: получение более глубокого понимания опыта и ожиданий пользователей через интервью с преподавателями и административным персоналом, работающим с занятостью кабинетов;
* анализ конкурентов: изучение существующих решений для управления занятостью кабинетов, анализ их функциональности и пользовательского интерфейса для выявления лучших практик и возможных недостатков;
* пользовательское тестирование: создание прототипов приложения и их тестирование на целевой аудитории для получения обратной связи о удобстве использования и функциональности на ранних этапах разработки;
* моделирование сценариев использования: разработка сценариев, описывающих разные пути взаимодействия пользователей с приложением, что позволит выявить возможные проблемы и улучшить интерфейс;
* кросс-дисциплинарные исследования: вовлечение специалистов из различных областей (дизайнеров, разработчиков, методистов) для обеспечения комплексного подхода к разработке приложения;
* объектно-ориентированный анализ: проектирование архитектуры и функциональности приложения, построение моделей поведения системы;
* анализ современного инструментария для разработки программного обеспечения: изучение актуальных инструментов и технологий, которые могут быть использованы при создании приложения;
* методы математической статистики: использование статистических методов для анализа данных с целью выявления закономерностей в потребностях пользователей;
* алгоритмы применения инструментальных средств: разработка алгоритмов и методик применения инструментальных средств на различных этапах разработки приложения.

Практическая значимость курсового проекта заключается в возможности использования приложения для учета занятости кабинетов в учебном процессе, что значительно повысит эффективность управления распределением кабинетов как для преподавателей, так и для административного персонала.

Основными пользователями разработанного приложения станут преподаватели и диспетчеры, которые занимаются организацией учебного процесса в колледже.

Аспекты проектирования приложения будут рассмотрены в первом разделе дипломного проекта.

Результаты практической реализации спроектированного приложения будут представлены во втором разделе дипломного проекта.

# **1 ТЕХНИЧЕСКИЙ ПРОЕКТ**

## Анализ предметной области

В современном образовательном процессе эффективное управление ресурсами, такими как учебные аудитории, становится важным аспектом для обеспечения качественного обучения. В колледже, где существует множество дисциплин и преподавателей, необходимость в оптимизации использования учебных кабинетов становится особенно актуальной. Преподаватели часто сталкиваются с проблемой недостатка информации о доступности кабинетов, что может привести к конфликтам и неэффективному использованию ресурсов.

Приложение учет занятости кабинетов позволяет решить эту проблему, предоставляя пользователям удобный интерфейс для просмотра расписания, подачи заявок на использование кабинетов и отслеживания статуса этих заявок. Аудитории, как ключевые ресурсы образовательного процесса, должны быть доступны для преподавателей в соответствии с их потребностями и расписанием занятий.

Некоторые из этих преимуществ включают:

* доступность и удобство использования – сайт доступен для преподавателей и диспетчеров в любое время и на любом устройстве. Он имеет интуитивно понятный интерфейс и простую навигацию, что облегчает процесс бронирования кабинетов;
* поиск по расписанию и заявкам – удобная функция поиска позволяет быстро находить нужные расписания занятий или заявки, что экономит время и улучшает доступ к информации;
* широкая аудитория – сайт будет полезен для всех преподавателей и диспетчеров колледжа, независимо от их уровня технической подготовки. Это делает его универсальным инструментом для упрощения процесса распределения кабинетов.  
    
  Приложение призвано решать ряд актуальных задач, таких как:
* упрощение коммуникации – обеспечивает быструю и прямую связь между преподавателями и диспетчерами, позволяя обмениваться сообщениями и оставлять заявки на аудитории;
* сокращение времени на поиск информации – пользователи могут быстро находить необходимые расписания занятий и информацию о кабинетах, а также отслеживать статус своих заявок;
* организация использования кабинетов – поддерживает систему бронирования кабинетов, что улучшает взаимодействие между преподавателями и диспетчерами;
* улучшение корпоративной культуры – способствует более прозрачному и доступному взаимодействию между преподавателями и административным персоналом, что укрепляет командный дух и повышает уровень вовлеченности;
* обратная связь и одобрение заявок – позволяет диспетчерам быстро реагировать на заявки преподавателей, обеспечивая более эффективное распределение ресурсов;
* поддержка удаленной работы и гибкости – обеспечивает связь и доступ к информации для преподавателей в различных условиях работы, что делает процесс бронирования более удобным для всех участников.

## Постановка задачи

Необходимо разработать web-приложение для образовательного учреждения.

Приложение должно будет содержать все необходимые сведения об образовательном учреждении, иметь удобный и интуитивно понятный пользователю графический интерфейс

Приложение необходимо разработать с использованием трехуровневой архитектуры. Для бизнес-логики необходимо будет разработать API.

Web-приложение должна обеспечивать выполнение следующих функций:

* Авторизация пользователей, имеющих доступ к платформе;
* предоставление в главном окне основной информации: расписание пар, а также занятость кабинетов;
* возможность загружать расписание пар;
* возможность преподавателю зайти в свой профиль;
* возможность подачи заявки диспетчеру на бронирование (освобождение) кабинета на определенную пару и дату;
* отображение списка кабинетов с их описанием, включая характеристики (вместимость, наличие мультимедийного оборудования и т.д.);
* возможность отслеживания статуса поданных заявок (ожидает одобрения, одобрена, отклонена) в специальной вкладке «Заявки»;
* функция фильтрации и поиска по аудиториям для удобства выбора подходящего кабинета;
* хранение материалов в БД.

Программный продукт должен включать в себя:

* API для бизнес-логики;
* дизайнерскую часть, которая обеспечит удобную работу с сайтом, обеспечит удобный и интуитивно понятный интерфейс для навигации по сайту;
* Web-приложение для управления портала.

Требования по входным данным:

* запросы пользователей – отправка заявки на бронирование кабинета;
* данные диспетчера – включает информацию, необходимую для управления данными приложения.
* данные аутентификации и учетной записи – включают информацию, предоставляемую администраторами входе в приложение. Эти данные включают идентификационную информацию, такую как логин, имя пользователя и пароль.

Требования по выходным данным:

* заявки пользователей – выходные данные представляют собой заявки, отправленные преподавателями на использование кабинетов. Эти данные могут содержать информацию о дате, времени и названии кабинета;
* результаты поиска – включают результаты поиска по расписанию пар и списку кабинетов, которые соответствуют запросам пользователей. Эти данные содержат информацию о доступных аудиториях и их расписании;
* интерфейс диспетчера – выходные данные для диспетчеров включают интерфейс управления заявками и расписанием, который позволяет им одобрять или отклонять заявки, а также добавлять и редактировать информацию о кабинетах;
* данные ответов на запросы от API – предоставляют внешним приложениям доступ к информации о расписании занятий и статусе заявок. Эти данные должны быть в формате JSON и могут содержать информацию о свободных и занятых кабинетах;
* данные учетных записей – включают информацию о профилях преподавателей и диспетчеров, их настройках и правах доступа;
* уведомления и сообщения – включают уведомления для пользователей о статусе их заявок, подтверждении бронирования кабинетов, изменениях в расписании и ошибках системы.

Запросы должны обрабатываться через API и передаваться внешним приложениям в формате JSON.

Хранение и передача данных должны осуществляться в безопасной и защищенной форме. Доступ к данным для администраторов должен быть строго контролируемым с использованием аутентификации. Информация администраторов должна храниться в зашифрованном виде, и необходимо принять меры для обеспечения их конфиденциальности, чтобы предотвратить несанкционированный доступ к правам администратора.

**1.3 Проектирование функциональности web-приложения**

Для описания взаимодействия преподавателей и диспетчера с приложением «Учет занятости кабинетов» воспользуемся методологией UML.

Рассмотрим на диаграмме вариантов использования взаимодействие пользователей с разрабатываемым приложением. Для этого построим диаграмму Use Case, в которой имеются:

* актеры («Преподаватель», «Диспетчер», «API», «DB»);
* прецеденты («Авторизация», «Отправка заявки», «Получение заявки», «Одобрение заявки», «Отклонение заявки», «Обновление информации о пользователе», «Обработка запроса API», «Обработка запроса БД»).

Диаграмма вариантов использования представлена в приложении А.

Рассмотрим ключевые моменты диаграммы:

* Пользователь может выполнять прецеденты «Авторизация», «Отправка сообщения», «Получение сообщения», «Создание группового чата»;
* «Администратор» может выполнять прецеденты «Добавление участников в чат», «Удаление участников из чата» и также имеет доступ к прецедентам «Авторизация», «Отправка сообщения», «Получение сообщения», «Создание группового чата»;
* API обрабатывает запросы от актеров и взаимодействует с БД;
* DB обрабатывает запросы от API и доступ к данным.

Все прецеденты выполняются после прецедента «Авторизация», что обеспечивает контроль доступа к управлению участниками чатов.

Прецедент «Обработка запроса API» проверяет роль пользователя, а затем выполняет запрос. «Обработка запроса API» включает в себя прецедент «Обработка запроса БД», который выполняется при выполнении условий «Обработка запроса API» и получает необходимые данные из базы данных для удовлетворения запроса пользователя.

**1.4 Проектирование структуры программного продукта**

С целью удостоверения предъявляемых к программе требований, приведенных в разделах 1.1-1.2, при реализации портала будет использоваться трехзвенная архитектура. Данная архитектура представляет собой организацию приложения на трех уровнях, каждый из которых выполняет определенные функции:

* представление этот уровень отвечает за отображение пользовательского интерфейса и взаимодействие с пользователем. Он включает компоненты, которые занимаются визуализацией данных и обработкой действий пользователей. В этот уровень будет входить веб-приложение;
* логика приложения – на этом уровне осуществляется обработка бизнес-логики, связанной с функциональностью системы управления кабинетами. Здесь располагаются компоненты, отвечающие за обработку пользовательских запросов, проверку прав доступа, обработку данных и взаимодействие с базой данных;
* слой данных – уровень отвечает за хранение и доступ к данным. Здесь находится БД, содержащая информацию, и компоненты, реализующие операции чтения и записи данных. На портале этот слой будет представлять СУБД и компоненты, работающие с ней для получения и записи данных.

СХЕМА РАЗВЕРТЫВАНИЯ

Разрабатываемое Web-приложение будет состоять из следующих составляющих:

* аутентификация и авторизация – форма входа для аутентификации администраторов с разделением прав доступа;
* главная страница – центральная страница, на которой пользователи могут просматривать расписание пар и занятость аудиторий;
* страница список кабинетов – на этой странице находится список кабинетов с их характеристиками;
* страница занять кабинет – страница, на которой можно отправлять заявку на бронирование кабинета;
* страница заявок – страница, на которой преподаватели будут видеть свои заявки, а также их состояние (одобрено, отклонено, на рассмотрении), а диспетчер на этой странице может одобрять или отклонять эти заявки;
* страница пользователя – на этой странице можно изменять некоторые данные о пользователе.

Тестирование программ планируется проводить путём выбора каждого пункта меню и проверки корректности получаемого отклика от программы. При тестировании будет производиться эмуляция различных вариантов пользовательских действий.

Функциональные модули веб-приложения

**1.5 Проектирование базы данных**

Для хранения данных о сообщениях будет разработана БД. Она будет состоять из следующих таблиц:

* User (user\_id, password, surname, name, lastname, role\_id);
* Role (role\_id, role\_name);
* Class (class\_id, user\_id, lesson\_number, subject\_id, date);
* Auditorium (auditorium\_id, number, description, computers, interactive\_whiteboard);
* Subject (subject\_id, name, description);
* Request (requests \_id, user\_id, auditorium\_id, class\_id, request\_date, status, message);
* Teachers\_and\_subjects (user\_id, subject\_id); // для хранения статуса уведомлений о новых сообщениях.

Логическая модель БД представлена в приложении Б.

Ниже приведён словарь данных:

Таблица 1 – Таблица «Пользователь» (User)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Столбец** | **Тип** | **Атрибуты** | **Null** | **Дополнительно** |
| user\_id | bigint(20) | UNSIGNED | Нет | auto\_increment |
| login | varchar(255) |  | Нет | UNIQUE |
| password | varchar(255) |  | Нет |  |
| surname | varchar(50) |  | Нет |  |
| name | varchar(50) |  | Нет |  |
| lastname | varchar(50) |  | Да |  |
| role\_id | int(11) | UNSIGNED | Нет | FK (ссылается на Роль.role\_id) |

Таблица 2 – Таблица «Роль» (Role)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Столбец** | **Тип** | **Атрибуты** | **Null** | **Дополнительно** |
| role\_id | bigint(20) | UNSIGNED | Нет | auto\_increment |
| role\_name | varchar(255) |  | Нет | UNIQUE |

Таблица 3 – Таблица «Пара» (Class)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Столбец** | **Тип** | **Атрибуты** | **Null** | **Дополнительно** |
| class\_id | bigint(20) | UNSIGNED | Нет | auto\_increment |
| user\_id | int(11) |  | Нет | FK (ссылается на Пользователь.user\_id) |
| lesson\_number | bigint(20) |  | Нет |  |
| surname | varchar(50) |  | Нет |  |
| name | varchar(50) |  | Нет |  |
| lastname | varchar(50) |  | Да |  |
| role\_id | int(11) | UNSIGNED | Нет | FK (ссылается на Роль.role\_id) |

**ПРИЛОЖЕНИЕ А**

(Диаграмма вариантов использования**)**

**ПРИЛОЖЕНИЕ Б**

(Логическая модель БД)

