Оглавление

[Введение 2](#_Toc39053175)

[1. Постановка задачи и обзор аналогичных решений 3](#_Toc39053176)

[1.1 Постановка задачи 3](#_Toc39053177)

[1.2 разработать геоинформационную систему Негорельского лесхозаОбзор аналогов 3](#_Toc39053178)

[1.3 Патентный поиск по теме дипломного проекта 5](#_Toc39053179)

[1.3.1 Программа автоматизации учебного процесса (Программный Комплекс «Вега-Сфинкс») 6](#_Toc39053180)

[1.3.2 Модуль «Управление учебным центром» 6](#_Toc39053181)

[1.4 Вывод по разделу 1 7](#_Toc39053182)

[2 Проектирование структуры приложения 8](#_Toc39053183)

[2.1 Функциональное наполнение интернет-сервиса 8](#_Toc39053184)

[2.2 Архитектура интернет-сервиса 9](#_Toc39053185)

[2.3 База данных интернет-сервиса 11](#_Toc39053186)

[2.3 Выводы по разделу 2 14](#_Toc39053187)

[3. Разработка интернет-сервиса 15](#_Toc39053188)

[3.1 Разработка клиентской части интернет-сервиса 15](#_Toc39053189)

[3.2 Разработка серверной части интернет-сервиса 17](#_Toc39053190)

[3.3 Разработка базы данных интернет-сервиса 18](#_Toc39053191)

Введение

На кафедре информационных систем и технологий Белорусского государственного технологического университета ежегодно защищают дипломные проекты и работы около 100 выпускников. Контроль за ходом дипломного проектирования представляет собой отдельную не простую задачу,

Целью дипломного проекта является повышение эффективности работы заведующего кафедрой и руководителей дипломных проектов.

Для достижения цели сформированы следящие разделы:

1. Постановка задач и обзор аналогичных решений;
2. Проектирования интернет-сервиса;
3. Реализация интернет сервиса;
4. Тестирование интернет сервиса;

Для реализации данного интернет-сервиса используется база данных PostgreSQL, фреймворк для разработки динамических веб-приложений Angular, а также для разработки серверной части язык программирования Golang. Выбор фреймворка для разработки веб-приложения пал в пользу Angular по причине того, что он позволяет создавать быстродействующие, легко оптимизируемые приложения. Он также предоставляет возможность широкой настройки компонентов. Также фреймворк обзавёлся довольно большим сообществом разработчиков.

База данных PostgreSQL была выбрана, т.к. она является объектно-реляционной, поддерживает пользовательские объекты и их поведение, включая типы данных, функции, операции, домены и индексы. Это делает Posgres невероятно гибкой и надежной.

При разработке данного приложения был произведен обзор аналогичных продуктов с целью устранения слабых и добавления сильных сторон приложений схожей тематики в дипломный проект. Структура приложения спроектирована таким образом, чтобы позволить использовать интернет-сервис для различных кафедр университета.

1. Постановка задачи и обзор аналогичных решений

## Постановка задачи

Для достижения цели были поставлены следующие задачи:

* + исследовать преимущества и недостатки аналогичных приложений;
  + разработать интерфейс приложения;
  + разработать прототип приложения на основе интерфейса приложения;
  + разработать структуру базы данных;
  + разработать архитектуру программной реализации приложения.

## разработать геоинформационную систему Негорельского лесхозаОбзор аналогов

Модуль «GS-Ведомости: Online» - это web-интерфейс системы «GS-Ведомости», который позволяет работать с базой данных, установленной на сервере в образовательном учреждении, через сеть Интернет. Модуль работает на HTTP-сервере Apache. Пользователи веб-интерфейса системы подразделяются на следующие категории: администратор, преподаватель, студент, родители студента, абитуриент. Права доступа каждой категории пользователей настраиваются в модуле «Администратор» системы «GS-Ведомости».

В число реализованных на данный момент онлайн-модулей входят:

* модуль «Online: Абитуриент». Основной функционал онлайн-модуля заключается в возможности для абитуриентов подавать заявление на поступление на сайте образовательного учреждения. Заполненное заявление автоматически попадает в базу данных системы «GS-Ведомости». Данные о ходе приемной кампании, а также поля анкеты абитуриента можно настраивать исходя из потребностей и задач образовательного учреждения;
* модуль «Online: Контингент учащихся». С помощью web-интерфейса можно отслеживать данные об успеваемости и посещаемости студентов, итоги сессий, выполненные контрольные работы, лицевые счета студентов, обучающихся на платной основе. В данный модуль также входит онлайн-интерфейс расширения «Журнал успеваемости»;
* модуль «Online: Расписание занятий» отображает расписание занятий в разрезе группы студентов. Информацию о расписании занятий через web-интерфейс могут получать пользователи категории «студент» и «родитель студента». Пользователи категории «преподаватель» могут смотреть расписание своих занятий во всех группах;
* модуль «Online: Учебные планы». Отображает учебный план по специальности для пользователя категории «студент». Если войти в модуль «Online: Учебные планы» под логином преподавателя, отобразится перечень учебных планов по группам соответственно распределенной учебной нагрузке;
* модуль «Блокнот» можно использовать в качестве планировщика и дневника;

Интерфейс, вышерассмотренного аналога, представлен на рисунке 1.1

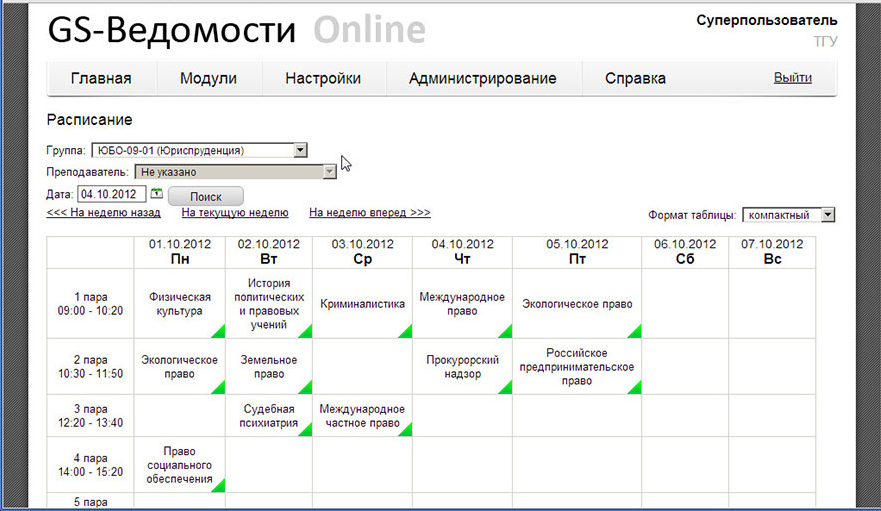


Рисунок 1.1 – Обзор аналога GS-Ведомости: Online

Автоматизация управления учебным процессом usu – это специализированное программное обеспечение, действие которого направлено на оптимизацию всего устройства. Автоматизация контроля учебного процесса возьмет на себя все прежде контролируемые единицы организации, станет напоминать об истекающих товарах, необходимых для обучения. Контроль результативности проводимых занятий и их посещаемость. Возможность составления расписания занятий внутри ПО, позволит составить его правильно, в соответствии с рациональным и последовательным использованием аудиторий.

Автоматизация учебного процесса usu подходит как для небольших образовательных отделов, мини-центров, обучающих дошколят, курсов по английскому, математике, физике и других интересных предметов, так и для ВУЗов, колледжей, лицеев и самих школ. Управление внутри системы осуществляет администратор (руководитель и/или бухгалтер), именно он распределяет обязанности и полномочия внутри софта Автоматизации. И может ограничивать доступ к некоторой информации для определенных подчиненных.

В целом, интерфейс программы максимально прост и представлен на рисунке 1.2.

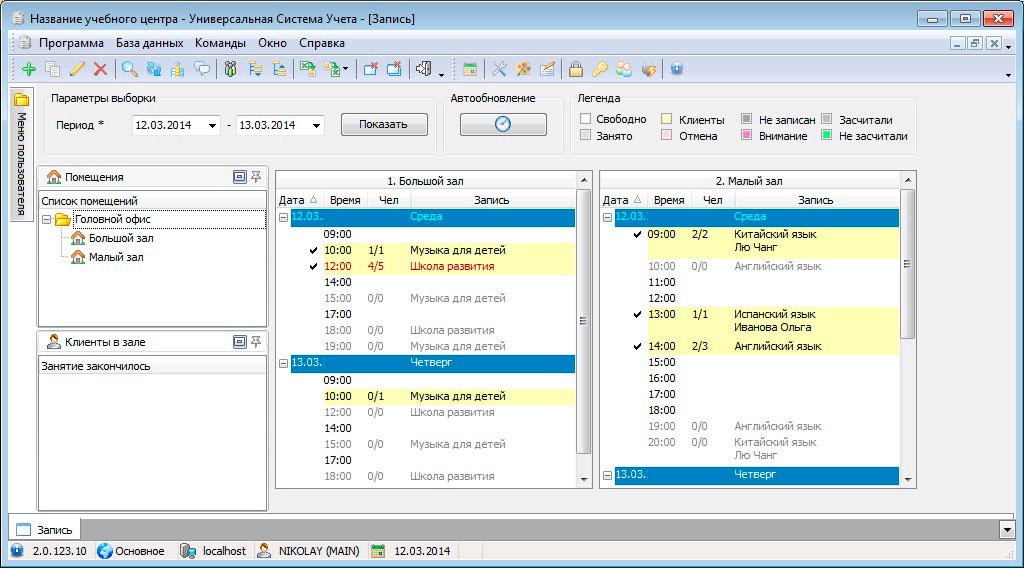


Рисунок 1.2 – Обзор аналога usu

## Патентный поиск по теме дипломного проекта

По определению патент — это документ, удостоверяющий государственное признание технического решения изобретением, полезной моделью, промышленным образцом и закрепляющий за лицом, которому он выдан, исключительное право на использование указанных объектов.

Патентный поиск — это процесс отбора соответствующих запросу документов или сведений по одному или нескольким признакам из массива патентных документов или данных.

Предмет поиска определяют исходя из конкретных задач патентных исследований категории объекта (устройство, способ, вещество), а также из того, какие его элементы, параметры, свойства и другие характеристики предполагается исследовать.

Патентный поиск является трудоёмким, но необходимым мероприятием. Он необходим не только лицам или организациям, желающим запатентовать изобретение, но и промышленным предприятиям, желающим это изобретение использовать.

При выполнении дипломного проекта основной целью проведения патентного поиска является определение уровня новизны имеющегося технического решения. При этом в процессе поиска определяется, как решалась данная задача ранее, какие технические решения защищены авторскими свидетельствами и патентами и каковы перспективы разработки темы.

В соответствии с темой дипломного проекта был проведен патентный поиск в области программных систем для автоматизации учебного процесса.

Осуществив патентный поиск в области приложений для автоматизации учебного процесса, были выявлены следующие аналоги: Программа автоматизации учебного процесса (Программный Комплекс «Вега-Сфинкс»), Модуль «Управление учебным центром».

## Программа автоматизации учебного процесса (Программный Комплекс «Вега-Сфинкс»)

Тип реализующей ЭВМ: IBM PC - совмест. ПК.

Язык программирования: PHP; Java Script.

Вид и версия операционной системы: ALT Linux; Astra Linux.

Номер регистрации (свидетельства): [2015660631](https://www1.fips.ru/registers-doc-view/fips_servlet?DB=EVM&DocNumber=2015660631&TypeFile=html).

Дата регистрации: [20.11.2015](https://www1.fips.ru/Archive/EVM/2015/2015.11.20/DOC/RUNW/000/002/015/660/631/document.pdf).

Реферат:

Программа предназначена для получения знаний, выработки умений и навыков у обучаемых и может применяться в обучающих системах. Программа обеспечивает выполнение следующих функций: подготовка материалов автоматизированных учебных занятий и формирование учебных планов занятий, регистрация обучаемых, формирование учебных групп и составление расписания занятий, проведение занятий с учебными группами под руководством преподавателя, выполнение контроля полученных обучаемыми знаний, умений и навыков.

## Модуль «Управление учебным центром»

Тип реализующей ЭВМ: IBM PC.

Язык программирования: PHP, JavaScript, HTML.

Вид и версия операционной системы: кроссплатформенная.

Номер регистрации (свидетельства): [2018618320](https://www1.fips.ru/registers-doc-view/fips_servlet?DB=EVM&DocNumber=2018618320&TypeFile=html).

Дата регистрации: [11.07.2018](https://www1.fips.ru/ofpstorage/Doc/PrEVM/RUNWPR/000/002/018/618/320/2018618320-00001/document.pdf).

Реферат:

Программа является подсистемой для платформы eLearningServer 4G. Предназначена для автоматизации процесса планирования и управления очным и дистанционным учебным процессом . Учебный план формируется на основе входного потока заявок на обучение от сотрудников организаций и внешних заказчиков обучения. Решает следующие задачи: контроль входного потока заявок на обучение; динамическое календарное планирование графиков обучения; подбор преподавателей для проведения обучения; резервирование помещений и ресурсов для обеспечения учебного процесса ; информирование участников о начале обучения и/или изменениях сроков обучения.

## Вывод по разделу 1

В основном, оценка интернет-сервиса для учета и контроля выполнения дипломного проектирования в вузе сводится к удобству пользовательского интерфейса, а также простоте использования. В данном разделе был произведён обзор аналогов, а также был произведен патентный поиск аналогов по теме дипломного проекта. При этом каждый из аналогов был подробно описан, и было показано, для чего предназначено то или иное программное средство.

Как следует из представленного краткого обзора, большинство ресурсов имеют общие функциональные возможности. Также это касается и базовых функций. По этой причине пользователь сталкивается с проблемой выбора нужного ресурса. Аналоги демонстрируют действительно интересные решения, некоторые функции можно позаимствовать из данных приложений и взять себе на заметку при разработке дипломного проекта.

2 Проектирование структуры приложения

2.1 Функциональное наполнение интернет-сервиса

Интернет-сервис, разрабатываемый в рамках данного дипломного проекта, в первую очередь, предназначен для создания удобной в использовании системы учета и контроля выполнения дипломного проектирования в вузе в виде веб-приложения для максимальной совместимости с устройствами.

Юз-кейс диаграмма интернет-сервиса представлена на рисунке 2.1.

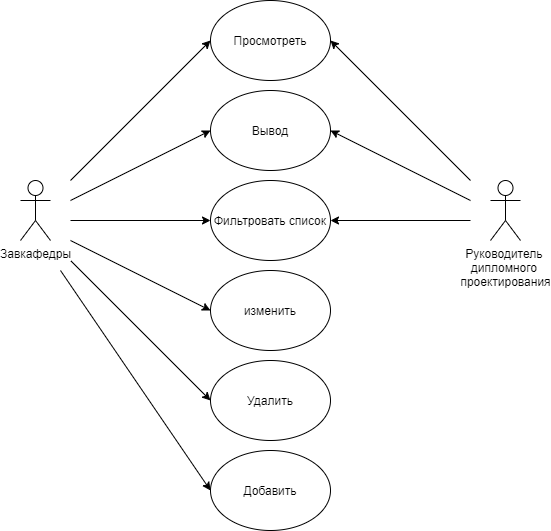


Рис.2.1 – Юз-кейс диаграмма приложения

В таблице 2.1 представлены роли интернет-сервиса и их назначение.

Таблица 2.1 — Виды ролей в приложении

|  |  |
| --- | --- |
| Роль | Назначение |
| Заведующий кафедры | Просмотр, изменение, удаление, добавление данных. Фильтрация данных. Вывод данных в документ. |
| Руководитель дипломного проектирования | Просмотр и фильтрация данных. Вывод данных в документ. |

В таблице 2.2 представлены функции доступные пользователю в интернет-сервисе.

Таблица 2.2 — Функции интернет-сервиса

|  |  |
| --- | --- |
| Функция | Пояснение |
| Просмотр | Просмотр информации в виде списка. |
| Вывод | Вывод отфильтрованных списков в виде документа |
| Фильтровать список | Изменение списка для просмотра по параметрам |
| Изменить | Изменение данных в списках. |
| Удалить | Удаление данных из списков. |
| Добавить | Добавление новых данных в списки. |

2.2 Архитектура интернет-сервиса

Целью дипломного проекта является разработка интернет-сервиса для учета и контроля выполнения дипломного проектирования в вузе для оптимизации контроля за ходом дипломных проектов или работ. Интернет-сервис должен иметь серверную часть, представляющую REST-сервисы к которым будет обращаться клиентская часть интернет-сервиса. Также должна иметься база данных для хранения данных о дипломных проектах или работ, руководителях дипломных проектов, нормконтролерах, рецензентах, председателях, приказах, комиссии и специальностях.

На рисунке 2.2 представлена архитектура интернет-сервиса данного дипломного проекта.

Клиентская часть представлена веб-приложением, которое создано для кроссбраузерной работы для учета и контроля выполнения дипломного проектирования в вузе. Клиентская часть имеет связь с серверной частью при помощи протокола http.

Серверная-часть - это часть системы, представленная REST-сервисами, предоставляющими функциональность описанную в пункте 2.1 и связующая между базой данных при помощи протокола tcp/ip и клиентской частью при помощи протокола http.

База данных – это часть интернет-сервиса ,которая предназначена для хранения информации связанной с дипломным проектированием. Данная часть должна реализовывать примерно следующий функционал: сохранение информации в формате utf-8, возможность «достать» эту информацию при помощи серверной части. Также данная часть должна поддерживать большой набор типов объектов.

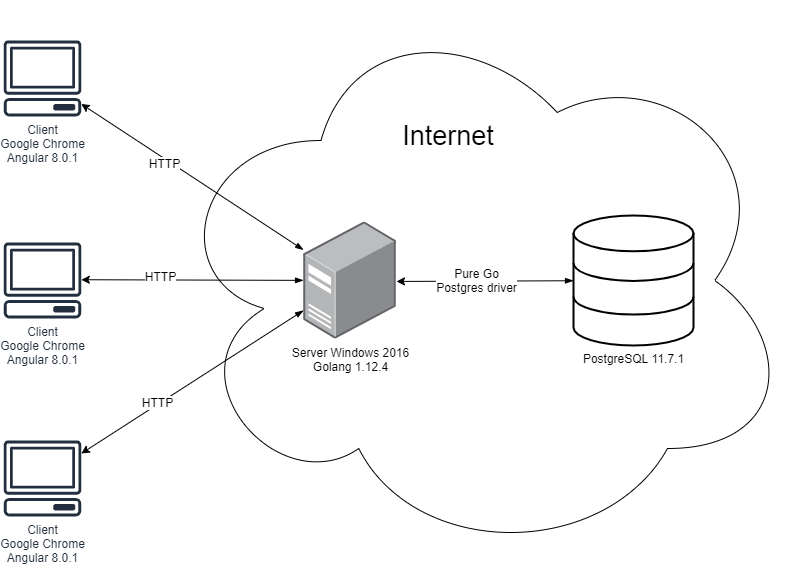


Рис.2.2 – Общая структура компонентов приложения

Основные параметры клиентской части приведены в таблице 2.3.

Таблица 2.3 — Описание клиентской части приложения

|  |  |
| --- | --- |
| Технология | Angular |
| Протокол взаимодействия | HTTP |
| Назначение | Отображение данных о дипломных проектах и работах, а так же манипуляции с ними |
| Дополнительные библиотеки | RxJs, ngx-bootstrap |

Основные параметры серверной части приведены в таблице 2.4.

Таблица 2.4 — Описание back-end части приложения

|  |  |
| --- | --- |
| Технология | Go |
| Протокол взаимодействия | HTTP, TCP/IP |
| Назначение | Связь клиентского приложения с базой данных, работа в качестве сервера статических файлов, в том числе и клиентского приложения |
| Дополнительные библиотеки | Gorilla, PQ |

Основные параметры базы данных приведены в таблице 2.5.

Таблица 2.5 — Описание базы данных приложения

|  |  |
| --- | --- |
| Технология | PostgreSQL |
| Протокол взаимодействия | TCP\IP |
| Назначение | Хранение информации о дипломном проектировании |

2.3 База данных интернет-сервиса

На рисунке 2.3 представлена схема таблиц базы данных.

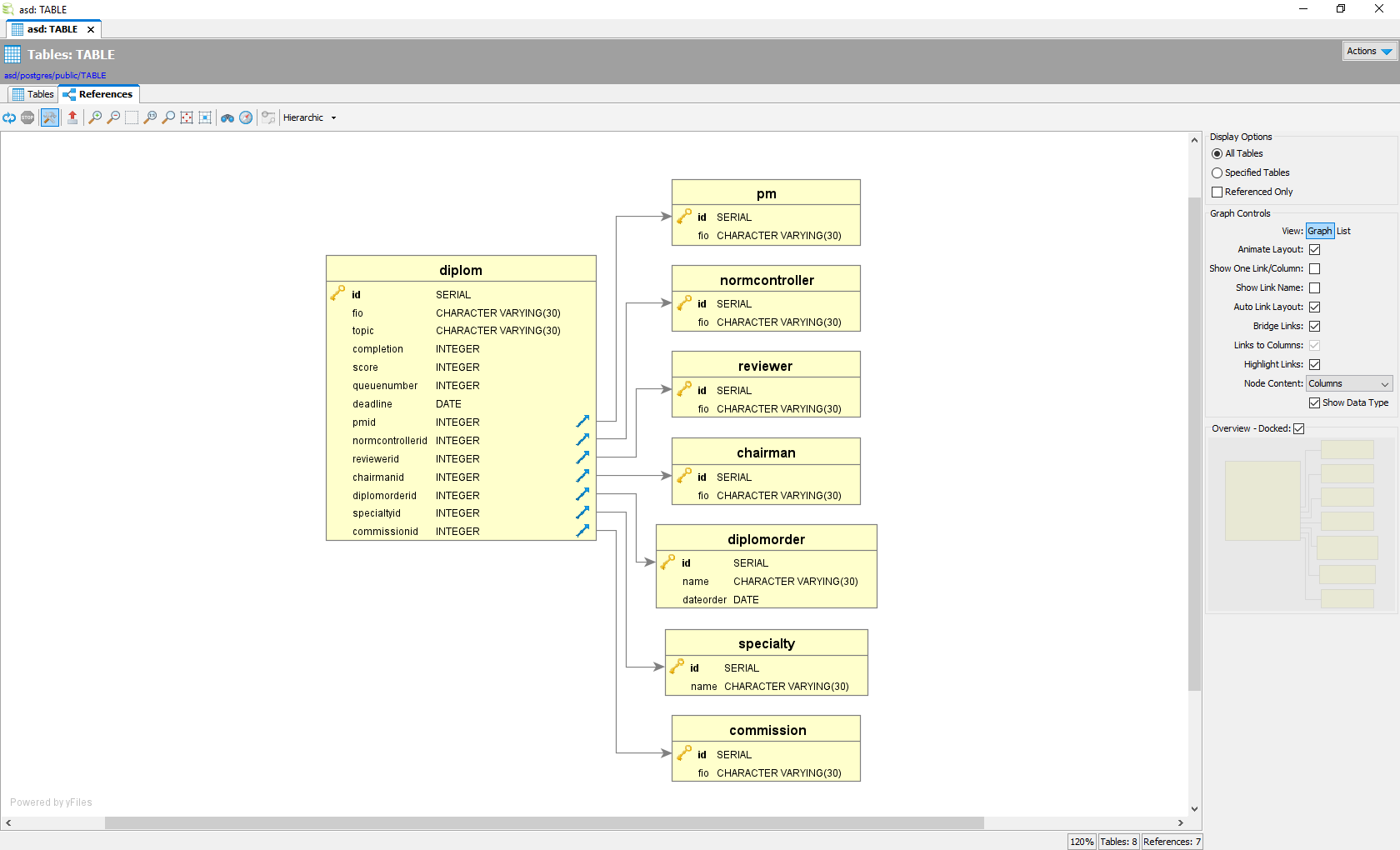


Рис.2.3 – Общая схема таблиц базы данных

Таблица «pm» предназначена для хранения руководителей дипломных проектов. Описание полей «pm» приведено в таблице 2.6.

Таблица 2.5 — Описание полей таблицы «pm»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id | Integer | Первичный ключ |
| fio | Varchar(30) | Имя руководителя |

Таблица «normcontroller» предназначена для хранения нормоконтролеров. Описание полей «normcontroller» приведено в таблице 2.6.

Таблица 2.6 — Описание полей таблицы «normcontroller»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id | Integer | Первичный ключ |
| fio | Varchar(30) | Имя нормоконтролера |

Таблица «reviewer» предназначена для хранения рецензентов. Описание полей «reviewer» приведено в таблице 2.6.

Таблица 2.6 — Описание полей таблицы «reviewer»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id | Integer | Первичный ключ |
| fio | Varchar(30) | Имя рецензента |

Таблица «chairman» предназначена для хранения председателей. Описание полей «chairman» приведено в таблице 2.6.

Таблица 2.6 — Описание полей таблицы «chairman»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id | Integer | Первичный ключ |
| fio | Varchar(30) | Имя председателя |

Таблица «diplomorder» предназначена для хранения приказов. Описание полей «diplomorder» приведено в таблице 2.7.

Таблица 2.7 — Описание полей таблицы «diplomorder»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id | Integer | Первичный ключ |
| name | Varchar(30) | Название приказа |
| dateorder | Date | Дата приказа |

Таблица «specialty» предназначена для хранения специальностей. Описание полей «specialty» приведено в таблице 2.8.

Таблица 2.8 — Описание полей таблицы «specialty»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id | Integer | Первичный ключ |
| name | Varchar(30) | Название специальности |

Таблица «commission» предназначена для хранения комиссии. Описание полей «commission» приведено в таблице 2.9.

Таблица 2.9 — Описание полей таблицы «commission»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id | Integer | Первичный ключ |
| name | Varchar(30) | Имя комиссии |

Таблица «diplom» предназначена для хранения данных о дипломном проекте или работе. Описание полей «diplom» приведено в таблице 2.10.

Таблица 2.10 — Описание полей таблицы «diplom»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id | Integer | Первичный ключ |
| fio | Varchar(30) | Фамилия студента |

Продолжение таблицы 2.10

|  |  |  |
| --- | --- | --- |
| topic | Varchar(100) | Тема диплома |
| completion | Integer | Процент выполнения |
| score | Integer | Оценка |
| deadline | Date | Дата сдачи |
| queuenumber | Integer | Номер в очереди |
| pmid | Integer | Вторичный ключ  Руководитель проекта |
| normcontrollerid | Integer | Вторичный ключ  Нормокотролер |
| reviewerid | Integer | Вторичный ключ  Рецензент |
| chairmanid | Integer | Вторичный ключ  Председатель |
| diplomorderid | Integer | Вторичный ключ  Приказ |
| specialtyid | Varchar(30) | Вторичный ключ  Специальность |
| execution | Date | Срок |
| type | Integer | Имя комиссии |
| commissioncomment | Varchar(100) | Имя комиссии |

2.3 Выводы по разделу 2

Во время выполнения раздела 2 был сформирован основной функционал, соответствующий целям дипломного проекта, для этого была спроектирована юз-кейс диаграмма, описаны роли и соответствующие им функции.

Спроектирована архитектура интернет-сервиса, описаны основные компоненты которые позволяют реализовать весь функционал описанный ранее, их технологии и назначение.

Спроектирована база данных, предназначенная для хранения данных, описаны таблицы, поля этих таблиц, и связи между таблицами.

В результате выполнения раздела 2 были сформированы необходимые данные которые позволяют перейти к разработке интернет-сервиса.

3. Разработка интернет-сервиса

Архитектура интернет-сервиса представлена в главе 2 и включает 3 основных компонента: клиентская часть интернет-сервиса, серверная часть интернет-сервиса и база данных.

3.1 Разработка клиентской части интернет-сервиса

Клиентская часть представляет из себя веб-приложение написанное на TypeScript. Для ускорения разработки был выбран фреймворк Angular. Angular - фреймворк от компании Google для создания клиентских приложений. Одной из ключевых особенностей Angular является то, что он использует в качестве языка программирования TypeScript.

Основные структурные компоненты приведены в таблице 3.1.

Таблица 3.1 — Описание структурных компонентов клиентской части

|  |  |
| --- | --- |
| Компонент | Функция |
| Папка node\_modules | Папка с npm модулями |
| Папка src | Папка с исходным кодом приложения |
| Папка public | Папка с html файлом, к которому подключаются бандлы и остальная метаинформация |
| Папка app | Папка с компонентами |
| Папка assets | Папка с картинками |
| Файл main.ts | Инициализационный файл всего приложения |
| Файл Angular.json | Файл с настройками проекта |
| Файл package.json | Файл с информацией о проекте |

Структура веб-приложения представлена на рисунке 3.1

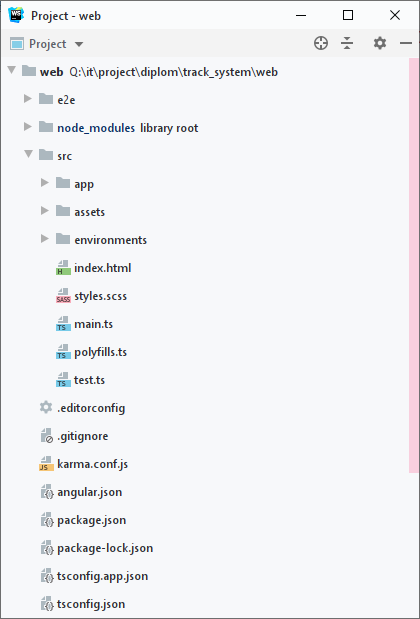


Рисунок 3.1 – Структура клиентской части приложения

Папка node\_modules хранит в себе необходимые модули для приложения, один из которых это сам Angular. Папка src содержит корневой компонент приложения, в который так или иначе подключаются другие компоненты. Папка assets содержит и прочие общие ассеты. В папке app находятся компоненты – маленькие элементы так называемого «конструктора», из которых позже собираются страницы. Файл main.ts содержит код инициализации приложения.

Остальные файлы являются конфигурационными и предназначены для разработки и развертывания, а также для написания правил проверки валидности и семантики кода.

Основные компоненты приведены в таблице 3.2.

Таблица 3.2 — Описание основных компонентов клиентской части

|  |  |
| --- | --- |
| Компонент | Функция |
| Diplom-layout | Компонент связывающий все компоненты, и пристраивающий роутинг |
| Diplom | Компонент содержащий в себе логику и отображение связанную с дипломных проектом или работой |

Продолжение таблицы 3.2

|  |  |
| --- | --- |
| diplom-chairman | Компонент содержащий в себе логику и отображение связанную с председателем |
| diplom-commission | Компонент содержащий в себе логику и отображение связанную с комиссией |
| diplom-normocontroller | Компонент содержащий в себе логику и отображение связанную с нормконтролером |
| diplom-order | Компонент содержащий в себе логику и отображение связанную с приказом |
| diplom-pm | Компонент содержащий в себе логику и отображение связанную с руководителем проекта |
| diplom-reviewer | Компонент содержащий в себе логику и отображение связанную с рецензентом |

Исходный код компонентов представлен в приложении А.

3.2 Разработка серверной части интернет-сервиса

Для реализации серверной части был выбран Go. Одной из ключевых возможностей языка Go является возможность работы с сетевыми сервисами: отправлять запросы к ресурсам в сети и, наоборот, обрабатывать входящие запросы. Основной функционал по работе с сетью представлен пакетом net. Этот пакет предоставляет различные низкоуровневые сетевые примитивы, через которые идет взаимодействие по сети.

Для упрощения маршрутизации был выбран Gorilla. Gorilla - пакет разработчика созданный для упрощения создания веб-приложений на языке Go, который, в свою очередь, включает ряд пакетов, мы будем использовать gorilla/mux который позволяет определять более сложные маршруты, которые могут использовать регулярные выражения.

Структура приложения представлена на рисунке 3.2.

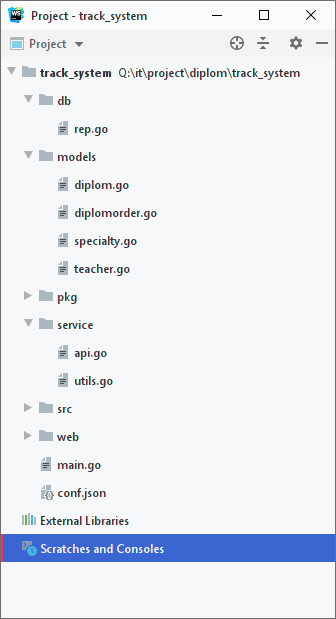


Рисунок 3.2 – Структура серверной части интернет-сервиса

Основные компоненты приведены в таблице 3.3.

Таблица 3.3 — Описание основных компонентов серверной части

|  |  |
| --- | --- |
| Компонент | Функция |
| Rep.go | Компонент описывающий работу с базой данных |
| Папка models | Содержит в себе файлы с описанием основных моделей интернет-сервиса |
| Api.go | Роутер для разделения обработки логики согласно маршруту |
| Main.go | Стартовая точка приложения |

Исходный код компонентов представлен в приложении Б.

3.3 Разработка базы данных интернет-сервиса

База данных была сформирована в разделе 2.3 и для ее реализации был создан скрипт, представленный на рисунке 3.3

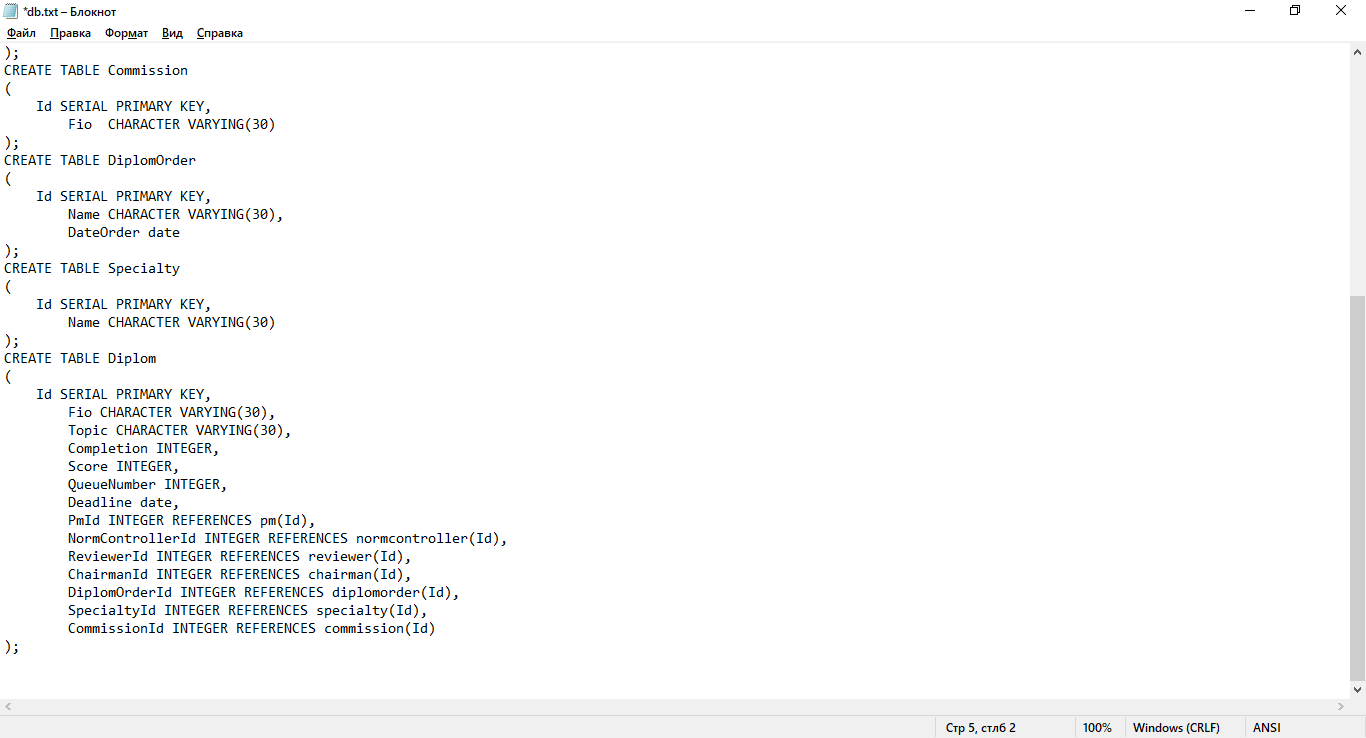


Рисунок 3.3 – Скрипт создания таблиц базы данных