

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

---

Кафедра «Прикладная математика и информатика»  
(наименование)

09.03.03 Прикладная информатика  
(код и наименование направления подготовки, специальности)

---

Корпоративные информационные системы  
(направленность (профиль) / специализация)

---

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка информационной системы интеграции сервисов учебного центра»

Студент

Е.Н. Моисеева

---

(И.О. Фамилия)

(личная подпись)

Руководитель

к.п.н., Е.А. Ерофеева

---

(ученая степень, звание, И.О. Фамилия)

Консультант

А. В. Москалюк

---

(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

---

Кафедра «Прикладная математика и информатика»  
(наименование)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

Студент Моисеева Екатерина Николаевна 4 курс гр. ПИп-1703а.

1. Тема «Разработка информационной системы интеграции сервисов учебного центра».
2. Срок сдачи студентом законченной выпускной квалификационной работы 27.06.2021.
3. Исходные данные к выпускной квалификационной работе разработать информационную систему для сотрудников учебного центра ООО «Неткрэкер» и его студентов.
4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов):

Введение;

Глава 1 Функциональное моделирование предметной области;

1.1 Технико-экономическая характеристика предметной области;

1.2 Концептуальное моделирование предметной области;

1.3 Анализ существующих разработок на предмет соответствия сформулированным

требованиям;

1.4 Постановка задачи на разработку проекта создания ИС;

1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»;

Выводы по главе 1;

Глава 2 Логическое проектирование ИС;

2.1 Выбор технологии логического моделирования ИС;

2.2 Логическая модель и описание ИС;

2.3 Информационное обеспечение ИС;

2.4 Проектирование БД ИС;

2.5 Требования к аппаратно-программному обеспечению ИС;

Выводы по главе 2;

Глава 3 Физическое проектирование ИС;

3.1 Выбор архитектуры ИС;

3.2 Выбор технологии разработки программного обеспечения ИС;

3.3 Выбор СУБД ИС;

3.4 Разработка физической модели данных ИС;

3.5 Разработка программного обеспечения ИС;

3.6 Описание функциональности ИС;

3.7 Тестирование программного проекта;

Выводы по главе 3;

Заключение;

Список используемой литературы и источников.

5. Ориентировочный перечень графического и иллюстративного материала иллюстрация диаграмм, скриншоты программного кода и системы, а также презентация.
6. Консультант по разделам Москалюк Александра Вячеславовна.
7. Дата выдачи задания «28» 12 2020 г.

Руководитель выпускной  
квалификационной работы

\_\_\_\_\_  
(подпись)

Е.А. Ерофеева  
(И.О. Фамилия)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование)

**КАЛЕНДАРНЫЙ ПЛАН**  
**выполнения выпускной квалификационной работы**

Студент Моисеева Екатерина Николаевна 4 курс группа ПИп-1703а  
по теме «Разработка информационной системы интеграции сервисов учебного центра».

Наименование работ	Плановый срок выполнения	Фактический срок выполнения	Отметка о выполнении	Подпись руководителя
Изучение предметной области	31.10.2020	31.10.2020		
Сбор требований	16.11.2020	15.11.2020		
Поиск и исследование литературы по технологиям разработки	20.11.2020	19.11.2020		
Составление плана работы	27.11.2020	27.11.2020		
Разработка и редактирование первой главы	15.01.2021	10.01.2021		
Проектирование системы	16.01.2021	16.01.2021		
Разработка и редактирование второй главы	15.03.2021	15.03.2021		
Разработка информационной системы	30.04.2021	30.04.2021		
Тестирование и отладка информационной системы	12.05.2021	12.05.2021		
Разработка и редактирование третьей главы	10.05.2021	10.05.2021		

Наименование работ	Плановый срок выполнения	Фактический срок выполнения	Отметка о выполнении	Подпись руководителя
Подготовка доклада и графического материала	20.05.2021	20.05.2021		
Проведение предварительной защиты ВКР	21.05.2021	21.05.2021		
Составление аннотации	07.06.2021	03.06.2021		
Исправление замечаний	15.06.2021	11.06.2021		

Руководитель выпускной  
квалификационной работы

\_\_\_\_\_

(подпись)

Е.А. Ерофеева  
(И.О. Фамилия)

## Аннотация

79 страниц, 53 рисунка, 14 таблиц, 45 источников, 10 приложений

### ИНФОРМАЦИОННАЯ СИСТЕМА ИНТЕГРАЦИИ СЕРВИСОВ УЧЕБНОГО ЦЕНТРА.

Разработан проект информационной системы интеграции сервисов учебного центра для автоматизации работы сотрудников компании ООО «Неткрэкер» и студентов учебного центра.

В работе представлено описание ООО «Неткрэкер» и его учебного центра. Был выполнен бизнес-анализ процессов подразделения, выявлены основные проблемы. Построены модели «как есть» и «как должно быть» с использованием методологии BPMN. Сформулированы цель и задачи проектирования и требования к информационной системе. Произведен выбор и обоснование методов проектирования информационной системы, проектных решений по СУБД и архитектуре системы, а также рассмотрен вопрос о разграничение прав доступа к ресурсам информационной системы.

Построена функциональная модель подразделения «как должно быть» по методологии UML. Описано программное обеспечение информационной системы интеграции существующих сервисов компании для пользователей, а также описана возможность удаленной коммуникации по электронной почте и мессенджеру Telegram. Было проведено автоматизированное и ручное системное тестирование, где проводились ручные проверки по 43 тестовым сценариям в двух браузерах Microsoft Edge и Google Chrome.

Работа находится на стадии приемочного тестирования. После данного этапа будет рассматриваться вопрос о реализации второго этапа разработки и внедрении программного решения.

## **Abstract**

The graduation work is devoted to developing an information system for integrating services of the training center of the IT company OOO Netcracker.

The company has created training centers to make a novice specialist a ready-made candidate for vacant positions. Students study on different platforms that are not connected to each other. In this regard, the company needs to create an information system for integrating the services of the training center. The company also requested to add a communication channel via the Telegram messenger.

To solve this problem, the following tasks were established: analysis of the training center's activities on the example of the company's Moscow and Togliatti offices; collection of requirements and design of the conceptual and logical model of the information system; implementation and testing of the system.

In order to analyze business processes of the training center the "AS-IS" models were built using the BPMN methodology and then the "TO-BE" models were built using the UML methodology. The logical modeling of the database was performed using the IDEF1X notation.

Thus the information system based on the constructed models was implemented using the framework ASP.NET Core in the C# programming language, and a chatbot for communication was implemented using the Python programming language. Microsoft SQL Server was chosen as the database management system.

To draw the conclusion, it should be observed that manual system testing showed a good indicator of the quality of the system. At the moment the system is undergoing acceptance testing at the customer.

In the future, it is planned to introduce it into use, expand the functionality and add new communication channels. In the long run, this system can help the company improve its learning and student engagement processes.

## Оглавление

Введение.....	4
Глава 1 Функциональное моделирование предметной области .....	5
1.1 Техничко-экономическая характеристика предметной области.....	5
1.2 Концептуальное моделирование предметной области .....	6
1.2.1 Выбор технологии концептуального моделирования предметной области.....	6
1.2.2 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ» и обоснование необходимости ИС.....	7
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям.....	9
1.4 Постановка задачи на разработку проекта создания ИС .....	11
1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ» .....	15
Глава 2 Логическое проектирование ИС .....	18
2.1 Выбор технологии логического моделирования ИС .....	18
2.2 Логическая модель и описание ИС.....	19
2.3 Информационное обеспечение ИС .....	23
2.3.1 Используемые классификаторы и системы кодирования .....	23
2.3.2 Характеристика нормативно-справочной и входной оперативной и выходной информации .....	25
2.4 Проектирование БД ИС.....	27
2.5 Требования к аппаратно-программному обеспечению ИС .....	29
Глава 3 Физическое проектирование ИС .....	32
3.1 Выбор архитектуры ИС.....	32



3.2 Выбор технологии разработки программного обеспечения ИС .....	34
3.3 Выбор СУБД ИС .....	36
3.4 Разработка физической модели данных ИС .....	37
3.5 Разработка программного обеспечения ИС .....	38
3.5.1 Схема взаимосвязи модулей приложения ИС.....	38
3.5.2 Описание модулей приложения ИС с примерами программного кода.....	40
3.6 Описание функциональности ИС .....	47
3.7 Тестирование программного проекта.....	48
Заключение .....	54
Список используемой литературы и источников .....	55
Приложение А Диаграмма BPMN «как-есть» процесса заочного обучения .....	60
Приложение Б Диаграмма BPMN «как-должно быть» процесса заочного обучения .....	61
Приложение В Диаграмма последовательности набора на курсы .....	62
Приложение Г Диаграмма последовательности заочного обучения.....	63
Приложение Д Описание классов .....	64
Приложение Е Дизайн-макеты форм ввода и вывода информационной системы.....	67
Приложение Ж Физическая диаграмма базы данных.....	72
Приложение И Описание графа сценариев.....	73
Приложение К Скриншоты страниц информационной системы.....	74
Приложение Л Список тестовых кейсов и результат их выполнения.....	77

## Введение

В настоящее время сложно найти кандидатов с высоким уровнем компетенций, связанные с предметной областью компании. Проблема удержания и создания квалификационных специалистов становится всё острее и острее. Так IT-компания ООО «Неткрэкер» создала учебные центры, чтобы сделать из начинающего специалиста готового кандидата на вакантные должности. Компания имеет свои учебные центры в разных городах России и стран СНГ. Часть процессов не автоматизирована во многих филиалах, студенты обучаются на разных платформах, которые между собой не связаны. В этой связи компания нуждается в создании информационной системы по интеграции сервисов учебного центра.

Объект исследования – учебный центр компании «Неткрэкер», а предметом – процессы по обучению и коммуникации со студентами и кандидатами.

Цель данной работы – спроектировать информационную систему для учебного центра IT-компания «Неткрэкер» с помощью CASE-технологии анализа и проектирования ИС, а также реализовать ее и провести тестирование.

Поставлены следующие задачи: анализ деятельности учебного центра на примере офисов компании в Москве и Тольятти; сбор требований и проектирование концептуальной и логической модели информационной системы; реализация первого этапа разработки и тестирование системы.

В первой главе будет представлена организация, ее учебный центр и его деятельность на примере офисов в Москве и Тольятти. Будут составлены диаграммы бизнес-процессов подразделения. Во второй главе будут составлены логической модели системы и базы данных. А в третьей: выбор архитектуры и системы управления базами данных, реализация первого этапа разработки и тестирование системы.

## **Глава 1 Функциональное моделирование предметной области**

### **1.1 Техничко-экономическая характеристика предметной области**

ООО «Неткрэкер» – международная компания-разработчик решений на основе информационных технологий для операторов связи, специализирующаяся на создании, внедрении и сопровождении систем поддержки операций (OSS), бизнеса (BSS).

Штаб-квартира расположена в США, но также офисы присутствуют в других странах, в том числе и в России. По исследованию Gartner несколько лет назад компанию выделили лидером в OSS и BSS решениях [5].

Также у компании есть свой учебный центр, где проходят курсы по разным направлениям. Обучаться могут разные люди, преимущественно студенты технических направлений, которые прошли собеседование [28].

Поскольку компания международная, у нее есть несколько офисов по миру и России. У каждого офиса есть свой менеджер, под его управлением местный отдел по работе с персоналом со своим руководителем и менеджерами. Менеджер отдела по работе с персоналом проводит набор IT специалистов на курсы учебного центра и активно взаимодействует с центром и продвигает его в студенческой среде.

Учебный центр сформирован в отдельную структуру компании. Его руководитель занимается административными задачами, развитием и продвижением центра, обновлением программного обеспечения, документооборотом.

Старший куратор производит набор кураторов из сотрудников компании, отвечает за программы обучения, разрабатывает расписание занятий и выпускных мероприятий. Куратор работает напрямую с каждым студентом, является его ментором, помогает усвоить программу и отвечает на вопросы вне занятий.

Несколько раз в год учебный центр объявляет набор в группы на учебные курсы. Заявки на них могут подать студенты и выпускники технических специальностей. После отбора кандидаты становятся студентами учебного центра. Они проходят заочно курсы и тесты на специальной платформе (системе 1), выполняют задания на внутренней платформе по автоматизированной проверке заданий (системе 2), и от компании назначаются кураторы, с которыми студенты общаются и создают проекты. Также администратор учебного центра делает срез по успехам студентов и сообщает результаты своим коллегам.

## **1.2 Концептуальное моделирование предметной области**

### **1.2.1 Выбор технологии концептуального моделирования предметной области**

Для разработки концептуальной модели была выбрана нотация BPMN. BPMN (нотация и модель бизнес-процессов) – графическая нотация моделирования бизнес-процессов, является промежуточным звеном между формализацией, визуализацией и воплощением бизнес-процесса.

Такая нотация представляет собой описание графических элементов, используемых для построения схемы протекания бизнес-процесса. Она нужна, чтобы:

- выстроить в соответствии с ней бизнес-процесс и регламентировать его для всех участников;
- провести автоматизацию бизнес-процессов в соответствии с имеющейся схемой [14].

Сравнение средств моделирования приведено в таблице 1.

Таблица 1 – Сравнение средств моделирования

Критерий	ELMA BPM [22]	Microsoft Visio [39]	BPMN Studio[4]
Свободно распространяемое ПО	+	-	+
Онлайн доступ	-	-	+
Документирование элементов	+	+	+
Итого	2	1	3

Программное обеспечение для разработки концептуальной модели выбиралась по нескольким критериям: свободно распространяемое ПО (программное обеспечение), онлайн доступ и документирование элементов. Из таблицы видно, что больше всего плюсов набрала программа BPMN Studio.

### **1.2.2 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ» и обоснование необходимости ИС**

Менеджер отдела по работе с персоналом проводит набор студентов на данные курсы, обрабатывает заявки, которые приходят на почту. Затем он вручную пишет письма будущим студентам, чтобы подтвердить желание и назначить интервью. После интервью сотрудники отбирают кандидатов.

В процессе обучения подключаются кураторы и студенты. Кураторы готовят материалы для курса, снимают видео, делают презентации, тесты и выкладывают в систему 1. Также в системе 2 можно найти задания, которые автоматически проверяются данной системой. Во время учебы кураторы проводят лекции и тестирование по материалу. Так студенты УЦ несколько месяцев проходят курсы заочно, и они вместе с кураторами создают проекты. После чего студенты могут стать выпускниками учебного центра и пройти собеседование в компанию на вакантную должность.

Сначала менеджер по работе с персоналом открывает прием заявок на курсы, кандидаты подают заявки на сайте, потом менеджер вручную

отправляет всем кандидатами письма на электронную почту о дате собеседования. По информации менеджеров на этом этапе уходят около 30 % кандидатов, поскольку они не смотрят почту и не знают о дате собеседования, а иногда ошибаются адресом почты, и информация до них не доходит.

После собеседования менеджер также отправляет сообщение на почту о начале курсов. Здесь также есть риск того, что новый студент УЦ пропустит данное письмо. На рисунке 1 представлена диаграмма BPMN процесса набора студентов на курсы учебного центра «как есть».

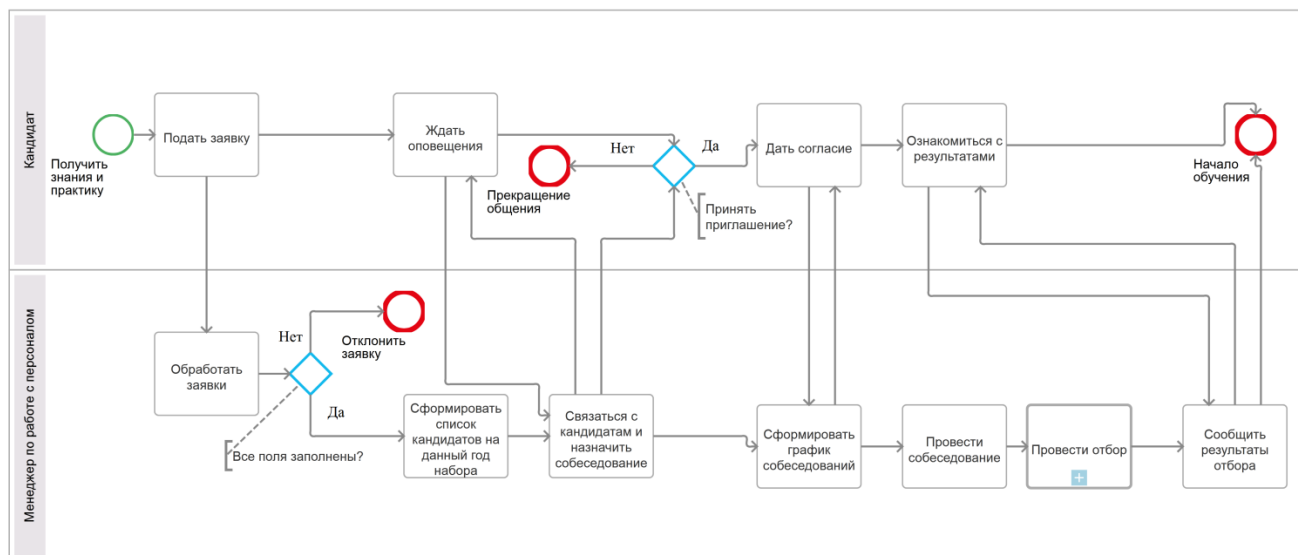


Рисунок 1 – Диаграмма BPMN процесса набора студентов на курсы

В процессе заочного обучения администратор открывает курсы и задания учебным группам. Студент УЦ изучает теорию и выполняет тесты в системе 1, прodelывает задания в системе 2. Системы сохраняют результаты студентов, а администратор может их выгрузить и сформировать отчет вручную, чтобы получить рейтинг студентов, который после публикации доступен всем.

В приложении А, на рисунке А.1 представлена диаграмма процесса обучения, где более подробно раскрыт данный процесс. Студенту УЦ

необходимо проходить аутентификацию в двух системах с разными учетными записями, чтобы выполнить необходимые элементы обучения.

Администратор вручную пишет письма-напоминания студентам, сопоставляет результаты с двух сервисов и формирует рейтинг, который обновляется всего 3 раза в месяц.

После проведенного исследования можно сказать, что в данный момент учебных центров города Москвы и Тольятти есть следующие недостатки:

- нечастое обновление рейтинга студентов для среза по контролю знаний;
- высокая трудоемкость обработки информации, порядка 400 студентов обучаются в потоке;
- несовершенство процессов оповещения, информирования кандидатов и студентов.

Компании необходима информационная система, которая разрешит данные проблемы.

### **1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям**

На рынке существует большое количество программных средств по управлению обучением (LMS). Это программное приложение для администрирования учебных курсов в рамках дистанционного обучения [23].

Для компании важны следующие критерии:

- управление большим количеством студентов;
- существование нескольких каналов коммуникации, интеграция существующих сервисов;
- создание новостей и формирование рейтинга и вывод баллов.

Можно сказать, что программное обеспечение рассматривалось по вышеуказанным критериям.

Для сравнения были выбраны программы из рейтинга на основе отзывов пользователей систем [27]:

- Looop – образовательная платформа для эффективной работы команды учебной организации;
- Learn Amp – платформа для образования и взаимодействия, улучшающая производительность команды учебной организации;
- iSpring Learn – облачная система управления обучением для обучения и оценки сотрудников или студентов в Интернете, российская разработка.

Сравнение LMS программ приведено в таблице 2.

Таблица 2 – Сравнение LMS

Критерий	Looop [45]	Learn Amp [38]	iSpring Learn [18]	Разрабатываемое приложение
Управление большим количеством студентов	+	+	+	+
Существование нескольких каналов коммуникации	-	-	-	+
Интеграция существующих сервисов	-	-	-	+
Создание новостей	+	+	+	+
Формирование рейтинга и вывод баллов	+	+	+	+
Итого	3	3	3	5



Из сравнительной таблицы видно, что существующие разработки имеют более общий формат, а разрабатываемое приложение возьмет уже существующие наработки и объединит их в личный кабинет студента и сотрудников компании.

#### **1.4 Постановка задачи на разработку проекта создания ИС**

Настоящее техническое задание распространяется на разработку информационной системы, которая должна отвечать требованиям сотрудников компании. ИС предназначена для автоматизации отбора кандидатов и обучения студентов. В базе должны содержаться сведения о персонале, кандидатах, студентах, курсах, сообщениях.

##### **1.4.1 Наименование и область применения**

###### **1.4.1.1 Наименование**

Информационная система «NC EDUcation».

###### **1.4.1.2 Область применения**

Данная разработка предназначена для применения в учебном центре компании ООО «Неткрэкер».

##### **1.4.2 Основание для разработки**

###### **1.4.2.1 Основание**

Необходимо автоматизировать процесс отбора и обучения в учебном центре компании.

###### **1.4.2.2 Тема разработки**

Разработка информационной системы «Учебный центр».

###### **1.4.2.3 Исполнитель:**

Моисеева Екатерина Николаевна.

##### **1.4.3 Назначение разработки**

Программа предназначена для использования работниками компании для автоматизации процесса отбора кандидатов, обучения и контроля успеваемости студентов. Также программа предназначена для

автоматической отправки сообщений студентам по нескольким каналам коммуникации.

#### **1.4.4 Технические требования к программе**

##### **1.4.4.1 Требования к функциональным характеристикам**

###### **1.4.4.1.1 Функциональные требования**

Программа должна обеспечивать возможность выполнения следующих функций:

- отправка сообщений по нескольким каналам коммуникации;
- обновление рейтинга из двух сервисов в автоматическом и ручном режиме;
- в одной системе формировать список модулей и заданий в автоматическом и ручном режиме;
- создание, редактирование и вывод новостей.

###### **1.4.4.1.2 Исходные данные**

Файл с данными о сотрудниках, студентах и кандидатах, программа обучения, готовые задания и ссылки на них.

###### **1.4.4.2 Требования к надежности**

В разрабатываемой системе необходимо предусмотреть следующие меры защиты:

- контроль вводимой информации;
- разграничение прав доступа, защиту от несанкционированного доступа посредством паролей;
- автоматического сохранения изменений после завершения транзакций.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем

операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы.

#### **1.4.4.3 Условия эксплуатации**

Минимальное количество персонала, требуемого для работы программы, должно быть не менее 3 штатных единиц – системный программист и конечный пользователь программы – работник компании и студент.

Системный программист должен иметь высшее техническое образование. Конечные пользователи программы должны обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы и работы в Интернете.

#### **1.4.4.4 Требования к составу и параметрам технических средств**

В состав технических средств для пользователей должен входить персональный компьютер, включающий в себя:

- процессор Intel Pentium 4 или более поздней версии;
- оперативную память объемом 4 Гбайт и жесткий диск объемом 50 Гбайт и выше;
- манипулятор типа «мышь»;
- операционную систему Windows 8 и выше;
- браузер Google Chrome или Microsoft Edge.

Для серверов приложения и базы данных:

- процессор Intel Core i-5, 2 ядра и 4 потока или более поздней версии;
- оперативную память объемом 8 Гбайт и жесткий диск объемом 1 ТБайт и выше.

#### **1.4.4.5 Требования к информационной и программной совместимости**

Системные программные средства, используемые программой, должны быть представлены локализованной версией операционной системы минимум Windows 8.

#### **1.4.4.6 Требования к маркировке и упаковке**

Не предъявляются.

#### **1.4.4.7 Требования к транспортированию и хранению**

Не предъявляются.

#### **1.4.4.8 Специальные требования**

Программа должна быть снабжена графическим интерфейсом.

#### **1.4.5 Техничко-экономические показатели**

Ориентировочная экономическая эффективность не рассчитывается. Предполагаемое число использования системы сотрудниками – ежедневное использование программы, за исключением выходных дней, в течение рабочего дня. Студенты имеют возможность круглосуточного использования.

#### **1.4.6 Стадии и этапы разработки**

##### **1.4.6.1 Стадии разработки**

Разработка должна быть проведена в три стадии: разработка технического задания, рабочее проектирование, внедрение.

##### **1.4.6.2 Этапы разработки**

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены следующие этапы работ: изучение предметной области, проектирование системы, разработка программы и программной документации, тестирование и отладка программы. Затем идет внедрение программы.

### 1.4.7 Порядок контроля и приемки

После проведения испытаний в полном объеме, на основании «Протокола испытаний» утверждают «Свидетельство о приемке», после чего программный продукт считается принятым.

### 1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»

На рисунке 2 показана диаграмма BPMN процесса приема «как должно быть». Когда кандидат заполняет заявку, он указывает e-mail и/или Telegram, чтобы получать информацию.

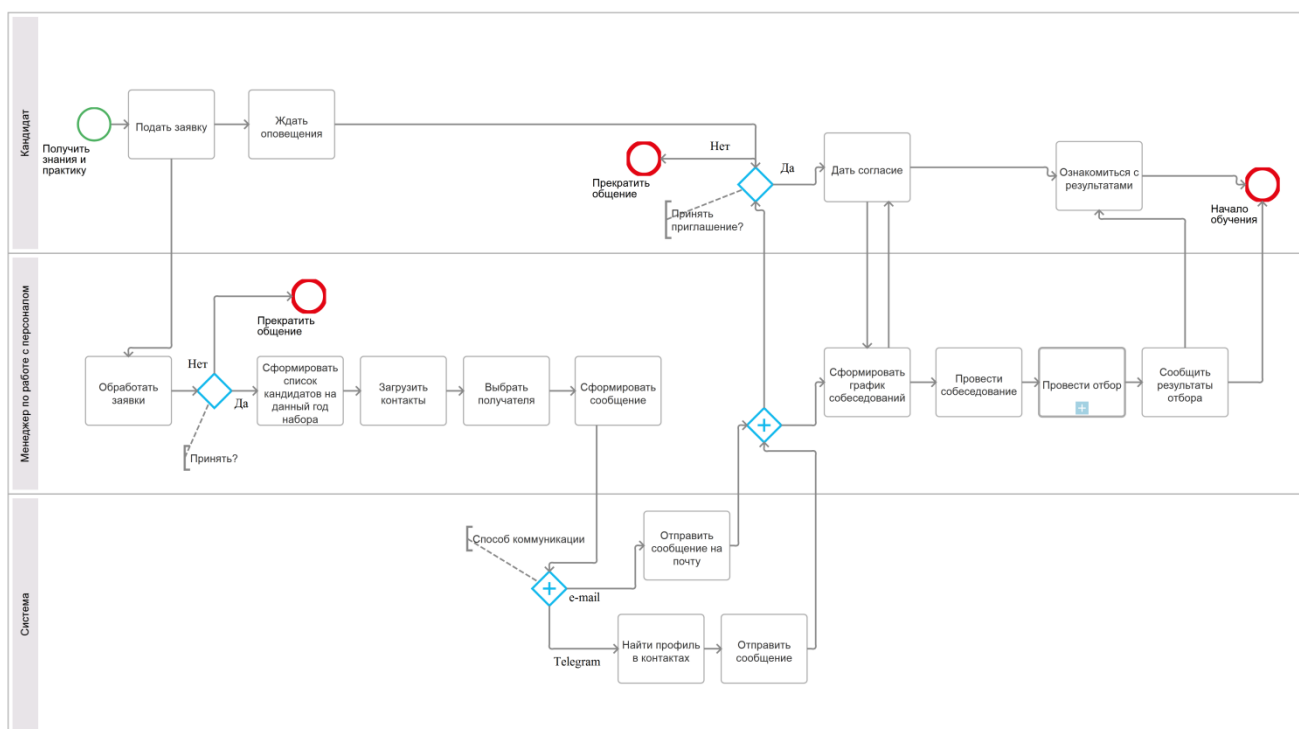


Рисунок 2 – Диаграмма BPMN процесса набора студентов на курсы

В настоящее время большая часть студентов УЦ родилась в конце 90-х и начале 2000-х годов. Электронной почтой они пользуются реже, для них актуальнее мессенджеры. Благодаря мессенджеру организация получает

ссылку на действующий аккаунт. Сообщения в мессенджерах прочитывают в 3,6 раз чаще, чем рассылки в электронной почте [3].

Можно выделить, что стоит добавить коммуникацию через мессенджер. На данном этапе был выбран Telegram, поскольку:

- у него богатый функционал;
- заботится о приватности, что важно для компании;
- есть возможность запустить бесплатные стикеры;
- есть возможность быстро и несложно создать бота, что полезно в маркетинговых целях [20].

Менеджер обрабатывает заявку, с помощью системы высылает сообщение в нужный канал коммуникации. Так он может сообщить о собеседовании и начале обучения. Здесь уменьшается риск того, что человек пропустит сообщение.

В приложении Б, на рисунке Б.1 показана диаграмма процесса заочного обучения. Здесь система автоматически отправляет сообщения студентам, объединяет всю программу курса в одном месте. А также она может формировать и публиковать рейтинг по запросу администратора.

## **Выводы по главе 1**

Были выделены основные бизнес-процессы учебного центра компании ООО «Неткрэкер». Изначально в компании некоторые процессы исполнялись людьми вручную. Например, менеджер по работе с персоналом вручную отправляет письма на электронную почту кандидатам и студентам учебного центра. Данные о почтовых адресах берутся из заявок кандидатов.

Есть риск, что кандидат или студент не прочитает письмо, так редко проверяет почту или неправильно указал электронный адрес. Коммуникация по мессенджеру снижает этот риск. Для менеджеров по работе с персоналом

данная система поможет отправлять сообщения по разным каналам коммуникации по запросу.

Администратор задает параметры по доступу к курсам в разрабатываемой системе, которая затем передает параметры в сервисы компании. После этого автоматически рассылаются приглашения и анонсы студентам. А также система автоматически формирует отчеты и рейтинг по результатам прохождения тестов и заданий студентов. Администратор также может отправлять сообщения по заданным каналам коммуникации по запросу.

Было проведено сравнение нескольких существующих программ, они имеют более общий формат, а разрабатываемая система возьмет уже существующие наработки и объединит их в личный кабинет студента и сотрудника УЦ. Информационная система должна взять эти функции на себя под управлением человека (администратора).

## Глава 2 Логическое проектирование ИС

### 2.1 Выбор технологии логического моделирования ИС

Для разработки логической модели была выбрана нотация UML – унифицированный язык моделирования. В нем использует система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. А также его можно использовать для визуализации, спецификации, конструирования и документирования программных систем [35].

Сравнение программ моделирования приведено в таблице 3.

Таблица 3 – Сравнение средств моделирования

Критерий	StarUML [42]	Microsoft Visio [39]	Enterprise Architect [36]
Свободно распространяемое ПО	+	-	-
Большое количество возможных диаграмм	+	-	+
Документирование элементов	+	+	+
Итого	3	1	2

Программное обеспечение для разработки концептуальной выбиралось по нескольким критериям:

- свободно распространяемое ПО;
- большое количество возможных диаграмм;
- документирование элементов.

Из таблицы видно, что больше всего плюсов набрала программа StarUML.



## 2.2 Логическая модель и описание ИС

На рисунке 3 и 4 изображены диаграммы вариантов использования системы. На данных диаграммах показаны актеры и их прецеденты:

- менеджер по работе с персоналом и администратор – сотрудники УЦ;
- студент – студент УЦ;
- разрабатываемая система (РС) – проект информационной системы;
- система 1 – система компании, где хранятся данные о тестах и лекциях с автоматической проверкой;
- система 2 – система компании с автоматической проверкой заданий по программированию.

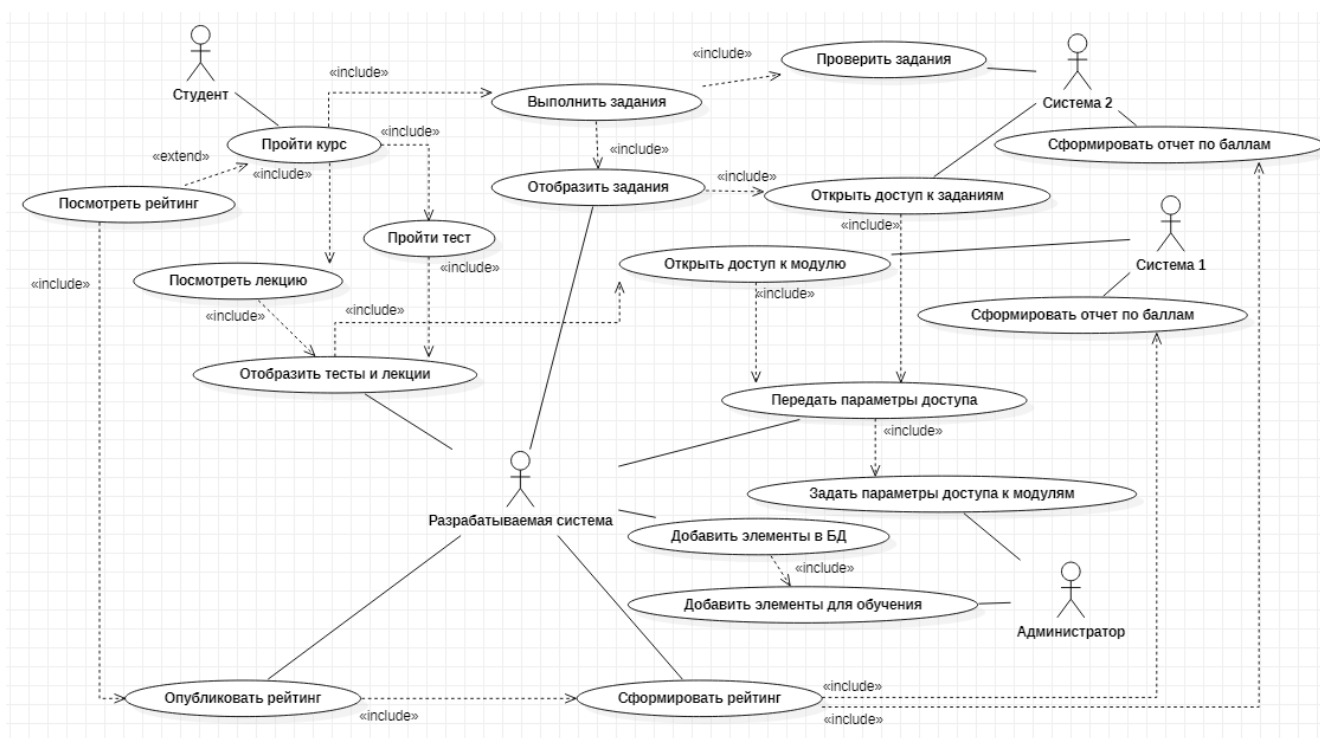


Рисунок 3 – Диаграмма вариантов использования «Прохождение учебного курса»

Администратор открывает доступ к модулям курса в РС, Студент просматривает модули и задания в РС, переходит по ссылкам, чтобы выполнить элементы курса в Системе 1 и Системе 2.

В таблице 4 представлено описание главный раздел сценария варианта использования «Прохождение учебного курса», а в таблице 5 – исключения.

Таблица 4 – Типичный ход событий сценария выполнения варианта использования «Прохождение учебного курса»

Действия актеров	Отклик системы
1. Администратор задает параметры для доступа к курсу Исключение №1: Параметры не заданы	2. РС передает параметры доступа в Систему 1 и Систему 2 3. Система 1 и Система 2 открывают доступ к заданиям по заданным параметрам
4. Администратор добавляет элементы для обучения вручную	5. РС добавляет элементы в базу данных
6. Студент проходит курс	7. РС отображает лекции, тесты и задания в зависимости от параметров доступа из Системы 1 и Системы 2
8. Студент выполняет лекции, тесты и задания в Системе 1 и Системе 2 9. Система 1 и Система 2 формируют отчет о выполнении, передают его в РС Исключение №2: Администратор редактирует баллы	10. РС формирует рейтинг из Системы 1 и Системы 2 11. РС публикует рейтинг для всех пользователей
12. Студент может посмотреть рейтинг	13. РС отображает рейтинг

Таблица 5 – Исключения сценария выполнения варианта использования «Прохождение учебного курса»

Действия актеров	Отклик системы
Исключение №1. Параметры не заданы	
1. Администратор не задает параметры для доступа к курсу	2. РС выводит сообщение о том, что параметры введены не верно

Продолжение таблицы 5

Действия актеров	Отклик системы
Исключение №2. Администратор хочет исправить баллы	
9. Система 1 и Система 2 формируют отчет о выполнении, передают его в РС, администратор редактирует баллы	10. РС обновляет рейтинг по новым значениям баллов

Также администратор или менеджер по работе с персоналом выбирают получателя, пишут сообщение и выбирают канал коммуникации. РС отправляет сообщение и выводит отправление.

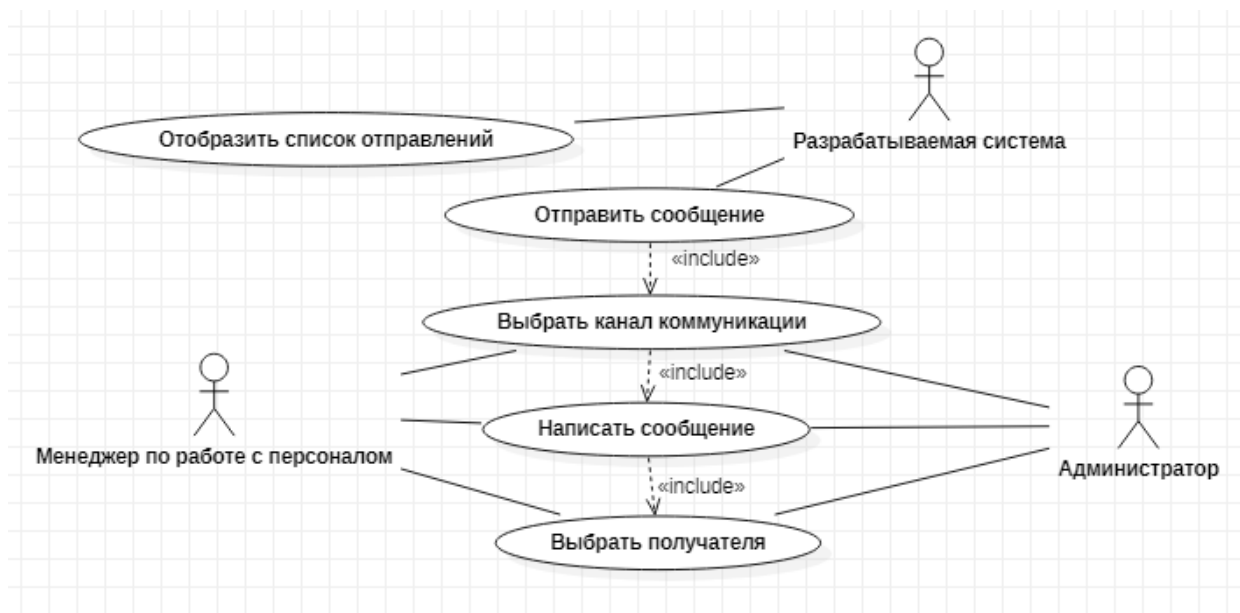


Рисунок 4 – Диаграмма вариантов использования «Отправление сообщений»

В таблице 6 представлено описание главный раздел сценария варианта использования «Отправление сообщений», а в таблице 7 – исключения.

Таблица 6 – Типичный ход событий сценария выполнения варианта использования «Отправление сообщений»

Действия актеров	Отклик системы
1. Администратор или Менеджер по работе с персоналом выбирает получателя 2. Администратор или Менеджер по работе с персоналом пишет сообщение 3. Администратор или Менеджер по работе с персоналом выбирает канал коммуникации 4. Администратор или Менеджер по работе с персоналом нажимает отправить  Исключение №1: Тест сообщения не введен Исключение №2: Произошла ошибка при отправлении	5. РС на основе заданных параметров отправляет сообщение 6. РС формирует и выводит записи об отправке сообщений

Таблица 7 – Исключения сценария выполнения варианта использования «Отправление сообщений»

Действия актеров	Отклик системы
Исключение №1. Тест сообщения не введен	
2. Администратор или Менеджер по работе с персоналом не вводит сообщение	5. РС выводит сообщение о том, что поле «Текст» не заполнено
Исключение №2. Произошла ошибка при отправлении	
4. Администратор или Менеджер по работе с персоналом нажимает отправить, почтовый сервер или Telegram недоступен в данный момент	6. РС формирует и выводит запись об отправке сообщений со статусом «Не отправлено»

В приложении В, на рисунке В.1 представлена диаграмма последовательности набора на курсы. Здесь более детально представлен процесс общения между менеджером по работе с персоналом и кандидатом на курсы.

Процесс начинается от объявления о наборе студентов УЦ и заканчивается началом обучения. Благодаря реализуемой системе менеджер

может сформировать послание, выбрать человека и выбрать канал коммуникации, по которому отправляет сообщение. Если это Telegram, то модуль находит необходимого человека по заранее указанному идентификатору (id) в Telegram, который хранится в базе данных и берется из заявки. Если это почта, то система берет почтовый адрес из базы данных и вставляет это в поле получателя письма.

В приложении Г, на рисунке Г.1 представлена диаграмма последовательности заочного обучения. После формирования учебных групп, администратор назначает права и роли новым студентам. Студент УЦ информируется о том, что занятия начались, получает ссылки и данные учетной записи. Затем студенты проходят курс, как только они закончили элементы обучения, они могут посмотреть рейтинг по мере выполнения заданий.

В приложении Д представлена диаграмма классов. В данной диаграмме показываются отношения между сущностями разрабатываемой системы, а также там перечислены ключевые элементы диаграммы.

В ходе работы были подготовлены дизайн-макеты. Они были выполнены в программе Figma – онлайн-сервисе для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени [37].

В ней были разработаны макеты интерфейса отдельно для каждой группы пользователей: студент УЦ, администратор и менеджер по работе с персоналом.

## **2.3 Информационное обеспечение ИС**

### **2.3.1 Используемые классификаторы и системы кодирования**

Всю информацию об учебном центре можно классифицировать по следующим информационным объектам (ИО):

- информация об учебном центре – ИО «Учебный центр»;

- информация о кандидатах – ИО «Кандидат»;
- информация о студентах – ИО «Студент»;
- информация о сотрудниках – ИО «Сотрудник»;
- информация о должностях – ИО «Должность»;
- информация о департаментах – ИО «Отдел»;
- информация о ролях – ИО «Роль»;
- информация о группах – ИО «Группа»;
- информация о курсах – ИО «Курс»;
- информация о модулях – ИО «Модуль»;
- информация о тестах – ИО «Тест»;
- информация о заданиях – ИО «Задание»;
- информация о рейтинге – ИО «Рейтинг»;
- информация о новостях – ИО «Новость»;
- информация о сообщениях – ИО «Сообщение»;
- информация об отправлениях сообщений – ИО «Отправление».

Для классификации была выбрана иерархическая система, потому что она обладает простотой построения и отличается использованием независимых классификационных признаков в различных ветвях иерархической структуры.

В разрабатываемой системе не планируется огромное количество объектов. Так для обеспечения удобной и эффективной обработки информации была выбрана порядковая регистрационная система кодирования. Порядковое регистрационное кодирование не требует предварительной классификации объектов. Такая система предполагает последовательную нумерацию объектов числами натурального ряда [11].

Также стоит отметить, что регистрационное кодирование некоторых элементов будет приходиться из других систем. Правила кодирования заданы в существующих системах компании, как правило, это числа натурального ряда. В этой связи идентификаторы (Id) объектов, которые впоследствии будут интегрированы из других систем компании и созданные в самой системе, имеют целочисленный тип данных.

### **2.3.2 Характеристика нормативно-справочной и входной оперативной и выходной информации**

Вначале необходимо создать организационную структуру и пользователей. Например, для работы с кандидатами и студентами УЦ администратору и менеджеру по работе с персоналом необходимо создать такие сущности в системе. В приложении Е представлены дизайн-макеты.

У сотрудников есть несколько вариантов создания:

- вручную на сайте, форма ввода представлена на рисунке Е.1;
- с помощью импорта из .xlsx файла, форма ввода представлена на рисунке Е.2.

Так на форме присутствуют ключевые поля: персональная информация о человеке, контакты для каждого типа коммуникации, курс (у кандидатов) и группа (у студентов). При добавлении с помощью импорта (массовом добавлении) необходимо загрузить .xlsx файл в определенном оформлении, справка есть на странице импорта. На рисунке 5 представлен скриншот .xlsx файла, который должен обрабатываться в системе.





После создания новости она появляется в разделе новостей, где новость можно отредактировать. На главной странице отображаются 3 последние новости по дате по убыванию.

После создания кандидата или студента УЦ в системе на выходе получается таблица с кратким перечислением их данных. Администратор, менеджер по работе с персоналом и студент УЦ могут видеть сводный рейтинг и отдельные транзакции после загрузки данных о баллах.

## 2.4 Проектирование БД ИС

При логическом моделировании использовались нотации унифицированного языка моделирования и IDEF1X.

IDEF1X является методом для разработки реляционных баз данных и использует условный синтаксис, специально разработанный для удобного построения концептуальной схемы.

Использование метода IDEF1X наиболее целесообразно для построения логической структуры базы данных после того, как все информационные ресурсы исследованы и решение о внедрении реляционной базы данных, как части корпоративной информационной системы, было принято. Средства моделирования IDEF1X специально разработаны для построения реляционных информационных систем.

Основным преимуществом IDEF1X является жесткая и строгая стандартизация моделирования. Установленные стандарты позволяют избежать различной трактовки построенной модели [17].

Ниже перечислены сущности базы данных:

- Roles – роли пользователей;
- Users – пользователи, относятся к Roles, как многие к одному;
- Departaments – департаменты;

- Positions – должности, относятся к Departments, как многие к одному;
- Employees – сотрудники компании, наследуется от Users, связь один к одному, а у Positions связь многие к одному;
- Courses – учебный курс. Связан с Groups, как один ко многим, и с Employees, как многие к одному;
- Modules – модуль курса, связан с Courses, как многие к одному;
- Services – сервисы учебного модуля, связан с Modules, как многие к одному;
- Tests – тест в рамках учебного модуля, наследуется от Services, связь один к одному;
- класс Tasks – задание в рамках учебного модуля, наследуется от Services, связь один к одному;
- Groups – учебная группа студентов. Связан с Students, один ко многим, а также связан с Courses, как многие к одному;
- Students – студенты УЦ, наследуется от Users, связь один к одному, а у Groups, как многие к одному;
- Scores – транзакции по выполнению заданий. Связан с Tests и Tasks, Students, как многие к одному;
- News – новость, связан с Employees, как многие к одному;
- Messages – сообщение, связан с Employees, как многие к одному;
- Sendings – отправление сообщения, связан с Messages, как многие к одному.

У каждой сущности имеется первичный ключ, практически у всех – это Id. Вышеперечисленные связи происходят по внешнему ключу. Затем можно

построить логическую модель данных по имеющимся сущностям, которая будет соответствовать второй нормальной форме [16].

Диаграмма IDEF1X логической модели данных представлена на рисунке 6, где отражены связи между сущностями.

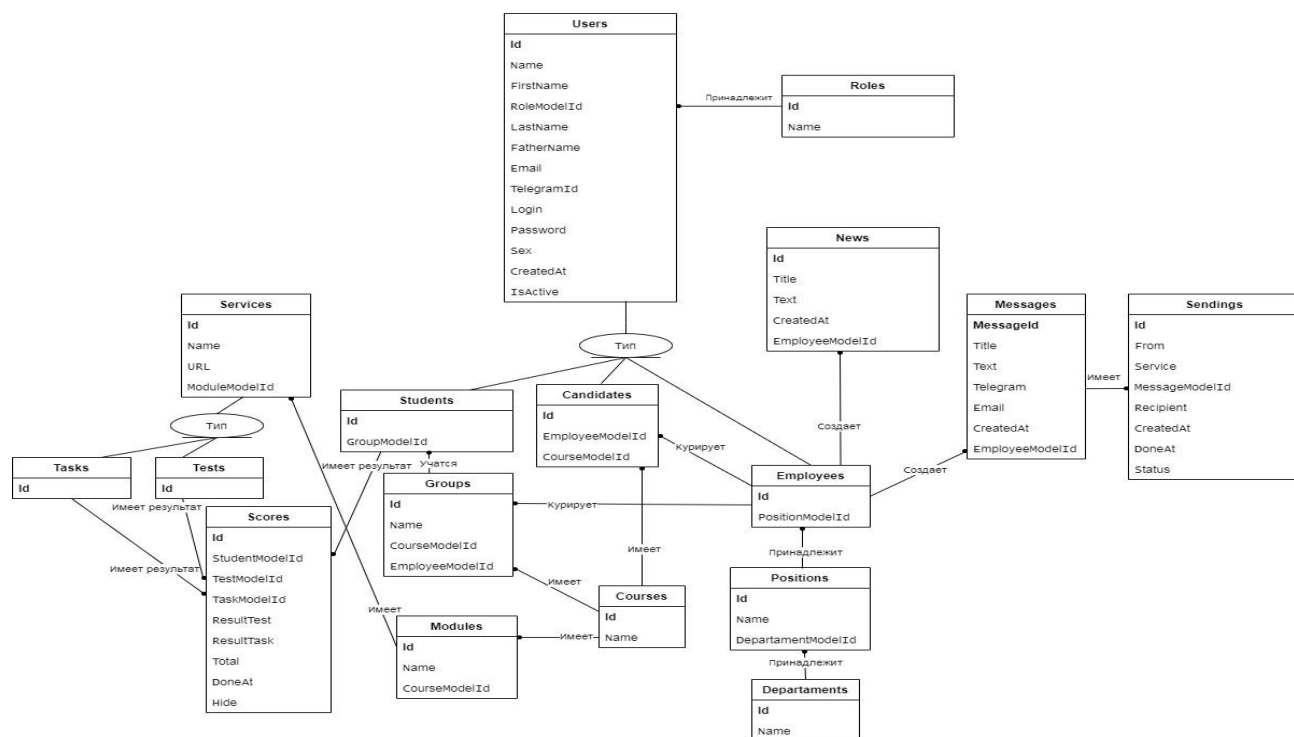


Рисунок 6 – Диаграмма IDEF1X логической модели данных

На данном рисунке можно увидеть логическую модель по методологии IDEF1X, чаще всего используется связь один ко многим, а также есть наследование со связью один к одному.

## 2.5 Требования к аппаратно-программному обеспечению ИС

Технология FURPS+ позволяет обозначить требования к приложению, в таблице 8 отражены содержания функциональности, удобства использования, надежности, производительности, поддерживаемости и проектные ограничения.

Таблица 8 – Требования к приложению

Раздел	Содержание
Functionality, функциональность	добавление новых пользователей, курсов, модулей, тестов, заданий в ручном режиме (на первом этапе разработки) и автоматическом (на втором этапе разработки); рассылка сообщений кандидатам и студентам УЦ по выбранным каналам коммуникации; формирование рейтинга на основе результатов тестов и заданий и отображение в сгруппированном виде или в виде отдельных транзакций для администраторов; отображение курса, модулей, заданий и суммарный балл для студентов; добавление и редактирование новостей
Usability, удобство использования	наличие справочной информации по импорту данных; наличие навигационного меню
Reliability, надежность	система должна сохранять работоспособность при 1000 запросах за 1 минуту
Performance, производительность	ожидание на создание элемента в одиночном режиме должно быть не более 5 секунд
Supportability, поддерживаемость	возможность масштабирования; расширение функционала по отправке сообщений и интеграции сервисов
Проектные ограничения	количество записей в таблице кандидатов и студентов должно быть не более 800

Сначала пользователь должен войти в систему, если это администратор, он может привязать курс, отправить сообщение, написать новость и следить за успеваемостью студентов с помощью общего рейтинга.

На первом этапе разработки необходимо разработать ручной ввод данных и с помощью электронной таблицы (файла .xlsx).

А студенты после выполнения заданий и модулей на других платформах могут просмотреть свои результаты. Все эти действия должны выполняться не более чем за 5 секунд.

## **Выводы по главе 2**

Во второй главе было рассмотрено логическое проектирование разрабатываемой информационной системы.

В качестве технологии проектирования была выбрана нотация UML за ее объектно-ориентированный подход. Были составлены диаграммы: вариантов использования, классов, последовательности, а также макет страниц информационной системы.

Для информационного обеспечения ИС в качестве классификатора была выбрана иерархическая модель, и по ней была выстроена система кодирования.

Для каждого потока входной информации были даны характеристики, показаны формы ввода данной информации в систему. Аналогично было сделано описание выходной информации и соответствующих форм.

Затем в пункте «Проектирование БД ИС» была выбрана технология IDEF1X, что является методом для разработки реляционных баз данных. В концептуальной модели были выделены сущности ИС и разработана логическая модель базы данных информационной системы. А также были составлены требования по технологии FURPS+.

## Глава 3 Физическое проектирование ИС

### 3.1 Выбор архитектуры ИС

В таблице 9 проведено сравнение архитектурных решений для информационной системы.

Таблица 9 – Сравнение архитектурных решений

Критерий	Файл-сервер	Двухзвенный «клиент-сервер»	Трехзвенный «клиент-сервер»	Модель-Представление-Контроллер (Model-View-Controller)
Минимальное количество передаваемых данных	-	+	+	+
Нахождение пользователей в разных сетях	-	+	+	+
Минимальная нагрузка на клиента	-	+	+	+
Кроссплатформенность	-	+	+	+
Защищенность	-	+/-	+	+
Масштабируемость	-	-	+	+
Представление одной и той же информации в разных формах	-	-	-	+
Итого	0	4,5	6	7

Для решаемых задач была выбрана архитектура Модель-Представление-Контроллер (MVC). Она похожа на трехзвенную модель «клиент-сервер», но немного отличается по принципу действия.

В ней содержатся следующие компоненты:

- Model – модель, содержит всю бизнес-логику приложения;

- View – представление (вид), отвечает за отображение данных из модели пользователю, не изменяя ее;
- Controller – контроллер, отвечает за обработку действий пользователя, через него пользователь вносит изменения в данные, которые хранятся в модели.

Основной принцип MVC: отделить реализацию бизнес-логики приложения от ее визуализации. Такое разделение повысит возможность повторного использования кода. При этом, не затрагивая реализацию видов, можно изменить реакции на действия пользователя (нажатие мышью на кнопку, ввод данных). Так можно упростить написание программ, повысить читаемость кода, сделать легче расширение и поддержку системы в будущем [29].

Диаграмма развертывания представлена на рисунке 7. На ней видно взаимодействие элементов информационной системы. Наиболее выгодно использовать данную архитектуру, когда пользователю нужно предоставлять одни и те же данные в разных формах.

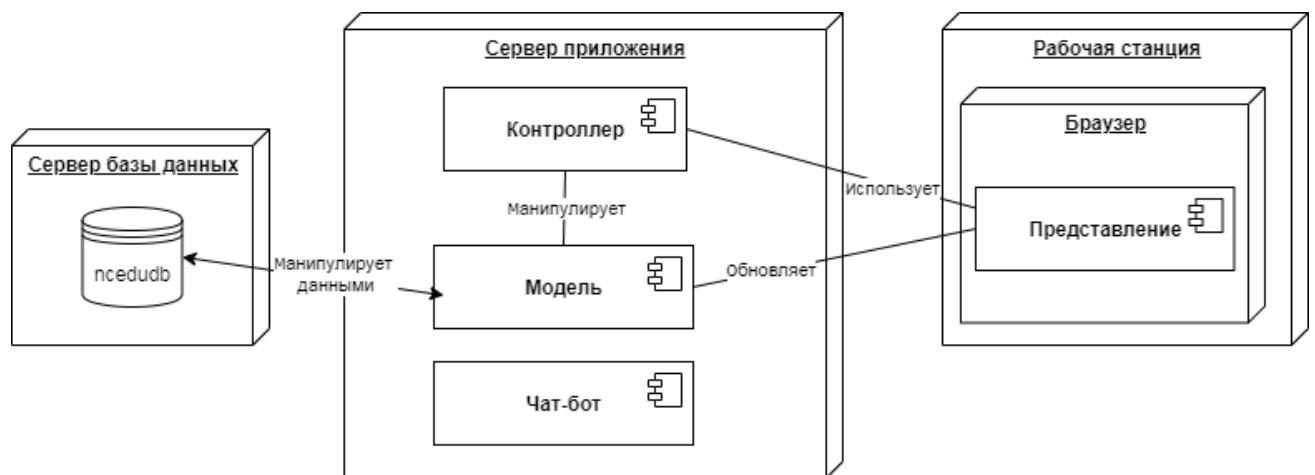


Рисунок 7 – Диаграмма развертывания

Чат-бот отвечает за обработку сообщений от пользователей в мессенджере Telegram, он отдельно запускается на сервере и не зависит от модели и контроллера.

### 3.2 Выбор технологии разработки программного обеспечения ИС

В таблице 10 проведено сравнение языков программирования Python, Java и C#.

Таблица 10 – Сравнение языков программирования

Критерий	Python	Java	C#
Кроссплатформенность	+	+	+
Большое количество библиотек	+	+	+
Скорость выполнения программ	-	+/-	+
Поддержка кириллицы	+	+	+
Итого	3	3,5	4

В результате сравнения был выбран язык программирования C#, поскольку он содержит больше положительных характеристик относительно решаемых задач [19].

У языка C# есть фреймворк ASP.NET Core, он свободно-распространяемый и кроссплатформенный. Данная платформа разрабатывается компанией Microsoft совместно с сообществом. Она имеет модульную структуру и совместима с разными операционными системами [10]. ASP.NET Core позволяет создавать высокопроизводительные кроссплатформенные веб-приложения [1].

ASP.NET Core является модульной платформой. Ее компоненты обновляются через менеджер пакетов NuGet, то есть можно обновлять ее модули по отдельности в отличие от .NET фреймворка. Приложение может



работать с разными модулями и не зависит от единого обновления платформы. Была выбрана версия 3.1, поскольку она является выпуском долгосрочной поддержки [31].

Один из популярных фреймворков – Spring Web MVC на языке Java по результатам рейтинга производительности фреймворков «Awesome Open Source» [44] и «TechEmpower» занимает позиции ниже, чем ASP.NET Core [43]. В этой связи в данном случае выгоднее использовать ASP.NET Core.

Представления разрабатывались на HTML и CSS, используя фреймворк Bootstrap. Его преимущества: адаптивность, кросс-браузерность, единство стилей и высокая скорость. Хорошо подходит для легкой настройки большого количества однотипных страниц [30].

Для обработки сообщений чат-бота в Telegram был выбран язык Python, поскольку на этом языке чаще разрабатываются такие программы. Сам Python имеет много документации по Telegram API и большой выбор библиотек. А также Python чаще используется для анализа данных в разных сферах из-за большого количества открытых библиотек. Анализ данных – важная часть работы подразделения, планируется внедрение инструментов работы с данными [6].

В качестве среды разработки на ASP.NET Core была выбрана среда Microsoft Visual Studio, поскольку в ней есть:

- большая коллекция расширений, которая постоянно пополняется;
- технология автодополнения IntelliSense;
- возможность настройки рабочей панели;
- поддержка разделенного экрана [2];
- разработана Microsoft (создателем языка и платформы);
- применены технология бэклога позволяет членам команды взаимодействовать при гибкой методологии разработки [14].

А для программы на Python в целях тестирования была выбрана онлайн интегрированная среда разработки (IDE) и веб-хостинг PythonAnywhere [40].

Также при работе с данным решением использовалась система контроля версий Git, которая поддерживается в Microsoft Visual Studio. Была выбрана платформа GitLab для управления репозиториями и отслеживанием задач. По версии Startpack.ru GitLab больше подойдет для размещения кода больших организациях в бесплатных репозиториях с приватным режимом, что очень важно для компании [25].

### 3.3 Выбор СУБД ИС

В таблице 11 проведено сравнение систем управления базами данных для реляционной базы данных.

Таблица 11 – Сравнение систем управления базами данных

Критерий	Oracle	MySQL	Microsoft SQL server
Надежность и стабильность	+	+	+
Наличие документации	+	+	+
Наибольшая производительность	-	+	+
Подключение к средам разработки	-	+	+
Отслеживание уровня производительности	-	-	+
Генератор форм, отчетов	-	-	+
Итого	2	4	6

В результате сравнения была выбрана СУБД Microsoft SQL server. Это система управления базами данных, ее движок работает на облачных

серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно. Также она является кроссплатформенной.

В последних версиях присутствуют: временная поддержка данных, которая позволяет отслеживать изменения данных с течением времени, и динамическая маскировка данных, которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные [26].

### **3.4 Разработка физической модели данных ИС**

В Microsoft SQL server используется T-SQL или Transact-SQL (диалект SQL) [34].

Первоначальный язык был расширен следующими возможностями:

- управляющие операторы;
- локальные и глобальные переменные;
- различные дополнительные функции для обработки строк;
- поддержка аутентификации Microsoft Windows [21].

Основные отличия:

- в T-SQL точка с запятой для разделения операторов необязательна;
- в T-SQL присутствует оператор go, применяется для отделения друг от друга групп операторов, а расширенный синтаксис означает, что соответствующий групповой оператор будет выполнен n раз;
- оператор use в T-SQL изменяет контекст базы данных, задавая другую базу данных.

Также T-SQL можно встраивать в SQL запросы. Пример запроса на T-SQL приведен на рисунке 8.

```

1 CREATE TABLE [dbo].[Candidates] (
2     [Id] NVARCHAR (450) NOT NULL,
3     [EmployeeModelId] NVARCHAR (450) NULL,
4     [CourseModelId] NVARCHAR (450) NULL,
5     CONSTRAINT [PK_Candidates] PRIMARY KEY CLUSTERED ([Id] ASC),
6     CONSTRAINT [FK_Candidates_Courses_CourseModelId] FOREIGN KEY ([CourseModelId]) REFERENCES [dbo].[Courses] ([Id])
7     CONSTRAINT [FK_Candidates_Employees_EmployeeModelId] FOREIGN KEY ([EmployeeModelId]) REFERENCES [dbo].[Employees]
8     CONSTRAINT [FK_Candidates_Users_Id] FOREIGN KEY ([Id]) REFERENCES [dbo].[Users] ([Id]) ON DELETE CASCADE
9 );

```

Рисунок 8 – Пример запроса на T-SQL

На данном рисунке показан запрос создания таблицы «Candidates». Схема dbo представляет собой схему по умолчанию для вновь созданной базы данных. Владельцем схемы dbo является учетная запись пользователя dbo [924].

Затем полям таблицы назначаются строковый тип nvarchar для данной СУБД. Первичный ключ задается внизу запроса с помощью оператора «CONSTRAINT [PK...] PRIMARY ([поле] ASC)». Внешний ключ через «CONSTRAINT [FK...] FOREIGN KEY ([поле]) REFERENCES [dbo].[внешняя таблица] ([поле])». Оператор «ON DELETE CASCADE» означает каскадное удаление [24].

Диаграмма физической модели базы данных приведена в приложении Ж, на рисунке Ж.1. Она была построена с помощью средств программы SQL Server Management Studio.

### 3.5 Разработка программного обеспечения ИС

#### 3.5.1 Схема взаимосвязи модулей приложения ИС

Схему модулей ИС можно представить в виде дерева функций (рисунок 9), где функции делятся на основные и системные.

Основные функции отвечают за обработку и управление данными. Это функции: управление пользователям; управление курсами; отправка сообщений; управление новостями; отображение заданий; отображение рейтинга.

Системные функции отвечают за взаимодействие с системой. Это функции: разграничение прав доступа; аутентификация; авторизация.



Рисунок 9 – Дерево функций ИС

На рисунке 10 представлен сценарий диалога ИС, который отображает последовательность действий системы и переход из одного состояния в другое.

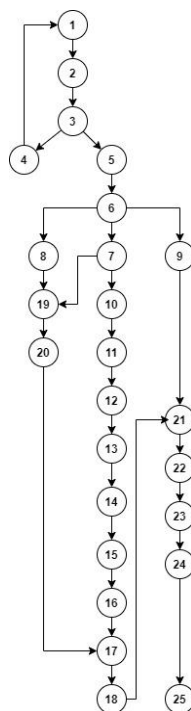


Рисунок 10 – Сценарий диалога ИС

В приложении И, таблице И.1 представлено табличное описание графа сценариев. Вначале пользователь входит в систему, идет проверка введенных данных на наличие в базе данных. Затем производится авторизация, в зависимости от роли у пользователя есть определенные права на модули.

Так администратор может создать, просмотреть отредактировать, удалить: пользователя, курс, модуль, задания, новость, сообщение и рейтинг. Менеджер по работе с персоналом (HR-менеджер) может просмотреть список кандидатов и студентов, создать, просмотреть, отредактировать, удалить новости, сообщения, просмотреть сводный рейтинг.

Для коммуникации по Telegram кандидату (студенту) необходимо указать свой Telegram Id в анкете, который можно узнать в чат-боте УЦ командой «/start», поскольку Telegram не позволяет отправлять сообщения пользователям, которые не писали боту. Как только контактные данные были получены, менеджер может отправить сообщение в Telegram с помощью ИС.

А студент просмотреть модули, задания, тесты, новости и сводный рейтинг. Затем пользователь может выйти из системы.

### **3.5.2 Описание модулей приложения ИС с примерами программного кода**

Для работы моделей с базой данной был использован Entity Framework Core (EF), который позволяет разработчикам работать с базой данных с помощью объектов .NET, а также EF самостоятельно генерирует код для доступа к данным, чтобы уменьшить время разработки [33].

В полях класса DataBaseCont, наследуемый от DbContext, перечислены таблицы и к какой модели они относятся через DbSet. Далее в конструкторе класса происходит передача базового класса объекта DbContextOptions, который инкапсулирует параметры конфигурации [1313]. А OnModelCreating позволяет задать значения для вычисляемого поля (рисунок 11).

```

Ссылка: 10
public DbSet<SenderMesModel> Sendings { get; set; }
Ссылка: 7
public DbSet<RatingModel> Scores { get; set; }

Ссылка: 0
public DataBaseCont(DbContextOptions<DataBaseCont> options) : base(options)
{
    Database.EnsureCreated();
}
Ссылка: 0
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<RatingModel>()
        .Property(u => u.Total)
        .HasComputedColumnSql("[ResultTest] + [ResultTask]");
}

```

Рисунок 11 – Скриншот кода класса DataBaseCont

После того, как сущности были описаны, была подключена аутентификация и авторизация. В .NET Core есть встроенная поддержка аутентификации на основе cookie (куки). При получении запроса от клиента происходит их валидация и инициализация свойства User объекта HttpContext [7].

В классе Startup в методе Configure(IApplicationBuilder app, IWebHostEnvironment env) подключается авторизация через app.UseAuthorization() и аутентификация и через app.UseAuthentication(). Затем в методе ConfigureServices(IServiceCollection services) происходит установка конфигурации подключения, где метод AddCookie() настраивает аутентификацию. Здесь используется свойство LoginPath, по которому будет перенаправляться анонимный пользователь при доступе к страницам.

А если необходимых прав нет у авторизованного пользователя, то свойство AccessDeniedPath перенаправляет на страницу с соответствующим сообщением (рисунок 12).

```

ссылка: 0
public void ConfigureServices(IServiceCollection services)
{
    string connection = Configuration.GetConnectionString("DefaultConnection"); if (connection.Contains("%CONTENTROOTPATH%")) { connec
services.AddDbContext<DataBaseCont>(options => options.UseSqlServer(connection));
services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)
    .AddCookie(options =>
        {
            options.LoginPath = new PathString("/Account/Login");
            options.AccessDeniedPath = new PathString("/NoAccess");
        });
services.AddControllersWithViews();
}

```

Рисунок 12 – Скриншот кода метода ConfigureServices(IServiceCollection services)

Как только пользователь успешно заходит в систему, ему назначаются права с помощью объектов claim. В данном случае они представляют роль пользователя. В зависимости от значений ролей можно предоставлять пользователю доступ к той или иной странице [32].

Так при входе метод Authenticate(string login) контроллера AccountController получает логин пользователя, ищется строка с таким логином, и по Id пользователя находится имя роли. Затем происходит установка аутентификационных куки. Claims содержит роль для логина, соответствующего пользователя (рисунок 13).

```

ссылка: 1
private async Task Authenticate(string login)
{
    UserModel userModel = db.Users.FirstOrDefault(groupName => groupName.Login == login);
    RoleModel roleModel = db.Roles.FirstOrDefault(groupName => groupName.Id == userModel.RoleModelId);

    var claims = new List<Claim>
    {
        new Claim(ClaimsIdentity.DefaultNameClaimType, login),
        new Claim(ClaimsIdentity.DefaultRoleClaimType, roleModel.Name)
    };

    ClaimsIdentity id = new ClaimsIdentity(claims, "ApplicationCookie", ClaimsIdentity.DefaultNameClaimType, ClaimsIdentity.DefaultRole
await HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, new ClaimsPrincipal(id));
}

```

Рисунок 13 – Скриншот кода метода Authenticate(string login)

Администратор может создать разные сущности по одному или через импорт данных из .xlsx файла. В одиночном режиме пользователь вводит информацию в форму, она выводится с помощью содержимого HttpGet метода контроллера. После нажатия на кнопку «Создать» вызывается



HttpPost метод в контроллере. Для примера на рисунке 14 показан код метода CreateModuleOne(ModuleModel moduleModel) для создания модуля курса в одиночном режиме. Здесь с помощью EF в таблицу Modules добавляется новая запись со значениями, который ввел пользователь. Команда db.SaveChanges() сохраняет полученные записи в базе данных.

```
[HttpPost]
Ссылка: 0
public IActionResult CreateModuleOne(ModuleModel moduleModel)
{
    try
    {
        db.Modules.Add(moduleModel);
        db.SaveChanges();
        TempData["message"] = "Модуль '" + moduleModel.Name + "' добавлен успешно";
    }
    catch (DbUpdateException)
    {
        TempData["message"] = "Произошла ошибка в сохранении данных, убедитесь, что Id '" + moduleModel.Id + "' уже не существует в таб.
    }
    return RedirectToAction("CreateModuleOne");
}
```

Рисунок 14 – Скриншот кода метода CreateModuleOne(ModuleModel moduleModel)

Также пользователь может загрузить .xlsx файл с данными. Для примера на рисунке 15 показана блок-схема алгоритма метода CreateModule(IFormFile file, ModuleModel moduleModel) для создания модулей, заданий и тестов в рамках этих же модулей. По определенным правилам пользователь должен сформировать .xlsx файл и загрузить его на специальную страницу, а также указать принадлежность к курсу. На вход поступает file – .xlsx файл и moduleModel – данные о модели модуля.

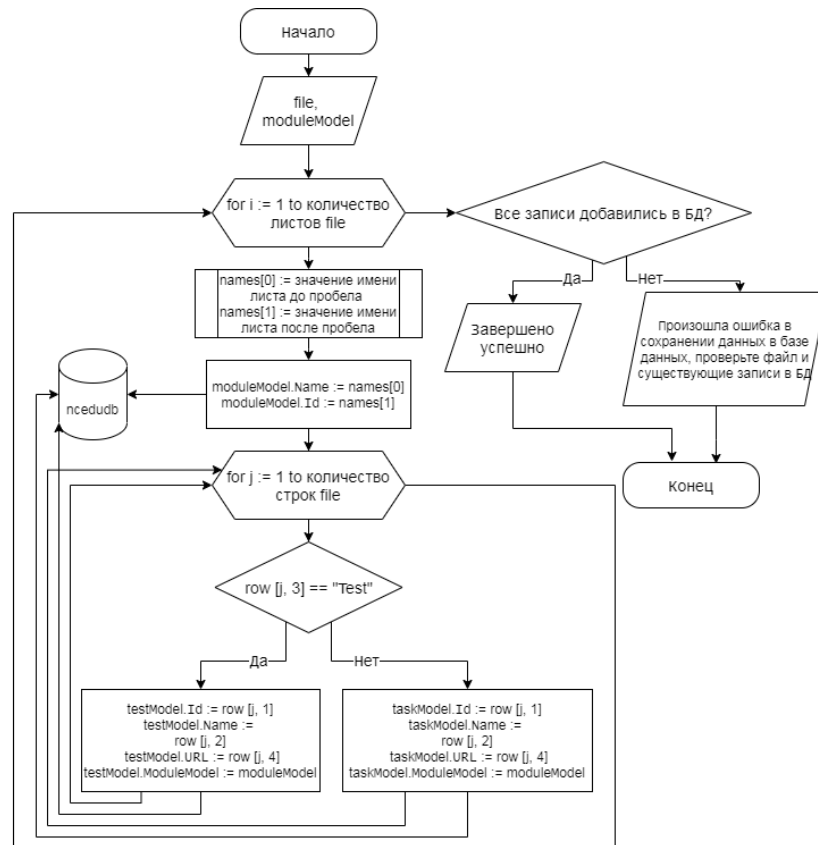


Рисунок 15 – Блок-схема алгоритма метода CreateModule(IFormFile file, ModuleModel moduleModel)

Во внешнем цикле идет проход по листам таблицы. У каждого листа есть имя, которое формируется по принципу: «Название модуля» пробел «id модуля». Имя разбивается по пробелу, левая часть присваивается moduleModel.Name, а правая moduleModel.Id. Затем данные модели moduleModel заносятся в базу данных систему таблицу «Modules».

Во внешнем цикле идет проход по используемым строкам таблицы, первый индекс массива row – номер строки, второй индекс – ячейка строки. Третья ячейка имеет всего 2 значения «Test» или «Task». Если третья ячейка равна значению «Test», то testModel заполняет значения для своих полей из соответствующих ячеек. Затем данные модели testModel заносятся в таблицу «Tests». Иначе taskModel заполняет значения для своих полей из соответствующих ячеек. Затем данные модели taskModel заносятся в таблицу «Tasks». Как только все циклы пройдены, то выводится сообщение

«Завершено успешно» или «Произошла ошибка в сохранении данных в базе данных, проверьте файл и существующие записи в БД» в зависимости от результата работы алгоритма.

Для работы с таким алгоритмом была подключена библиотека ClosedXML, где данные считываются в цикле по листам файла, затем по строкам. В каждой итерации цикла выбираются и добавляются необходимые данные для сущностей БД в режиме импорта (массовом режиме).

Отправка писем осуществляется с помощью библиотеки MailKit и Smtplib протокола. Как только сообщение было сформировано, в класс MimeMessage записываются данные об отправителе, получателе, теме и тексте сообщения. Smtplib позволяет отправить письмо по протоколу Smtplib (рисунок 16).

```
ссылка:1
private bool EmailSender(string recipient, string title, string text)
{
    try {
        var emailMessage = new MimeMessage();
        emailMessage.From.Add(new MailboxAddress("TEST", BotConfig.Address));
        emailMessage.To.Add(new MailboxAddress("recipient", recipient));
        emailMessage.Subject = title;
        emailMessage.Body = new TextPart("plain")
        {
            Text = text
        };
        using (var client = new SmtplibClient())
        {
            client.Connect("smtp.mail.ru", 465);
            client.Authenticate(BotConfig.Address, BotConfig.EmailPass);
            client.Send(emailMessage);
            client.Disconnect(true);
        }
    }
    return true;
}
```

Рисунок 16 – Скриншот кода метода по отправке e-mail

Для того чтобы получить id пользователя в Telegram, человеку необходимо написать команду «/start» в чат-бот, ему отправится его id, которое он должен занести в заявку (рисунок 17).

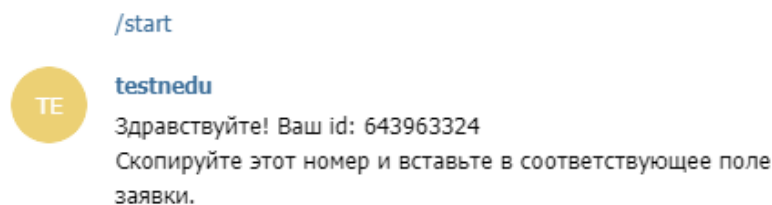


Рисунок 17 – Скриншот чат-бота

Бот для работы в Telegram был написан на Python. На рисунке 18 представлен скриншот кода по отправке id пользователя.

```
1 # -*- coding: utf-8 -*-
2 import telebot
3
4
5 bot = telebot.TeleBot("1713477090:AAE4rYJoSjq1Svj-3DJHgIwcx5NDXnNiCCY")
6
7
8 @bot.message_handler(commands=['start'])
9 def start(message):
10     user_id = message.chat.id
11     text = 'Здравствуйте! Ваш id: ' + str(user_id) + '\nСкопируйте этот номер и вставьте в соответствующее поле заявки.'
12     bot.send_message(message.chat.id, text)
13
14 bot.polling()
15
16
```

Рисунок 18 – Скриншот кода чат-бота

Для отправки сообщения в Telegram создается объект класса TelegramBotClient, затем вызывается метод TelgSender(string chatID, string text), в который передается id получателя и текст сообщения. Метод SendTextMessageAsync получает данные параметры и отправляет получателю (рисунок 19).

```
private static readonly TelegramBotClient client = new TelegramBotClient(BotConfig.Token);
ссылка:1
private static async void TelgSender(string chatID, string text)
{
    if (text != null)
    {
        await client.SendTextMessageAsync(chatID, text);
    }
}
```

Рисунок 19 – Скриншот кода метода по отправке сообщения в Telegram

Если отправление прошло успешно, то происходит обновление записи таблицы отправок на статус «Отправлено». Затем в Razor странице – представлении представлена разметка на HTML коде, и там можно найти вставки кода на языке С# (рисунок 20).

```
<div>
  <a asp-action="MessagesList">Список сообщений</a>
</div>
@using (Html.BeginForm("Sendings"))
{
  <form method="get">
    <div class="form-inline">
      <label class="control-label">Сервис: </label>
      @Html.DropDownList("service", Model.Service as SelectList,
        htmlAttributes: new { @class = "form-control" })
      <label class="control-label">Статус: </label>
      @Html.DropDownList("status", Model.Status as SelectList,
        htmlAttributes: new { @class = "form-control" })
      <input type="submit" value="Фильтр" class="btn btn-outline-secondary" />
    </div>
    <p></p>
  </form>
  <hr />
  <table class="table table-striped">
    <tr>...</tr>
    @foreach (var item in Model.Sendings)
    {
      <tr>
        <td>
          @Html.DisplayFor(modelItem => item.Id)
        </td>
        <td>

```

Рисунок 20 – Скриншот кода представления

Здесь представлена разметка таблицы с перечислением отправок в цикле объектов модели. А также выше расположена ссылка на список сообщений и поля для фильтров данных.

### 3.6 Описание функциональности ИС

Вход в систему осуществляется с помощью браузера (рекомендуется Google Chrome или Microsoft Edge). Затем пользователь попадает на главную страницу. Скриншоты страниц представлены в приложении К. Вид для администратора представлен на рисунке К.1, для менеджера по работе с персоналом на рисунке К.2, а вид для студента на рисунке К.3.

Администратор может создать модуль (рисунок К.4) в одиночном режиме или через импорт файла модули и задания (рисунок К.5). Стоит отметить, что предварительно необходимо создать курсы, которые можно выбрать при создании модуля.

Администратор и менеджер по работе с персоналом могут написать и отправить сообщение. Для этого выбирается получатель (рисунок К.6), затем выходит форма написания сообщений. Список отправлений можно отследить на специальной странице.

Пользователи всех ролей могут просмотреть сводный рейтинг. Здесь отдельно суммируются все баллы за тесты и отдельно за задания, в колонке «Всего» суммируются результаты за задания и тесты. Отредактировать баллы может только администратор, зайдя по ссылке «Отдельные транзакции». Страница «О сервисе» представлена на рисунке 21.

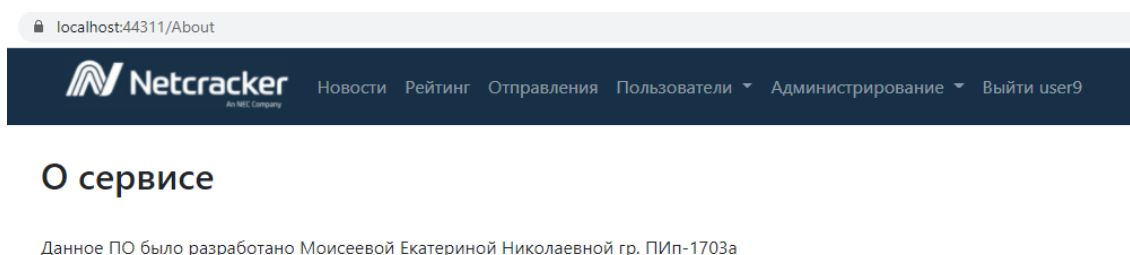


Рисунок 21 – Скриншот страницы «О сервисе»

Данный скриншот подтверждает, что ИС была выполнена автором данной работы.

### 3.7 Тестирование программного проекта

Тестирование программного обеспечения – это проверка соответствия между реальным и ожидаемым поведением программы. Ниже перечислены некоторые виды тестирования:

- функциональный базируется на функциях и особенностях системы;
- нефункциональный необходим для определения характеристик ПО, которые могут быть измерены различными показателями;
- связанный с изменениями используется после введения новых функций или модификации программного кода [8].

В данной работе активно использовалось функциональное тестирование, поскольку программный продукт переходит на стадию приемочного тестирования.

Были протестированы элементы пользовательского интерфейса в автоматическом режиме. Данный вид тестирования проверяет взаимодействие компонентов между собой и увеличивает надежность приложения.

Для автоматического тестирования пользовательского интерфейса использовался Selenium. Это инструмент с открытым исходным кодом для автоматизации тестирования веб-приложения [41].

Был создан проект на фреймворке xUnit.net для тестирования решения, где был подключен драйвер для браузера Google Chrome. Для кейсов работы входа в систему были созданы классы, которые отвечают за поля и возможные методы для проверок. В конструкторе класса вынесена инициализация драйвера, который позволяет работать с HTML элементами страниц (рисунок 22).

```

class LoginPage
{
    private readonly IWebDriver _driver;
    public LoginPage(IWebDriver driver)
    {
        _driver = driver;
    }
    public string PageTitle => _driver.Title;
    private const string URI = "https://localhost:44311";
    private IWebElement LoginElement => _driver.FindElement(By.Id("Login"));
    private IWebElement PasswordElement => _driver.FindElement(By.Id("inputPassword"));
    private IWebElement LogInButton => _driver.FindElement(By.Id("loginsys"));
    public string Source => _driver.PageSource;
}

```

Рисунок 22 – Скриншот класса LoginPage

В классе AutomatedUITests находятся поля для работы с автотестом, а также методы – сами тестовые сценарии. В конструкторе класса происходит инициализация драйвера и навигация на страницу, которая указана в классе, например, LoginPage (рисунок 23).

```
public class AutomatedUITests : IDisposable
{
    private readonly string path = @"C:\Test\";
    private readonly IWebDriver _driver;
    private readonly LoginPage _page;
    Ссылка: 0
    public AutomatedUITests()
    {
        _driver = new ChromeDriver(path);
        _page = new LoginPage(_driver);
        _page.Navigate();
    }
}
```

Рисунок 23 – Скриншот класса AutomatedUITests

Для проверки сценария с попыткой входа по неправильному логину или паролю был создан метод Incorrect\_Login(). В поле логина и пароля вводятся значения, которых нет в базе данных, затем нажимается кнопка «Войти». Ожидаемый результат: сообщение «Некорректные логин и(или) пароль» (рисунок 24).

```
[Fact]
Ссылка: 0
public void Incorrect_Login()
{
    _page.PopulateLogin("user99");
    _page.PopulatePassword("12345684");
    _page.ClickLogIn();

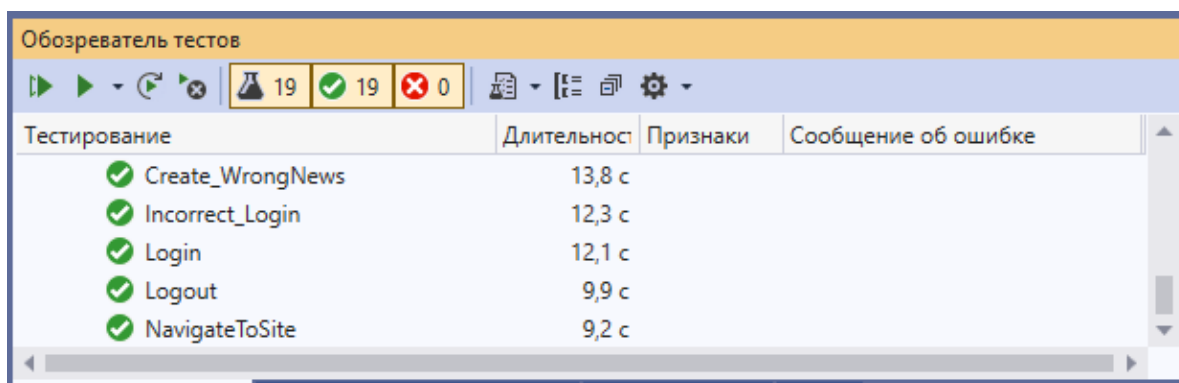
    TestScreenShot("Incorrect_Login");
    Assert.Contains("Некорректные логин и(или) пароль", _page.Source);
}
```

Рисунок 24 – Скриншот метода Incorrect\_Login()

В Microsoft Visual Studio существует обозреватель тестов, который позволяет управлять тестами. Так, например, можно запустить один или



несколько тестов и посмотреть статус их выполнения. Пример такого окна на рисунке 25.



Тестирование	Длительность	Признаки	Сообщение об ошибке
✔ Create_WrongNews	13,8 с		
✔ Incorrect_Login	12,3 с		
✔ Login	12,1 с		
✔ Logout	9,9 с		
✔ NavigateToSite	9,2 с		

Рисунок 25 – Скриншот обозревателя тестов

Когда тест начинает выполняться, запускается окно браузера, в данном случае в Google Chrome. Пример на рисунке 26.

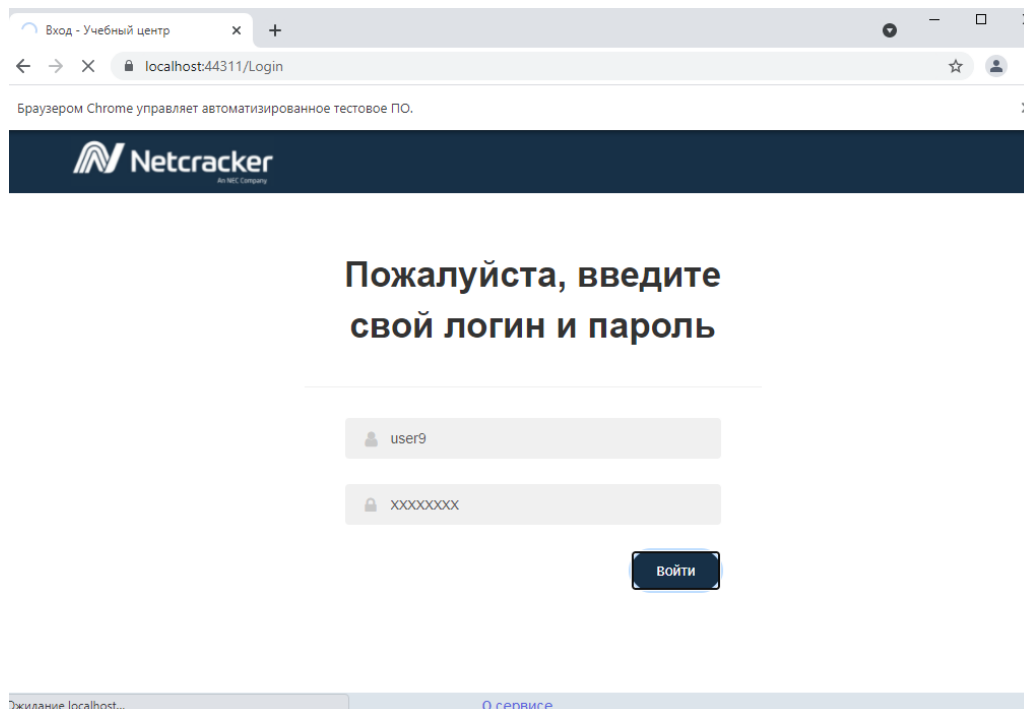


Рисунок 26 – Скриншот браузера под управлением автотеста

Когда автотест заканчивает работу, то окно браузера закрывается, и определяется результат прохождения теста.

Также были составлены тестовые кейсы (ТК) для ручного прохождения. Список тестовых кейсов с приоритетами и статусами выполнения представлен в приложении Л, таблице Л.1. Пример тестового кейса представлен в таблице Л.2.

Системное тестирование в основном было выполнено в мануальном режиме, поскольку требования часто менялись, и автотесты покрывали самые базовые требования [12].

На рисунке 27 представлены столбчатые диаграммы итогов прохождения тестовых сценариев в браузере Google Chrome, Microsoft Edge.

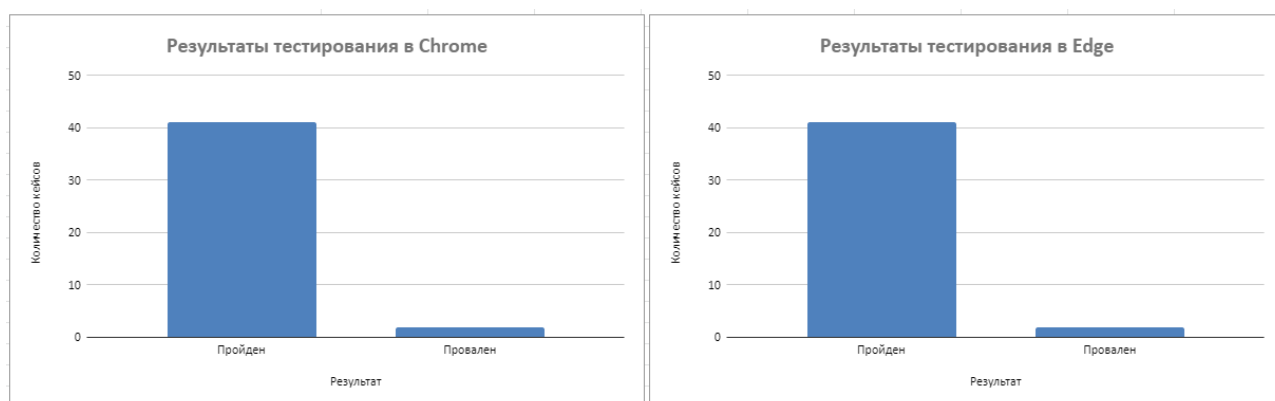


Рисунок 27 – Диаграммы результатов тестирования

Тестирование в двух браузерах показало себя успешно. Однако были найдены ошибки в негативных сценариях при отправке сообщений, поскольку не было проверки на существование данных о получателе.

### **Выводы по главе 3**

В третьей главе были рассмотрены разные подходы к архитектуре программного решения. В результате сравнения разных архитектур был выбран подход «Модель-Представление-Контроллер». Затем для поставленных задач был выбран язык программирования C# и платформа для разработки ASP.NET Core также в результате сравнения с другими решениями.

Также была выбрана система управления базами данных Microsoft SQL Server и построена физическая модель базы данных, была работа с Transact-SQL.

Затем были выделены основные и системные функции. Основные отвечают за обработку и управление данными, а системные отвечают за взаимодействия с системой.

Также был составлен граф сценариев диалога информационной системы с табличным описанием, где отображен основной порядок действий пользователей с разными ролями.

Были описаны ключевые функции системы с примерами программного кода. Использовались разные библиотеки, например, Entity Framework, ClosedXML, MailKit, Telegram.Bot. А также были представлены скриншоты интерфейса программы.

Было проведено системное тестирование программного обеспечения. Данный этап тестирования был выполнено вручную с помощью списка тестовых кейсов и автотестов на Selenium, которые опирались на требования к функционалу программы.

## Заключение

Информационная система для учебного центра компании ООО «Неткрэкер» была спроектирована с помощью CASE-технологии, а также реализован первый этап разработки (ручное добавление и редактирование объектов в одиночном режиме и с помощью импорта) и проведено системное тестирование.

Задачи по анализу деятельности учебного центра на примере московского и тольяттинского офисов компании, сбора требований и проектирования концептуальной и логической модели информационной системы, реализации первого этапа и тестированию системы были выполнены в нужные сроки.

Вначале был представлен учебный центр организации на примере офиса городов Москвы и Тольятти, были составлены логические модели информационной системы и базы данных, сформулированы требования.

На архитектуре «Модель-Представление-Контроллер» и на языке программирования C#, и платформе для разработки ASP.NET Core была реализована данная система. Была выбрана система управления базами данных Microsoft SQL Server. А также проведено системное ручное и автоматизированное тестирования, которое показало хороший показатель качества программного продукта. Суммарный показатель успешно пройденных тестов составляет более 90 %.

Компания получила прототип информационной системы интеграции сервисов учебного центра, которая вскоре пройдет приемочное тестирование и внедрение в организацию, затем будет проходить второй этап разработки с интеграцией сервисов УЦ с помощью программного интерфейса приложения (API). Сотрудники учебного центра уже отмечают значимость системы для оптимизации рутинных задач. Затем систему планируется расширять, путем связки с другими сервисами компании, расширять функции чат-бота, проводить анализ данных.

## Список используемой литературы и источников

1. .NET Core [Электронный ресурс] : Авторы Википедии // Википедия, свободная энциклопедия. Сан-Франциско: Фонд Викимедиа, 2021. URL: <https://ru.wikipedia.org/?curid=68866&oldid=108935326/> (дата обращения: 12.01.2021);
2. 10 лучших IDE [Электронный ресурс] : Комьюнити. URL: <https://timeweb.com/ru/community/articles/5-luchshih-ide-1/> (дата обращения: 12.01.2021);
3. Email или мессенджеры: какие рассылки эффективнее [Электронный ресурс] : Нетология-групп, 2011-2021. URL: <https://netology.ru/blog/email-vs-messenger/> (дата обращения: 20.11.2020);
4. BPMN Studio [Электронный ресурс] : ELMA. URL: <https://bpmn.studio/ru/> (дата обращения: 10.11.2020);
5. Netcracker Technology [Электронный ресурс] : Авторы Википедии // Википедия, свободная энциклопедия. URL: [https://ru.wikipedia.org/wiki/Netcracker\\_Technology/](https://ru.wikipedia.org/wiki/Netcracker_Technology/) (дата обращения: 10.11.2020);
6. Анализ данных с использованием Python [Электронный ресурс] : «Habr», 2006 – 2021. URL: <https://habr.com/ru/post/353050/> (дата обращения: 15.01.2021);
7. Аутентификация и авторизация [Электронный ресурс] : metanit.com, 2012-2021. URL: <https://metanit.com/sharp/aspnet5/15.1.php/> (дата обращения: 15.01.2021);
8. Виды Тестирования Программного Обеспечения [Электронный ресурс] : ProTesting.ru 2008-2021. URL: <http://www.protesting.ru/testing/testtypes.html> (дата обращения: 05.04.2021);
9. Владение и отделение пользователей от схем в SQL Server [Электронный ресурс] : Microsoft 2021. URL: <https://docs.microsoft.com/ru->

ru/dotnet/framework/data/adonet/sql/ownership-and-user-schema-separation-in-sql-server (дата обращения: 10.01.2021);

10. Гайдлайны архитектуры .NET приложений [Электронный ресурс] : Goodwillgroup 2021. URL: <https://gwgr.ru/blog/4/gajdlajny-arhitektury-net-prilozhenij/> (дата обращения: 12.01.2021);

11. Классификация и кодирование информации [Электронный ресурс] : Ресурсный центр физико-математического факультета 2021. URL: [http://phys.bspu.by/static/lib/inf/posob/stu\\_m/glaves/glava2/gl\\_2\\_2.htm](http://phys.bspu.by/static/lib/inf/posob/stu_m/glaves/glava2/gl_2_2.htm) (дата обращения: 30.11.2020);

12. Когда нужны тесты и автотесты, взгляд из надсистемы [Электронный ресурс] : «Habr», 2006 – 2021. URL: <https://habr.com/ru/post/470375/> (дата обращения: 17.03.2021);

13. Конфигурация подключения [Электронный ресурс] : metanit.com, 2012-2021. URL: <https://metanit.com/sharp/entityframeworkcore/2.2.php> (дата обращения: 22.01.2021);

14. Лучшие IDE для разработки на C# [Электронный ресурс] : GeekBrains 2021. URL: [https://gb.ru/posts/c\\_sharp\\_ides](https://gb.ru/posts/c_sharp_ides) (дата обращения: 12.01.2021);

15. Умнова, Е. Г. Моделирование бизнес-процессов с применением нотации BPMN : учебно-методическое пособие / Е. Г. Умнова. – Саратов : Вузовское образование, 2017. – с. 5 – Текст : электронный // Электронно-библиотечная система IPR BOOKS. URL: <https://www.iprbookshop.ru/67840.html> (дата обращения: 05.11.2020);

16. Описание основных принципов нормализации баз данных [Электронный ресурс] : Microsoft 2021. URL: <https://docs.microsoft.com/ru-ru/office/troubleshoot/access/database-normalization-description> (дата обращения: 10.01.2021);

17. Основы методологии IDEF1X [Электронный ресурс] : Интернет-проект «Корпоративный менеджмент», 1998–2021. URL: <https://www.cfin.ru/vernikov/idef/idef1x.shtml> (дата обращения: 10.11.2020);

18. Платформа дистанционного обучения iSpring Learn [Электронный ресурс] : ООО «Ричмедиа», 2001–2021. URL: <https://www.ispring.ru/ispring-learn> (дата обращения: 16.11.2020);
19. Плюсы и минусы языков программирования [Электронный ресурс] : ITVDN, 2021. URL: [https://teletype.in/@itvdn\\_it/zDfuiuHU](https://teletype.in/@itvdn_it/zDfuiuHU) (дата обращения: 12.01.2021);
20. Почему Telegram – главный мессенджер в России? программирования [Электронный ресурс] : medium, 2021. URL: <https://clck.ru/UiYjB> (дата обращения: 20.11.2020);
21. Различия в синтаксисе SQL и Transact-SQL [Электронный ресурс] : serversql, 2021. URL: <http://serversql.ru/razlichija-v-sintaksise-sql-i-transact-sql.html> (дата обращения: 18.01.2021);
22. Система управления бизнес-процессами ELMA BPM – Как работает BPM-система [Электронный ресурс] : ELMA, 2021. URL: [https://www.elma-bpm.ru/product/bpm/system\\_work.html](https://www.elma-bpm.ru/product/bpm/system_work.html) (дата обращения: 10.12.2020);
23. Система управления обучением [Электронный ресурс] : Авторы Википедии // Википедия, свободная энциклопедия. Сан-Франциско: Фонд Викимедиа, 2021. URL: <https://clck.ru/9nn8J> (дата обращения: 10.11.2020);
24. Создание первичных ключей данных [Электронный ресурс] : Microsoft 2021. URL: <https://docs.microsoft.com/ru-ru/sql/relational-databases/tables/create-primary-keys?view=sql-server-ver15> (дата обращения: 02.02.2021);
25. Сравнение сервисов GitHub и GitLab [Электронный ресурс] : Startpack 2021. URL: <https://startpack.ru/compare/github/gitlab#u61goizx> (дата обращения: 19.01.2021);
26. Сравнение современных СУБД [Электронный ресурс] : drach 2021. URL: <https://drach.pro/blog/hi-tech/item/145-db-comparison> (дата обращения: 12.01.2021);

27. Топ-20 систем управления обучением (LMS) на основе отзывов пользователей систем [Электронный ресурс] : hr-elearning.ru 2014-2020. URL: <https://hr-elearning.ru/top-20-sistem-upravleniya-obucheniem-lms-sistemy-obucheniya-personala/> (дата обращения: 20.11.2020);
28. Учебный центр [Электронный ресурс] : ООО «Неткрэкер» 2021. URL: <https://msk.edu-netcracker.com/> (дата обращения: 10.11.2020);
29. Часть 7. Знакомство с паттерном MVC (Model-View-Controller) [Электронный ресурс] : JavaRush 2021. URL: <https://javarush.ru/groups/posts/2536-chastjh-7-znakomstvo-s-patternom-mvc-model-view-controller> (дата обращения: 05.01.2021);
30. Что такое Bootstrap и зачем он нужен? [Электронный ресурс] : Александр Мальцев, 2014-2021. URL: <https://itchief.ru/bootstrap/introduction> (дата обращения: 08.01.2021);
31. ASP.NET Core [Электронный ресурс] : Авторы Википедии // Википедия, свободная энциклопедия. Сан-Франциско: Фонд Викимедиа, 2020. URL: [https://ru.wikipedia.org/wiki/ASP.NET\\_Core](https://ru.wikipedia.org/wiki/ASP.NET_Core) (дата обращения: 12.01.2021);
32. ClaimPrincipal и объекты Claim [Электронный ресурс] : metanit.com, 2012-2021. URL: <https://metanit.com/sharp/aspnet5/15.4.php> (дата обращения: 30.01.2021);
33. Entity Framework Core [Электронный ресурс] : Microsoft 2021. URL: <https://docs.microsoft.com/ru-ru/ef/core/> (дата обращения: 22.01.2021);
34. Transact-SQL [Электронный ресурс] : Авторы Википедии // Википедия, свободная энциклопедия. Сан-Франциско: Фонд Викимедиа, 2021. URL: <https://ru.wikipedia.org/wiki/Transact-SQL> (дата обращения: 15.03.2021);
35. UML [Электронный ресурс] : Авторы Википедии // Википедия, свободная энциклопедия. Сан-Франциско: Фонд Викимедиа, 2020. URL: <https://ru.wikipedia.org/?curid=68866&oldid=108935326> (дата обращения: 20.11.2020);



36. Enterprise Architect [Электронный ресурс] : Sparx Systems Pty Ltd, 2000 – 2021. URL: <https://sparxsystems.com/> (дата обращения: 10.11.2020);
37. Figma [Электронный ресурс] : Figma, 2021. URL: <https://www.figma.com/> (дата обращения: 10.12.2020);
38. Learn Amp [Электронный ресурс] : learnamp, 2021. URL: <https://learnamp.com/> (дата обращения: 10.11.2020);
39. Microsoft Visio [Электронный ресурс] : Microsoft 2021. <https://www.microsoft.com/en-gb/microsoft-365/visio/flowchart-software/> (дата обращения: 10.11.2020);
40. PythonAnywhere [Электронный ресурс] : PythonAnywhere LLP, 2011-2021. URL: <https://www.pythonanywhere.com/> (дата обращения: 05.02.2021);
41. Selenium Automation Test Cases For The .NET Web Application [Электронный ресурс] : C# Corner 2021. URL: <https://www.c-sharpcorner.com/article/selenium-automation-test-cases-for-the-net-web-application/> (дата обращения: 05.04.2021);
42. StarUML [Электронный ресурс] : MKLabs Co.,Ltd, 2014-2021. URL: <https://staruml.io/> (дата обращения: 10.11.2020);
43. Web Framework Benchmarks [Электронный ресурс] : techempower 2021. URL: <https://www.techempower.com/benchmarks/> (дата обращения: 10.01.2021);
44. Web Frameworks [Электронный ресурс] : awesomeopensource 2021. URL: <https://awesomeopensource.com/project/the-benchmark/web-frameworks> (дата обращения: 10.01.2021);
45. Why Looor? [Электронный ресурс] : Looor 2021. URL: <https://www.looor.co/why-looor> (дата обращения: 10.11.2020).

## Приложение А

### Диаграмма BPMN «как-есть» процесса заочного обучения

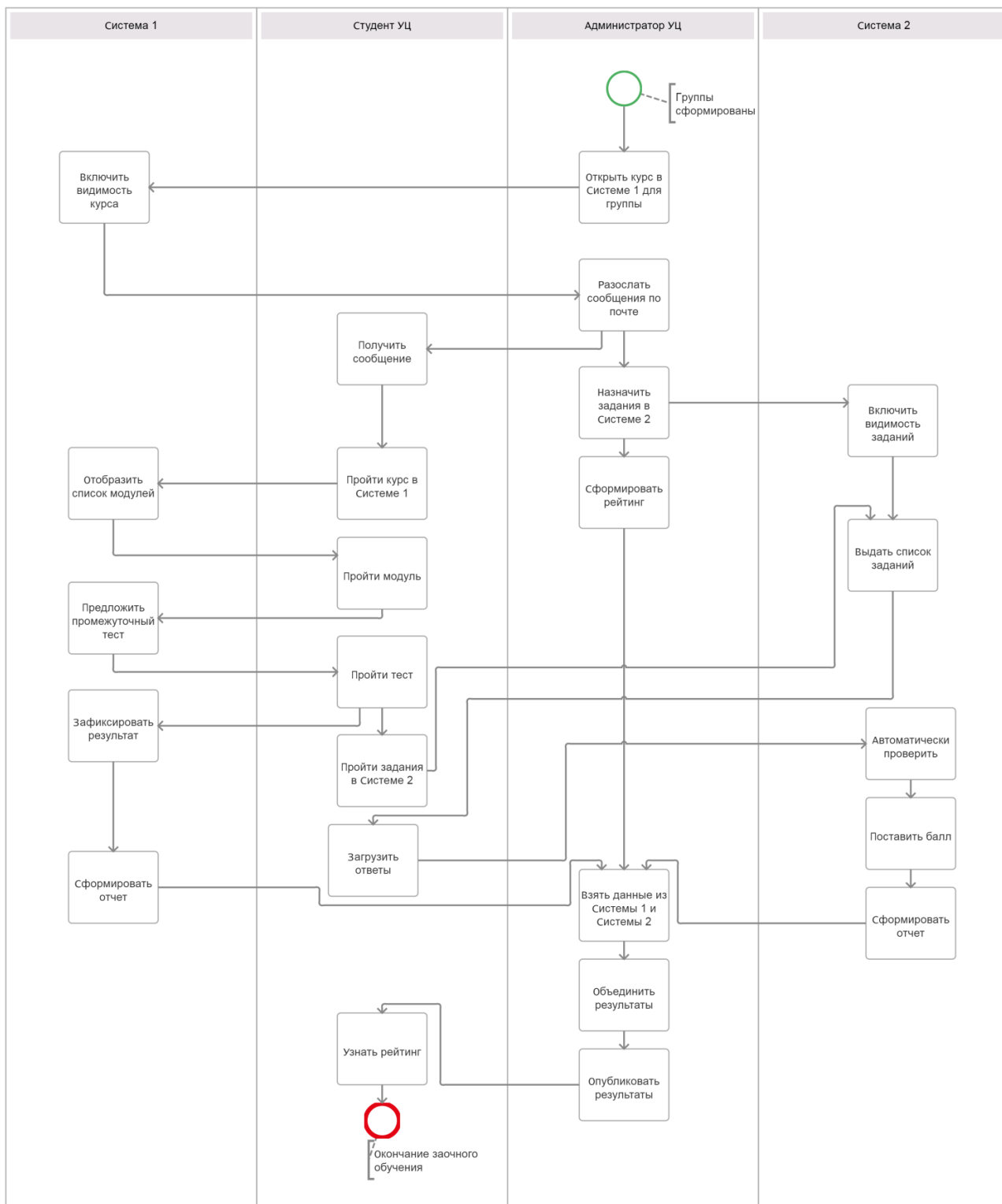


Рисунок А.1 – Диаграмма BPMN «как-есть» процесса заочного обучения

## Приложение Б

### Диаграмма BPMN «как-должно быть» процесса заочного обучения

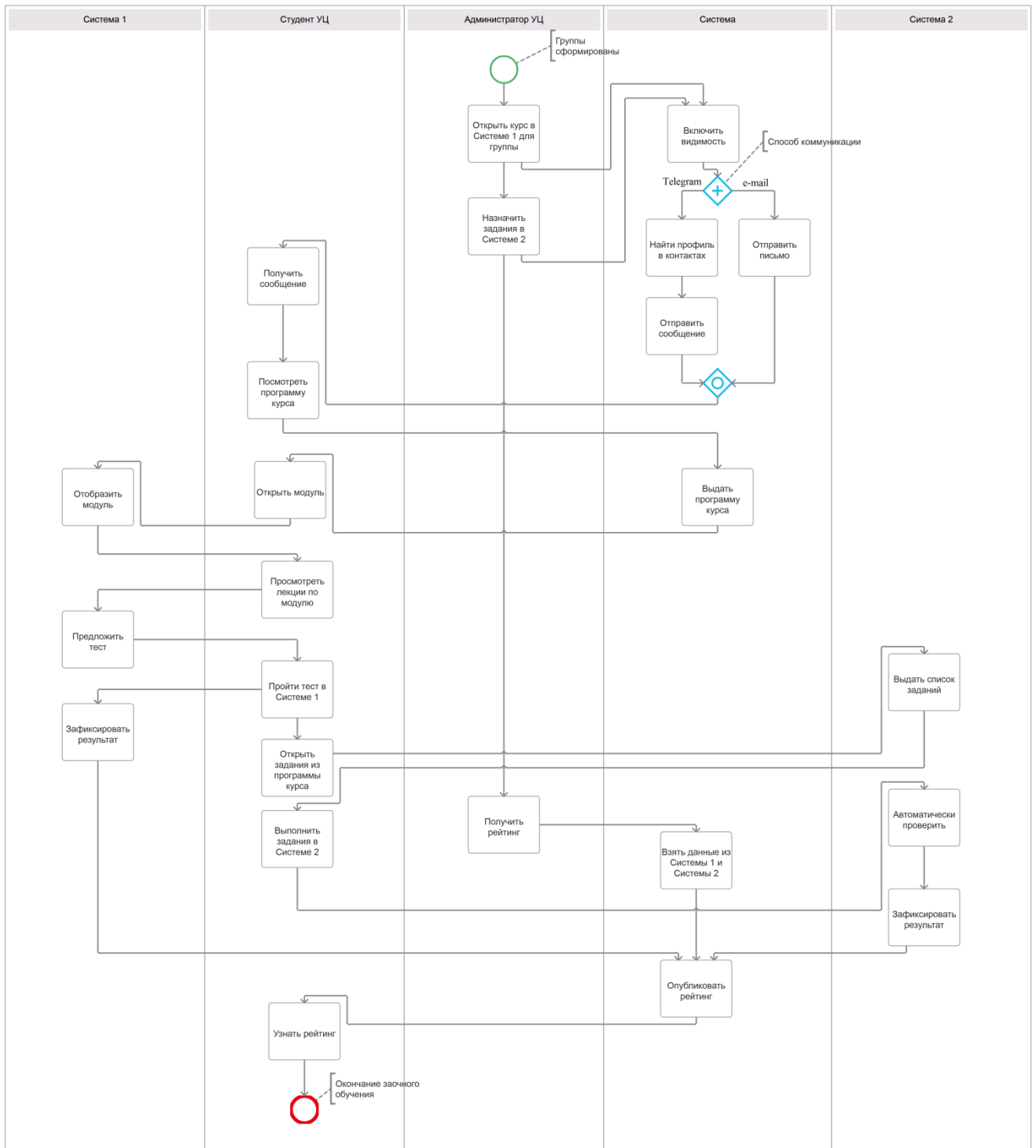


Рисунок Б.1 – Диаграмма BPMN «как-должно быть» процесса заочного обучения

## Приложение В

### Диаграмма последовательности набора на курсы

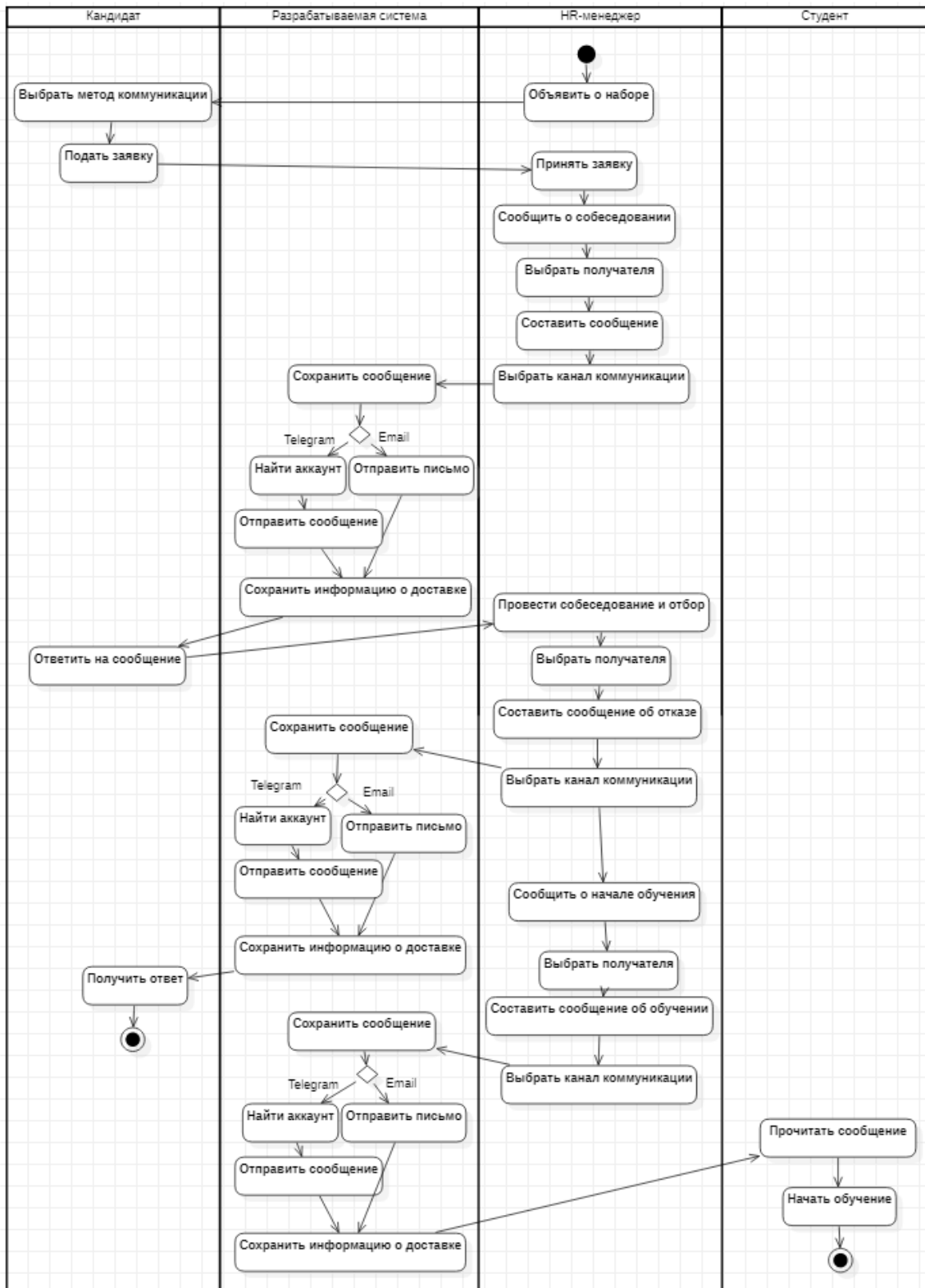


Рисунок В.1 – Диаграмма последовательности набора на курсы

## Приложение Г

### Диаграмма последовательности заочного обучения

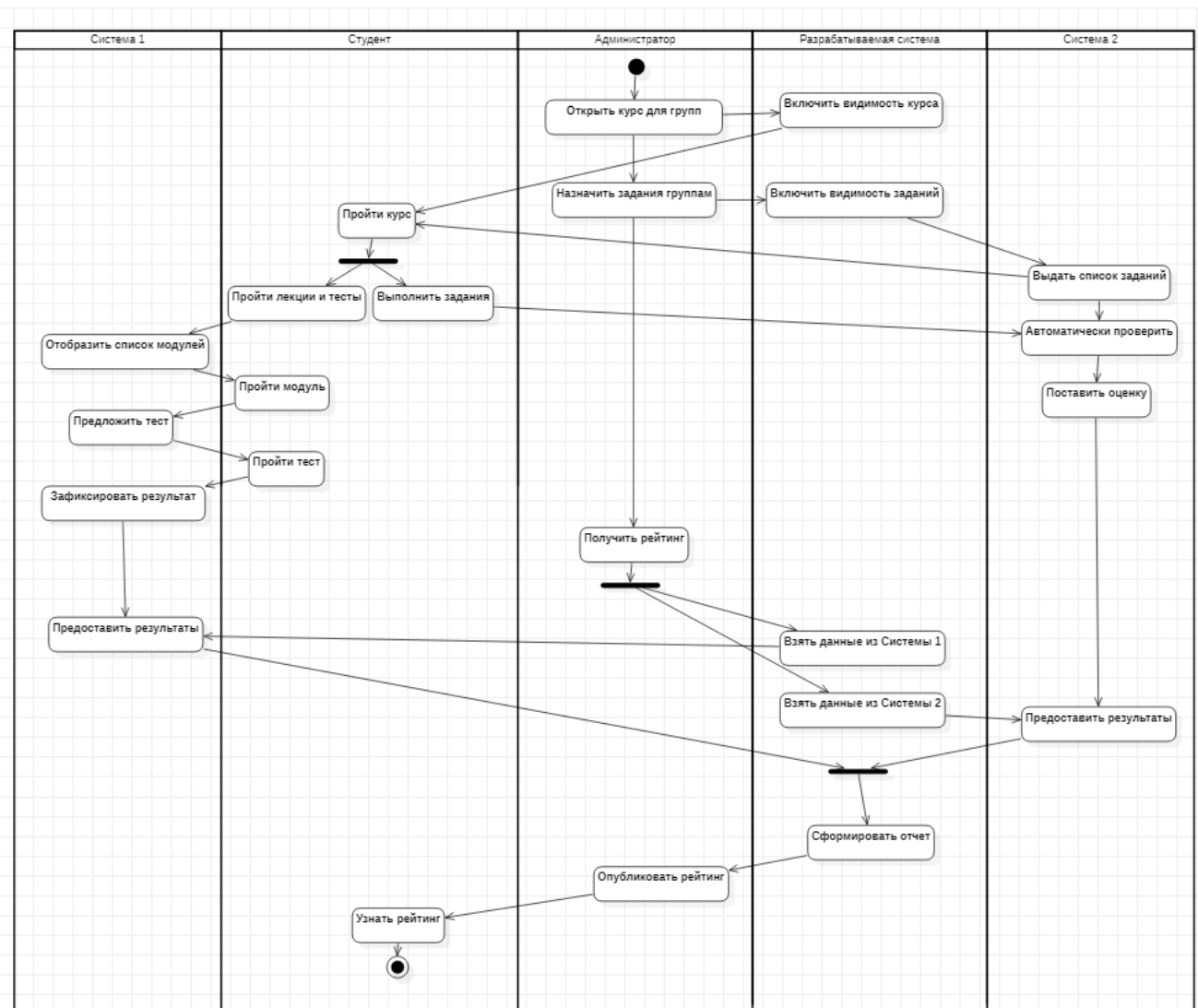


Рисунок Г.1 – Диаграмма последовательности заочного обучения

# Приложение Д Описание классов

На рисунке Д.1 представлена диаграмма классов.

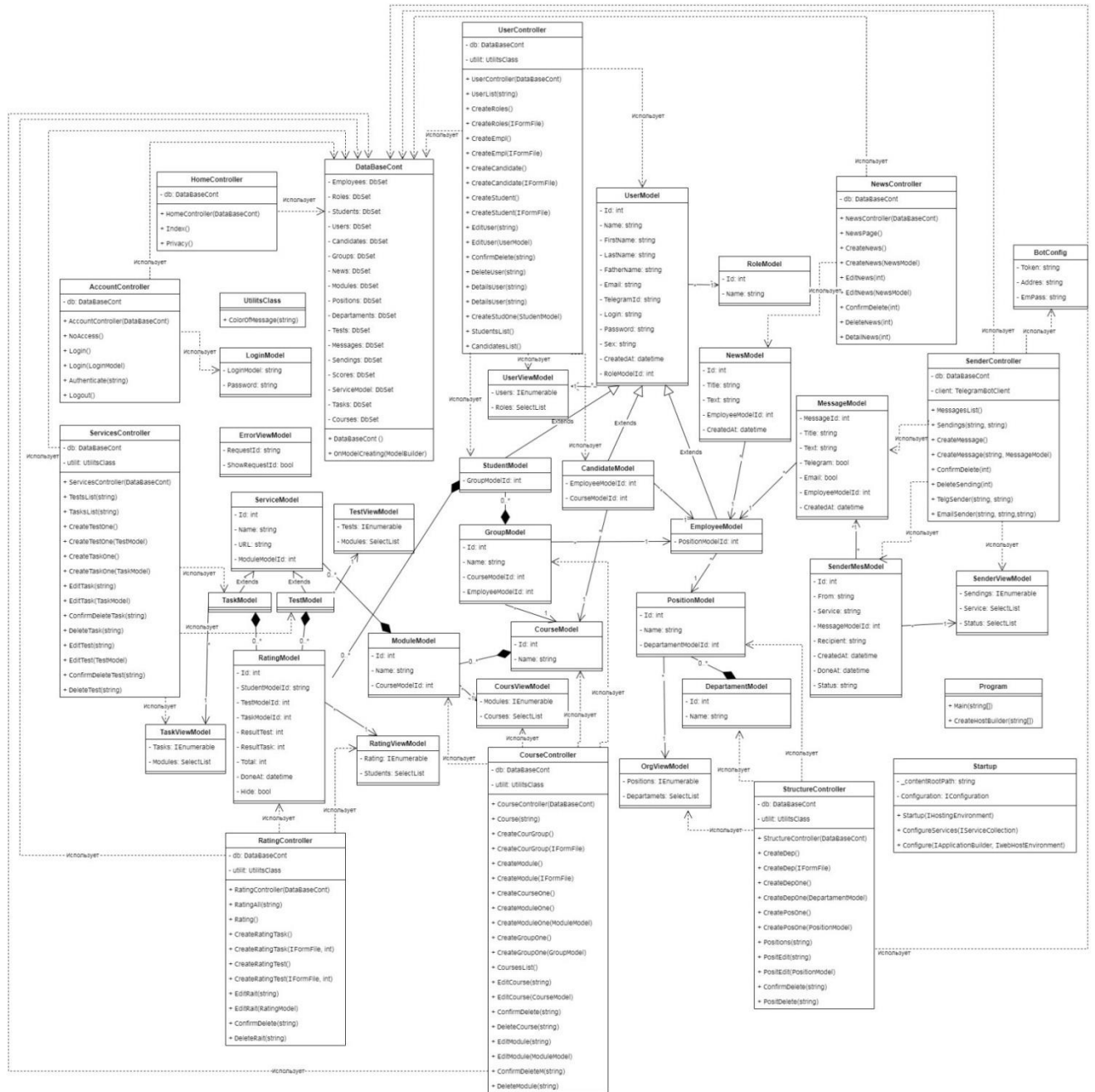


Рисунок Д.1 – Диаграмма классов

## Продолжение Приложения Д

Описание классов перечислено ниже:

- класс `UserModel` содержит в себе поля, которые необходимы при работе как с сотрудниками, так и со студентами. Класс содержит персональную информацию и данные для работы в системе;
- класс `StudentModel` – студент учебного центра. Наследует свойства `UserModel`, добавляется принадлежность к учебной группе;
- класс `EmployeeModel` – сотрудники компании. Наследует свойства и методы `UserModel`, добавляется принадлежность к должности. Должность имеет связь с департаментом;
- класс `GroupModel` – учебная группа студентов. Связан с классом `StudentModel` агрегацией, один ко многим соответственно. Содержит `id` студентов, куратора, курса;
- класс `CourseModel` – учебный курс. Связан с классами `GroupModel` и `EmployeeModel` ассоциацией. Несколько групп проходят один курс. Один сотрудник назначен на несколько курсов. Содержит поля: имя, `id`;
- класс `ModuleModel` – модуль курса, связан с `CourseModel` композицией. Содержит поля: имя, `id`, ссылку на курс;
- класс `ServiceModel` – общий класс сервисов учебного модуля. Связан с классом `ModuleModel` агрегацией, многие к одному соответственно. Содержит в себе `id`, ссылку, название, ссылку на модуль;
- класс `TestModel` – тест в рамках учебного модуля. Наследует свойства и методы `ServiceModel`;
- класс `TaskModel` – задание в рамках учебного модуля. Наследует свойства и методы `ServiceModel`;

## Продолжение Приложения Д

- класс RatingModel содержит в себе ссылку на задания и тесты, ссылку студента с результатом выполнения. Связан с классом TestModel и TaskModel агрегацией;
- класс NewsModel – новость, содержит в себе заголовок, текст, ссылку на сотрудника. Связан с классом EmployeeModel зависимостью;
- класс MessageModel – сообщение, содержит в себе тему, текст, ссылку на отправителя и тип коммуникации. Связан с классом EmployeeModel зависимостью;
- класс SenderMesModel – отправление сообщения, содержит в себе ссылку на сообщение, ссылку на отправителя, получателя и тип коммуникации, когда создано и отправлено. Связан с классом MessageModel зависимостью;
- класс UtilitsClass содержит функции, которые часто используются в контроллерах;
- Класс DataBaseCont предназначен для работы вышеперечисленных сущностей с базой данных;
- класс LoginModel – содержит в себе логин и пароль для входа в систему;
- контроллеры и ViewModel используют экземпляры классов, в том числе класса DataBaseCont;
- класс BotConfig отвечает за данные для чат-бота и e-mail рассылок;
- класс Program запускает приложение, а класс Startup управляет маршрутизацией и запускает СУБД.



## Приложение Е

### Дизайн-макеты форм ввода и вывода информационной системы

The form titled "Создать кандидата" (Create Candidate) contains the following fields and controls:

- Id:
- Имя:
- Email:
- Telegramid:
- Пол:
- Курс:
- Сотрудник:
- Создать:

Рисунок Е.1 – Форма заполнения данных вручную

The form titled "Добавить кандидатов" (Add Candidates) includes the following elements:

- Описание структуры файла: [Описание структуры файла](#)
- Пример листа .XLSX: [Пример листа .XLSX](#)
- Загрузите файл:
- Загрузить:

Рисунок Е.2 – Форма заполнения данных в массовом режиме

The form titled "Редактирование пользователя" (Edit User) contains the following fields and controls:

- Id:
- ФИО:
- Email:
- Telegram Id:
- Логин:
- Пароль:
- Пол:
- Роль:
- Сохранить:

Рисунок Е.3 – Форма редактирования данных

## Продолжение Приложения Е

### Отправить сообщение

Получатель: Иванов Олег

Тема

Текст

Telegram

Email

Отправить

Сбросить

[Список сообщений](#)

## Рисунок Е.4 – Форма создания сообщения

### Создать новость

Заголовок

Текст

Создать

Сбросить

## Рисунок Е.5 – Форма создания новости

### Редактирование новости

Заголовок

новость 3

Текст

новость 3

Создано

12.11.2020 21:36:05,928



Сохранить

## Рисунок Е.6 – Форма редактирования новости

## Продолжение Приложения Е

Пожалуйста, введите свой  
логин и пароль

Логин

Пароль

Войти

Рисунок Е.7 – Форма входа в систему

### Последние новости

<b>новость 1.1</b> 12.01.2021 Не следует, однако забывать, что укрепление и развитие структуры обеспечивает широкому кругу (специалистов) участие в формировании направлений прогрессивного развития. <a href="#">Подробнее</a>	<b>новость 3</b> 12.01.2021 новость 3 <a href="#">Подробнее</a>	<b>новость 2</b> 12.01.2021 новость 2 <a href="#">Подробнее</a>
--	--	--

Рисунок Е.8 – Форма вывода новостей (главная страница)

### Все новости

[Добавить новость](#)

<b>новость 1.1</b> 12.01.2021 Не следует, однако забывать, что укрепление и развитие структуры обеспечивает широкому кругу (специалистов) участие в формировании направлений прогрессивного развития. <a href="#">Редактировать</a> <a href="#">Удалить</a>	<b>новость 3</b> 12.01.2021 новость 3 <a href="#">Редактировать</a> <a href="#">Удалить</a>	<b>новость 2</b> 12.01.2021 новость 2 <a href="#">Редактировать</a> <a href="#">Удалить</a>
--	--	--

Рисунок Е.9 – Форма вывода новостей (все новости)

## Продолжение Приложения Е

### Список сообщений

Id сообщения	Тема	Текст	Телеграм?	E-mail?	Отправший сотрудник	Создано
1	Тестовое сообщение	Тестовое сообщение	<input type="checkbox"/>	<input checked="" type="checkbox"/>	9	01.12.2020 00:00:00

Рисунок Е.10 – Форма вывода списка сообщений

### Список отправлений

[Список сообщений](#)

Сервис:  Статус:

Id транзакции	Создано	Сервис	От	Кому	Id сообщения	Статус	Отправлено	Действия
2	01.12.2020 00:00:00	Email	9	m@yandex.ru	1	Отправлено	01.12.2020 00:00:10	<a href="#">Удалить</a>
1	01.12.2020 00:00:00	Telegram	9	6111124	1	Отправлено	01.12.2020 00:00:10	<a href="#">Удалить</a>

Рисунок Е.11 – Форма вывода списка отправлений

### Список кандидатов

[Добавление записей](#)

#### Кандидаты

Курс:

Id пользователя	ФИО	Пол	Курс	Email	Telegram	Действия
1	Алексеева Маргарита Руслановна	ж	Enterprise Java Development	caterina15@yandex.ru	643963324	<a href="#">Связаться</a>   <a href="#">Подробнее</a>   <a href="#">Редактировать</a>   <a href="#">Удалить</a>
2	Новикова Мария Глебовна	ж	Technical Sales	caterina15@yandex.ru	643963324	<a href="#">Связаться</a>   <a href="#">Подробнее</a>   <a href="#">Редактировать</a>   <a href="#">Удалить</a>

Рисунок Е.12 – Форма вывода списка кандидатов

## Продолжение Приложения Е

### Список студентов

[Добавление записей](#)

Группа:

Id пользователя	ФИО	Пол	Группа	Email	Telegram	Действия
3	Михайлова Вероника Романовна	ж	DEV-2020_a	moiseewa.caterina15@yandex.ru	643963324	<a href="#">Связаться</a>   <a href="#">Подробнее</a>   <a href="#">Редактировать</a>   <a href="#">Удалить</a>
4	Савельева София Ивановна	ж	BA-2020_a	caterina15@yandex.ru	643963324	<a href="#">Связаться</a>   <a href="#">Подробнее</a>   <a href="#">Редактировать</a>   <a href="#">Удалить</a>

Рисунок Е.13 – Форма вывода списка студентов

### Общий рейтинг

[Отдельные транзакции](#)

Студент	Тесты	Задачи	Всего
Михайлова Вероника Романовна	412	446	858
Михеев Георгий Александрович	520	522	1042
Савельева София Ивановна	508	528	1036

Рисунок Е.14 – Форма вывода сводного рейтинга студентов УЦ

## Приложение Ж Физическая диаграмма базы данных

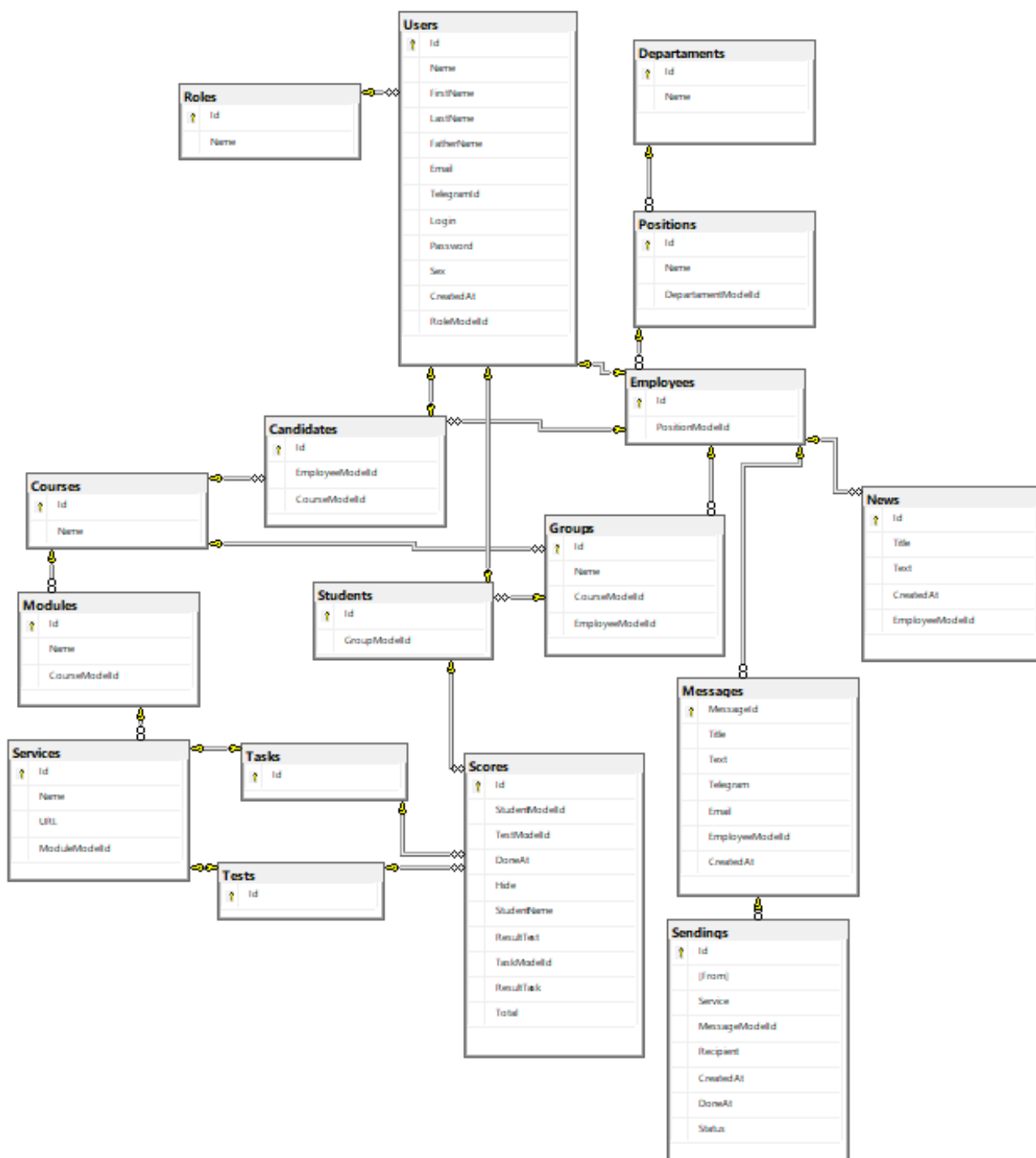


Рисунок Ж.1 – Физическая диаграмма базы данных

**Приложение И**  
**Описание графа сценариев**

Таблица И.1 – Описание графа сценариев

№ вершины графа	Сценарий
1	Добавление пользователей
2	Вход в систему
3	Проверка пользователя в базе данных
4	Такого пользователя не существует в базе данных
5	Такой пользователь есть в базе данных
6	Проверка роли (уровня доступа)
7	Роль – администратор
8	Роль – HR-менеджер
9	Роль – студент
10	Создание, просмотр, редактирование, удаление пользователей
11	Создание, просмотр, редактирование, удаление курсов
12	Создание, редактирование, удаление модулей
13	Создание, редактирование, удаление тестов
14	Создание, редактирование, удаление заданий
15	Создание, просмотр, редактирование, удаление транзакций рейтинга
16	Просмотр рейтинга
17	Создание, просмотр, редактирование, удаление новости
18	Создание, просмотр, отправка сообщения пользователю
19	Просмотр списка студентов и групп
20	Просмотр списка кандидатов
21	Просмотр модулей
22	Просмотр тестов
23	Просмотр заданий
24	Просмотр новостей
25	Выход из системы

## Приложение К

### Скриншоты страниц информационной системы

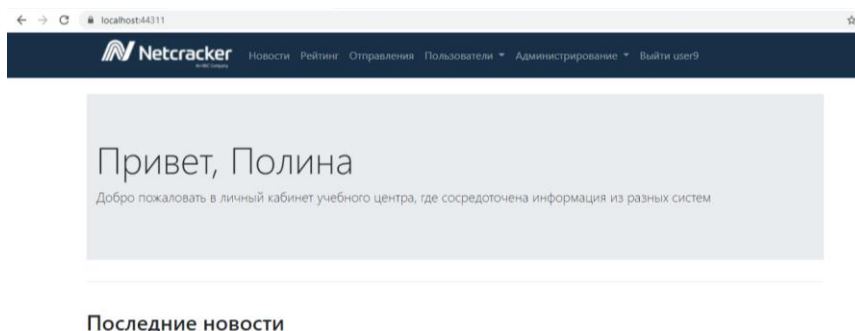


Рисунок К.1 – Вид главной страницы у администратора

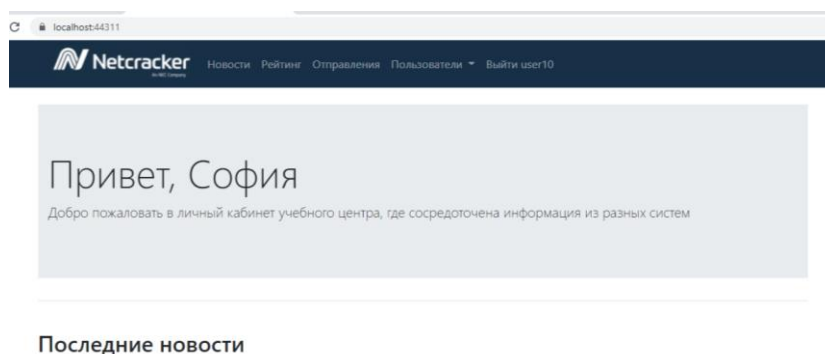


Рисунок К.2 – Вид главной страницы у HR-менеджера

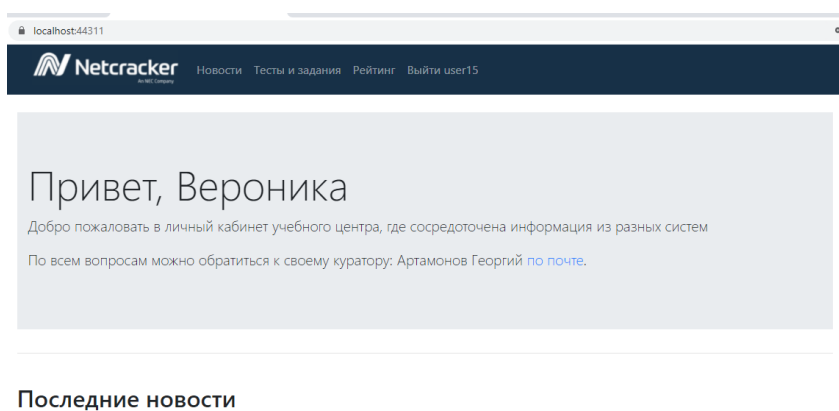


Рисунок К.3 – Вид главной страницы у студента



## Продолжение Приложения К

The screenshot shows a web browser window with the address bar displaying 'localhost:44311/Course/CreateModuleOne'. The page header includes the Netcracker logo and navigation links: 'Новости', 'Рейтинг', 'Отправления', 'Пользователи', 'Администрирование', and 'Выйти user9'. The main heading is 'Создать модуль'. Below it, there are three input fields: 'Id', 'Название', and 'Курс'. The 'Курс' dropdown menu is set to 'Enterprise Java Development'. A blue 'Создать' button is positioned below the form. At the bottom, there is a link 'Список модулей'.


Рисунок К.4 – Вид страницы создания модуля

The screenshot shows a web browser window with the address bar displaying 'localhost:44311/Course/CreateModule'. The page header is identical to the previous screenshot. The main heading is 'Добавить модули и задания'. Below it, there is a section titled 'Описание структуры файла' which contains instructions on how to format the XLSX file: 'Название модуля формируется по принципу "Название id" (обязательно id должен быть отделен пробелом от названия) и помещается в заголовок листа .XLSX.', 'Задания помещаются со второй строки', and a list of columns: '1я колонка - Id;', '2я колонка - Название;', '3я колонка - Тип (Test или Task);', and '4я колонка - URL.'. It also states 'Список заданий модуля помещается на соответствующий лист .XLSX.'. Below this is a link 'Пример листа .XLSX'. The next section is 'Загрузите файл', featuring a file selection input with a 'Browse' button. Below that is a 'Выберите курс' dropdown menu set to 'Enterprise Java Development' and a 'Загрузить' button.

Рисунок К.5 – Вид страницы создания модуля и заданий

## Продолжение Приложения К

localhost:44311/UserList

 [Новости](#) [Рейтинг](#) [Отправления](#) [Пользователи](#) [Администрирование](#) [Выйти user9](#)

### Список пользователей

[Добавление записей](#)

Роль:

Id	ИО	Логин	Пол	Роль	Действия
1	Степанов Георгий Степанович	user1	м	Tutor	<a href="#">Связаться</a>   <a href="#">Подробнее</a>   <a href="#">Редактировать</a>   <a href="#">Удалить</a>
2	Волков Егор Григорьевич	user2	м	Tutor	<a href="#">Связаться</a>   <a href="#">Подробнее</a>   <a href="#">Редактировать</a>   <a href="#">Удалить</a>
3	Корнеев Григорий Адамович	user3	м	Tutor	<a href="#">Связаться</a>   <a href="#">Подробнее</a>   <a href="#">Редактировать</a>   <a href="#">Удалить</a>

Рисунок К.6 – Вид страницы списка пользователей

**Приложение Л**  
**Список тестовых кейсов и результат их выполнения**

Таблица Л.1 – Список тестовых кейсов и результат их выполнения

№ ТК	Модуль	Приоритет	Название кейса	Статус выполнения	Результат
1	Вход в систему	Высокий	Войти в систему, как администратор (позитивный кейс)	Закончен	Пройден
2	Вход в систему	Средний	Войти в систему, как администратор (негативный кейс)	Закончен	Пройден
3	Вход в систему	Высокий	Войти в систему, как HR-менеджер	Закончен	Пройден
4	Вход в систему	Высокий	Войти в систему, как студент	Закончен	Пройден
5	Новости	Высокий	Создание новости от администратора (позитивный кейс)	Закончен	Пройден
6	Новости	Средний	Создание новости от администратора (негативный кейс)	Закончен	Пройден
7	Новости	Высокий	Редактирование новости от администратора (позитивный кейс)	Закончен	Пройден
8	Новости	Средний	Редактирование новости от администратора (негативный кейс)	Закончен	Пройден
9	Новости	Высокий	Удаление новости администратором	Закончен	Пройден
10	Новости	Высокий	Создание новости от HR-менеджера (позитивный кейс)	Закончен	Пройден
11	Новости	Средний	Создание новости от HR-менеджера (негативный кейс)	Закончен	Пройден
12	Новости	Высокий	Редактирование новости от HR-менеджера (позитивный кейс)	Закончен	Пройден
13	Новости	Средний	Редактирование новости от HR-менеджера (негативный кейс)	Закончен	Пройден
14	Новости	Высокий	Удаление новости HR-менеджером	Закончен	Пройден
15	Новости	Средний	Создание новости студентом	Закончен	Пройден
16	Новости	Средний	Редактирование новости студентом	Закончен	Пройден
17	Новости	Средний	Удаление новости студентом	Закончен	Пройден
18	Сообщения	Высокий	Отправить сообщение кандидату в Telegram и E-mail от администратора (позитивный кейс)	Закончен	Пройден
19	Сообщения	Высокий	Отправить сообщение студенту в Telegram и E-mail от администратора (негативный кейс)	Закончен	Провален
20	Сообщения	Высокий	Отправить сообщение студенту в Telegram от администратора (позитивный кейс)	Закончен	Пройден

Продолжение Приложения Л

Продолжение таблицы Л.1

№ ТК	Модуль	Приоритет	Название кейса	Статус выполнения	Результат
21	Сообщения	Высокий	Отправить сообщение кандидату в Telegram и E-mail от HR-менеджера (позитивный кейс)	Закончен	Пройден
23	Сообщения	Высокий	Отправить сообщение студенту в Telegram от HR-менеджера (позитивный кейс)	Закончен	Пройден
24	Пользователи	Высокий	Создание студента администратором в одиночном режиме	Закончен	Пройден
25	Пользователи	Высокий	Создание студента администратором в массовом режиме	Закончен	Пройден
26	Пользователи	Высокий	Создание кандидата администратором в одиночном режиме	Закончен	Пройден
27	Пользователи	Высокий	Создание кандидата администратором в массовом режиме	Закончен	Пройден
28	Курсы	Высокий	Создание курса администратором в одиночном режиме	Закончен	Пройден
29	Курсы	Высокий	Создание курсов администратором в массовом режиме	Закончен	Пройден
30	Модули	Высокий	Создание модуля администратором в одиночном режиме	Закончен	Пройден
31	Модули	Высокий	Создание модуля администратором в массовом режиме	Закончен	Пройден
32	Тесты	Высокий	Создание теста администратором в одиночном режиме	Закончен	Пройден
33	Тесты	Высокий	Создание тестов администратором в массовом режиме	Закончен	Пройден
34	Задания	Высокий	Создание задания администратором в одиночном режиме	Закончен	Пройден
35	Задания	Высокий	Создание заданий администратором в массовом режиме	Закончен	Пройден
36	Рейтинг	Высокий	Создание транзакции администратором	Закончен	Пройден
37	Рейтинг	Высокий	Просмотр транзакций администратором	Закончен	Пройден
38	Рейтинг	Высокий	Просмотр сводного рейтинга администратором	Закончен	Пройден
39	Рейтинг	Высокий	Просмотр сводного рейтинга HR-менеджером	Закончен	Пройден

Продолжение Приложения Л

Продолжение таблицы Л.1

№ ТК	Модуль	Приоритет	Название кейса	Статус выполнения	Результат
40	Рейтинг	Высокий	Просмотр сводного рейтинга студентом	Закончен	Пройден
41	Рейтинг	Высокий	Редактирование транзакций студентом	Закончен	Пройден
42	Орг. структура	Средний	Создание департамента администратором в одиночном режиме	Закончен	Пройден
43	Орг. структура	Средний	Создание департаментов и должностей администратором в массовом режиме	Закончен	Пройден

Таблица Л.2 – Пример тестового кейса

№ ТК	Модуль	Название кейса	Шаги	Ожидаемый результат
18	Сообщения	Отправить сообщение кандидату в Telegram и E-mail от HR-менеджера (позитивный кейс)	<ol style="list-style-type: none"> <li>1. Войти в систему, как HR-менеджер; нажать Пользователи&gt;Кандидаты; у первой строки нажать на «Связаться»;</li> <li>2. Ввести в поле Тема «18_Тестовая тема»;</li> <li>3. Ввести в поле Текст «18_Тестовое сообщение»;</li> <li>4. Выбрать Email и Telegram;</li> <li>5. Нажать «Отправить»;</li> <li>6. Проверить E-mail в почтовом сервисе и сообщение в Telegram</li> </ol>	<ol style="list-style-type: none"> <li>1. Вход в систему выполнен успешно; открылась страница со списком кандидатов; открылась форма создания сообщения;</li> <li>5. Открылась страница отправлений. Первые 2 строки имеют статус «Отправлено»;</li> <li>6. Получено письмо с темой «18_Тестовая тема» и текстом «18_Тестовое сообщение»; получено сообщение «18_Тестовое сообщение»</li> </ol>