



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра практической и прикладной информатики (ИиППО)

ОТЧЁТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ
Преддипломная практика

приказ Университета о направлении на практику от «__» апреля 2023 г. №

Отчет представлен к
рассмотрению:
Студент группы ИКБО-13-19

«__» мая 2023

Заикин В.П.
(подпись и расшифровка подписи)

Отчет утвержден.
Допущен к защите:

Руководитель практики
от кафедры

«__» мая 2023

Жматов Д.В.
(подпись и расшифровка подписи)

Москва 2023 г.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра практической и прикладной информатики (ИиППО)

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРОИЗВОДСТВЕННУЮ ПРАКТИКУ
Преддипломная практика

Студенту 4 курса учебной группы ИКБО-13-19
Заикину Валентину Павловичу

Место и время практики: РТУ МИРЭА кафедра ИиППО, с 20 апреля 2023 г. по 17 мая 2023 г.

Должность на практике: студент

1. СОДЕРЖАНИЕ ПРАКТИКИ:

1.1. Изучить: научную и техническую литературу, электронные информационно-образовательные ресурсы, применяемые для профессиональной деятельности по теме практики

1.2. Практически выполнить:

1.2.1 Спроектировать бизнес-процессы исследуемой предметной области, для их автоматизации в информационной системе ИС;

1.2.2 Сформировать требования к ИС;

1.2.3 Осуществить концептуальное, функциональное и логическое проектирование клиент-серверных ИС среднего и крупного масштаба сложности;

1.2.4 Описать реализацию клиентской и серверной части ИС.

1.2.5 Описать обеспечение информационной безопасности при эксплуатации ИС;

1.2.6 Описать работу пользователя с ИС

1.2.7. Оценить технические характеристики практической разработки;

1.2.8. Выполнить тестирование разрабатываемого проекта, собрать метрики и оценить результат проведенного тестирования.

1.3. Ознакомиться: с актуальными нормативно-правовыми документами, международными и отечественными стандартами, с СУБД и инструментальными средствами разработки ИС

2. ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ: Подготовить презентацию и доклад по результатам практики.

3. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ: в процессе практики рекомендуется использовать периодические издания и отраслевую литературу годом издания не старше 5 лет от даты начала прохождения практики

Руководитель практики от кафедры
«20» апреля 2023 г.

Подпись (Жматов Д.В.)

Задание получил
«20» апреля 2023 г.

Подпись (Заикин В.П.)

СОГЛАСОВАНО:
Заведующий кафедрой:

«20» апреля 2023 г.

Подпись (Болбаков Р.Г.)

Проведенные инструктажи:

Охрана труда:

«20» апреля 2023 г.

Инструктирующий

Подпись

Болбаков Р.Г., зав.
кафедрой
ИиППО
Заикин В.П.

Инструктируемый

Подпись

Техника безопасности:

«20» апреля 2023 г.

Инструктирующий

Подпись

Болбаков Р.Г., зав.
кафедрой
ИиППО
Заикин В.П.

Инструктируемый

Подпись

Пожарная безопасность:

«20» апреля 2023 г.

Инструктирующий

Подпись

Болбаков Р.Г., зав.
кафедрой
ИиППО
Заикин В.П.

Инструктируемый

Подпись

С правилами внутреннего распорядка ознакомлен:

«20» апреля 2023 г.

Подпись Заикин В.П.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

**РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

студента Заикина В.П. 4 курса группы ИКБО-13-19 очной формы обучения, обучающегося по направлению подготовки 09.03.04 Программная инженерия, профиль «Разработка программных продуктов и проектирование информационных систем».

Неделя	Сроки выполнения	Этап	Отметка о выполнении
1	20.04.2023	Подготовительный этап, включающий в себя организационное собрание (Вводная лекция о порядке организации и прохождения производственной практики, инструктаж по технике безопасности, получение задания на практику)	
1	21.04.2023- 26.04.2023	Аналитический этап (разработка логической модели базы данных; формирование таблиц сущностей с атрибутами и типами данных; описание входной информации, применяемых классификаторов и справочников; построение макетов выходной информации; описание математического обеспечения)	
2	27.04.2023	Представление руководителю структурированного материала согласно содержанию выше указанного этапа	
2	28.04.2023- 03.05.2023	Технологический этап (Обоснование выбора средств разработки; описание реализации базы данных (логическая и физическая модели базы данных); построение экранных форм)	
3	04.05.2023	Представление руководителю материала согласно содержанию выше указанного этапа, реализуемого на 2 неделе; а также иного ранее непредставленного материала предыдущих этапов в случае наличия отметок о невыполнении	

3	05.05.2023- 10.05.2023	Технологический этап (Описание реализации клиентской части ИС (дерево функций, сценарий диалога, структурная схема); описание работы пользователей с ИС; описание обеспечения информационной безопасности эксплуатации ИС)	
4	11.05.2023	Представление руководителю материала согласно содержанию выше указанного этапа, реализуемого на 3 неделе; а также иного ранее непредставленного материала предыдущих этапов в случае наличия отметок о невыполнении	
4	14.05.2023	Представление руководителю предварительной версии отчета с обеспечением согласованности материала по всем его частям, полученных на предыдущих этапах	
4	15.05.2023- 16.05.2023	Подготовка окончательной версии отчета по практике (Оформление материалов отчета в полном соответствии с требованиями на оформление письменных учебных работ студентов)	
4	17.05.2023	Представление окончательной версии отчета по практике руководителю	

Руководитель практики от
кафедры

_____/ Жматов Д.В., к.т.н., доцент/

Обучающийся

_____/Заикин В.П./

Согласовано:

Заведующий кафедрой

_____/Болбаков Р.Г., к.т.н., доцент/

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	7
ВВЕДЕНИЕ.....	8
1. Аналитический раздел	9
1.1. Разработка логической модели данных.....	9
1.2 Основные объекты в базе данных.....	10
2. Технологический раздел	11
2.1. Разработка клиентской части системы.....	11
2.2. Разработка серверной части информационной системы.....	11
2.3. Структура проекта.....	12
2.4. Конфигурационные классы приложения	12
2.4.1. Базовые классы приложения	12
2.4.2. Прикладные классы приложения	14
2.5. Тестирование приложения	15
ЗАКЛЮЧЕНИЕ	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	18

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

АИС — Автоматизированная информационная система;

ИС — Информационная система;

БД — База данных;

ВВЕДЕНИЕ

Одним из качественных показателей вуза является уровень использования информационных технологий, процессы их развития и совершенствования. Автоматизированные информационные системы (АИС) обеспечивают четкость функционирования и совершенствование управленческих механизмов[3]. Практика успешной автоматизации организационно-управленческих процессов вузов постоянно расширяется.

Эффективное построение единого информационного пространства вуза, комплексной АИС во многом определяется выбором соответствующей технологической платформы, конкретных программных решений. При создании единого информационно-образовательного пространства вуза одной из самых сложных и жизненно важных частей работы на сегодняшний день является сбор планово-отчетной документации обучающихся.

Целью работы является разработка автоматизированной информационной системы для сбора, обработки и представления планово-отчетной документации обучающихся.

Для достижения цели выпускной квалификационной работы обозначены следующие задачи:

- проведение анализа предметной области, в том числе существующих аналогов;
- определение и формализация принципов проектирования и разработки информационных систем в сфере образования;
- проектирование архитектуры информационной системы
- определение информационных, технических, программных средств для разработки программного приложения;
- разработка информационной системы

1. Аналитический раздел

1.1. Разработка логической модели данных

На рисунке 1.1 изображена спроектированная логическая схема разрабатываемой информационной системы.

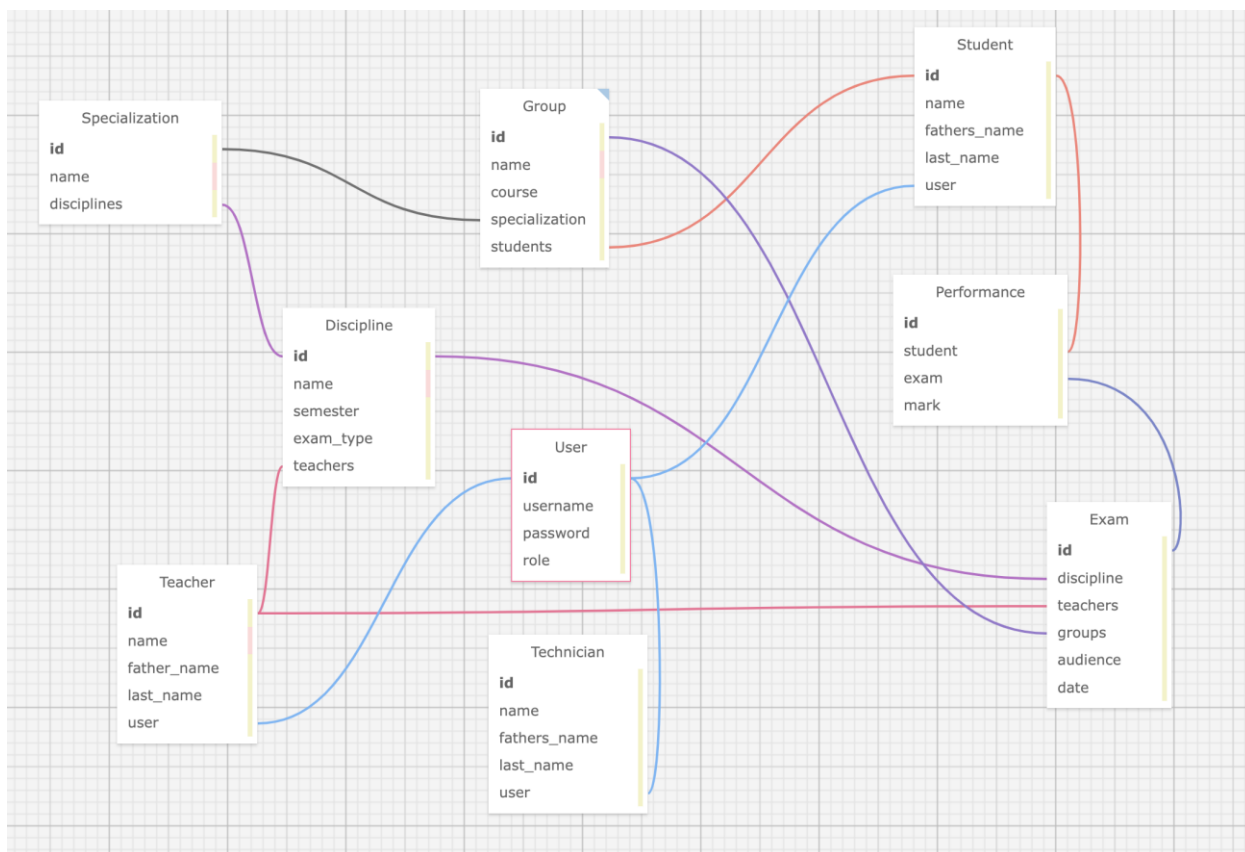


Рисунок .1 – Логическая модель ИС

Логическая модель базы данных состоит из 9 сущностей:

- Специализация (Specialization). Хранит в себе информацию об учебном плане и дисциплинах, которые ему принадлежат;
- Дисциплина (Discipline). Хранит в себе информацию о дисциплинах университета и о преподавателях данной дисциплины;
- Группа (Group). Хранит в себе информацию об учебной группе и студентах, которые в нее входят;
- Студент (Student). Хранит в себе информацию о конкретном студенте: ФИО и пользователе с авторотационными данными;
- Преподаватель (Teacher). Хранит в себе информацию о преподавателе;

1.2 Основные объекты в базе данных

Определим основные объекты проектируемой системы. Поскольку Веб-сайт в сфере планетологии нацелен на пользователей, то они и являются основными объектами системы.

О пользователях в системе необходимо хранить следующую информацию:

- ID пользователя;
- фамилию;
- имя;
- отчество.

Всю модерацию и изменение информации проводят админы, поэтому информация о них хранится в отдельной таблице. Она содержит следующие поля:

- ID админа;
- фамилия;
- имя;
- отчество.

Также необходимо содержать данных о дисциплинах ВУЗа

- Код дисциплины
- Название
- Кафедра
- преподаватели

2. Технологический раздел

2.1. Разработка клиентской части системы

При разработке интерфейса были созданы шапка страницы на которой располагаются кнопки: преподаватели, предметы.

При нажатии на любую другую появляется список, отображающий элементы, в зависимости от нажатой кнопки. На рисунке 1.1 показан вывод списка учителей, а также панель редактирования учителя. Идентичным образом происходит навигация и по остальным элементам.

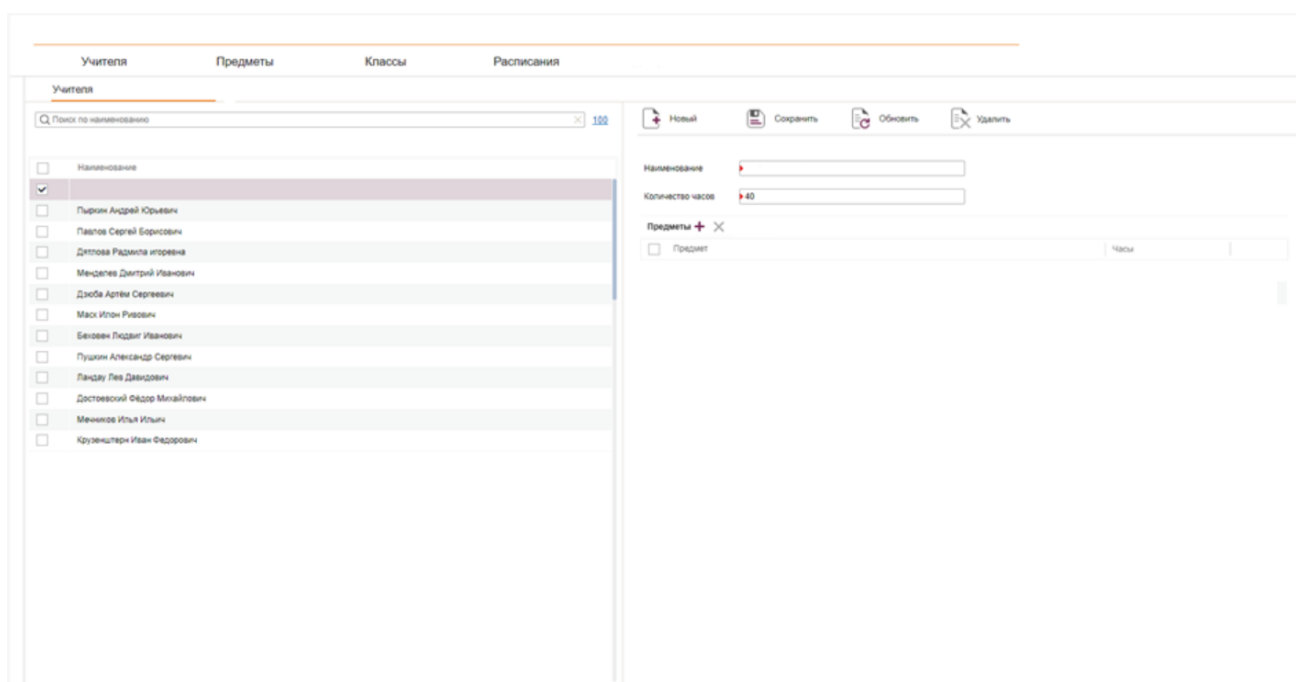


Рисунок 2.1 – Интерфейс информационной системы

2.2. Разработка серверной части информационной системы

Выбор технологий разработки серверной части ИС

Для разработки серверной части был выбран Spring Boot. Spring делает программирование на Java более быстрым, простым и безопасным. Гибкий и набор расширений и сторонних библиотек Spring позволяет разработчикам создавать практически любые приложения. Spring Boot же позволяет упростить процессы конфигурации и развертывания приложения.

2.3. Структура проекта

Для удобства разработки и повышения простоты расширения разделили проект на несколько пакетов. Структура проекта показана на рисунке 1.2.

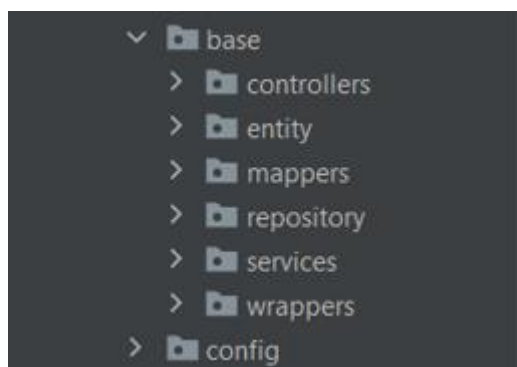


Рисунок 2.2 – Структура проекта

В пакете base находятся базовые классы, реализующих функционал, который переиспользуется, что позволяет не дублировать уже существующий код. Пакет config содержит классы, отвечающие за механизм авторизации и отображение веб-страниц.

2.4. Конфигурационные классы приложения

Пакет config содержит классы MvcConfig и WebSecurityConfig. Класс MvcConfig [8] отвечает за отображение веб-страниц и работу с ресурсами веб-интерфейса, имплементируя методы addViewControllers и addResourceHandlers интерфейса WebMvcConfigurer фреймворка Spring Web MVC. Класс WebSecurityConfig [8] наследуется от класса WebSecurityConfigurerAdapter фреймворка Spring Security и реализует процесс авторизации, переопределяя метод configure. В WebSecurityConfig настраиваются права доступа без авторизации для адресов «/login/**», а все остальные только после того, как введен логин и пароль. Для обеспечения доступа пользователям к ресурсам сервера используется встроенный в Spring механизм – базовая HTTP аутентификация.

2.4.1. Базовые классы приложения

Пакет base хранит классы отвечающий за базовый функционал работы сервера. Классы, реализованные в пакете base показаны на рисунке 3.4.

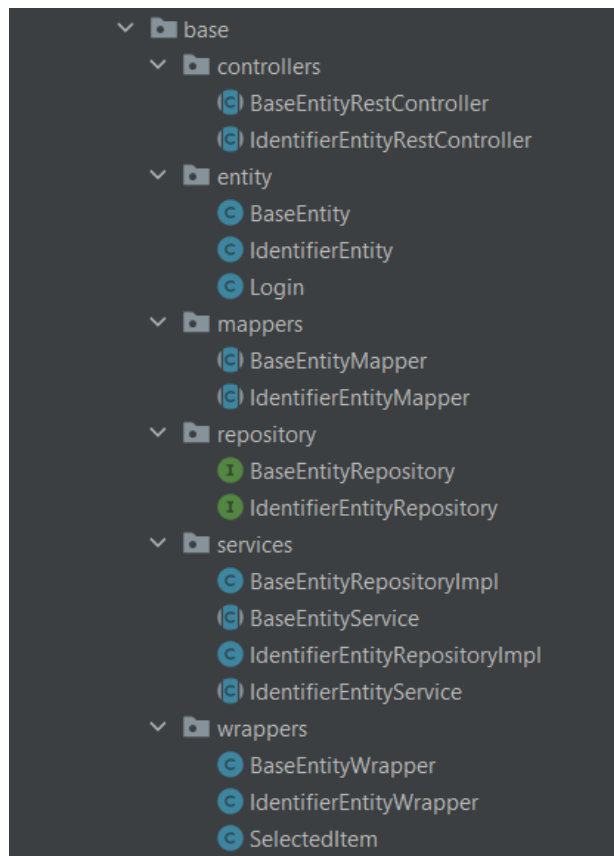


Рисунок 2.3 – Подробная структура пакета base

В пакете `base.entity` находятся базовые классы для сущностей, такие как `IdentifierEntity` с полем `id`, и его наследник `BaseEntity`, в котором добавлено поле `label`. Поле `id`, содержащее первичный ключ, помечено аннотацией `@Id`, которая отвечает за работу с первичным ключом таблицы, и аннотацией `@GeneratedValue(strategy = GenerationType.SEQUENCE)` для того, чтобы поле `id` последовательно генерировалось базой. Поле `label` из `BaseEntity` помечено аннотацией `@Column`, означающей, что поле `label` соответствует атрибуту сущности БД. Эти классы помечены аннотацией `@MappedSuperclass`, позволяющей включить класс и JPA аннотации в производный класс.

Также в `base.entity` хранится сущность `Login`, наследуемая от `BaseEntity`, которая отвечает за хранение информации о логине и пароле пользователя.

Доступ к БД осуществляется посредством встроенного в фреймворк Spring механизма репозитория. Они представляют собой набор интерфейсов, наследующих интерфейс `JpaRepository`.

Интерфейсы `IdentifierEntityRepository` и `BaseEntityRepository` из `base.repository` наследуются от `JpaRepository`. Эти интерфейсы вместе со своими имплементациями `IdentifierEntityRepositoryImpl` и `BaseEntityRepositoryImpl` расширяют `JpaRepository` новыми методами поиска информации по сущностям в БД.

Пакет `base.services` содержит классы, реализующие базовую бизнес-логику. В целом, работа сервисов заключается в обращении к репозиториям для получения данных и преобразовании этих данных для передачи в контроллеры. Аналогично репозиториям, каждому сервису соответствует сущность.

При создании серверного приложения использовался архитектурный стиль REST, который подразумевает использование протокола HTTP и предоставление списка URL-адресов, с помощью которых сервер может принимать запросы на получение, сохранение, обновление и удаление данных. В пакете `base.controllers` реализованы базовые контроллеры, которые содержат несколько URL-адресов, каждый из которых привязан к определенному методу контроллера с помощью следующих аннотаций `@GetMapping` (для GET-запросов), `@PostMapping` (для POST-запросов) и т.п. Получив запрос на определенный адрес, контроллер вызывает метод, по этому адресу. Также как репозитории и сервисы, контроллеры ассоциированы с сущностями.

Абстрактные классы из пакета `base.mappers` отвечают за сопоставление сущностей классам обёрток, расположенным в пакете `base.wrappers`, для этого у классов сущностей и обёрток должны быть реализованы методы `get` и `set` для полей, которые будут сопоставляться друг другу, а так же написать в классе маппере абстрактный метод, в сигнатуре которого будут указаны классы сущности и обёртки. Для явного указания целевого класса при сопоставлении используется аннотация `@MappingTarget`.

2.4.2. Прикладные классы приложения

Для классов сущностей нужно указать аннотации `@Entity` и `@Table`, чтобы определить для фреймворка Spring, что данные классы являются сущностями, в котором соответствует таблица в БД. Над классами-сервисами нужно указать

аннотацию `@Service` [18], а над классами Mappers аннотацию `@Mapper`. Для контроллеров нужно прописать аннотации `@RestController`, которая обозначит соответствующие классы, как контроллеры, и аннотации `@RequestMapping` с указанием адреса конкретного контроллера. В соответствии с требованиями бизнес-логики приложения в разработанные классы были добавлены дополнительные методы, которых нет в базовых классах.

2.5. Тестирование приложения

Нагрузочное тестирование – это тестирование работы системы с реальной рабочей нагрузкой.

Тестирование было проведено с помощью программы ApacheBench . С её помощью можно узнать максимальное количество запросов, обрабатываемых в одно и то же время на веб-сервере.

Результат выполнения стресс тестирования представлен на рисунке 2.5.

```
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:
Server Hostname:      localhost
Server Port:          8000

Document Path:        /
Document Length:       36803 bytes

Concurrency Level:     50
Time taken for tests:   107.168 seconds
Complete requests:     1000
Failed requests:        0
Total transferred:     37974000 bytes
HTML transferred:      36803000 bytes
Requests per second:    9.33 [#/sec] (mean)
Time per request:       5358.377 [ms] (mean)
Time per request:       107.168 [ms] (mean, across all concurrent requests)
Transfer rate:          346.04 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    0  0.3      0    1
Processing: 96 5118 723.2   5170 5919
Waiting:    95 5118 723.3   5170 5918
Total:      97 5119 723.2   5170 5919

Percentage of the requests served within a certain time (ms)
 50%  5170
 66%  5344
 75%  5401
 80%  5454
 90%  5598
 95%  5640
 98%  5803
 99%  5905
100%  5919 (longest request)
```

Рисунок 2.4 – Нагрузочное тестирование

ЗАКЛЮЧЕНИЕ

В ходе выполнения практики на основе анализа предметной области были сформулированы требования к разрабатываемой информационной системе, обоснована необходимость разработки и разработана информационная система для управление планово-отчетной документации обучающихся

Для достижения поставленной цели была проведена следующая работа:

- 1) Проанализированы принципы построения информационных систем в сфере образования;
- 2) Сформулированы основные функциональные и нефункциональные требования к разрабатываемому продукту;
- 3) Смоделирована функциональная и логическая схемы информационной системы контроля процесса промежуточной аттестации;
- 4) Разработаны программный и пользовательский интерфейсы;
- 5) Проведено тестирование разработанной информационной системы;

Все вышеописанные задачи были выполнены. Поставленная цель в полной мере достигнута. информационная система для сбора, обработки и представления планово-отчетной документации обучающихся реализована в полном объеме в соответствии с указанными в техническом задании требованиями.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Проектирование интерфейса [Электронный ресурс]. — URL: <https://skillbox.ru/media/design/proektirovanie-interfeysa-8-printsipov/> - (дата обращения 05.04.22).
3. ORM [Электронный ресурс]. — URL: <https://simpleone.ru/glossary/orm/> (Дата обращения 11.04.2023).
4. Уоллс К. Spring в действии. — М.: ДМК Пресс, 2013 — 752 с.: ил. ISBN 978-5-94074-568-6.
5. Бинарный поиск [Электронный ресурс]. — URL: <https://prog-cpp.ru/search-binary/> - (Дата обращения 03.04.23).
6. Сложность бинарного поиска [Электронный ресурс]. — URL: project.algowiki.org/ru/Двоичный_поиск - (Дата обращения 01.04.23).
7. Tasty mocking framework for unit tests in Java [Электронный ресурс]. — URL: <https://site.mockito.org/> (Дата обращения 11.04.2023).