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, требующую назначение темы, распределение студентов руководителям дипломных работ, назначение руководителей, комиссии, нормоконтролеров, председателей дипломной комиссии, рецензентов, а также мониторинг процесса выполнения работы студентом.

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

Для достижения цели сформированы следящие 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 интернет-сервис для учета и контроля выполнения дипломного проектирования в вузе.

Целевой аудиторий данного дипломного проекта являются заведующие кафедрой и руководители дипломных проектов.

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.

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.

1. Постановка задачи и обзор аналогичных решений

## Постановка задачи

Целью дипломного проекта является повышение эффективности работы заведующего кафедрой и руководителей дипломных проектов путем автоматизации контроля за ходом дипломного проектирования представляющего собой отдельную не простую задачу, требующую назначение темы, распределение студентов руководителям дипломных работ, назначение руководителей, комиссии, нормоконтролеров, председателей дипломной комиссии, рецензентов, а также мониторинг процесса выполнения работы студентом.

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 интернет-сервис для учета и контроля выполнения дипломного проектирования в вузе.

## Обзор аналогичных решений

Для того, чтобы создать интернет-сервис, который соответствовал бы современным стандартам, выглядел согласно новым тенденциям в области дизайна, а также имел исчерпывающий набор функционала, был изучен ИТ-рынок и проведен небольшой обзор аналогов разрабатываемого интернет-сервиса.

В этой главе описаны некоторые существующие аналоги веб-приложения для управления турами, также были поставлены цели и задачи дипломного проекта.

**GS-Ведомости: Online** - это web система, которая 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

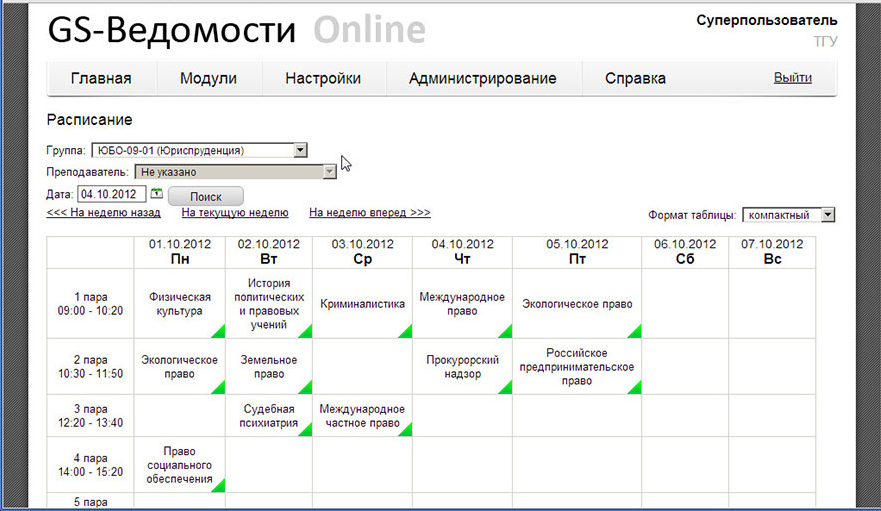


Рисунок 1.1 – Обзор аналога GS-Ведомости: Online

Преимущества приложения:

* большой функционал;
* доступ через интернет.

Недостатки приложения:

* довольно высокая цена;
* большая сложность обучения;
* утаревший дизайн.

**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.

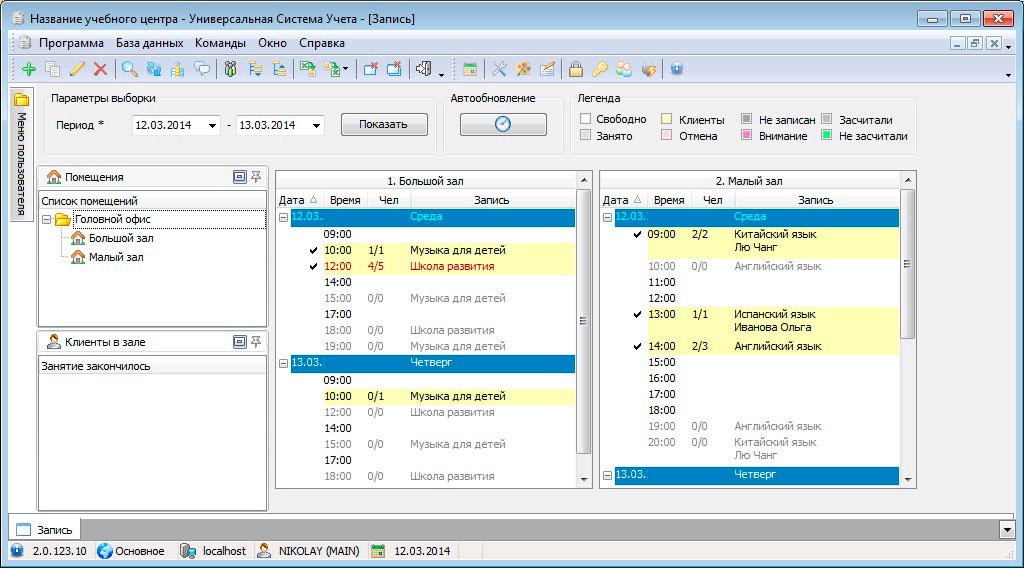


Рисунок 1.2 – Обзор аналога usu

Преимущества приложения:

* большой функционал;
* поддержка нескольких языков.

Недостатки приложения:

* высокая цена;
* устаревший дизайн;
* десктоп приложение.

## Патентный поиск по теме дипломного проекта

В соответствии с темой дипломного проекта был проведен патентный поиск в области программных систем для автоматизации учебного процесса. Для проведения патентного поиска был использован «Федеральный институт промышленной собственности» который позволяет провести поиск патентов программ для с 2013 года.

В результате патентного поиска было найдено два патента. Предметом поиска (объектом исследования или его составной части) была указана “автоматизация учебного процесса”.

Первым найденным патентом является патент системы автоматизации учебного процесса «Автор-ВУЗ» за номером RU[2016611794](https://www1.fips.ru/registers-doc-view/fips_servlet?DB=EVM&DocNumber=2016611794&TypeFile=html) от 10.02.2016, заявителем которого являются Губенко Игорь Олегович, Меренков Антон Сергеевич и Тареев Денис Сергеевич. Программа предназначена для автоматизации действий работников образовательных организаций. Программа обеспечивает выполнение следующих функций: формирование штатного приказа, анализ наличия вакантных ставок на кафедрах; создание учебных планов, анализ их на соответствие стандартам, выгрузка для отправки на экспертизу; формирование в программе методических материалов, анализ обеспеченности материалами дисциплин, анализ качества их подготовки; составление расчета разных видов нагрузки, учет фактически выполненной нагрузки, контроль выполнения норм нагрузки; составление расписаний учебных занятий, возможность выгрузки занятости преподавателей в календари; выполнение функций электронной информационно-образовательной среды. Для реализации данного программного обеспечения использовались такие языка программирования как PHP и JavaScript.

Вторым найденным патентом является автоматизированная информационная система «Управление учебным процессом» за номером RU[2019664129](https://www1.fips.ru/registers-doc-view/fips_servlet?DB=EVM&DocNumber=2016611794&TypeFile=html) от 30.10.2019. Программа предназначена для автоматизации деятельности институтов повышения квалификации и учреждений дополнительного образования. Область применения: управление учебным процессом. Функциональные возможности: ведение базы данных слушателей, прием заявок слушателей на курсы и мероприятия через личный кабинет в Интернете, формирование групп слушателей и учебно-тематических планов, составление расписания, планирование занятости аудиторий с учётом имеющегося оборудования, ведение межкурсовой подготовки, учёт преподавательского состава и их часовой нагрузки. Программа не содержит персональные данные. Тип ЭВМ: IBM PC-совмест. ПК; ОС: Windows XP/7/8/8.1/10. Для реализации данной информационной системы использовались такие языка программирования как C#, HTML, CSS и JavaScript.

## Выбор программной платформы и инструментария для разработки

Основной операционной системой для приложения была выбрана Windows Server 2016. Разработка веб-интерфейса выполняется в среде Web Storm с помощью платформы VueJS. Языком программирования для написания кода является Java script. Для хранения и предоставления данных системы разработана база данных. В данном случае выбрана реляционная база данных PostgreSQL. В качестве ключевого звена приложения, выбрано программное средство GeoServer.Для связи веб-приложения с базой данных является среда выполнения Go.

Основные инструменты, технологии, библиотеки и их назначение приведены в таблице 1.1.

Таблица 1.1 — Инструменты, технологии, библиотеки

|  |  |  |
| --- | --- | --- |
| Тип | Наименование | Назначение |
| Среда разработки | WebStorm | Написание в данной среде пользовательской части интернет-сервиса |
| Среда разработки | GoLand | Написание в данной среде серверной части интернет-сервиса |
| Среда разработки | DataGrip | Написание в данной среде базы данных для интернет-сервиса |
| Язык | TypeScript | Язык для написания пользовательской части веб-приложения |
| Язык | Go | Язык для написания сервисов для веб-приложения |
| Фреймворк | Angular | Фреймворк для упрощения и упорядочения пользовательской части веб приложения |
| Библиотека | AngularMaterial | Библиотека компонентов для клиетской части веб-приложения |

WebStorm — это одна из самых популярных сред разработки для JavaScript, разработанная на основе платформы IntelliJ IDEA.

WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями. Важным преимуществом интегрированной среды разработки WebStorm является работа с проектами (в том числе, рефакторинг кода JavaScript, находящегося в разных файлах и папках проекта, а также вложенного в HTML). Поддерживается множественная вложенность (когда в документ на HTML вложен скрипт на Javascript, в который вложен другой код HTML, внутри которого вложен Javascript) — то есть в таких конструкциях поддерживается корректный рефакторинг. Данная среда разработки доступна для Windows, OS X и Linux.

Angular — это прогрессивный фреймворк для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Angular создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления (view), что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, Angular полностью подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications), если использовать его совместно с современными инструментамии дополнительными библиотеками.

Важной концепцией Angular являются компоненты. Эта абстракция позволяет собирать большие приложения из маленьких «кусочков». Они представляют собой пригодные к повторному использованию объекты. Если подумать, почти любой интерфейс можно представить как дерево компонентов.

Angular Material — библиотека компонентов для Angular, которая является интеграция между спецификациями Angular и Material Design. Её можно легко настроить для удовлетворения всех потребностей с помощью простого API.

Преимущества Angular Material:

* компоненты интернационализированы и доступны, чтобы все пользователи могли их использовать;
* простые API, в которых невозможно запутаться;
* поведение хорошо протестировано как с модульными, так и с интеграционными тестами;
* настраивается в пределах спецификации Material Design;
* сведение к минимуму эксплуатационных затрат;
* код является чистым и хорошо документированным, чтобы служить хорошим примером для Angular разработчиков;
* поддержка большинства браузеров.

Go – компилируемый многопоточный язык программирования, разработанный внутри компании Google. Разработка Go началась в сентябре 2007 года, его непосредственным проектированием занимались Роберт Гризмер, Роб Пайк и Кен Томпсон, занимавшиеся до этого проектом разработки операционной системы Inferno. Официально язык был представлен в ноябре 2009 года. На данный момент поддержка официального компилятора, разрабатываемого создателями языка, осуществляется для операционных систем FreeBSD, OpenBSD, Linux, macOS, Windows, DragonFly BSD, Plan 9, Solaris, Android. Также Go поддерживается набором компиляторов gcc, существует несколько независимых реализаций. Ведётся разработка второй версии языка.

Язык Go разрабатывался как язык программирования для создания высокоэффективных программ, работающих на современных распределённых системах и многоядерных процессорах. Он может рассматриваться как попытка создать замену языкам Си и C++. По словам Роба Пайка, «Go был разработан для решения реальных проблем, возникающих при разработке программного обеспечения в Google».

Go создавался в расчёте на то, что программы на нём будут транслироваться в объектный код целевой аппаратной и программной платформы и в дальнейшем исполняться непосредственно, не требуя виртуальной машины, поэтому одним из критериев выбора архитектурных решений была возможность обеспечить быструю компиляцию в эффективный объектный код и отсутствие чрезмерных требований к динамической поддержке.

В результате получился язык, «который не стал прорывом, но тем не менее явился отличным инструментом для разработки крупных программных проектов».

Хотя для Go доступен и интерпретатор, практически в нём нет большой потребности, так как скорость компиляции достаточно высока для обеспечения интерактивной разработки.

## Вывод по разделу

1. Осуществлена постановка задач в которой сформированы цели дипломного проекта и задачи для ее достижения.
2. Выполнен обзор 2 аналогичных решений, выявлены основные недостатки, такие как высокая цена, устаревший дизайн, и преимущества, который является их функционал.
3. Выполнен патентный поиск который позволил выявить насколько изучена тема по автоматизации учебного процесса.
4. Выполнен выбор программной платформы и инструментария для разработки, в результате которого была выбрана платформа Angular, выбраны языки программирования TypeScript и Go. Для дизайна пользовательского интерфейса веб-приложения был выбран AngularMaterial так как он реализует принципы Google Material Design. В качестве серверной части выступает среда выполнения Go. База данных, предназначенная для хранения все нужной информации, предоставляемой веб-приложению будет использоваться PostgreSQL.

2 Проектирование интернет сервиса

2.1 Функциональное наполнение интернет-сервиса

Интернет-сервис, разрабатываемый в рамках данного дипломного проекта предназначен для создания удобной в использовании системы учета и контроля хода выполнения дипломного проектирования в вузе.

Диаграмма вариантов использования для интернет-сервиса представлена на рисунке 2.1.

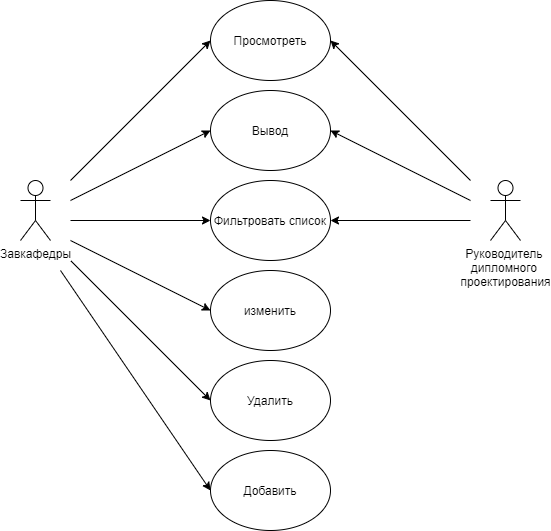


Рис.2.1 – Диаграмма вариантов использования

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 пользователям в интернет-сервисе.

Таблица 2.2 — Функции интернет-сервиса

|  |  |
| --- | --- |
| Функция | Пояснение |
| Просмотр | Просмотр дипломных проектов или работ, руководителей дипломного проектирования, комиссии, нормоконтрлеров, председателей, рецензентов, приказов и специальностей. |
| Вывод | Вывод списка дипломных проектов или работ в текстовом формате. |
| Фильтровать список | Фильтрация списка дипломных проектов или работ по приказам, специальностям, студентам, темам, руководителям, рецензентам, нормоконтролерам, председателям, комиссии, сроку исполнения, дате защиты, номеру в очереди и оценке. |
| Изменить | Изменение дипломных проектов или работ, руководителей дипломного проектирования, комиссии, нормоконтрлеров, председателей, рецензентов, приказов или специальностей. |
| Удалить | Удаление дипломных проектов или работ, руководителей дипломного проектирования, комиссии, нормоконтрлеров, председателей, рецензентов, приказов или специальностей. |
| Добавить | Добавление дипломных проектов или работ, руководителей дипломного проектирования, комиссии, нормоконтрлеров, председателей, рецензентов, приказов или специальностей. |

2.2 Архитектура интернет-сервиса

Архитектура интернет-сервиса представляет собой трехуровневую модель, включающую в себя три компонента: клиентская часть, серверная часть и база данных. На рисунке 2.2 представлена общая схема взаимодействия компонентов интернет-сервиса.

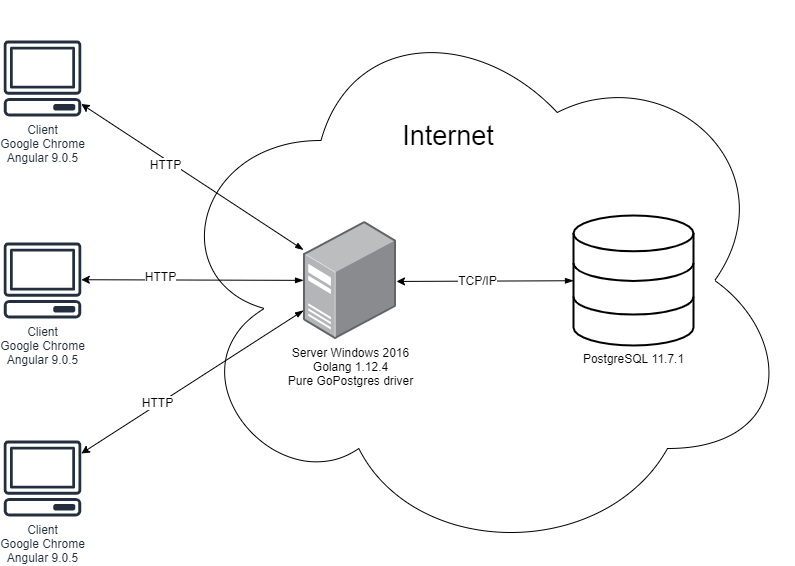


Рис.2.2 – Общая схема взаимодействия компонентов

Клиентская часть представляет из себя браузер, который при помощи http запроса обращается к серверной части, которая возвращает статические данные формирующие Angular. После инициализации веб-приложение взаимодействует с серверной частью

Серверная часть представляет из себя приложение

Интернет-сервис должен иметь серверную часть, представляющую REST-сервисы к которым будет обращаться клиентская часть интернет-сервиса, должна иметься база данных для хранения информации о дипломных проектах или работах, руководителях дипломных проектов, нормконтролерах, рецензентах, председателях, приказах, комиссии и специальностях.

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.

2.3 База данных интернет-сервиса

На рисунке 2.3 представлена логическая схема базы данных.

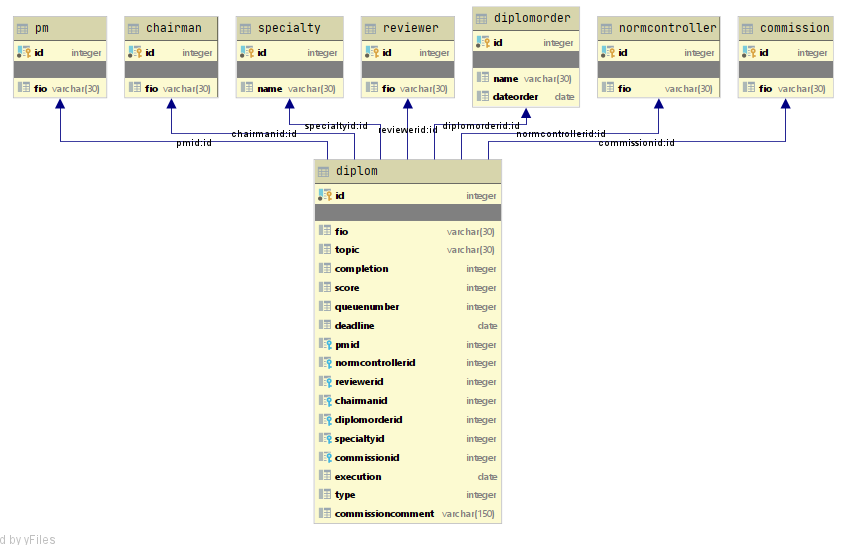


Рис.2.3 – Логическая схема базы данных

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 |
| 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 | Дата исполнения |
| type | Integer | Тип диплома |
| commissioncomment | Varchar(100) | Комментарий от комиссии |

2.4 Выводы по разделу 2

1. Разработана диаграмма вариантов использования которая включает в себя 2 роли: заведующий кафедры и руководитель дипломного проектирования, а также 6 функций: просмотреть, вывод, фильтровать список, изменить, удалить, добавить.
2. Разработана архитектура интернет-сервиса представляющая из себя трёхуровневую модель и состоит из трех частей: клинская, серверная и база данных.
3. Разработана логическая схема базы данных состоящая из 8 таблиц. Описаны поля, их типы данных и описание.

3. Разработка интернет-сервиса

Архитектура интернет-сервиса представлена в главе 2 и включает 3 основных части: клиентскую часть, серверную часть и базу 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. Структура клиентской части представлена на рисунке 3.1

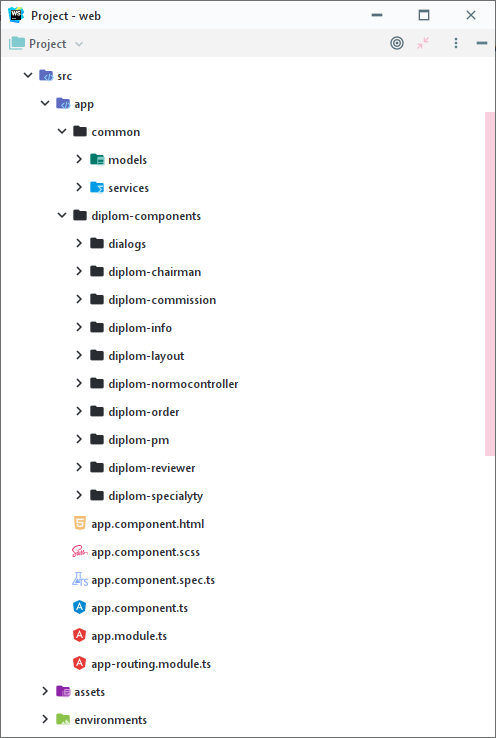


Рисунок 3.1 – Структура клиентской части интернет-сервиса

Разработка веб-приложение сводится к созданию компонентов для отображения информации в окне браузера и классов для вспомогательных функций, таких как взаимодействия по протоколу http или модели объектов. Папка common предназначена для хранения вспомогательных файлов, так в папке services находятся TypeScript файлы при помощи которых можно обращается к серверной части по протоколу http, а в папке models модели объектов. Компоненты находятся в папке diplom-components и приведены в таблице 3.1.

Таблица 3.1 — Описание основных компонентов клиентской части

|  |  |
| --- | --- |
| Компонент | Функция |
| diplom-layout | Родительский компонент, контролирующий отображение необходимого компонента при помощи роутера |
| diplom-info | Компонент содержащий в себе логику связанную с добавлением, обновление, удалением и отображением дипломных проектов или работ |
| diplom-chairman | Компонент содержащий в себе логику связанную с добавлением, обновление, удалением и отображением председателей |
| diplom-commission | Компонент содержащий в себе логику связанную с добавлением, обновление, удалением и отображением комиссии |
| diplom-normocontroller | Компонент содержащий в себе логику связанную с добавлением, обновление, удалением и отображением нормоконтролеров |
| diplom-order | Компонент содержащий в себе логику связанную с добавлением, обновление, удалением и отображением приказов |
| diplom-pm | Компонент содержащий в себе логику связанную с добавлением, обновление, удалением и отображением дипломных руководителей |
| diplom-reviewer | Компонент содержащий в себе логику связанную с добавлением, обновление, удалением и отображением рецензентов |
| diplom-specialyty | Компонент содержащий в себе логику связанную с добавлением, обновление, удалением и отображением специальностей |
| dialogs | Компонент содержащий в себе логику связанную с добавлением, обновление, удалением и отображением дипломных проектов или работ |

Исходный код компонентов представлен в приложении А.

3.2 Разработка серверной части интернет-сервиса

Для реализации серверной части был выбран Golang. Структура приложения представлена на рисунке 3.2.

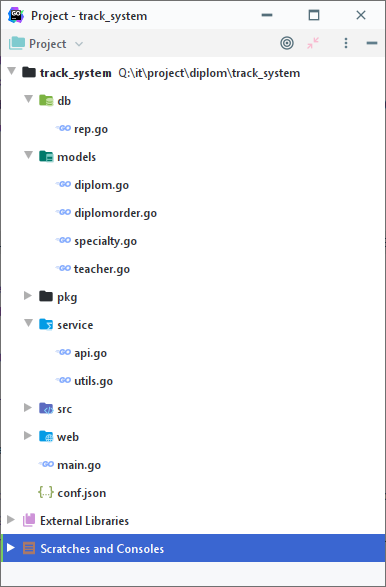


Рисунок 3.2 – Структура серверной части интернет-сервиса

Основные пакеты приведены в таблице 3.3.

Таблица 3.3 — Описание основных пакетов серверной части

|  |  |
| --- | --- |
| Компонент | Функция |
| main | Стартовая точка приложения |
| db | Компонент описывающий работу с базой данных |
| service | Роутер для разделения обработки логики согласно маршруту и создание файла регистрации |
| models | Создание необходимых структур обьектов |

Первоначально запускаемым файлом является *main.go*, это стартовая точка приложения. Первоначально создается файл регистрации при помощи InitConfiguration из пакета service для отслеживания состояния сервера после чего для обработки входящих запросов создаётся обработчик маршрутов роутера при помощи вызова RunRest из пакета service.

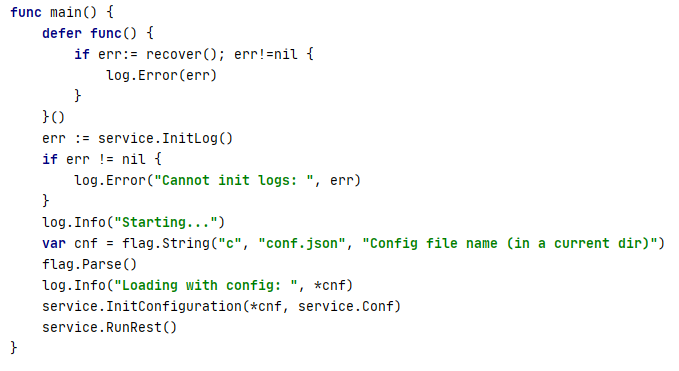


Рисунок 3.2 – Создание сервера

При вызове RunRest из пакета server, происходит создание обработчиков маршрутов. При указании в url описанных запроса происходит вызов обработчиков. Также создаётся обработчик, который обращается к серверу для раздачи статических файлов, по этому пути будет раздаваться клиентской части интернет-сервиса.

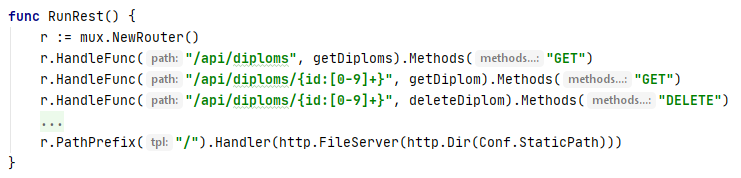


Рисунок 3.2 – Создание роутера и привязка обработчиков

Для реализации работы с базой данных в данном дипломном проекте был выбран пакет pq. Пакет pq - это драйвер Go Postgres для пакета database/sql. Для работы с базой данных с помощью pq необходимо выполнить следующие этапы.

Создать подключение к базе данных, создав строку подключения указав необходимы параметры.



Рисунок 3.3 – Создание подключения к базе данных

Данный объект подключения в дальнейшем используется для работы с базой данных. Например, для выполнения sql запроса используется метод db.Query.

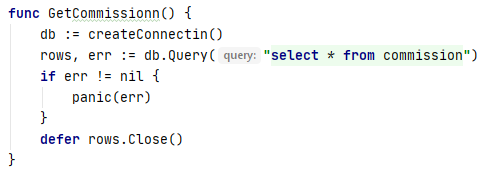


Рисунок 3.3 – Вызов sql конструкции

3.3 Разработка базы данных интернет-сервиса

База данных сформированная в разделе 2.3 имеет 8 таблиц, для реализации таблицы Diplom создан скрипт представленный на рисунке 3.3

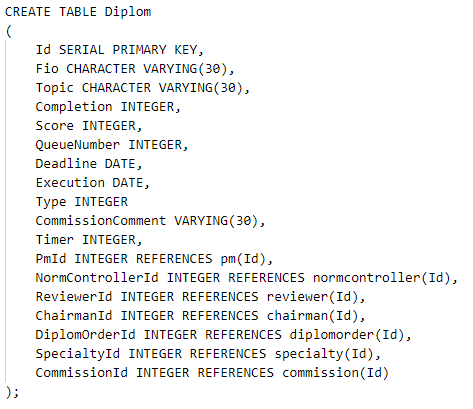


Рисунок 3.3 – Скрипт создания таблиц Diplom

Для создания таблицы Pm скрипт представлен на рисунке 3.4

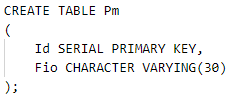


Рисунок 3.3 – Скрипт создания таблиц Pm

Для создания таблицы NormController скрипт представлен на рисунке 3.4

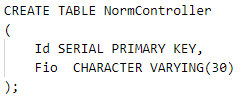


Рисунок 3.3 – Скрипт создания таблиц NormController

Для создания таблицы Reviewer скрипт представлен на рисунке 3.4

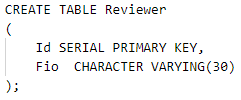


Рисунок 3.3 – Скрипт создания таблиц Reviewer

Для создания таблицы Chairman скрипт представлен на рисунке 3.4

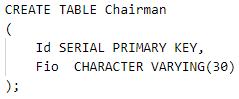


Рисунок 3.3 – Скрипт создания таблиц Chairman

Для создания таблицы Commission скрипт представлен на рисунке 3.4

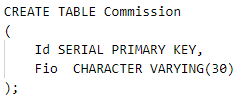


Рисунок 3.3 – Скрипт создания таблиц Commission

Для создания таблицы DiplomOrder скрипт представлен на рисунке 3.4

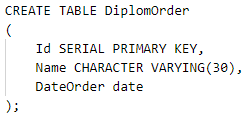


Рисунок 3.3 – Скрипт создания таблиц DiplomOrder

Для создания таблицы Specialty скрипт представлен на рисунке 3.4

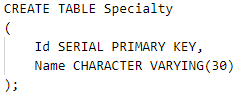


Рисунок 3.3 – Скрипт создания таблиц Specialty

Данные в базу данных будут записывается непосредственно из клиентской части при помощи серверной части, описанной в разделе 3.2.

3.4 Выводы по разделу 3

1. Разработана клиентская часть интернет-сервиса

4 Тестирование

4.1 Тестирование приложения

Для тестирования был выбран тест-кейс[4.1] план, представленный в таблице 4.1.

Таблица 4.1 — Тест-кейсы

|  |  |  |
| --- | --- | --- |
| Номер тест-кейса | Описание тест-кейса | Результат |
| 1 | Открыть интернет-сервис и отфильтровать дипломы по специальности. | Интернет-сервис открылось успешно, после фильтрации отобразились только необходимые дипломы |
| 2 | Добавить новый диплом | После заполнения формы и нажатия на кнопку добавить новый диплом появился в списке |

4.2 Вывод по разделу

Были протестированы основные функции, которые должны быть предоставлены приложением. Для тестирования был выбран метод прохождения тест-кейсов. В результате пройденных тест-кейсов можно убедится, что основной функционал интернет-сервиса выполняется.

5 Руководство программиста

В данной программе предусмотрены пользователи двух видов:

завкафедры

руководитель дипломного проектирования

Руководитель дипломного проектирования — это преподаватель, который имеет право только просматривать информацию в данном интернет-сервисе. Завкафедры обладает большим набором привилегий, таких как:

Измененить.

Удалить.

Добавить.

5.1 Просмотреть

Стартовая страница интернет-сервиса для завкафедры представлена на рисунке 5.1.

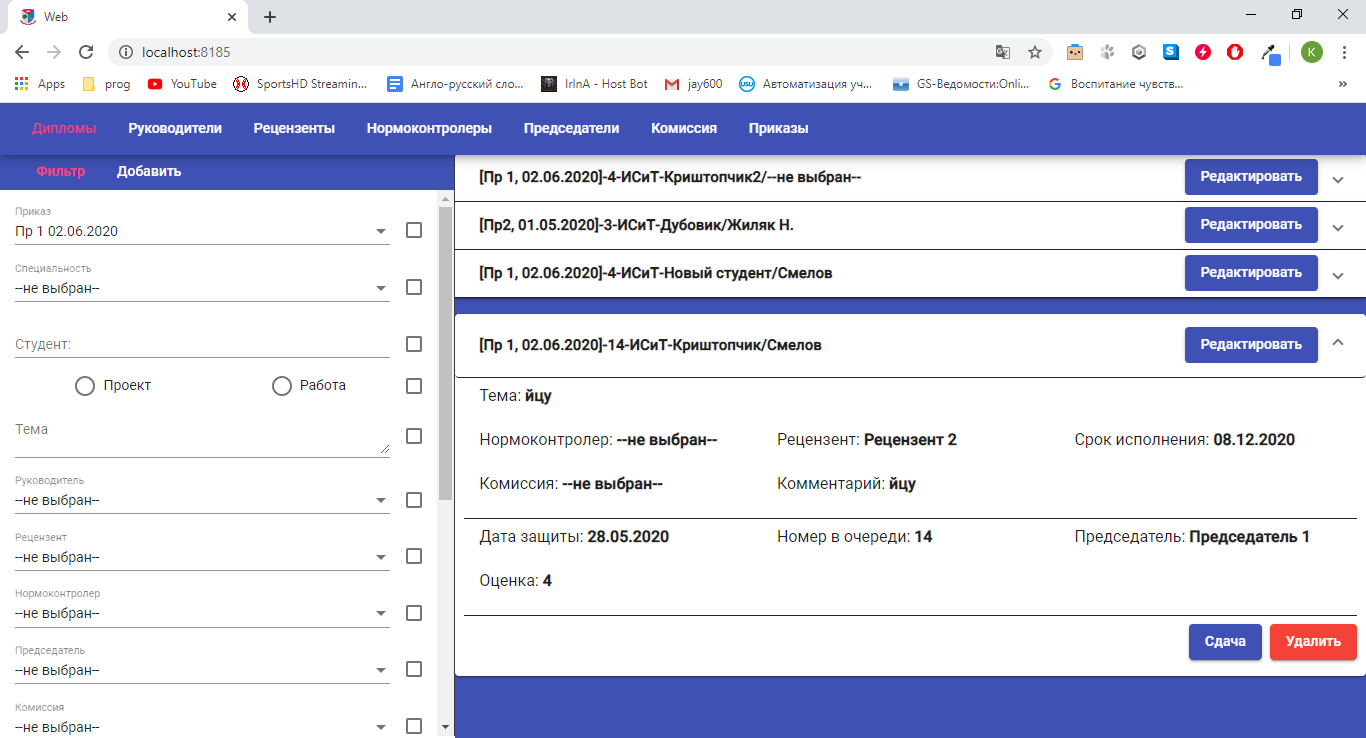


Рисунок 5.1 – Стартовая страница интернет-сервиса для завкафедры

На данной страницы, завкафедры может просмотреть имеющиеся дипломные проекты и работы, изменить их данные, удалить или добавить новые.

Стартовая страница для руководителей дипломных проектов выглядит иначе, и представлена на рисунке 5.2.

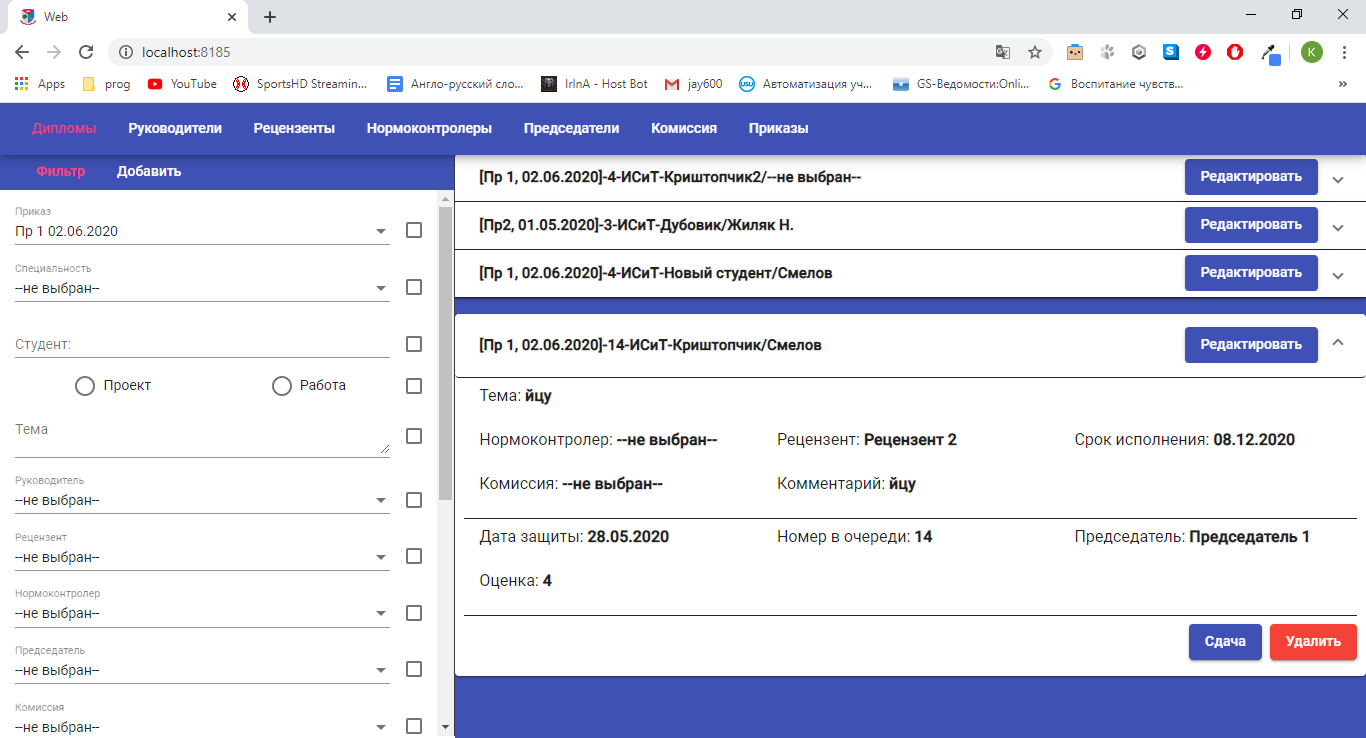


Рисунок 5.2 – Стартовая страница интернет-сервиса для руководителей дипломного проектирования

На данной страницы, руководителей дипломного проектирования может просмотреть имеющиеся дипломные проекты и работы.

Для просмотра руководителей, необходимо перейти на вкладку **Руководители**, данная вкладка представлена на рисунке 5.3.

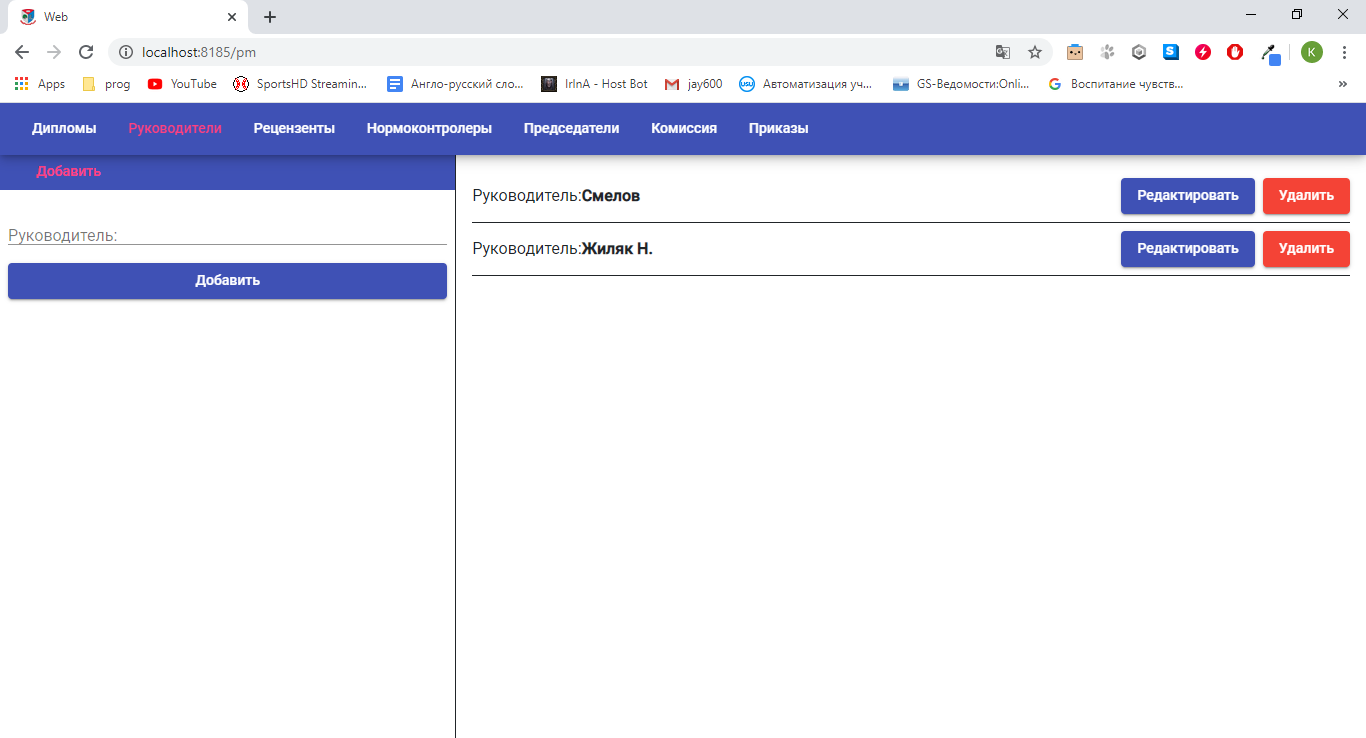


Рисунок 5.3 – Вкладка руководителей для завкафедры

На данной страницы, завкафедры может просмотреть имеющихся дипломных руководителей, изменить, удалить или добавить новых.

Вкладка руководителей для руководителей дипломных проектов выглядит иначе, и представлена на рисунке 5.4.

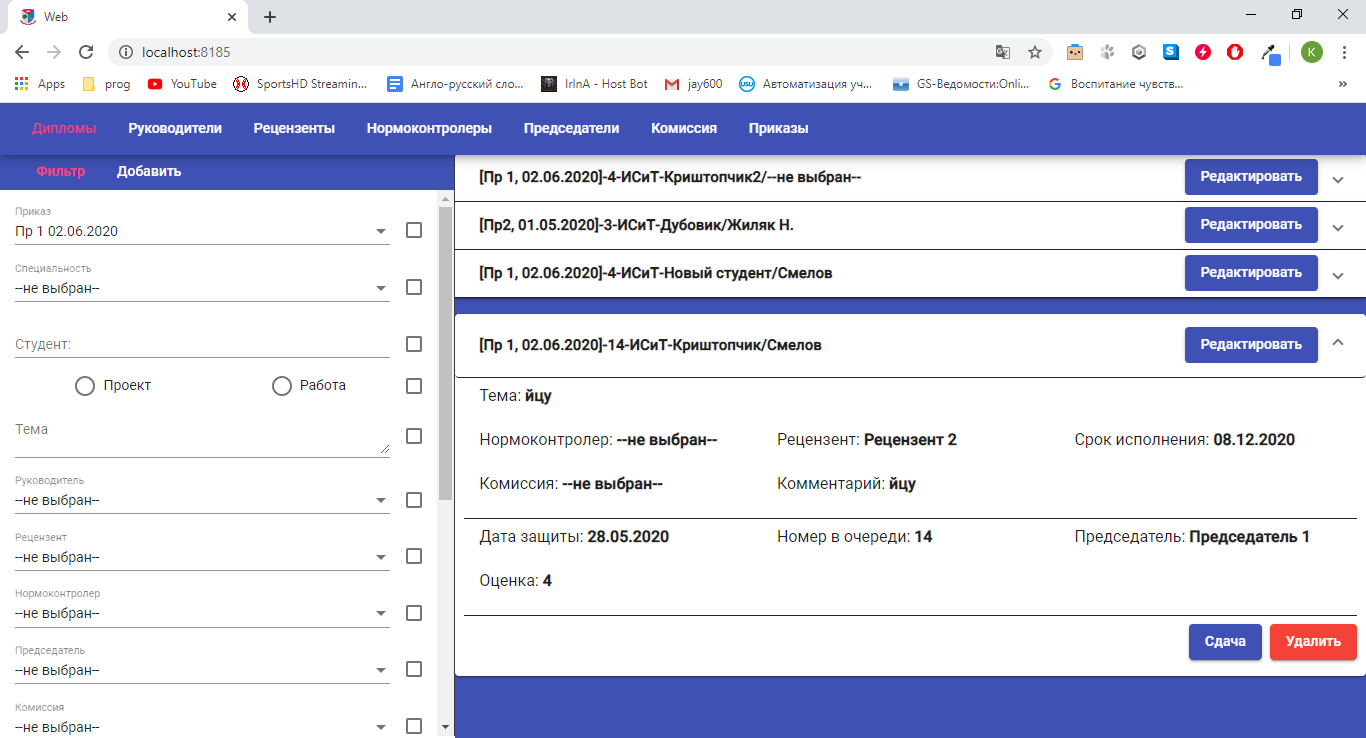


Рисунок 5.4 – Стартовая страница интернет-сервиса для руководителей дипломного проектирования

5.2 Добавить

Для добавления новой дипломов, завкафедры необходимо перейти на необходимую вкладку, после чего открыть форму для добавления которая представлена на рисунке 5.5.

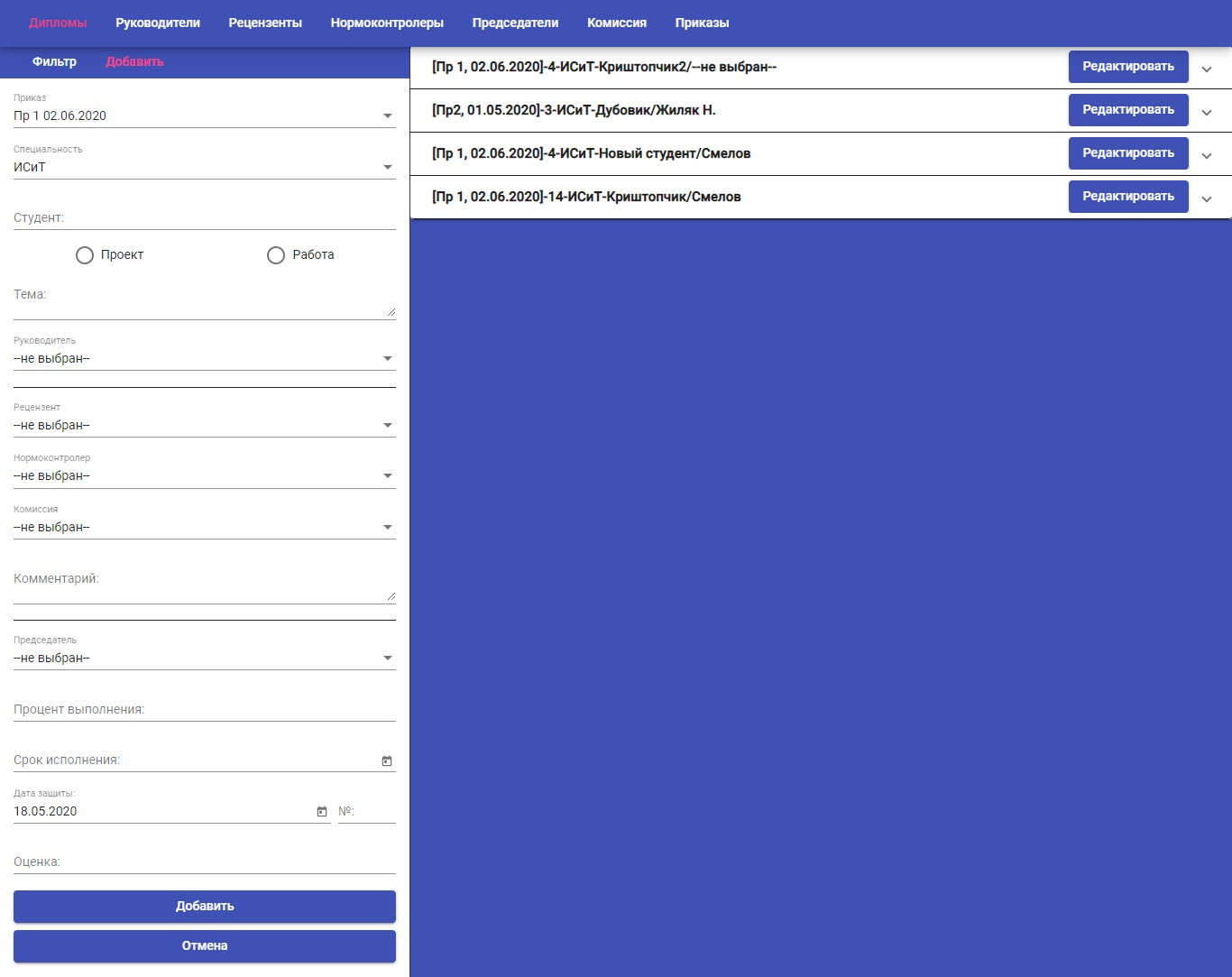


Рисунок 5.5 – Форма для добавления диплома

Заполнив необходимые поля и нажав кнопку добавить, диплом появится в списке всех дипломов.

Для добавления руководителей необходима ..

…….

…………

5.3 Вывод по разделу

1. Описан способ просмотра дипломов, руководителей, нормоконтролеров, рецензентов, председателей, комиссии, приказов в интернет-сервисе для руководителей дипломного проектирования и завкафедры.

2. Описан способ добавления новой информации, данная функция имеется только у завкафедры и включает в себя добавление новых дипломов, руководителей, нормоконтролеров, рецензентов, председателей, комиссии, приказов.