Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный индустриальный университет»

Кафедра прикладных информационных технологий и программирования

**КУРСОВОЙ ПРОЕКТ**

по дисциплине «Проектная деятельность 4»

на тему «Разработка веб-сервиса для старост по учету посещаемости студентов»

Выполнил (а):

Обучающийся гр. ИСП-20

(аббревиатура группы)

12.04.2023 \_\_\_\_\_\_\_\_ Токмаков А. К.

(дата) (подпись) (инициалы, фамилия)

Руководитель курсового проекта:

Преподаватель Белавенцева Д. Ю.

(уч. степень, звание, инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (дата) (подпись)

Новокузнецк

2024г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Сибирский государственный индустриальный университет»

Кафедра прикладных информационных технологий и программирования

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_\_\_\_ Рыбенко И. А.

(подпись) (инициалы, фамилия)

« 01 » сентября 2023г.

**ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА**

по дисциплине «Проектная деятельность 4»

на тему «Разработка веб-сервиса для старост по учету посещаемости студентов»

обучающегося: Токмакова Антона Константиновича

1. Согласно структуре курсового проекта, произвести инициацию и планирование проекта.

2. Провести анализ предметной области учета пропусков студентов.

3. Разработать back-end для веб-сервиса по учету посещаемости студентов для старост.

4. Разработать front-end для веб-сервиса по учету посещаемости студентов для старост.

5. Оформить отчёт по курсовому проектированию, подготовить презентацию по результатам проектирования и доклад.

Задание к курсовому проектированию принял:

«01» сентября 2023г. Токмаков А.К.

Руководитель курсового проекта \_\_\_\_\_\_\_\_ Белавенцева Д.Ю.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc156744278)

[1. Инициация и планирование проекта 6](#_Toc156744279)

[1.1. Устав проекта 6](#_Toc156744280)

[1.2. Календарное планирование 10](#_Toc156744281)

[1.3. Выбор и обоснование инструментов решения поставленных задач 13](#_Toc156744282)

[1.3.1. Выбор баз данных 13](#_Toc156744283)

[1.3.2. Выбор средства разработки интерфейса 15](#_Toc156744284)

[1.3.3. Выбор языка программирования 19](#_Toc156744285)

[1.4. Техническое задание 22](#_Toc156744286)

[1.4.1. Общие сведения 22](#_Toc156744287)

[1.4.2. Назначение и цели создания системы 23](#_Toc156744288)

[1.4.3. Характеристика объекта автоматизации 24](#_Toc156744289)

[1.4.4. Требования к автоматизированной системе 24](#_Toc156744290)

[1.4.5. Порядок разработки автоматизированной системы 31](#_Toc156744291)

[1.4.6. Порядок контроля и приемки автоматизированной системы 34](#_Toc156744292)

[1.4.7. Требования к подготовке объекта автоматизации к вводу АС в действие 35](#_Toc156744293)

[1.4.8. Требования к документированию 36](#_Toc156744294)

[1.4.9. Источники разработки 37](#_Toc156744295)

**ВВЕДЕНИЕ**

Целью проекта «Веб-сервис для старост по учету посещаемости студентов» является упростить и тем самым ускорить работу старосты и куратора группы по составлению аттестационной ведомости студентов группы. В настоящее время учет посещаемости студентов ведется через бумажный журнал, что может привести к его потере и полной потере информации о посещаемости студентов группы. Для предоставления отчета или передачи справки, старосте группы и классному руководителю необходимо каждый раз сверять свое расписание, чтобы назначить встречу для передачи информации о пропусках студентов или справке, далее информацию из отчета по пропускам переносится в специальный бланк, выданный дирекцией, только после этого он пойдет на проверку к заместителю директора, который будет проверять бланк на корректность подсчетов и ошибки человеческого фактора.

Используя информационную систему учета посещаемости студентов, можно избежать проблем, связанных с бумажным журналом, таких как потеря журнала, ошибки в подсчетах и т.д. Веб-сервис позволит старостам вносить данные об отсутствующих студентах на каждой паре, а затем формировать отчеты по посещаемости студентов, которые можно посмотреть в любое время или распечатать готовый аттестационный лист. Старосты смогут прикреплять справки об уважительном отсутствии студентов в информационную систему, после чего справка попадет к классному руководителю.

Веб-сервис для учета посещаемости студентов в образовательных учреждениях должен выполнять следующие функции:

* Внесение данных об отсутствующих студентах на каждой паре.
* Хранение данных о посещаемости студентов.
* Формирование отчетов по посещаемости студентов.
* Прикрепление справок об уважительном отсутствии студентов в информационную систему.

Использование информационной системы для учета посещаемости студентов в образовательных учреждениях позволит:

* Избежать проблем, связанных с бумажным журналом.
* Сократить время, затрачиваемое на составление учета посещаемости студентов старостами группы.
* Автоматизировать работу куратора группы, старосты и заместителя директора в учебных заведениях.
* Улучшить качество учета посещаемости студентов

Таким образом, проект «Веб-сервис для старост по учету посещаемости студентов» позволяет автоматизировать бизнес-процессы учета посещаемости студентов, что повышает эффективность работы старост, кураторов групп и заместителя директора в учебных заведениях.

1. **Инициация и планирование проекта**
   1. **Устав проекта**

Наименование проекта: Разработка веб-сервиса для старост по учету посещаемости студентов.

Краткое наименование: веб-сервис по учет посещаемости студентов.

Дата начала проекта: 01.09.2023

Дата окончания проекта: 12.04.2024

Цель проекта высшего уровня –упростить работу старосты и куратора группы по составлению аттестационной ведомости студентов группы.

Подцели:

1. Сократить время формирование аттестационной ведомости;
2. Предотвратить лишние встречи куратора группы и старосты;

Задачи проекта

1. Анализ процесса учета посещаемости студентов в колледже СибГИУ;
2. Составить документацию по выбору и обоснованию инструментов, используемых в проекте;
3. Разработать техническое задание для проекта;
4. Разработать информационное, алгоритмическое, программное и техническое обеспечение проекта на разработку веб-сервиса по учету посещаемости студентов;
5. Спроектировать и реализовать базу данных для проекта веб-сервиса по учету посещаемости старост;
6. Разработать back-end веб-сервиса по учету посещаемости студентов;
7. Разработать front-end веб-сервиса по учету посещаемости студентов;
8. Протестировать веб-сервис по учету посещаемости студентов на конечный пользователях.

Основные результаты проекта

* Разработана база данных для веб-сервиса по учету и посещаемости студентов;
* Разработанный back-end веб-сервиса по учету посещаемости студентов;
* Разработанный front-end веб-сервиса по учету посещаемости студентов

Продукт проекта

Продукт проекта - разработанный веб-сервис для старост по учету посещаемости студентов.

Этот веб-сервис позволяет старостам вести учет посещаемости студентов в колледже. С помощью этого сервиса старосты могут легко отслеживать посещаемость студентов и не носить с собой журнал посещаемости. Если студент отсутствует по уважительной причине, то староста может загрузить справку, предоставленную студентов, после веб-сервис сам проставит УП в дни, указанные в справке.

После прошествия дня, староста не может редактировать присутствующих на паре, это можно сделать, только через запрос куратору группы, который должен разрешить изменения.

Веб-сервис также может предоставлять отчеты о посещаемости студентов в нужном для дирекции виде, так же веб-сервис может быть использованы для анализа и улучшения учебного процесса.

Продукт проекта может быть полезен для старост, кураторов групп и заместителей директора в колледже СибГИУ.

Ограничения проекта

1. Ограниченные финансовые ресурсы на разработку веб-сервиса по учету посещаемости студентов, выделено 0 рублей;
2. Ограниченное количество времени на реализацию проекта, выделено 7 месяцев;

Риски проекта

* Возможны срывы сроков выполнения заданий по проекту из-за неправильного планирования и управления проектом. Решение - правильное планирование, использование современных методов управления проектами и контроль за выполнением работ.
* Недостаток времени на реализацию поставленной задачи из-за плотного графика. Решение - правильное планирование времени.
* Недостаток знаний для реализации поставленных задач. Решение - помощь одногруппников, интернет-курсы.
* Нестабильная работа информационной системы. Решение - покрытие информационной системы различными тестами.
* Аппаратные сбои, утеря важных данных по проекту. Решение - использование системы контроля версий и облачных хранилищ, которые обслуживаются крупными и надежными компаниями, для минимизации риска.
* Риск неправильной работы с данными. Решение - обеспечение безопасности данных и регулярное резервное копирование данных.
* Другие риски, которые невозможно предсказать. Решение - планирование сроков сдачи проекта заранее, чтобы учесть возможные риски.

Заинтересованные стороны проекта

Таблица 1 – Заинтересованные стороны проекта

|  |  |  |
| --- | --- | --- |
| №  п/п | Заинтересованная сторона | Интерес |
| 1 | Разработчик проекта | 1. Веб-сервис внедрен в колледж СибГИУ; 2. Обеспечение удобного интерфейса для пользователей; 3. Получение новых навыков по разработке веб-сервисов; 4. Получение подробной информации по работе предметной области; 5. Решение существующей проблемы. |

*Продолжение таблицы 1*

|  |  |  |
| --- | --- | --- |
| 2 | Заказчик (СибГИУ) | 1. Автоматизирована работа по вопросам посещаемости студентов; 2. Повышена эффективность работы преподавателей и старосты по вопросам посещаемости студентов; 3. Перевод от бумажного учета посещаемости студентов к электронной посещаемости; 4. Повышение мотивации студентов к посещаемости; |
| 3 | Пользователь | 1. Более интуитивный и понятный интерфейс, чем в существующих аналогов; 2. Сокращение рутинной работы; 3. Быстрая печать отчета. |

Ресурсы проекта

1. Временные ресурсы: 225 календарных дней;
2. Материальный ресурсы: ноутбук, сеть интернет, ProjectLibre, MS Office, браузер, IDE InteliJ IDEA, СУБД PostgreSQL, хостинг для сайта;

Последовательность работ по проекту

Таблица 2 – Последовательность работ по проекту

|  |  |
| --- | --- |
| № п/п | Этапы проекта |
|  | Инициализация проекта. Разработка устава проекта |
|  | Планирование проекта |
|  | Определение последовательности работ |
|  | Составления диаграммы Ганта |
|  | Реализация проектов |
|  | Анализ предметной области |

*Продолжение таблицы 2*

|  |  |
| --- | --- |
|  | Выбор программных и технических средств |
|  | Обоснование выбора программных и технических средств |
|  | Техническое задание |
|  | Разработка базы данных для проекта |
|  | Реализация базы данных в СУБД PostgreSQL |
|  | Демо версия front-end веб-сервиса |
|  | Реализация back-end веб-сервиса |
|  | Тестирование |
|  | Реализация front-end веб-сервиса |
|  | Тестирование |
|  | Описание по видам обеспечения проекта |
|  | Анализ и доработка |
|  | Завершение проекта |
|  | Создание отчета и презентации для защиты |
|  | Демонстрация и защита проекта |

Команда проекта

Заказчик: СибГИУ

Научный руководитель проекта: Белавенцева Дарья Юрьевна

Разработчик проекта: Токмаков Антон Константинович

* 1. **Календарное планирование**

Календарное планирование является важным этапом в реализации проекта. Оно позволяет определить цели, задачи, ресурсы и сроки выполнения проекта, что помогает управлять временем и ресурсами, а также контролировать прогресс и достигать поставленных целей.

Для планирования проекта можно использовать различные инструменты, такие как диаграмма Ганта, которая позволяет визуализировать все задачи и сроки их выполнения (рисунок 1). Также можно использовать современные методы управления проектами, которые помогают управлять проектом и контролировать его прогресс.

Для успешной реализации проекта необходимо определить все задачи, необходимые для выполнения, чтобы достичь поставленных целей, а также определить время на реализацию каждой задачи. На рисунке 2 изображен список задач, который является важным этапом в реализации проекта, позволяющий определить все задачи, необходимые для выполнения, чтобы достичь поставленных целей, а также определить время на реализацию каждой задачи.

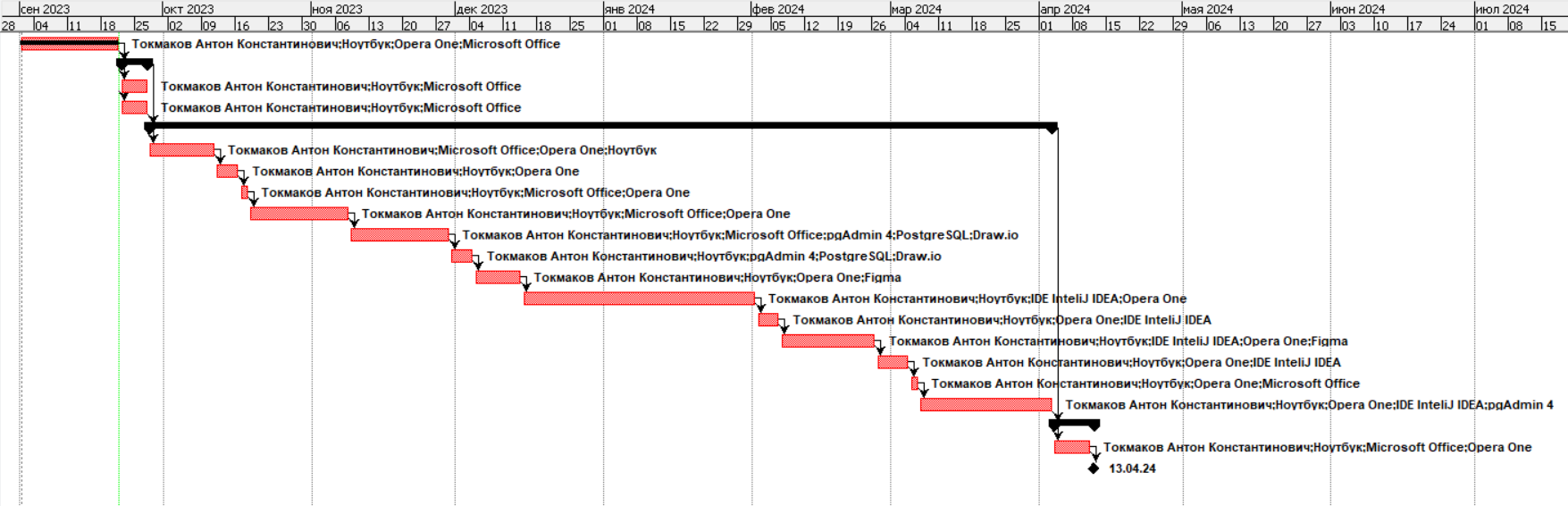


Рисунок 1 – Диаграмма Ганта проекта

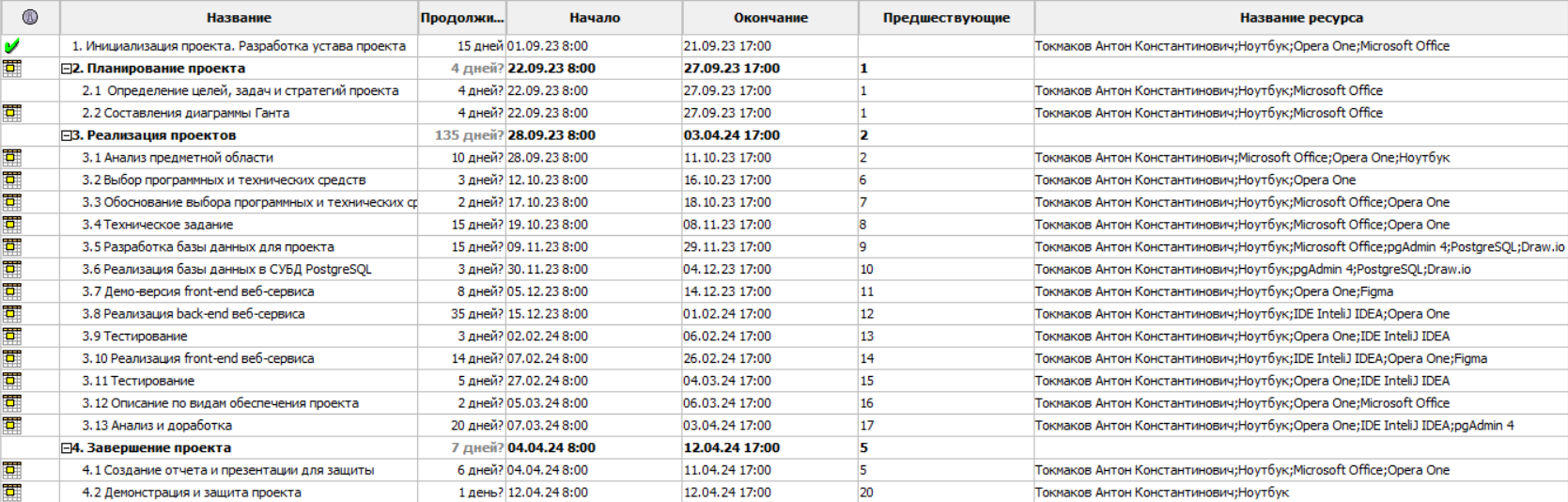


Рисунок 2 – Список задач для разработки проекта

* 1. **Выбор и обоснование инструментов решения поставленных задач**
     1. **Выбор баз данных**

В выборе БД было выделено 3 базы данных:

* Postgresql
* Oracle
* Microsoft SQL Server

PostgreSQL – это мощная объектно-реляционная система управления базами данных (СУБД), которая отличается открытым исходным кодом и активным сообществом разработчиков. Она предлагает расширенные возможности, включая поддержку сложных запросов, транзакций, хранимых процедур, репликации данных и многих других функций. PostgreSQL широко используется для разработки приложений, веб-сервисов, аналитики данных и других задач, где требуется надежное хранение и обработка информации.

Oracle Database – это одна из самых популярных коммерческих систем управления базами данных, разработанная компанией Oracle. Она предлагает широкий спектр возможностей, включая высокую производительность, расширенные возможности аналитики, масштабируемость, безопасность данных и многое другое. Oracle Database широко используется в корпоративной среде для хранения и обработки критически важных данных.

Microsoft SQL Server – это система управления базами данных, разработанная компанией Microsoft. Она предлагает широкий спектр функций, включая поддержку транзакций, хранимых процедур, аналитики данных, интеграции с другими продуктами Microsoft и многое другое. SQL Server широко используется в корпоративной среде, особенно в средах, где используются другие продукты Microsoft, такие как Windows Server и .NET Framework.

В таблице 3 расписаны характеристики для сравнения разных систем управления базами данных.

Таблица 3 – Сравнение СУБД

|  |  |  |  |
| --- | --- | --- | --- |
|  | Postgresql | Oracle | Microsoft SQL Server |
| Модель хранения данных | Объектно-реляционная | Объектно-реляционная | Реляционная |
| Кроссплатформенность | Да | Да | Да |
| Поддержка XML | Да | Да | Да |
| Подготовка специалистов | Легко | Сложно | Легко |
| Разработчик | PostgreSQL Global Development Group | Oracle | Microsoft |
| Языки SQL | PL and PG SQL | PL SQL | SQL, T-SQL |
| Тип лицензии | Открытый исходный код | Коммерческая | Коммерческая |
| Место в РСУБД | №4 | №1 | №2 |
| Текущий выпуск | 15.2, февраль 2023 г | 21c, январь 2021 г | 4 ноября 2019 г |
| Надежность | Высокая | Высокая | Высокая |
| Страна | Россия | США | США |
| Инструменты администрирования | pgAdmin4, pg\_archivecleanup, initdb и другие | Oracle Enterprise Manager, Oracle Wallet Manager | MS SQL Server |

Для выбранной предметной области лучшим решением будет использование PostgreSQL, так как эта СУБД отлично покрывает все требования к веб-сервису для старост по учету посещаемости студентов. Так же она отлично подходит для веб-сервиса такого вида, которые не оказывают высокой нагрузки на базу данных и не требуется функционал как у базы данных от компании Oracle. PostgreSQL является открытым программным обеспечением, что положительно влияет на его сравнение с MS SQL, следовательно данную СУБД можно изменять под потребности бизнеса.

PostgreSQL имеет несколько преимуществ для выбранной предметной области:

Удобство использования: PostgreSQL имеет простой и интуитивно понятный интерфейс, что делает его удобным для использования как для опытных разработчиков, так и для новичков. Он также поддерживает широкий спектр языков программирования и платформ, что позволяет разработчикам выбирать наиболее подходящие инструменты для своих проектов.

Открытость и гибкость: PostgreSQL является открытым программным обеспечением, что означает, что его исходный код доступен для всех и может быть изменен и адаптирован под конкретные потребности бизнеса. Это также означает, что PostgreSQL имеет большое сообщество разработчиков, которые работают над его улучшением и поддержкой.

Таким образом, PostgreSQL является удобной и гибкой системой управления базами данных для выбранной предметной области, что делает его хорошим выбором для веб-сервиса для учета посещаемости студентов. Кроме того, PostgreSQL является российской СУБД, поэтому поддерживая курс на импортозамещение ПО мы выбрали именно эту СУБД.

* + 1. **Выбор средства разработки интерфейса**

Для разработки интерфейса приложения будет использована одна из нижеперечисленных программ:

– Figma;

– Adobe Photoshop;

– CoreIDRAW;

Figma – веб-сервис для дизайна. С помощью нее можно создавать не только прототипы, но и сами интерфейсы (сайты, приложения, панель управления). Весомым преимуществом данной программы является ее простота.

На рисунке 3 представлен интерфейс данного веб-сервиса.

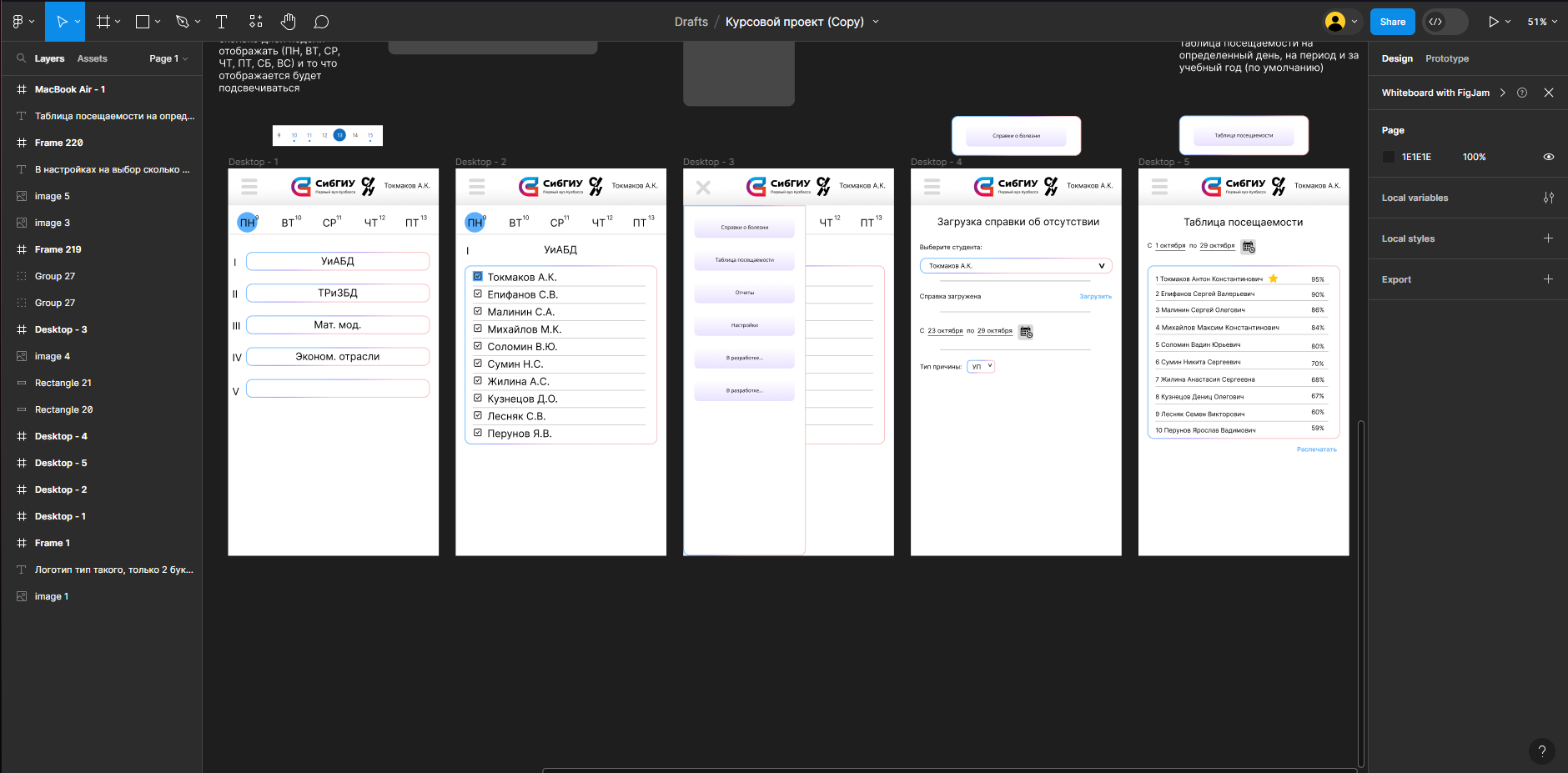


Рисунок 3 – интерфейс figma

Photoshop - это мощный графический редактор, который позволяет создавать и редактировать изображения любой сложности. Он был разработан компанией Adobe Systems и является одним из наиболее популярных и распространенных инструментов для работы с растровыми изображениями.

Photoshop предоставляет широкий спектр инструментов и функций, которые позволяют создавать и редактировать изображения, включая ретуширование, наложение эффектов, коррекцию цвета и яркости, работу с масками и слоями, создание графических элементов и многое другое.

Одной из главных особенностей Photoshop является его способность работать с большим количеством форматов файлов, включая JPEG, PNG, GIF, TIFF, PSD и многие другие. Это позволяет пользователям импортировать и экспортировать изображения в различных форматах, что делает Photoshop универсальным инструментом для работы с графикой.

Интерфейс Photoshop интуитивно понятен и легок в использовании, что делает его доступным для пользователей с различным уровнем опыта. Он состоит из панелей инструментов, меню и панелей настроек, которые можно настроить в соответствии с потребностями пользователя.

На рисунке 4 изображен интерфейс photoshop.

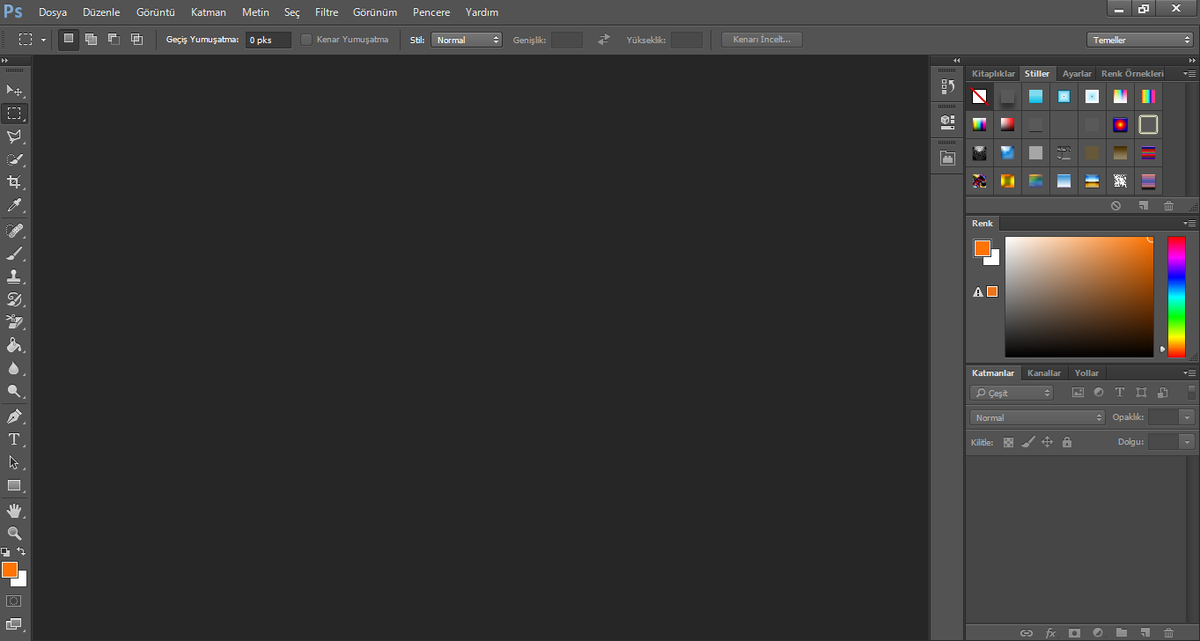


Рисунок 4 – интерфейс Photoshop

CoreIDRAW – графический редактор векторной графики, разработанный канадской корпорацией Corel. В пакет CorelDRAW также входит редактор растровой графики Corel. Photo-Paint и другие программы – например, для захвата изображений с экрана – Corel Capture. Программа векторизации растровой графики Corel Trace до 12 версии входила в пакет как самостоятельная программа.

В состав пакета CorelDRAW также входит редактор растровой графики Corel Photo-Paint, который обеспечивает возможности для редактирования растровых изображений, а также другие программы, такие как Corel Capture, предназначенная для захвата изображений с экрана. Кроме того, до 12 версии в пакет входила программа векторизации растровой графики Corel Trace, которая позволяла преобразовывать растровые изображения в векторный формат.

Интерфейс CorelDRAW является интуитивно понятным и удобным в использовании. Он включает в себя панели инструментов, меню и настраиваемые панели, что делает работу с программой удобной и эффективной.

На рисунке 5 изображен интерфейс данной программы.

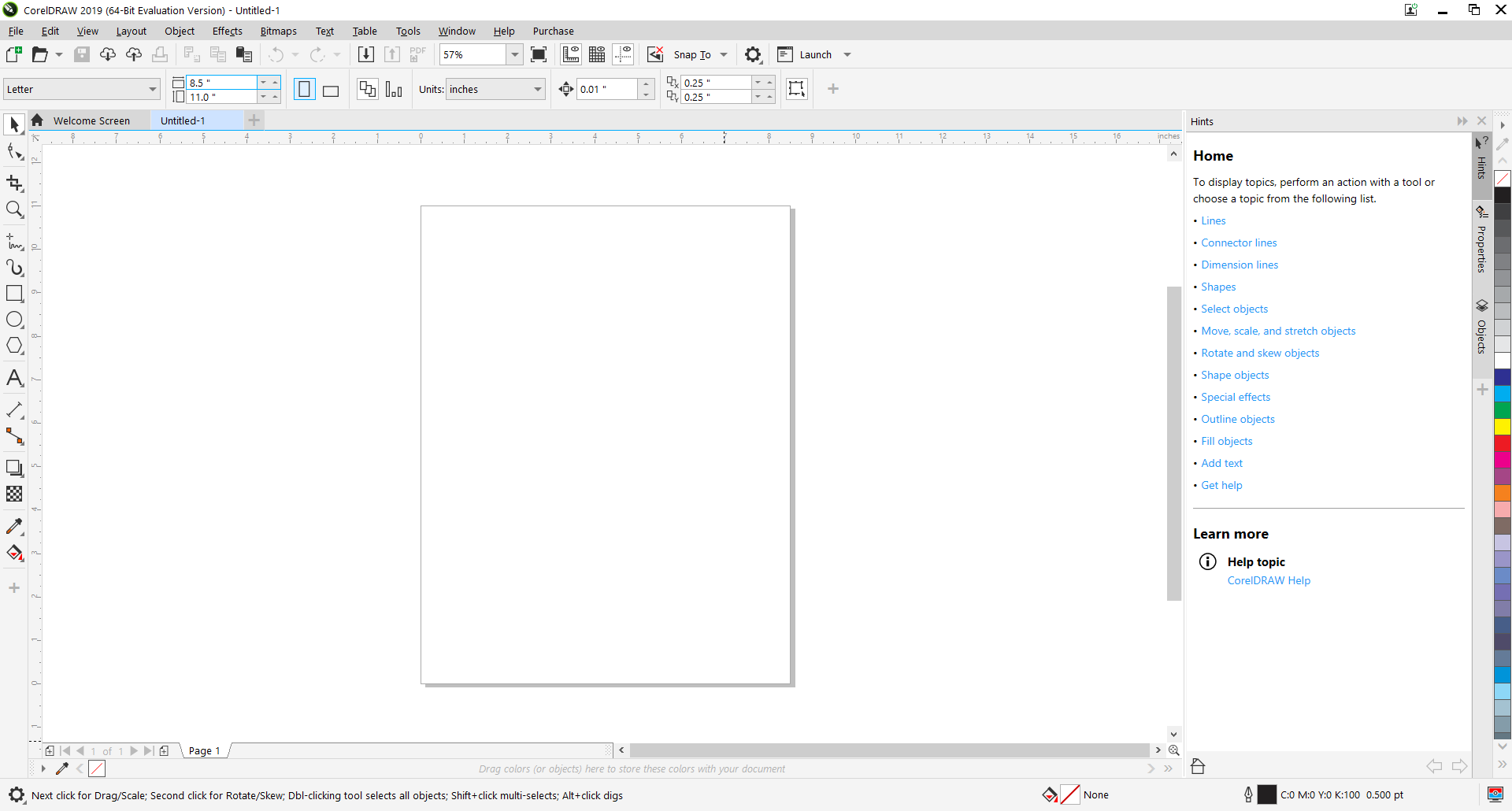


Рисунок 5 – интерфейс CoreIDRAW

В таблице 4 прописаны сравнительные характеристики всех программных средств для разработки интерфейса.

Таблица 4 – выбор программного средства для разработки интерфейса.

|  |  |  |  |
| --- | --- | --- | --- |
| Средство разработки интерфейса | Figma | Adobe Photoshop | CoreIDRAW |
| Доступность программного обеспечения | Есть бесплатная версия  Figma Professional  $12 | Только платная версия  $22.99 | Есть бесплатная версия  $27.42 |
| Опыт работы | 4 года | Не работал | Полгода |
| Предназначен для веб-дизайна | Да, разрабатывался под удобство веб дизайнеров | Нет, разрабатывался для графических дизайнеров | Нет, разрабатывался для графических дизайнеров |
| Платформа | Веб платформа, удобно открывать в любом месте | Только desktop | Только desktop |

Для создания интерфейса проекта был выбран графический редактор Figma. Этот выбор обоснован тем, что Figma является мощным инструментом для создания интерфейсов и прототипов для веб-сервисов, который позволяет работать над проектом в режиме реального времени и совместно с другими участниками команды, так как находится в облачном хранилище.

Figma обладает широким набором функций и инструментов, которые позволяют создавать высококачественные и профессиональные интерфейсы. Кроме того, Figma является онлайн-сервисом, что позволяет работать с проектом из любой точки мира и на любом устройстве.

В целом, выбор Figma для создания интерфейса проекта «Разработка веб-сервиса для старост по учету посещаемости студентов» обоснован его функциональностью, удобством использования и возможностью совместной работы над проектом.

* + 1. **Выбор языка программирования**

Из всех языков можно выделить три языка программирования, для выделенной предметной области.

* C#
* Java
* Go

C# – это объектно-ориентированный язык программирования, разработанный компанией Microsoft. Он является частью платформы .NET и широко используется для создания приложений под Windows, веб-приложений, игр, мобильных приложений и других программных решений. C# сочетает в себе простоту использования с мощными возможностями, такими как сильная типизация, управление памятью и поддержка многопоточности.

Java – это высокоуровневый объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (позднее приобретенной компанией Oracle). Он известен своей кроссплатформенностью, что позволяет запускать программы на различных операционных системах без изменений в их исходном коде. Java широко используется для создания веб-приложений, мобильных приложений, встраиваемого программного обеспечения, игр и других прикладных программ.

Go – это язык программирования, разработанный компанией Google. Он известен своей простотой, эффективностью и возможностью параллельного выполнения задач. Go часто используется для создания высоконагруженных веб-сервисов, распределенных систем, облачных приложений и других программ, где требуется эффективная обработка параллельных задач.

А таблице 5 представлено сравнение языков программирования и выбор наилучшего для предметной области.

Таблица 5 – сравнение языков программирования

|  |  |  |  |
| --- | --- | --- | --- |
|  | C# | Java | Go |
| Кроссплатформенность | - Ограниченная поддержка на других платформах | + Широкая поддержка на различных операционных системах | + Кроссплатформенность изначально встроена в язык |
| Экосистема и библиотеки | + Хорошая интеграция с платформой .NET и обширная библиотека | + Обширная экосистема и множество библиотек, включая Spring Framework | - Относительно молодая экосистема и меньшее количество библиотек |
| Надежность и масштабируемость | + Хорошая надежность и масштабируемость | + Известная надежность и масштабируемость, поддерживаемая Spring Boot | + Изначально разработан для создания масштабируемых систем |
| Сообщество и поддержка | + Большое сообщество разработчиков и обширная документация | + Большое сообщество разработчиков и обширная документация | + Активное сообщество и поддержка со стороны Google |
| Безопасность и управление памятью | + Управление памятью через сборку мусора | + Высокий уровень безопасности и управление памятью | + Безопасность и управление памятью изначально встроены в язык |
| Интеграция с существующими системами | + Хорошая интеграция с платформой .NET | + Хорошая поддержка для интеграции с существующими системами | + Хорошая поддержка для интеграции с другими языками и системами |
| Поддержка многопоточности | + Поддержка многопоточности через асинхронное программирование | + Встроенная поддержка многопоточности | + Эффективная поддержка многопоточности через горутины |

Java была выбрана из-за своей кроссплатформенности, обширной экосистемы и библиотек, надежности, поддержки сообщества, безопасности и управления памятью, интеграции с существующими системами, а также поддержки многопоточности. Кроме того, использование Spring Boot облегчает создание самостоятельных, производственно-готовых приложений, что делает Java подходящим выбором для данного проекта.

* 1. **Техническое задание**
     1. **Общие сведения**
        1. **Наименование системы**
           1. **Полное наименование системы**

Полное наименование системы: Разработка веб-сервиса для старост по учету посещаемости студентов.

* + - * 1. **Краткое наименование системы**

Краткое наименование системы: веб-сервис по учету посещаемости.

* + - 1. **Основания для проведения работ**

Основанием для разработки является задание на курсовой проект по дисциплине «Проектная деятельность 4».

* + - 1. **Наименование организаций – Заказчика и Разработчика**

Заказчик: учебное заведение СибГИУ.

Разработчик: Токмаков Антон Константинович.

* + - 1. **Плановые сроки начала и окончания работы**

Дата начала работ: 21.09.2023.

Дата окончания работ: 12.04.2024.

* + - 1. **Источники и порядок финансирования**

Источники финансирования не имеется.

* + 1. **Назначение и цели создания системы**
       1. **Назначение системы**

Основное назначение системы является упростить и тем самым ускорить работу старосты и куратора группы по составлению аттестационной ведомости студентов группы.

* + - 1. **Цели создания системы**

Система разрабатывается со следующими задачами:

* Анализ процесса учета посещаемости студентов в колледже СибГИУ;
* Составить документацию по выбору и обоснованию инструментов, используемых в проекте;
* Разработать техническое задание для проекта;
* Разработать информационное, алгоритмическое, программное и техническое обеспечение проекта на разработку веб-сервиса по учету посещаемости студентов;
* Спроектировать и реализовать базу данных для проекта веб-сервиса по учету посещаемости старост;
* Разработать back-end веб-сервиса по учету посещаемости студентов;
* Разработать front-end веб-сервиса по учету посещаемости студентов;
* Протестировать веб-сервис по учету посещаемости студентов на конечный пользователях.
  + 1. **Характеристика объекта автоматизации**
       1. **Основные сведения об объекте автоматизации или ссылки на документы с этими сведениями**

Объектом автоматизации является процесс по отметкам посещаемости студентов.

Система предназначена для упрощения работы старосты и куратора группы по выставлению пропусков, а также составлению аттестационной ведомости студентов группы.

* + - 1. **Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды**

Дополнительные требования к эксплуатации и характеристикам окружающей среды не предъявляются.

* + 1. **Требования к автоматизированной системе**
       1. **Требования к структуре АС в целом**
          1. **Перечень подсистем, их назначение и основные характеристики**
* Подсистема хранения данных – централизованная база данных, обеспечивающая доступ к данным о пропусках.
* Подсистема обработки информации о посещаемости – сохраняет, записывает, изменяет и читает данные из базы.
* Подсистема логирования – собирает ошибки с другой подсистемы и предоставляет удобный просмотр и отправку уведомлений разработчикам о критических ошибках.
  + - * 1. **Требования к способам и средствам обеспечения информационного взаимодействия компонентов АС**

Между подсистемой логирования и обработки информации о посещаемости был выбран способ взаимодействия компонентов на основе GRPC.

* + - * 1. **Требования к характеристикам взаимосвязей создаваемой АС со смежными АС**

Дополнительные требования к характеристикам взаимосвязей со смежными системами не предъявляются.

* + - * 1. **Требования к режимам функционирования АС**

Система должна иметь возможность функционировать в следующих режимах:

* Штатный режим;
* Режим системного администрирования.

Штатный режим должен являться основным режимом функционирования, обеспечивающим выполнение задач Веб-сервиса по учету посещаемости студентов.

Режим системного администрирования должен являться технологическим режимом и использоваться для сопровождения Веб-сервиса по учету посещаемости студентов.

* + - * 1. **Требования по диагностированию АС**

Дополнительные требования по диагностированию системы при развитии системы не предъявляются.

* + - * 1. **Перспективы развития, модернизации АС**

Дополнительные требования по перспективам развития и модернизации системы не предъявляются.

* + - 1. **Требования к функциям АС**

а) Хранение данных – централизованное и безопасное хранение данных о посещаемости студентов.

б) Выставление пропусков – возможность произвести выставление пропусков студентам группы.

в) Создание аттестационной ведомости по окончанию аттестации.

д) Возможность старосте загружать справки пропусков по уважительной причине.

е) Контроль доступа и безопасность – обеспечение безопасности данных и контроля доступа.

* + - 1. **Требования к видам обеспечения**
         1. **Требования к математическому обеспечению**

Дополнительные требования к математическому обеспечению не предъявляются.

* + - * 1. **Требования к информационному обеспечению**

Дополнительные требования к информационному обеспечению не предъявляются.

* + - * 1. **Требования к лингвистическому обеспечению**

Дополнительные требования к лингвистическому обеспечению не предъявляются.

* + - * 1. **Требования к программному обеспечению**

Дополнительные требования к программному обеспечению не предъявляются.

* + - * 1. **Требования к техническому обеспечению**

Минимальные и рекомендованные системные требования клиентского компьютера для запуска информационной системы управления проектами в образовательной сфере представлена в таблице 7.

Таблица 7 – Требование к клиентской машине

|  |  |
| --- | --- |
| Минимальные системные требования | |
| Операционная система | Windows 10 |
| Процессор | Intel i3 7100 |
| Свободное место | не менее 128 Gb |
| Объем оперативной памяти | не менее 4 Gb |
| Видеокарта | Nvidia GeForce G210 |
| Рекомендуемые системные требования | |
| Операционная система | Windows 10 |
| Процессор | Intel i3 9100 |
| Свободное место | 256 Gb |
| Объем оперативной памяти | 16 Gb |
| Видеокарта | Nvidia GeForce 1050 |

Для реализации информационной системы, также требуется серверный компьютер со следующими минимальными или рекомендованными системными требованиями, представленными в таблице 8.

Таблица 8 – Требование к серверу

|  |  |
| --- | --- |
| Минимальные системные требования | |
| Операционная система | Windows 10 |
| Процессор | Intel i3 9100 |
| Свободное место | 512 Гб |
| Объем оперативной памяти | 8 Гб |
| Видеокарта | Nvidia GeForce G210 |
| Рекомендованные системные требования | |
| Операционная система | Windows 10 |
| Процессор | Intel i3 10100 |
| Свободное место | 1 Тб |
| Объем оперативной памяти | 16 Гб |
| Видеокарта | Nvidia GeForce GT1030 или AMD Radeon R7 |

* + - * 1. **Требования к метрологическому обеспечению**

Дополнительные требования к метрологическому обеспечению не предъявляются.

* + - * 1. **Требования к организационному обеспечению**

Дополнительные требования к организационному обеспечению не предъявляются.

* + - * 1. **Требования к методическому обеспечению**

Дополнительные требования к методическому обеспечению не предъявляются.

* + - 1. **Общие технические требования**
         1. **Требования к численности и квалификации персонала и пользователей АС**

Дополнительные требования к численности и квалификации персонала не предъявляются.

* + - * 1. **Требования к показателям назначения**

Дополнительные требования к показателям назначения не предъявляются.

* + - * 1. **Требования к надежности**

Дополнительные требования к надежности не предъявляются.

* + - * 1. **Требования к безопасности**

Дополнительные требования к безопасности не предъявляются.

* + - * 1. **Требования к эргономике и технической эстетике**

Дополнительные требования к эргономике и технической эстетике не предъявляются.

* + - * 1. **Требования к транспортабельности для подвижных АС**

Дополнительные требования к транспортабельности не предъявляются.

* + - * 1. **Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов АС**

Дополнительные требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов АС не предъявляются.

* + - * 1. **Требования к защите информации от несанкционированного доступа**

Доступ к аппаратным средствам Веб-сервиса по учету посещаемости должен быть ограничен технологическими и организационными мерами.

* + - * 1. **Требования по сохранности информации при авариях**

Необходимо гарантировать безопасность и предотвращение утраты информации путем внедрения организационных и технических мер. Требуется обеспечить возможность создания резервных копий и восстановления информации с использованием стандартных средств программного обеспечения от поставщиков решений и/или средств от сторонних разработчиков, с учетом требований к защите информации.

* + - * 1. **Требования к защите от влияния внешних воздействий**

Дополнительные требования к защите от влияния внешних воздействий не предъявляются.

* + - * 1. **Требования к патентной чистоте и патентоспособности**

Дополнительные требования к патентной чистоте не предъявляются.

* + - * 1. **Требования по стандартизации и унификации**

Дополнительные требования по стандартизации и унификации не предъявляются.

* + - * 1. **Дополнительные требования**

Дополнительные требования не предъявляются

* + 1. **Порядок разработки автоматизированной системы**
       1. **Порядок организации разработки АС**

Список задач для разработки АС представлена на рисунке 6. На рисунке 7 изображена диаграмма Ганта.

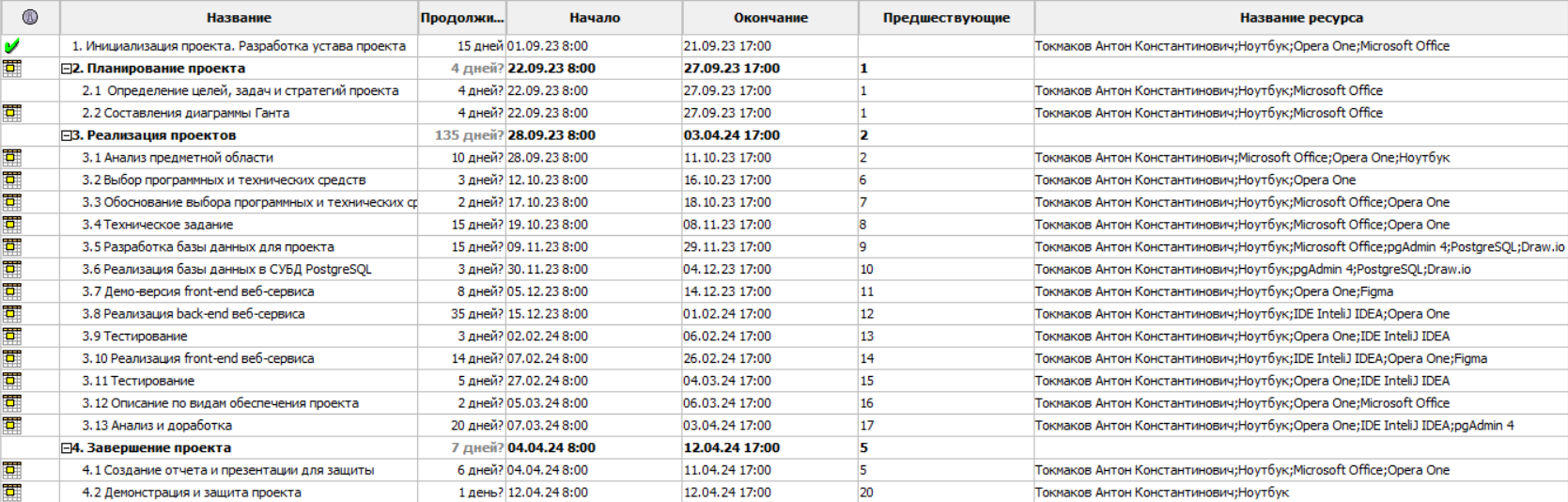


Рисунок 6 – Список задач для разработки проекта

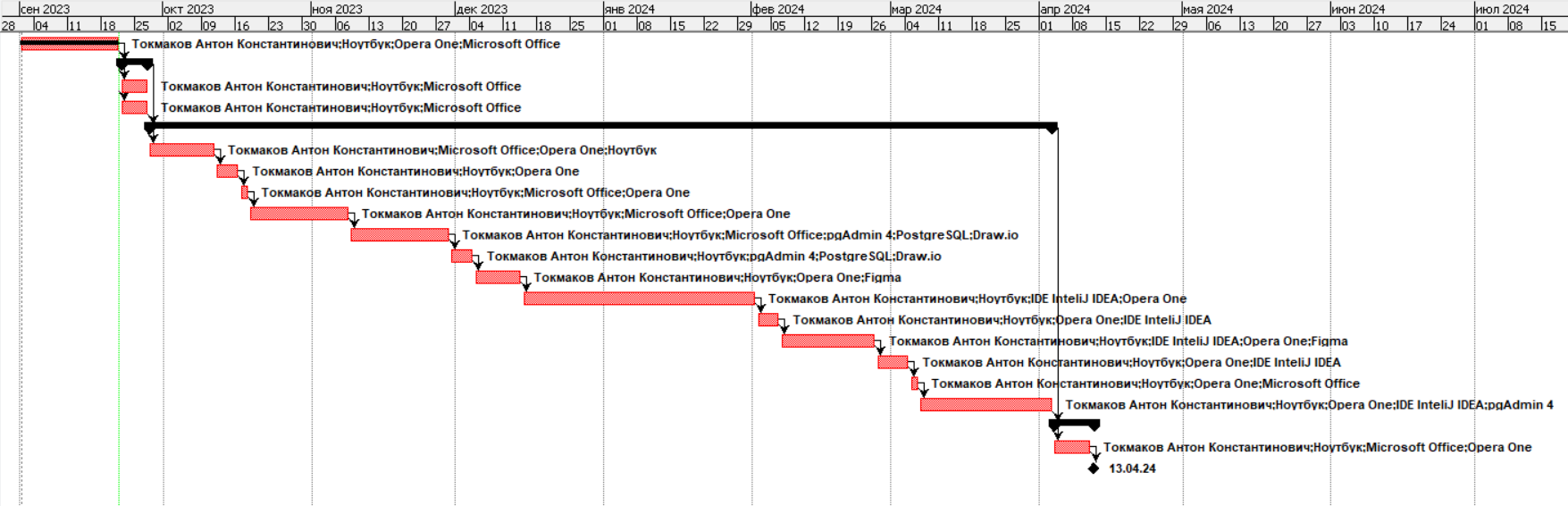


Рисунок 7 – Диаграмма Ганта проекта

* + - 1. **Перечень документов и исходных данных разработки АС**

Перечень документов:

* Техническое задание на разработку АС;
* Технические спецификации и стандарты.
  + - 1. **Перечень документов, предъявляемых по окончании соответствующих этапов работ**

В качестве документов, предъявляемых по окончанию этапов работ, служат промежуточные отчеты по выполнению работ.

* + - 1. **Порядок проведения экспертизы технической документации**
* Проверка соответствия документации действующим стандартам и нормативным актам;
* Оценка технических аспектов документации;
* Подведение итогов экспертизы.
  + - 1. **Перечень макетов, порядок их разработки, изготовления, испытаний, необходимость разработки на них документации, программы и методик испытаний**

Дополнительных макетов не требуется.

* + - 1. **Порядок разработки, согласования и утверждения плана совместных работ по разработке АС**

Поскольку разработка ведется одним лицом, нет необходимости в установлении порядка разработки, согласования и утверждения плана совместных работ по созданию автоматизированной системы.

* + - 1. **Порядок разработки, согласования и утверждения программы работ по стандартизации**

Проводимые мероприятия:

* Определение стандартов, которым должна соответствовать АС;
* Разработка программы работ по стандартизации;
* Согласование программы со стейкхолдерами и утверждение.
  + - 1. **Требования к гарантийным обязательствам разработчика**

Дополнительных требований к гарантийным обязательствам не предъявляется.

* + - 1. **Порядок проведения технико-экономической оценки разработки АС**

Дополнительного проведения технико-экономической оценки разработки АС не требуется.

* + - 1. **Порядок разработки, согласования и утверждения программы метрологического обеспечения программы на надежность**

Дополнительного утверждения программы метрологического обеспечения программы на надежность не требуется.

* + 1. **Порядок контроля и приемки автоматизированной системы**
       1. **Виды, состав и методы испытаний АС и ее составных частей**

Типы тестирования:

* Проверка функциональности – оценка соответствия функциональным требованиям системы;
* Оценка производительности и стабильности системы при максимальных нагрузках;
* Проверка уровня защищенности системы и соблюдение стандартов безопасности.
  + - 1. **Общие требования к приемке работ, порядок согласования и утверждения приемочной документации**

При приемке АС должны произвестись следующие действия:

* Сравнение функционала системы с требованиями, предъявленными заказчиком;
* Проверка соответствия документации системы и реальной реализации;
* Оценка удовлетворенности пользователя.
  + - 1. **Статус приемочной комиссии**

Приемочная комиссия отсутствует.

* + 1. **Требования к подготовке объекта автоматизации к вводу АС в действие**
       1. **Перечень мероприятий подготовки объекта автоматизации**

В перечень мероприятий подготовки объекта автоматизации входят:

* Обеспечение необходимых вычислительных ресурсов, сетевой инфраструктуры, и обеспечение безопасности данных;
* Подготовка рабочих мест пользователей с установкой необходимого программного обеспечения;
* Тестирование окружающей среды для поддержки функционирования АС.
  + - 1. **Создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой АС требованиям ТЗ**

Дополнительных требований по условиям функционирования объекта автоматизации не предъявляется.

* + - 1. **Проведение необходимых организационно-штатных мероприятий**

Среди необходимых организационных мероприятий следует выделить:

* Разработку и утверждение организационной структуры, включая определение ответственных за поддержку и эксплуатацию автоматизированной системы;
* Обеспечение наличия технического персонала для поддержки и обслуживания системы;
* Утверждение процедур взаимодействия между сотрудниками, ответственными за автоматизированную систему, и пользователями.
  + - 1. **Порядок обучения персонала и пользователей АС**

В качестве средства обучения персонала и пользователей следует разработать руководство по эксплуатации и технической поддержке системы.

* + 1. **Требования к документированию**
       1. **Перечень подлежащих разработке документов**

В перечень подлежащих разработке документов входят:

* Техническое задание на разработку АС;
* Руководство пользователя;
* Техническая спецификация и функциональные требования.
  + - 1. **Вид представления и количество документов**

Вид представления:

* электронные версии документов для удобства распространения и архивирования.
* печатные копии для официальных целей.

Количество документов минимизировано, но обеспечено полнотой и понятностью представления информации.

* + - 1. **Требования по использованию ЕСКД и ЕСПД при разработке документов**

Применение стандартных символов и обозначений для унификации и упрощения восприятия документации. Обеспечение структурированности программной документации на всех этапах жизненного цикла системы. Использование стандартных схем и шаблонов для программной документации.

* + 1. **Источники разработки**

В настоящем документе использованы следующая литература и нормативные документы:

* ГОСТ 34.602-2020 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»;

1. **Виды обеспечения**
   1. **Информационное обеспечение**
      1. **Use-case**

На рисунке 8 представлен use-case веб-сервиса по посещаемости студентов. На нем изображено 5 актеров.

Староста в данном веб-сервисе главный актер системы, он может отметить пропуск, просмотреть посещаемость по паре, просмотреть список студентов, загрузить справку и запросить разрешение на редактирование информации по посещаемости. Так же староста обобщается от актера регистрация и отчеты.

Далее есть актер преподаватель, который может разрешить редактирование и просмотреть список запросов на редактирование, так же он обобщается от актера регистрация и отчет, чтобы зарегистрироваться в системе и просматривать отчеты соответственно.

Последний актер в веб-сервисе – это администратор, который может изменять информацию о дисциплине и информацию о преподавателе. Так же данный актер обобщается от регистрации и отчеты.

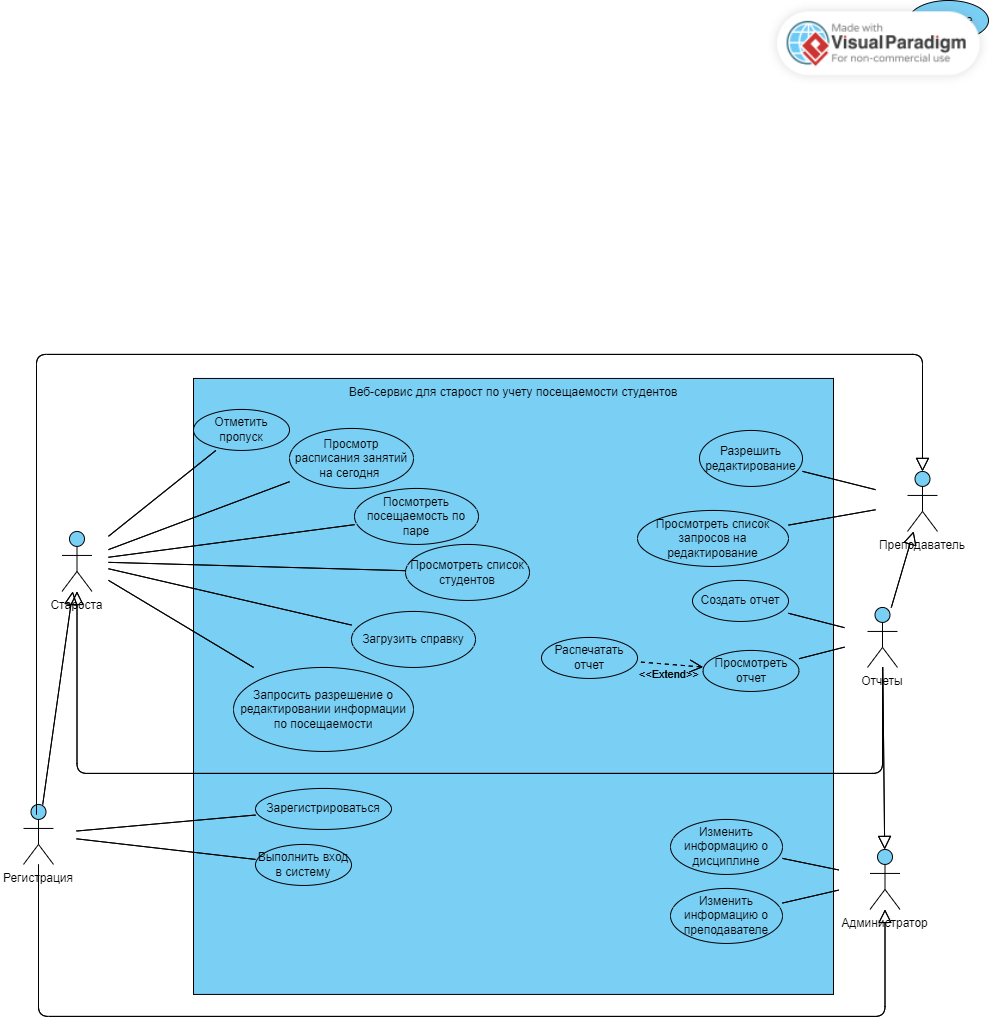


Рисунок 8 – use-case веб-сервиса

* + 1. **Iedf0**
    2. **Схема базы данных**

Сущность институты хранит в себе информацию о наименовании института, его аббревиатуры, кабинет института, а также отдельной сущностью контактную информацию для института, в которой есть такие атрибуты как код института, наименование контакта и сама контактная информация.

Обязательно существующему и только одному институту относится к множеству возможно не существующей контактной информации.

Кафедра хранит в себе такую информацию как наименование кафедры, аббревиатура кафедры и код института, которому принадлежит кафедра. Так же у кафедры есть контактная информация, которая хранит такие данные как идентификатор записи, код кафедры к которой относится данная информация, наименование и сама контактная информация.

К одному точно существующему институту относится множество точно существующих кафедр. И к одной точно существующей кафедре относится множество возможно не существующей контактной информации.

Сущность преподаватели хранит в себе код преподавателя, фамилию, имя, отчество и код кафедры.

В таблице дисциплины хранится данные о коде дисциплины, наименованию дисциплины, аббревиатура дисциплины, а также код кафедры, к которой относится дисциплина.

В сущности, учебные группы можно увидеть код учебной группы, наименование и аббревиатура группы, количество обучающихся, код кафедры, для определения к какой кафедре относится группа, а также есть атрибут специализация.

К одной точно существующей кафедре, относятся множество точно существующих преподавателей, дисциплин и учебных групп.

В учебной группе есть атрибут специализация, которая ссылается на таблицу специализация, в которой хранится такая информация как код специализации, наименование специализации, а также сокращение специализации.

К одной точно существующей специализации, относится множество точно существующих учебных групп.

Далее дисциплину и преподавателя надо связать одной сущностью с наименованием преподаватели\_дисциплины, в которой хранится уникальный идентификатор данной связи и внешние ключи на сущность преподавателя и дисциплины.

К одному точно существующему преподавателю относятся множество точно существующих связей преподаватель\_дисциплины.

Так же и на оборот, к одному точно существующей дисциплине относятся множество точно существующих связей преподаватель\_дисциплины.

Преподаватели могут быть кураторами группы нескольких групп, следовательно сущности преподаватели и учебные группы надо связать дополнительной сущностью кураторы групп, которая будет отвечать за связь определенного препода с группой внешними ключами и иметь свой уникальный идентификатор связи, что бы удовлетворять нормальной формы Бойса-Кодда.

Один точно существующий преподаватель относится к множеству точно существующих кураторов групп и такое же истинно для сущности учебные группы. Одина точно существующая группа относится к множеству точно существующих кураторов групп.

Студенты хранят в себе информацию о уникальном идентификаторе, фамилии, имени, отчестве, даты рождения, староста и код учебной группе, так как у каждого студента есть учебная группа.

К одной точно существующей учебной группе относится множество точно существующих студентов.

В сущности, посещаемость хранятся данные о уникальном идентификаторе записи, внешний ключ на связку преподавателя и дисциплину, студента, статуса посещаемости, дня недели, типы пары, а также атрибуты не являющиеся внешний ключом, дата и номер пары.

Таблица посещаемости связана с таблицей документ подтверждение уважительного пропуска один-к-одному, но документ-подтверждение при этом может не существовать. Он содержит уникальный идентификатор, наименование пропуска, скан документа, дата начала, дата окончания, код посещаемости к которому будет относится документ.

Так же есть таблицы справочники, такие как статус посещаемости, тип пары, дни недели.

На рисунке 10 изображена логическая схема БД в третьей нормальной форме, описанная выше.

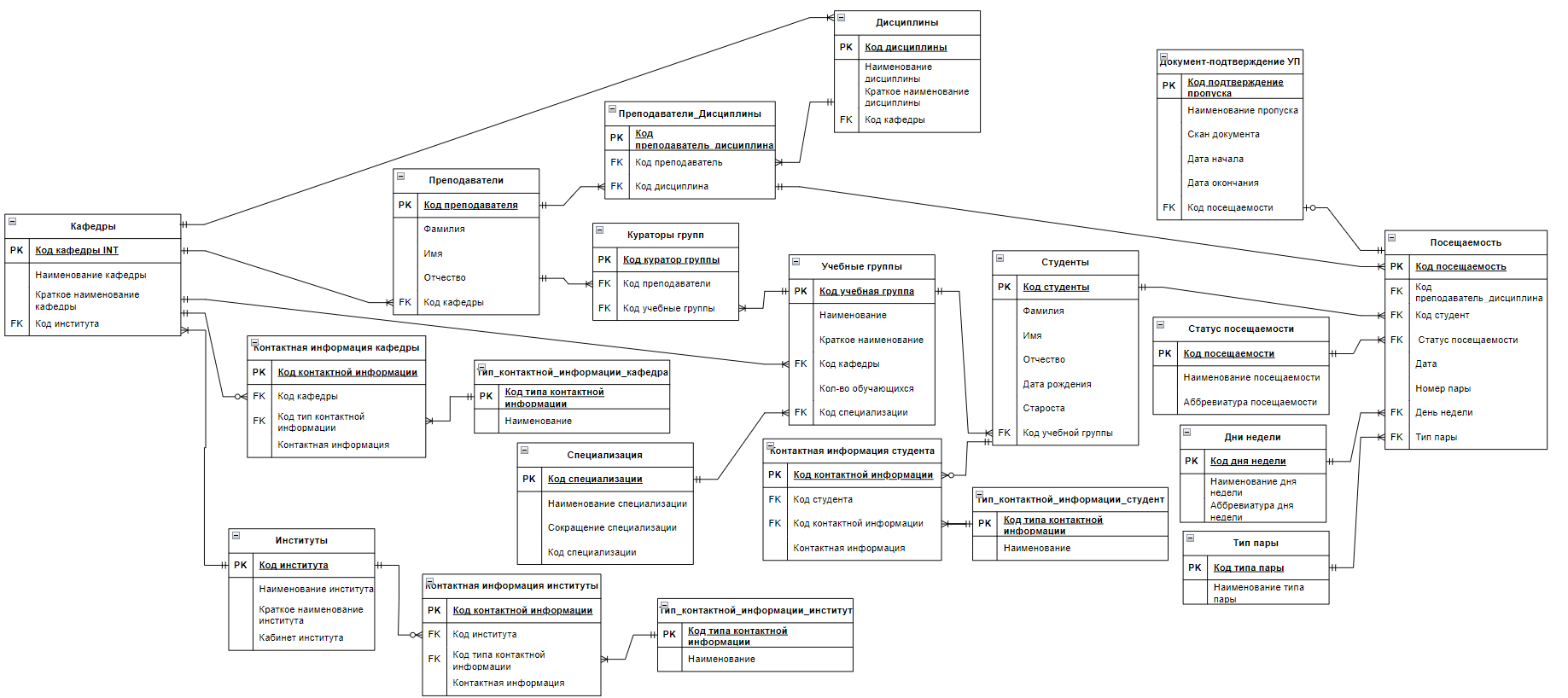


Рисунок 10 – Логическая схема БД