

С:\Users\WooDuFX\Documents\ МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра технологий обработки и защиты информации

Курсовой проект

Система учета успеваемости студентов «Атао»

09.03.02 Информационные системы и технологии

Информационные технологии управления

Обучающийся _____ *Д.Н.Лиманский, 3 курс, д/о*

Обучающийся _____ *Я.С.Неровный, 3 курс, д/о*

Обучающийся _____ *С.С.Смородинова, 3 курс, д/о*

Воронеж 2021

Содержание

Содержание	2
Введение	3
1 Постановка задачи	4
2 Глоссарий.....	5
3 Анализ предметной области	7
3.1 Актуальность.....	7
3.2 Анализ существующих решений	7
4 Анализ задачи.....	9
5 Анализ средств реализации	19
6 Воронки конверсии.....	21
7 Реализация	22
8 Тестирование	23
Заключение	24
Список используемых источников.....	Ошибка! Закладка не определена.

Введение

В настоящее время всё более и более актуальными становятся интернет-сервисы и электронные ресурсы. Люди стремятся упростить свою жизнь посредством современных технологий. Многим из них удобнее открыть телефон или компьютер, чтобы почитать любимую книгу, получить образовательную информацию или выполнить иную операцию. Раньше эти процессы занимали огромное количество времени, а сейчас мы можем их выполнить буквально за пару кликов. Данные новшества стали особенно актуальны в сфере науки и образования. В частности, для того, чтобы сократить количество бумажных ресурсов и перевести пользователей в Интернет. Это непосредственно затронуло и систему учета успеваемости школьников, студентов. Тем самым возросла необходимость в реализации удобных сервисов, которые могли бы заменить дневники и бумажные ведомости.

Данный курсовой проект посвящен разработке веб-приложения, которое позволит решить данную проблему.

Сайт должен облегчать ведение учета успеваемости студентов. Основная функциональность сайта должна включать в себя:

- Просмотр оценок,
- Выставление оценок в ведомости преподавателями,
- Процесс добавления и редактирования пользователей и учебных дисциплин.

1 Постановка задачи

Целью курсового проекта является создание веб-приложения, с помощью которого пользователи смогут вести и просматривать успеваемость студентов на факультете.

Данное веб-приложение должно отвечать следующим требованиям:

- Просмотр успеваемости студентов
- Возможность выставления оценок по учебным дисциплинам
- Разделение пользовательского интерфейса на 3 активные роли (администратор, преподаватель, студент)

Для достижения данной цели были выделены следующие подзадачи:

- Разработка Front-end части сервиса;
- Разработка Back-end части сервиса;
- Создание связи между Front-end и Back-end частями приложения;
- Разработка базы данных.

2 Глоссарий

Администратор - пользователь сайта, зарегистрированный и авторизованный в приложении, имеющий возможность редактировать данные студентов и преподавателей, редактировать и создавать учебные дисциплины, добавлять новых пользователей в систему, пользоваться личным кабинетом и общедоступными возможностями сайта.

Преподаватель - пользователь сайта, зарегистрированный и авторизованный в приложении, имеющий возможность выставлять оценки по преподаваемым предметам, пользоваться личным кабинетом и общедоступными разделами сайта.

Студент - пользователь сайта, зарегистрированный и авторизованный в приложении, имеющий возможность просматривать свои оценки, пользоваться личным кабинетом и общедоступными разделами сайта.

Гость - пользователь сайта, еще не зарегистрированный или не авторизованный в приложении.

Личный кабинет - это раздел сервиса, в котором Пользователь может получить доступ к своим данным и изменить некоторые из них.

Общедоступные разделы сайта - информационные разделы сайта, доступные всем пользователям («о нас», «контакты» и т.п.).

front-end - клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

back-end - программно-аппаратная часть сервиса.

Use Case диаграмма – диаграмма, которая позволяет описать функциональность приложения и поведения ролей для того чтобы заказчик и исполнитель могли совместно обсуждать проектируемую или существующую систему.

Инвайт-код – уникальный код, выдаваемый незарегистрированному пользователю администратором, который позволяет произвести регистрацию пользователя.

REST API - это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов.

GitHub - крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

Header - визуальный элемент, расположенный в верхней части страницы.

Footer - визуальный элемент, расположенный в нижней части страницы.

3 Анализ предметной области

3.1 Актуальность

Огромное число студентов еще со школьного времени привыкло следить за своей успеваемостью посредством электронных ресурсов, поэтому и в университете они зачастую хотят наблюдать за своими оценками путем отслеживания каких-нибудь интернет-ведомостей. Несмотря на актуальность поставленной задачи, в настоящее время на рынке веб-приложений существует небольшое количество сервисов, которые позволяют вести учет успеваемости студентов. Это создает широкий спектр действий для разработчиков, которые не ограничены конкуренцией и готовы предоставить на рынок свои уникальные решения.

Таким образом, очевидна потребность в сервисе, который бы позволял пользователям отслеживать свою успеваемость, а преподавателям – без особых проблем выставлять оценки по учебным дисциплинам.

3.2 Анализ существующих решений

На данный момент существует не так много сервисов для учета успеваемости студентов. Главным потенциальным конкурентом является сайт magellanius.ru – комплексная информационная система для университета, института, академии, курсов повышения квалификации.

Достоинства:

- Адаптивность и гибкость проекта
- Широкий спектр возможностей помимо учета успеваемости студентов
- Хранение информации по всему учебному процессу
- Быстрая и простая настройка сотрудниками образовательной организации

Недостатки:

- Высокая стоимость лицензии

- Учёт успеваемости является второстепенной задачей данного сервиса
- Отсутствие личного кабинета пользователя

Непрямым конкурентом также еще является сайт dnevnik.ru - цифровая образовательная платформа широкого спектра направленности. Его отличие заключается в том, что он направлен на сферу школьного образования. Данный сервис позволяет выяснить функции и возможности образовательных интернет ресурсов, пользующихся спросом, а также позволяет оценить перспективы развития подобных проектов.

4 Анализ задачи

4.1 Варианты использования системы

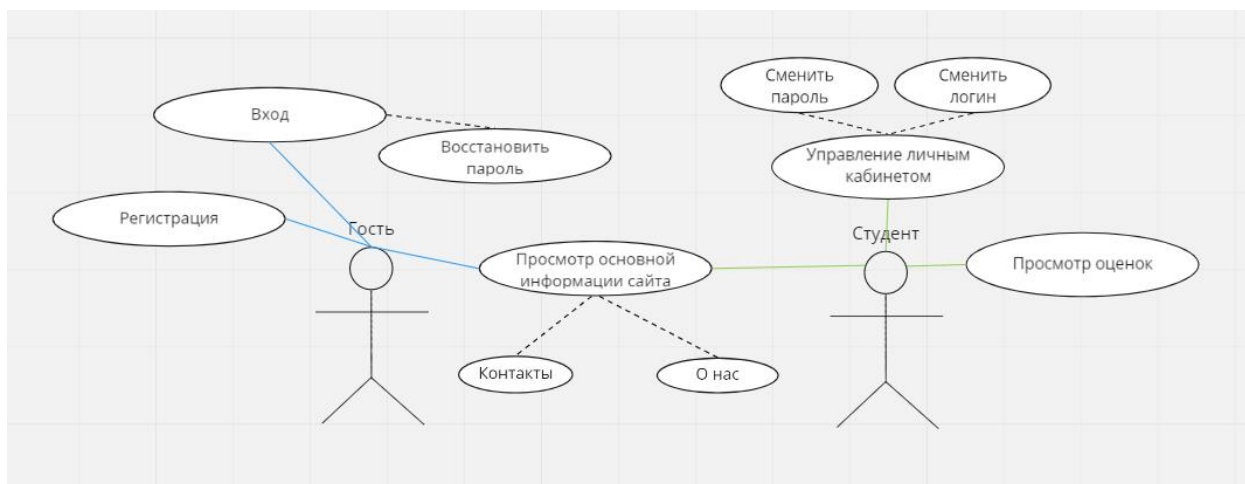


Рисунок 1 – Часть диаграммы прецедентов с актором Гость и Студент.

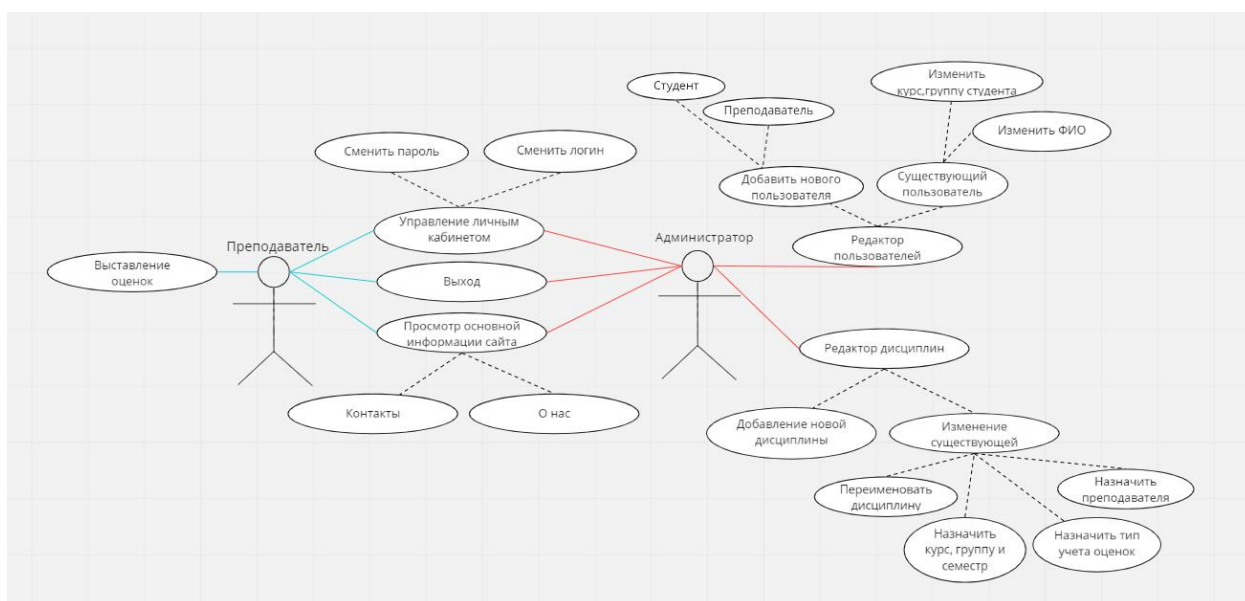


Рисунок 2 – Часть диаграммы прецедентов с актором Преподаватель и Администратор.

При взаимодействии с приложением выделяются четыре роли: гость, студент, преподаватель, администратор. Их роли наглядно изображены на рисунке 1 и на рисунке 2, а именно:

Гость:

- Авторизация и регистрации
- Просмотр общедоступных разделов сайта

Студент:

- Просмотра оценок
- Взаимодействие с личным кабинетом (смена логина и пароля)
- Просмотр общедоступных разделов сайта

Преподаватель:

- Выставления оценок по различным дисциплинам
- Взаимодействие с личным кабинетом (смена логина и пароля)
- Просмотр общедоступных разделов сайта

Администратор:

- Редактирование данных о пользователях
- Создание нового пользователя уникальной роли и выдача ему инвайт-кода
- Создание и редактирование учебных дисциплин
- Взаимодействие с личным кабинетом (смена логина и пароля)
- Просмотр общедоступных разделов сайта

4.2 Общая структура системы

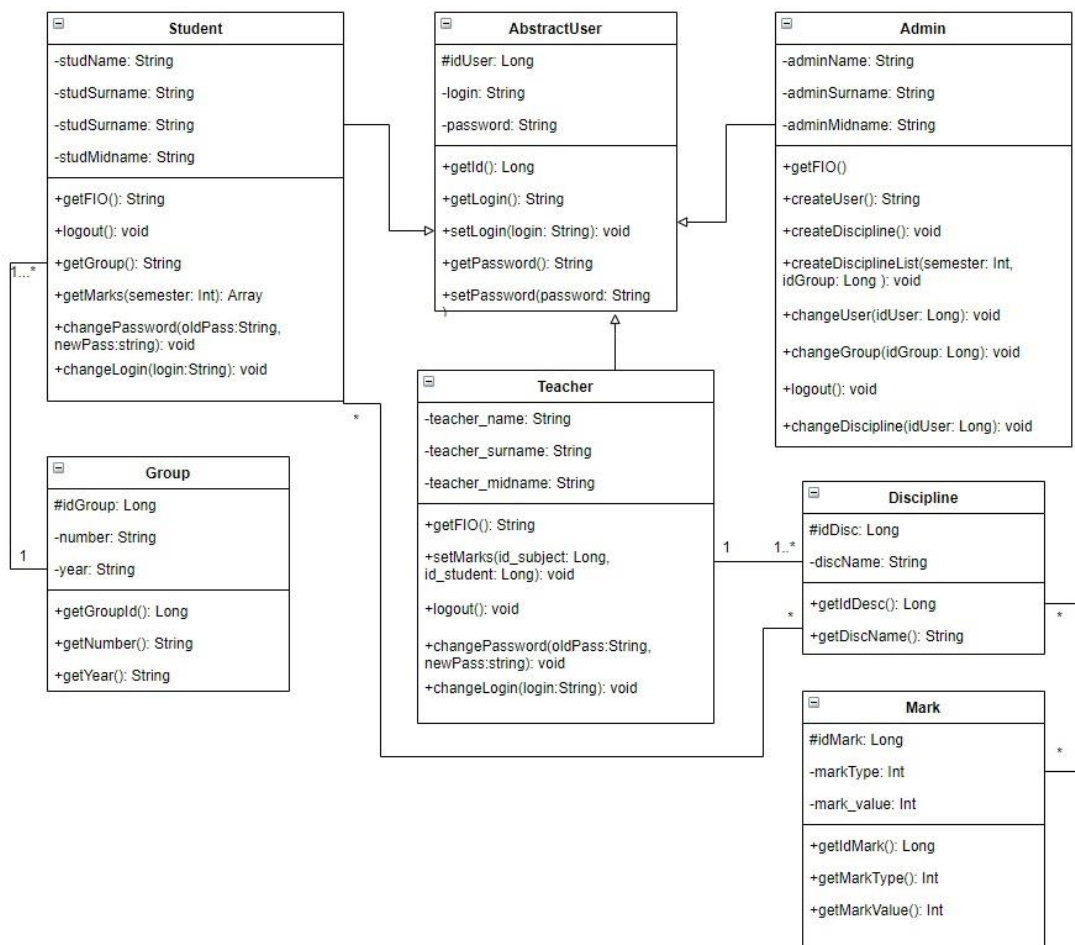


Рисунок 3 – Диаграмма классов.

На Рисунке 3 изображена диаграмма классов, отражающая их отношения.

Класс Student представляет собой отражение сущности Студент, Group – Группа, Teacher – Преподаватель, Admin – Администратор, Discipline – учебная дисциплина, Mark - оценка. Классы Admin, Student и Teacher наследуются от абстрактного класса AbstractUser. Все классы имеют

различные свойства и методы, указанные на рисунке 3.

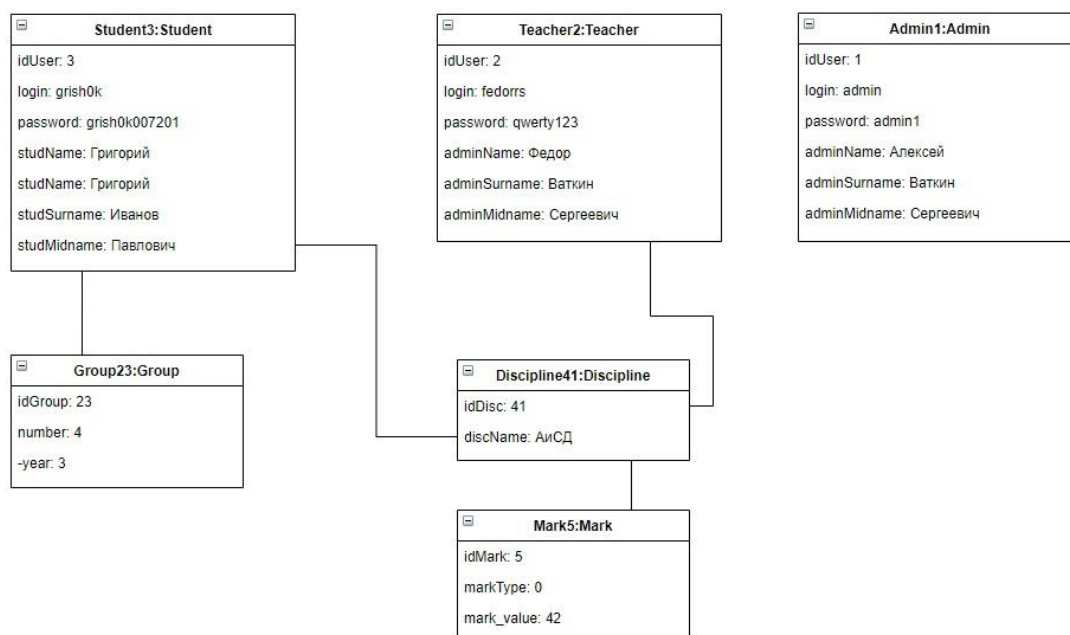


Рисунок 4 – Диаграмма объектов.

На Рисунке 4 изображена диаграмма объектов, которая отражает множество экземпляров классов и отношений между ними в некоторый момент времени. На ней изображён экземпляр класса Student, который хранит в себе ссылку на экземпляр класса Group, экземпляр класса Discipline, который хранит в себе ссылку на экземпляры классов Teacher и Student. Также на рисунке указан экземпляр класса Mark, который хранит в себе ссылку на экземпляр класса Discipline.

4.3 Взаимодействие компонентов системы

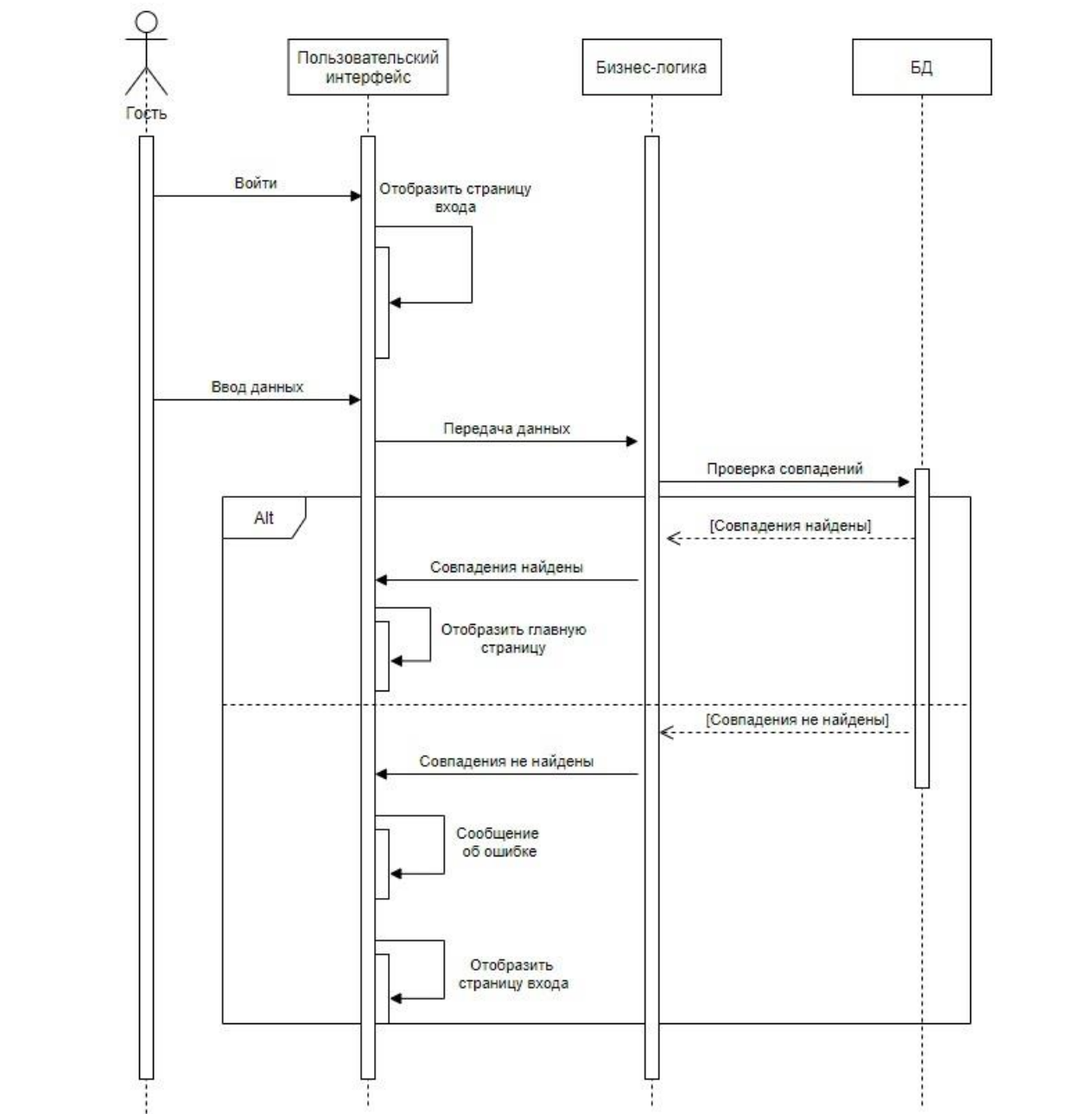


Рисунок 5 – Диаграмма последовательностей авторизации.

На рисунке 5 показана диаграмма последовательности, на которой изображено упорядоченное во времени взаимодействие объектов при авторизации пользователя.

Для того, чтобы авторизоваться пользователю, ему необходимо обратиться к графическому интерфейсу сайта, на котором отобразится окно

авторизации. После чего пользователь должен ввести свои данные и подтвердить их отправку нажатием на кнопку вход. Данный блок передает данные в блок бизнес логики, в котором осуществляется проверка корректности введенных данных. Если данные были введены корректно, то блок бизнес логики отправляет запрос в блок базы данных для того, чтобы сверить совпадения. В случае успеха пользователь попадает на главную страницу. В случае некорректных данных ему снова отображают страницу входа с выведенной ошибкой о некорректности введенных данных

4.4 Варианты состояния системы

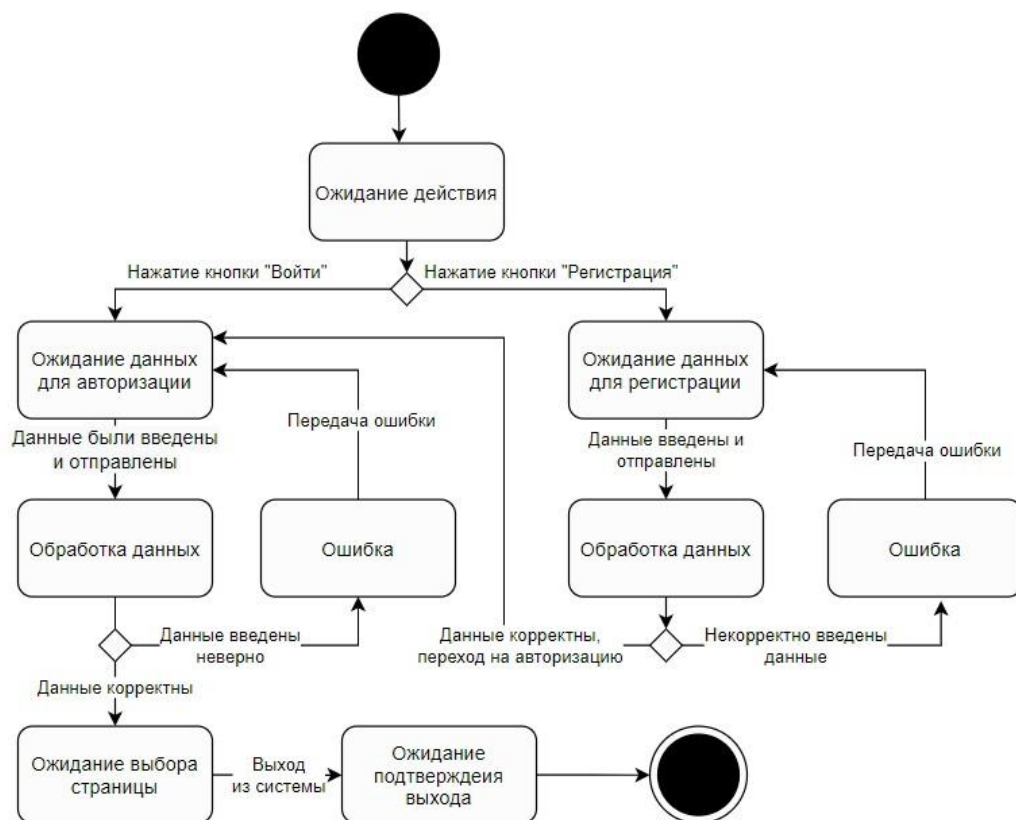


Рисунок 6— Диаграмма состояний для входа и регистрации.

На рисунке 6 приведена диаграмма состояний, которая отображает различные состояния при входе и регистрации. При переходе на сайт с целью авторизации и регистрации система первоначально находится в состоянии ожидания. После чего пользователь выбирает либо авторизацию, либо регистрацию. Данные действия ведут за собой следующие состояния: ожидание данных для авторизации, обработка данных. В случае успешной обработки данных пользователю не выдает ошибки, и он переходит на следующие состояния, которые указаны на схеме.

Диаграмма состояний, изображенная на рисунке 7, отображает возможные состояния системы для преподавателя.

Сначала система находится в ожидании какого-либо действия от преподавателя. Если он выбирает переход на страницу оценок, то система сначала находится в состоянии выбора дисциплины, семестра и группы, затем переходит в состояние системы выбора студента, а после – в состояние выставления оценки. Если оценка была введена корректно, то система переходит снова в состояние выбора студентов, если совершена ошибка при вводе данных оценки, то система возвращается на состояние выставления оценки.

Если преподаватель хочет просмотреть основные информационные разделы сайта, то система переходит в состояние ожидания выбора информационной страницы. После просмотра страницы состояние переходит к ожиданию действия.

Если преподаватель заходит в личный кабинет, то система переходит к состоянию ожидания действия от преподавателя. В случае необходимости редактирования данных система переходит к состоянию «данные изменены и отправленные», а после к состоянию «попытка записи в БД». Если всё

успешно, то возврат к состоянию ожидания действия. Если нет – состояние ошибки и переход на состояние ожидания действия.

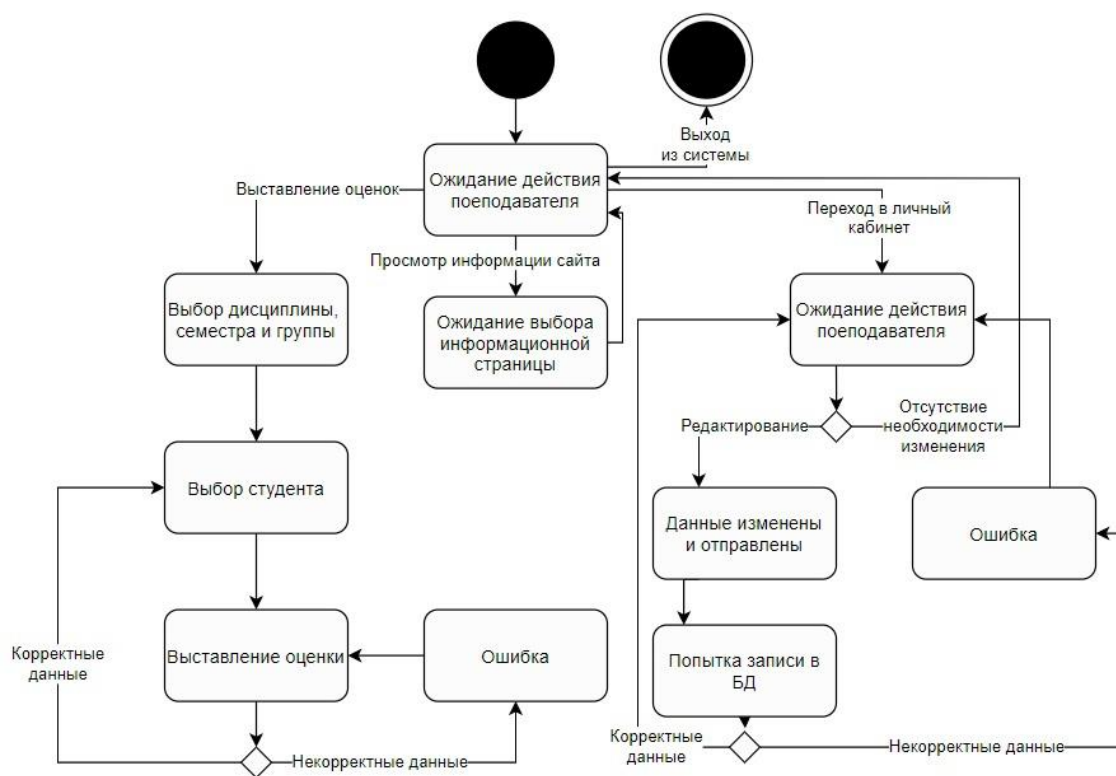


Рисунок 7 – Диаграмма состояний для преподавателя.

4.5 Варианты действия в системе

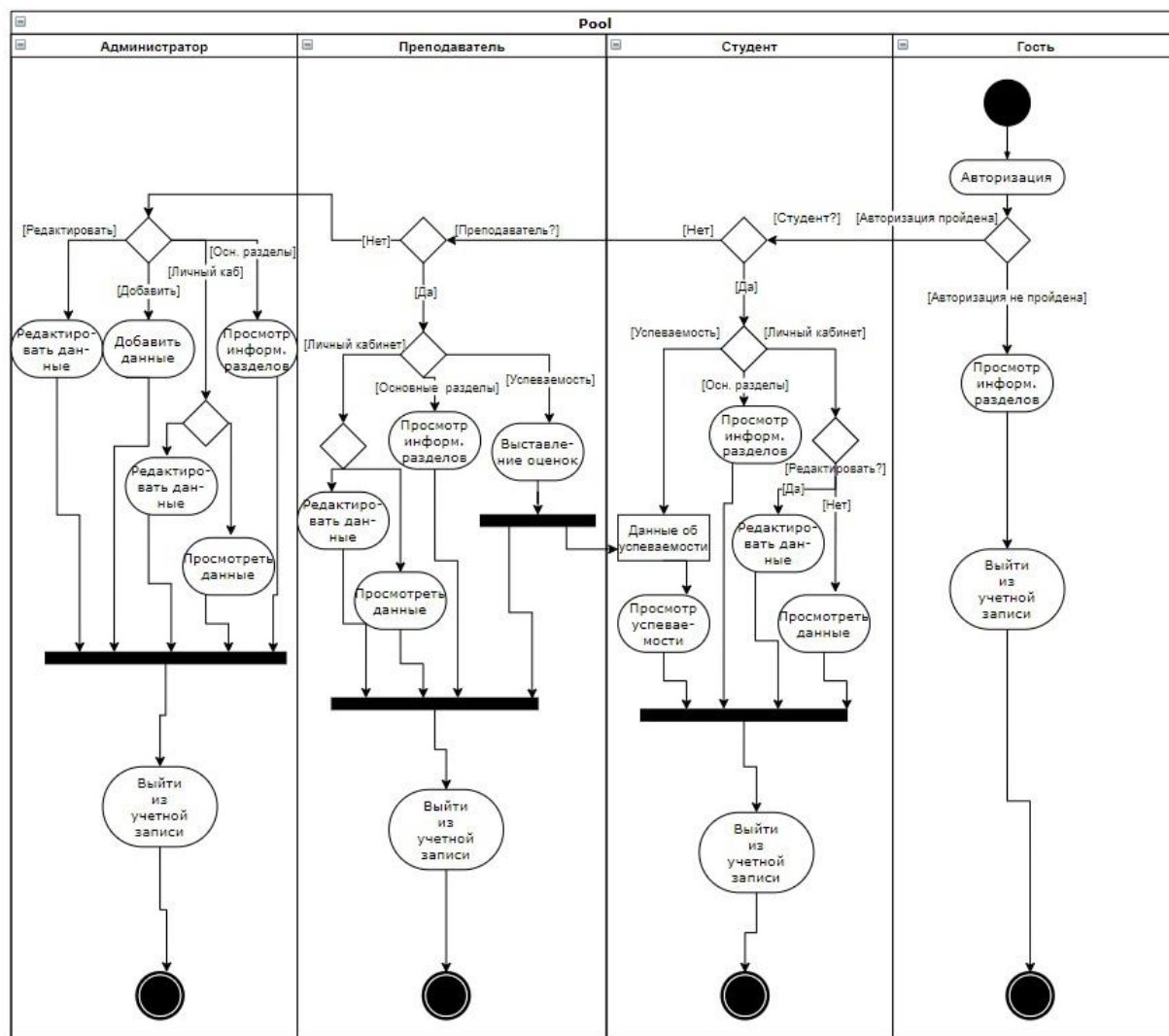


Рисунок 8 – Диаграмма активности.

Диаграмма активности, изображенная на Рисунке 8, отражает действия всех пользователей при использовании данного веб приложения с целью получения учета успеваемости.

4.6 Развертывания системы

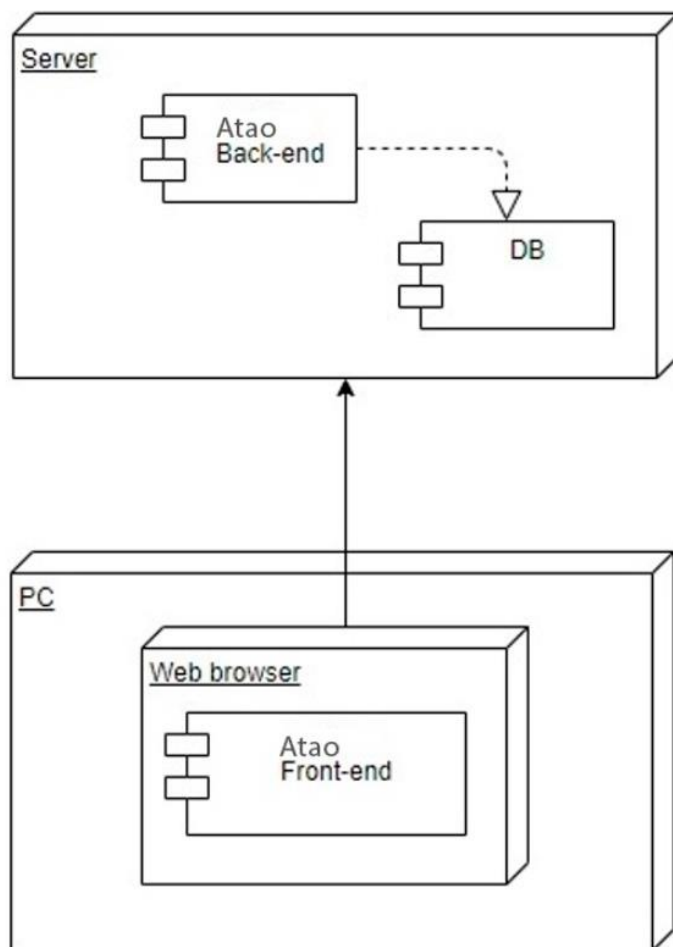


Рисунок 9 – Диаграмма развертывания.

На Рисунке 9 представлена диаграмма развертывания, чтобы определить какие аппаратные компоненты («узлы») существуют, какие программные компоненты работают на каждом узле и как различные части этого комплекса соединяются друг с другом. Для разрабатываемого web-приложения узлом устройства является персональный компьютер и сервер, а в качестве узла среды выполнения выступает web-браузер. В браузере развернут front-end приложения, а на серверной части back-end и база данных.

4.7 IDEF0

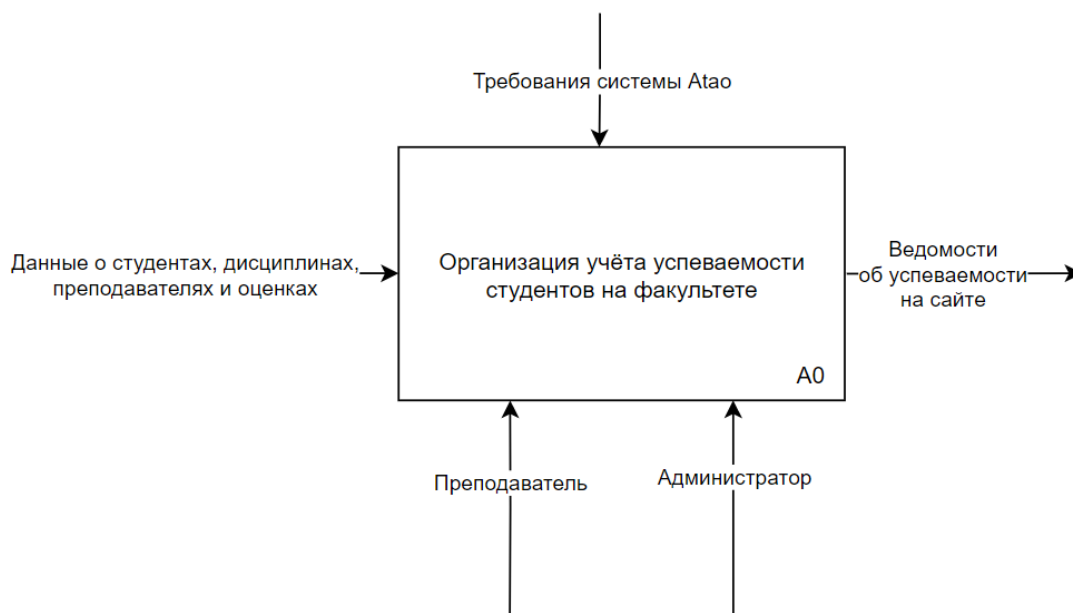


Рисунок 10 – Диаграмма IDEF0.

Диаграмма, изображенная на рисунке 10, является диаграммой нулевого уровня методологии IDEF0. Диаграмма приведена без декомпозиции. Входным потоком являются данные о студентах, дисциплинах, преподавателях и оценках. Механизмы – преподаватель и администратор. Управляющим потоком на данной диаграмме являются требования системы Атао. На выходе мы получаем готовые ведомости об успеваемости на сайте.

5 Анализ средств реализации

В качестве средств реализации приложения были выбраны следующие технологии:

— HTML, CSS, JS и сторонние библиотеки React JS.

ориентированные языки HTML и CSS необходимы для разработки front-end части. JS и сторонние библиотеки дают доступ к огромному количеству инструментов, заточенных под разные задачи и упрощающие процесс разработки.

— в качестве СУБД была выбрана MySQL. Она является хорошо масштабируемой, в равной степени легко может быть использована для работы, как с малыми, так и с большими объемами данных. А за счет упрощения некоторых используемых в ней стандартов система имеет высокую производительность,

— в качестве языка разработки в back-end использовался язык JS, а именно платформа Node JS.

6 Воронки конверсии

7 Реализация

8 Тестирование

Заключение

В результате проведенной работы было создано веб-приложение для учета успеваемости студентов. Были осуществлены следующие задачи:

- Разработка Front-end части сервиса;
- Разработка Back-end части сервиса;
- Создание связи между Front-end и Back-end частями приложения;
- Разработка базы данных.

Приложение отвечает всем требованиям, заявленным в техническом задании проекта.