

Министерство образования и науки Российской Федерации
ФГБОУ ВО РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ
(РГГМУ)

Институт Информационных систем и геотехнологий
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

БАКАЛАВРСКАЯ РАБОТА

На тему «Разработка приложения с индивидуальным расписанием обучения студентов»

Исполнитель Смотрин Григорий Александрович

(фамилия, имя, отчество)

Руководитель Доктор технических наук, профессор


(ученая степень, ученое звание)

Истомин Евгений Петрович

(фамилия, имя, отчество)

«К защите допускаю»

Заведующий кафедрой


(подпись)

К. Т. Н.

(ученая степень, ученое звание)

Александр Тарас Александрович

(фамилия, имя, отчество)

«03» июня 2025 г.

Санкт-Петербург

2025

Оглавление

Введение	3
Глава 1	5
1.1 Анализ предметной области.....	5
1.1.1 Исследование приложения.....	5
1.1.2 Сравнение с конкурентами.....	7
1.1.3 Необходимый функционал.....	9
1.2 Выбор методов решения задач.....	10
1.2.1 Выбор CASE-средств.....	10
1.2.2 SWOT-анализ.....	10
1.2.3 PEST-анализ.....	12
Глава 2	12
2.1 Проектирование информационной системы.....	12
2.2 Язык моделирования UML.....	13
2.3 Диаграмма прецедентов(Use Case Diagram).....	14
2.4 Сценарии на основе Use Case диаграммы.....	15
2.5 Диаграммы последовательности.....	20
2.6 Диаграмма компонентов.....	22
2.7 Диаграмма развертывания.....	24
2.8 Схема база данных.....	24
Глава 3	27
3.1 Реализация модуля “Выбор расписания” и “Просмотр расписания”.....	26
3.2 Макеты информационной системы.....	29
3.3 Оценка экономической эффективности.....	34
Заключение	41
Приложение А	42
Приложение В	64
Приложение С	73
Источники	73

Введение

В современном мире образовательные учреждения, сталкиваются с необходимостью повышения эффективности и в тоже время оптимизации обучения. Из-за постоянно растущей нагрузки на студентов и преподавателей, становится важной задача персонализация или корректировка обучения под себя, в том числе при помощи технологий. Одним из ключевых аспектов данной задачи является создания гибкой системы, позволяющей студенту подставить расписание под себя, учитывая его увлечения, нагрузку в тот или день и так далее.

Одной из главных проблем с которой сталкиваются студент в процессе обучения является неэффективное распределение учебной нагрузки. В основном расписание занятий формируется одно для всех, без учета личных предпочтений студента, их занятости, а также специфики их направления подготовки. Это приводит к тому, что студенты могут переутомиться, не иметь достаточно времени для подготовки, а также не имеют времени на личную жизнь. В результате всего этого, у студента снижается мотивация дальнейшего обучения, он хуже усваивает материал и увеличивает тем самым риск выгорания и плохой успеваемости.

Кроме этого, в последние годы наблюдается тенденция на персонализацию образования. Студенты все чаще стремятся выбирать, что им изучать, совмещать стажировки или просто работу с обучением. В таких условиях, становятся актуальными инструменты, позволяющие гибко настраивать расписания под себя, а также адаптировать его в зависимости от тех или иных обстоятельств.

Одной из главных причин для разработки данной системы, является еще то, что из-за пандемии COVID-19 сильно развилась сфера дистанционного и смешанного обучения. Введение различных онлайн-курсов, платформ для онлайн-обучения с видеоконференциями, открыли широкие возможности для трансформации образовательной среды.[20] Однако для того чтобы все это эффективно использовать, необходимы сервисы, которые смогут учитывать расписание занятий, занятость и нагрузку преподавателей и учитывать наличие свободных аудиторий.

С учетом всего вышеперечисленного, одной из актуальных задач, является создание платформы, позволяющей студентам самим формировать свое расписание с учетом, их личных предпочтений, приоритетов, занятости и нагрузки. Также приложение должно повысить эффективность обучающихся, за счет того, что они смогут распределить свою учебную нагрузку и больше времени тратить на профильные предметы. При этом важно, чтобы система была понятной и удобной в использовании, надежной, а так же смогла легко интегрироваться с внутренними системами вуза.

Актуальность темы определяется, необходимостью перехода от общих расписаний для всех студентов, к индивидуальным, что соответствует современным тенденциям в цифровизации и индивидуализации процесса обучения. Проблема проектирования и внедрения подобных решений требует глубокого анализа существующих подходов к планированию учебной деятельности, оценки возможностей современных технологий, а также создания практико-ориентированных решений с высоким уровнем юзабилити.

Целью данной дипломной работы является создание приложения, позволяющего формировать и поддерживать индивидуальное расписание студента. Данное приложение должно стать полезным для управления учебным процессом, предоставляя пользователям

оперативный доступ к информации о занятиях, их содержании, а также возможность интеграции с другими подсистемами учебного заведения.

Объектом исследования в данной работе являются процессы планирования и управления учебным временем студентов в контексте высшего образования. Предметом исследования является система формирования и управления индивидуальными расписаниями студентов на основе информационных технологий.

Решаемые задачи данной работы это:

1. Анализ предметной области и выбор решения методов задач.
2. Проектирование системы с составлением различных диаграмм и проектирование базы данных.
3. Реализация функциональных блоков и макетов интерфейса системы.

Глава 1

1.1 Анализ предметной области

Анализ предметной является одним из главных этапов в разработке информационной систем, потому что он предоставляет общее понимание целей, аспектов и проблем, связанных с предметной областью. В данной главе будет представлен анализ запросов пользователей, основных процессов и выявление проблем, связанных с разработкой данной информационной системы.

1.1.1 Исследование приложения

Данное приложение будет предназначено в основном для студентов, которые хотят подстроить расписание под себя в зависимости от их

предпочтений, возможностей и т.д. На основе своего индивидуального расписания студенты смогут более эффективно обучаться и лучше распределять силы между учебой, работой и личной жизнью.

Основные цели данного приложения можно сформулировать следующим образом:

1. Оптимизация учебного процесса: Приложение должно быть направлено на улучшение процесса обучения студента через управление учебным расписанием подстраивая его под индивидуальные потребности обучающегося.
2. Персонализация обучения: Система на основе, текущих курсов на которые записан студент, направления обучения, преподавателя и пройденных предметов будет предлагать новые программы для изучения
3. Улучшение образовательного процесса: Из-за того, что в данном приложении будет возможность просматривать информацию о количестве студентов на курсе и посещении их занятий, преподаватели смогут скорректировать учебный план. А студенты смогут эффективнее изучать дисциплины, потому что будут комфортно распределять нагрузку и фокусировать внимание на профильных предметах.
4. Обеспечение безопасности данных: Важно обеспечить безопасность данных пользователей системы, используя современные методы защиты, соответствующие современным стандартам.

Основные назначения приложения являются:

1. Автоматизация планирования и составления расписания: Это позволяет студентам создавать расписание с учетом их личных предпочтений, занятости и так далее. Это позволяет минимизировать перегрузки в время обучения и способствует более эффективному усвоению учебной программы.
2. Учет и анализ учебной нагрузки преподавателей и студентов: Данная система помогает отслеживать распределение нагрузки среди преподавателей и студентов,

что поможет обеспечить справедливое распределение учебных часов.

3. Персонализированное обучение: Система позволяет подстраивать студентам расписание под себя, что позволяет выделить больше времени на профильные предметы.
4. Безопасность и конфиденциальность данных: Система обеспечивает высокий уровень защиты данных пользователей и их конфиденциальности, в соответствии с современными стандартами безопасности.

1.1.2 Сравнение с конкурентами

В данном пункте проведем сравнение разрабатываемой системы с конкурентами. Ниже представлены таблицы с описанием возможностей и функционала а также с сравнением с конкурентами (см Табл. 1 и Табл.2)

№	Название	Возможности
1	Moddle	Платформа для создания онлайн-курсов с гибкими настройками расписания, поддерживает интеграцию с различными системами и плагинами.
2	ILIAS	Система управления обучением с возможностями настройки расписания, учета успеваемости и интеграции с внешними ресурсам
3	Эквио	Мобильная платформа

		для обучения и управления персоналом, включает функции создания расписаний и мониторинга учебного процесса.
4	jSolutions	Система управления учебным заведением с функционалом автоматизации планирования занятий, учета посещаемости и успеваемости студентов.
5	FET	Программное обеспечение для автоматического создания расписаний с учетом множества параметров, включая преподавателей, аудитории и ограничения.

Таблица 1-Описание возможностей конкурентов

Функционал	FET	ILIAS	jSolutions	Эквио	Moodle	Мое приложение
Аналитика и отчетность	-	+	-	+	+	+
Управление расписанием	-	-	-	-	-	+

студентом						
Интеграция с внешними ресурсами	-	+	-	+	+	+
Учет посещаемости	-	+	-	-	+	+
Создание курсов преподавателем	-	+	-	+	+	+

Таблица 2-сравнение функционала

Из данных таблиц видно, что в большинстве программ конкурентов нету учета посещаемости на занятия. К тому же на двух платформах нету возможности создания курсов преподавателями. Также в половине программ конкурентов нету какой-либо аналитики и отчетности, а также не реализована интеграция с внешними предложением. И главное, ни в одной из этих систем нету возможности управлять расписанием. Из этого мы можем сделать вывод, что разрабатываемая система решает большинство проблем в сфере систем расписания, что позволит ей быть более привлекательно для будущих пользователей на фоне конкурентов.

1.1.3 Необходимый функционал

В данном разделе будут представлены возможности разрабатываемой, системы.

1. Регистрация пользователя
2. Авторизация пользователя
3. Просмотр предложенных новых курсов
4. Просмотр описания курса
5. Возможность изменять и сохранять расписание занятий
6. Просмотр посещаемости преподавателем

1.2 Выбор методов решения задач

При разработки информационной системы важно выбрать методы решения задач, которые обеспечивают успешную реализацию проекта

1.2.1 Подбор CASE-средств

CASE-средства-, это набор различных программ, методов и инструментов, которые позволяют автоматизировать процессы проектирования и разработки системы, что повышает ее качество, и делает легче дальнейший процесс обслуживания. [1] Это делает процесс их выбора важной и обязательной частью разработки информационной системы.

Среди CASE-средств для нашего проекта можно выделить:

1. Средства проектирования баз данных (MySQL)
2. Средства анализа (BPwin)
3. Средства планирования и управления проектом (1C)
4. Средства анализа и проектирования (ERWin)

1.2.2 SWOT-анализ

SWOT-анализ-это процесс изучения текущей ситуации, выявление как сильных, так и слабых сторон. Также выявление возможностей, разрабатываемого продукта и угрозы, которые могут помешать его работе.[2]

S(strengths) - указывает на сильные стороны разрабатываемой ИС.[2]

W(weakness) - указывает на слабые стороны разрабатываемой ИС.[2]

O(opportunities) - указывает на возможности, которые предоставляет ИС.[2]

T(threats) - указывает на угрозы, которые могут помешать работе ИС. [2]

Ниже представлен SWOT-анализ для нашего приложения(см Табл.3)

	SWOT-анализ
S	1.Простота использования 2.Возможность корректировки расписания под себя 3.Выбор предметов на основе предпочтений и важности для каждого, отдельно взятого студента
W	1.Сбои в работе 2.Наличие не всех предметов внутри приложения
O	1.Бесплатное использование 2.Малое количество прямых аналогов 3.Поддержка IT-проектов с стороны государства
T	1.Реформы в сфере образования 2.Наличие уже проверенных временем систем 3.Нежелание учеников и преподавателей пользоваться неизвестным приложением

Таблица 4 -SWOT-анализ

1.2.3 PEST-анализ

PEST-анализ позволяет выявить основные аспекты внешней среды, влияющие на бизнес компании. Они подразделяются на политические(P), экономические(E), социальные(S), технологические(T). [3]

Ниже представлен PEST-анализ для нашей системы(см Табл 5)

	PEST-анализ
P	1.Реформы в системе образования 2.Поддержка IT-проектов с стороны государства
E	1.Рост инвестиций в развитие образовательных технологий
S	1.Стремление студентов к индивидуализации обучения 2. Высокий уровень цифровой грамотности у студентов
T	1.Быстрое развитие технологий 2.Возможность интеграции/совместимости с существующими образовательными платформами

Таблица 5 -PEST-анализ

Глава 2

2.1 Проектирование информационной системы

Проектирование является одним из ключевых этапов разработки информационной системы. Оно позволяет выделить основной функционал системы на основе задач и потребностей пользователя, сформировать архитектуру.

Есть несколько основных этапов проектирования информационной системы. Первый, это анализ требований к системе со стороны пользователя. Нужно выявить основные предпочтения пользователя, а также функции, которые будут в системе.[4] Этот этап помогает выявить

функциональные и нефункциональные требования.

Следующим этапом является разработка архитектуры системы. На этом этапе определяют структуру системы, ее основные компоненты, а также их взаимодействие друг с другом.[5] Выбор архитектуры важен, так как это помогает обеспечить высокую надежность, хорошую производительность и сильную масштабируемость.

Также одним из важных этапов является проектирование базы данных. Для ее создания необходимо определить структуру, формат данных, их организацию внутри базы данных, а также способы их защиты.[6]

Последним этапом проектирования является разработка интерфейса пользователя системы. Важно, чтобы интерфейс был удобен, понятен и прост в использовании для обеспечения доступа к всем возможностям системы.[7]

Целью данной главы является рассмотрение первых трех этапов проектирования для обеспечения эффективной работы системы, удовлетворения пользовательских требований и понимания структуры разрабатываемого продукта.

2.2 Язык моделирования UML

Язык моделирования UML-это язык служащий для описания и разработки программного обеспечения или систем. Он представляет из себя набор различных символом, фигур и так далее. В основном его используют для визуализации процессов и явлений в системе при помощи различных диаграмм и схем.[26]

Основными причинами выбора данного языка моделирования являются:

1. Универсальность и стандартизация: UML является одним из самых стандартных и распространенных языков моделирования. Из-за этого он широко используется среди большинства разработчиков и аналитиков, что делает его простым для понимания. [25]
2. Разнообразные средства моделирования: UML предлагает различные виды диаграммы, что позволяет визуализировать сложные части системы, что делает их понятнее для всех участников разработки.[25]
3. Доступность: при использовании UML мы можем создавать четкие и структурированные модели, которые легко понятны для всех участников разработки. Это важно, потому что при работе в команде каждый, сотрудник сможет легко понимать, что от него

требуется. Также это облегчает процесс общения с заказчиком, который, посмотрев на различные схемы и диаграммы, созданные на языке UML, может понять, чего он хочет от системы.[25]

4. Жизненный цикл: кроме того, что UML позволяет моделировать систему и ее архитектуру, он позволяет проводить анализ требований, тестирование и так далее. Все это позволяет поддерживать жизненный цикл системы. [27]

2.3 Диаграмма прецедентов(Use Case Diagram)

Диаграмма прецедентов (Use Case Diagram) описывает пользователей системы, а также то как они ее используют. В рамках данной диаграммы пользователей называют актерами.[22]

Актером называется любая сущность, взаимодействующая с системой извне.[18] Это может быть человек, программа, техническое устройство или все что определяет разработчик для взаимодействия с системой.

Прецедент служит для описания функций или сервисов, которые система предлагает актеру. Другими словами, каждый прецедент определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится как именно будет совершено взаимодействие между системой и актером[19]

В общем случае данная диаграмма является графом, который служит представлением прецедентов, актеров, некоторых интерфейсов а также отношением между этими элементами.

Ниже представлена диаграмма прецедентов для ИС (см.Рис 1)

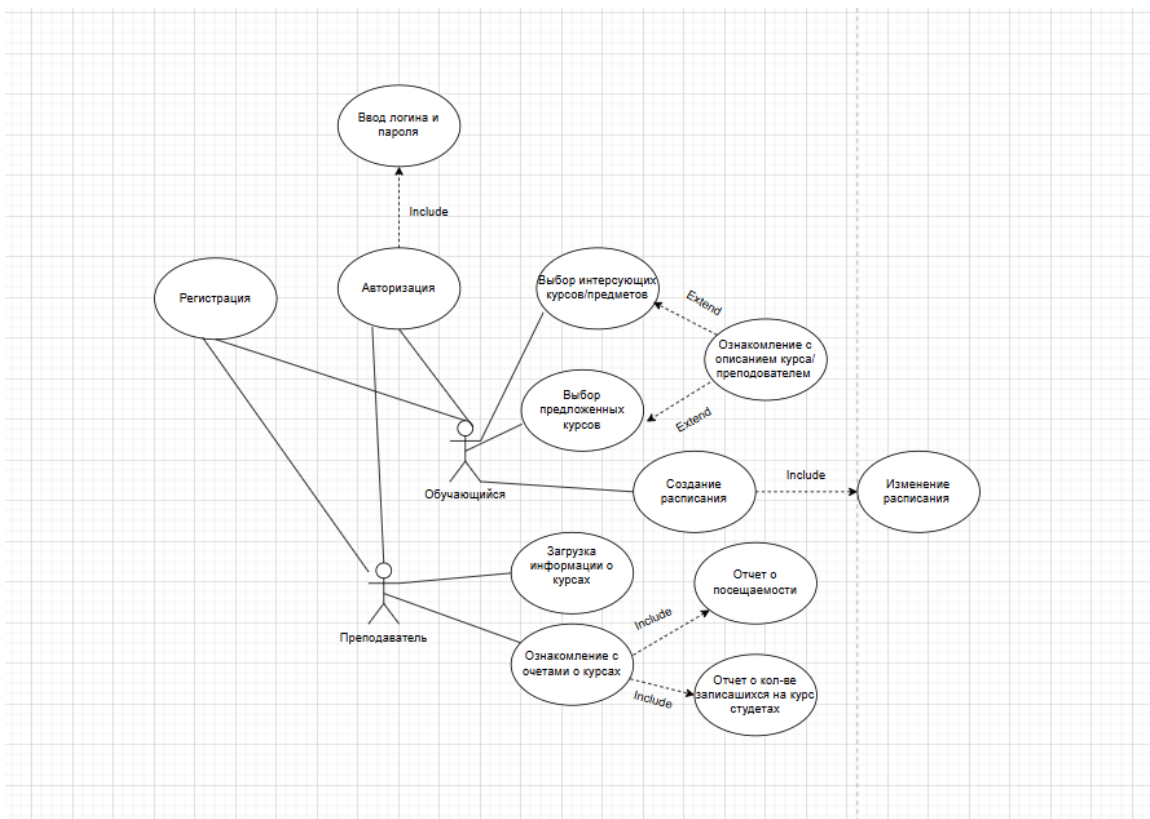


Рис 1 – Диаграмма прецедентов

2.4 Сценарии на основе Use Case диаграммы

Сценарии являются важной частью в процессе разработки информационной системы. Это описание того, как пользователь будет взаимодействовать с системой для достижения определенной цели. [17]

В данном пункте будут рассмотрены два основных сценария: “Авторизация” и “Регистрация” в виде таблиц, потому что я считаю, это будет более наглядно и удобно для восприятия (см. Табл). Остальные будут представлены в Приложении А

Главный раздел сценария выполнения варианта использования “Авторизация”
--

Вариант использования	Авторизация
Актеры	Пользователь
Краткое описание	Пользователь авторизуется в аккаунт
Цель	Пользователь успешно получает доступ к приложению
Тип	Базовый
Ссылки на другие варианты использования	Включает: Ввод логина и пароля

Табл 6 – Главный раздел сценария “Авторизация”

Типичный ход событий сценария выполнения варианта использования «Авторизация»
--

Действия актеров	Отклик системы
1.Пользователь заходит в приложение	
2. Пользователь нажимает на кнопку “Авторизация”	
	3.Переход к окну “Авторизации” с выводом строк “Логин” и “Пароль”
4.Пользователь вводит свои данные в строчках “Логин” и “Пароль”	

5.Пользователь нажимает кнопку “Войти”	
	6.Система отправляет запрос к базе данных для проверки наличия данных
	7.База данных проверяет наличие введенных данных
	Исключение 1: данные отсутствуют в базе данных
	8. База данных подтверждает наличие данных у себя
	9. Система предоставляет пользователю доступ к его аккаунту
10. Пользователь успешно входит в аккаунт	

Табл 7 - Типичный ход событий сценария выполнения варианта использования «Авторизация»

Исключение сценария выполнения варианта использования “Авторизация”

Исключение 1: данные отсутствуют в базе данных

Действие актеров	Отклик системы
	7.1 Система получает ответ о БД, что введенных данных не существует
	7.2 Система выдает ошибку и просит пользователя проверить введенные данные

8.Пользователь вводит данные повторно	
9.Пользователь нажимает кнопку “Войти”	

Табл 8 - Исключение 1: данные отсутствует в базе данных

Главный раздел сценария выполнения варианта использования “Регистрация”

Вариант использования	Регистрация
Актеры	Пользователь
Краткое описание	Пользователь регистрирует новый аккаунт
Цель	Пользователь успешно регистрирует новый аккаунт
Тип	Базовый
Ссылки на другие варианты использования	Не включает в себя другие варианты использования

Табл 9 -Главный раздел сценария выполнения варианта использования “Регистрация”

Типичный ход событий сценария выполнения варианта использования «Регистрация»

Действия актеров	Отклик системы
1.Пользователь заходит в приложение	

2. Пользователь нажимает кнопку “Регистрация”	
	3.Переход к окну “Регистрация” с полями “Логин”, “Пароль” и “Роль” и условиями для них. Для поля “Логин”: “Введите уникальные логин”. Для “Пароль”: “Пароль должен быть длиной 8-12 символов”. Для поля “Роль” выбор из двух ролей “Преподаватель” или “Студент/ученик”
4.Пользователь вводит логин и пароль	
	5.Система отправляет данные в БД для их проверки
	Исключение №1 “Не выполнены условия ввода”
	6.Система вносит данные в БД
	7.Система выводит сообщение, “Регистрация успешна завершена”
8.Пользователь нажимает кнопку “Окей”	
	9.Система переносит пользователя на главную страницу

Табл 10 - Типичный ход событий сценария выполнения варианта использования «Регистрация»

Исключение сценария выполнения варианта использования “Регистрация”

Исключение 1: Не выполнены условия ввода данных

Действие актеров	Отклик системы
	5.1 Система получает ответ о БД, что введенные данные не соответствуют условиям
	5.2 Система выдает ошибку и просит пользователя изменить логин или пароль
7.Пользователь вводит новые данные	
8.Пользователь нажимает кнопку “Зарегистрироваться”	

Табл 11 - Исключение сценария выполнения варианта использования “Регистрация”

2.5 Диаграммы последовательности

Диаграмма последовательностей (Sequence Diagram) в UML является типом диаграммы, который показывает как объекты взаимодействуют и обмениваются сообщениями между собой в течение времени. [9]

Данная диаграмма состоит из:

1.Объекты — это различные части системы, которые взаимодействуют друг с другом.[11]

2.Линии жизни - вертикальные линии, показывающие, течение времени, а также время исполнения функций объект или его активность. [11]

3.Сообщения - горизонтальные стрелки, показывающие обмен информацией между объектами.[11]

Ниже представлены диаграммы последовательности для двух блоков (Рис 2 и Рис 3). Остальные диаграммы будут представлены в Приложении В.

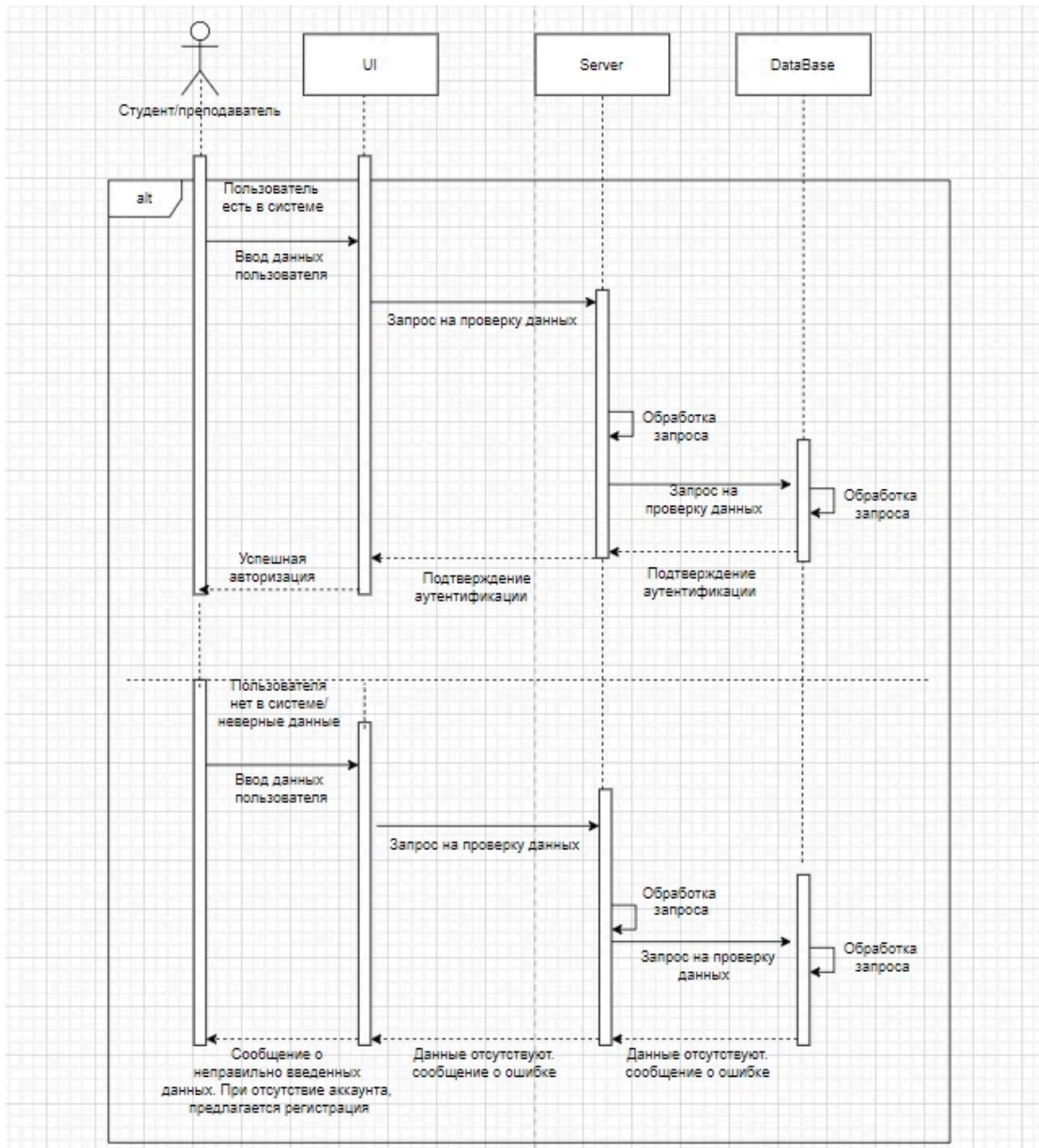


Рис 2 - Диаграмма блок "Авторизация"

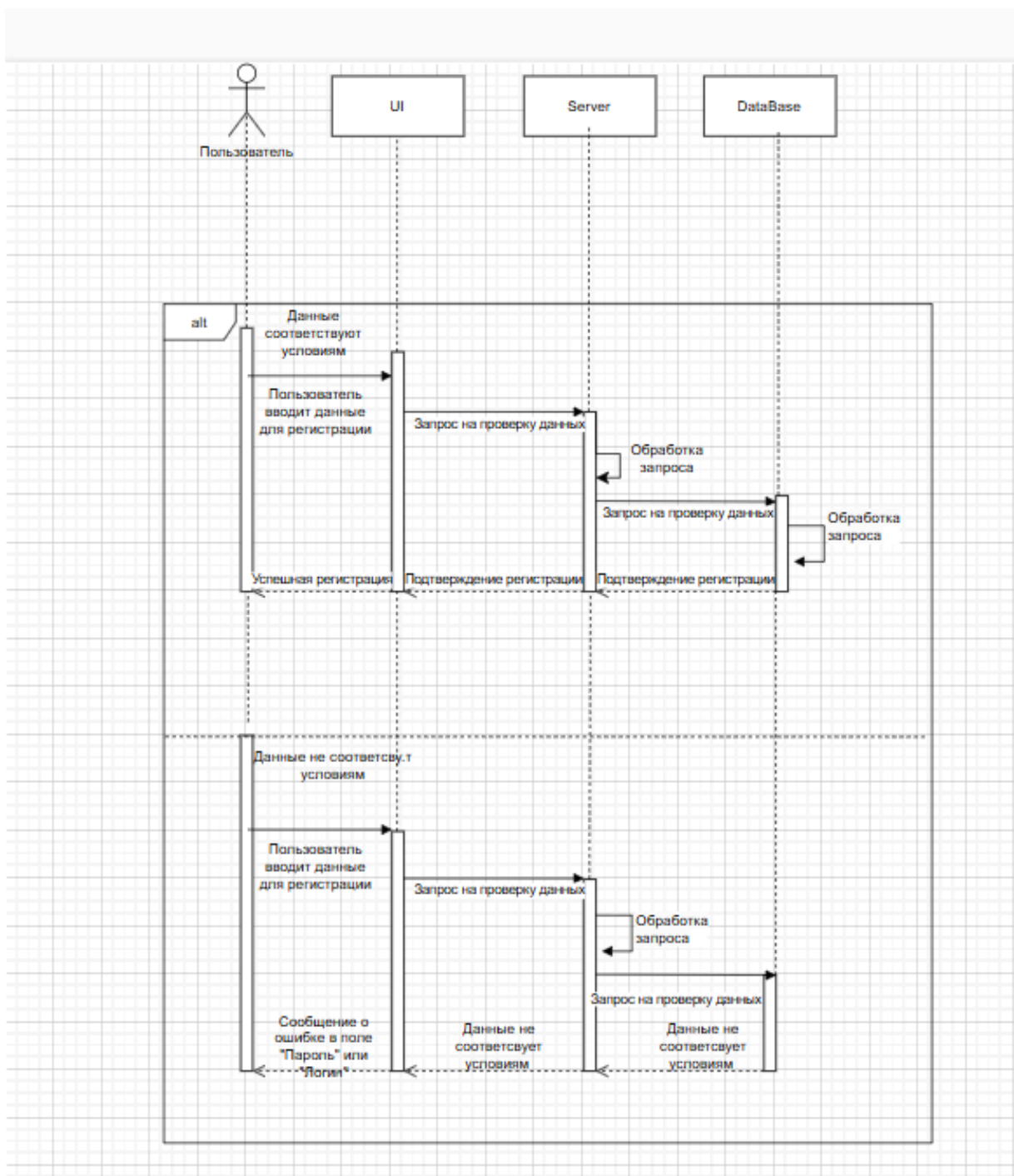


Рис 3 - Диаграмма блок “Регистрация”

2.6 Диаграмма компонентов

Диаграмма компонентов, это еще один тип UML диаграмм. Она показывает, то, как будет выглядеть система физически.[12] При помощи этой диаграммы можно определить архитектуру будущей системы, установив то, как разные компоненты взаимосвязаны между собой.[12]

Диаграмма компонентов позволяет представить общую картину архитектуры систем. Это помогает лучше понять ее структуру и взаимосвязи разработчикам. А также облегчает процесс документирования решений, связанных с архитектурой системы и коммуникации в процессе ее проектирования.[12]

Диаграмма компонентов разрабатывается для следующих целей:

1. Визуализация общей структуры исходного кода программной системы.[12]
2. Спецификация исполняемого варианта программной системы:[12]
3. Обеспечение многократного использования отдельных фрагментов программного кода:[12]
4. Представление концептуальной и физической схем баз данных.[12]

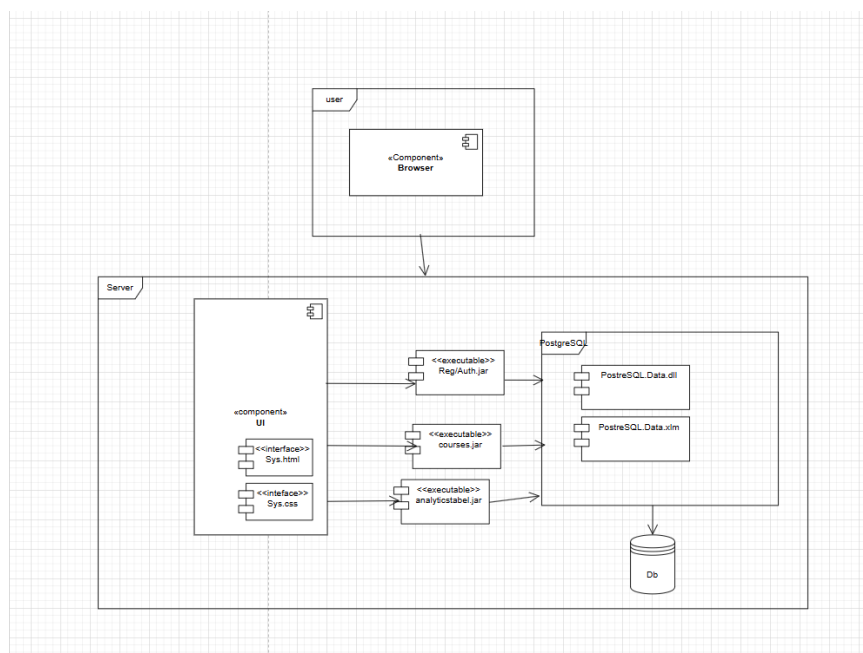


Рис 5 - Диаграмма компонентов

2.7 Диаграмма развертывания

Диаграмма развертывания – это тип UML-диаграммы, которая показывает архитектуру исполнения системы, включая такие узлы, как аппаратные или программные среды исполнения, а также промежуточное программное обеспечение, соединяющее их.[16]

Диаграммы развертывания обычно используются для визуализации физического аппаратного и программного обеспечения системы.

Используя его, вы можете понять, как система будет физически развернута на аппаратном обеспечении.[16]

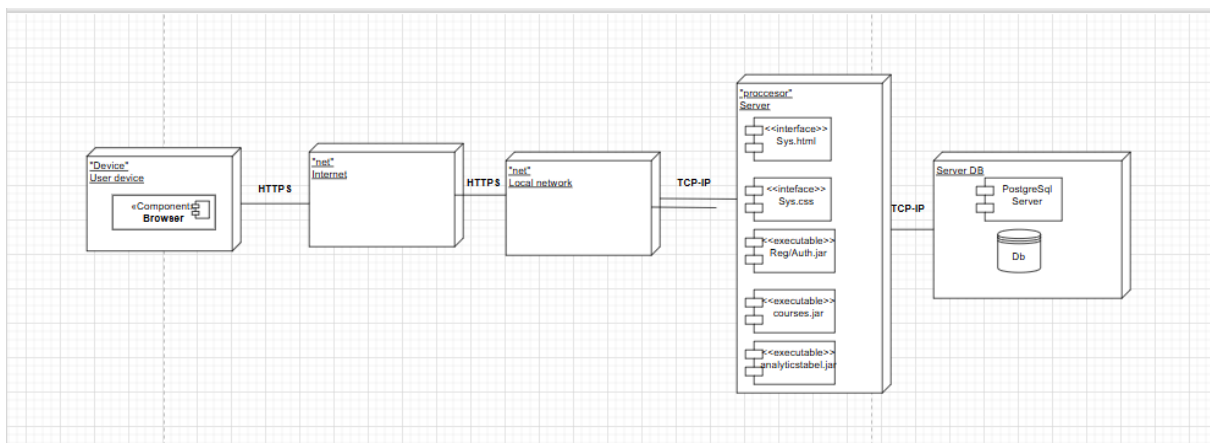


Рис 6 - Диаграмма развертывания

2.8 Схема база данных

База данных (БД) - совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.[13]

Система управления базами данных (СУБД) – это комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными.[14] Другими, словами, СУБД — это набор программ, позволяющий организовывать, контролировать и администрировать базы данных.

В современном мире ни одна из программ не может обойтись без БД. Благодаря им мы можем удобно и понятно составить связь между различными элементами ИС. Для проектирования ИС мы представим схему базы данных

Разработка данной схемы позволяет наглядно представить организацию данных, определить ключевые поля таблиц, типы связей, а также обеспечить целостность и непротиворечивость хранимой информации. Схема служит основой для проектирования базы данных, необходимой для реализации функционала системы, а также является важным инструментом при дальнейшем сопровождении и масштабировании приложения. Ниже представлена схема(Рис.6)

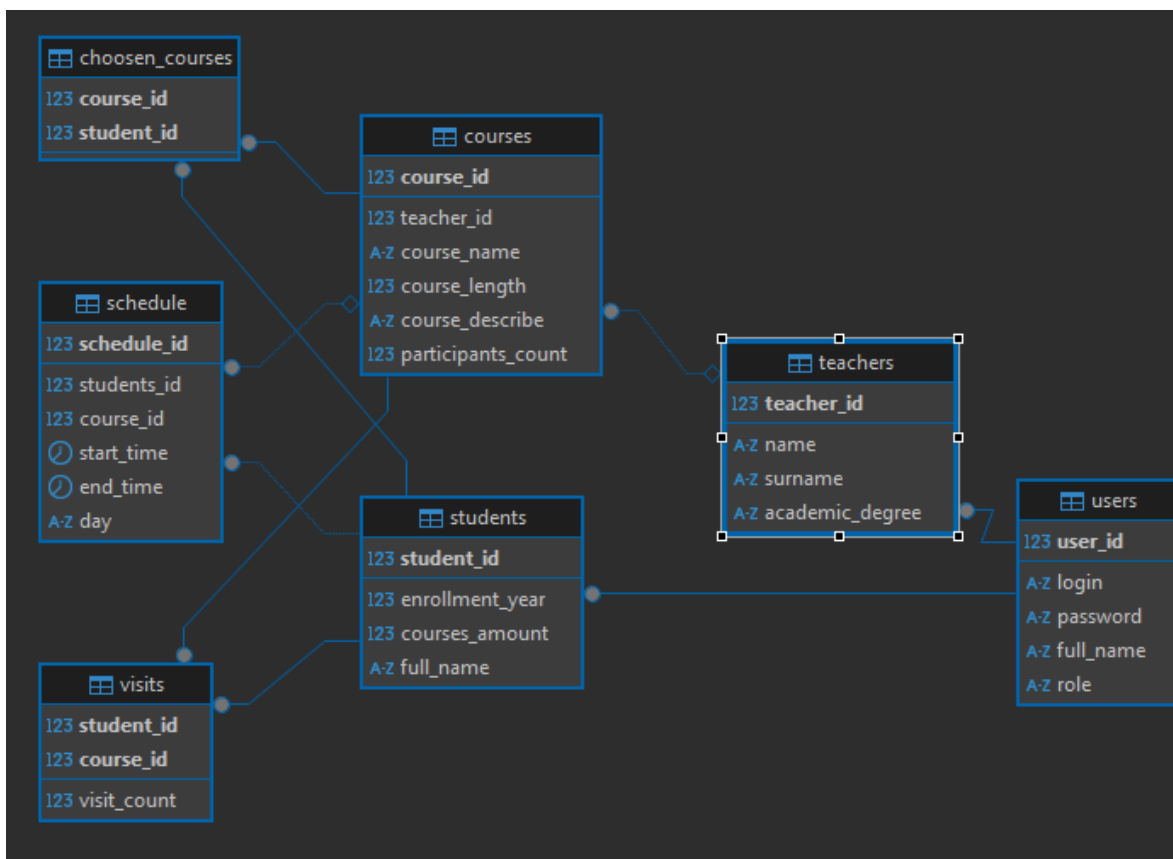


Рис.6 – Схема базы данных

Здесь представлены описание полей, базы данных.

login - логин пользователя

password-пароль пользователя

Course_length- длина курсы (с начальной и конечной датой, обычно ограниченная семестром)

Participants_count- кол-во участников на курсе

Visit_count- кол-во посещенных занятий студентом

Enrollment_year – год поступления

Role – роль(преподаватель/студент)

start_time - начало занятий

end_time - окончание занятия

day -день в который проходит занятие

courses_amount - кол-во курсов на которые, записан студент

user_id - уникальный идентификатор пользователя

student_id и teacher_id- соответственно уникальные идентификаторы студента и преподавателя

visit_count- кол-во посещенных занятий студентом

course_id - уникальный идентификатор курса

course_name - название курса

academic_degree - ученая степень преподавателя

schedule_id - уникальный идентификатор расписания каждого студента на определенный день

Глава 3

3.1 Реализация модуля “Изменений расписания” и “Просмотр расписания”

В данном пункте будет продемонстрирована реализация модулей “Изменение расписания” и “Просмотр расписания” на языке Python, с использованием библиотеки psycopg2, для подключения к базе данных. Первым делом в Dbeaver была создана база данных на основе схемы на Рис. 6. Далее будут представлены скриншоты кода и результат работы каждого из

модулей.(СмРис 7 - Рис11)

```
import psycopg2

conn = psycopg2.connect(
    host="localhost",
    port="5432",
    user="postgres",
    password="grush@2!33",
    dbname="schedule_sys"
)

# usage
def schedule_get(student_id):
    cur = conn.cursor()

    try:
        cur.execute( query: "SELECT 1 FROM system.students WHERE student_id = %s", vars: (student_id,))
        if not cur.fetchone():
            print(f"Студент с ID={student_id} не найден")
            return
        query = """
            SELECT s.full_name, c.course_name, sc.day, sc.start_time, sc.end_time
            FROM system.students s
            JOIN system.schedule sc ON s.student_id = sc.students_id
            JOIN system.courses c ON sc.course_id = c.course_id
            WHERE s.student_id = %s
            ORDER BY sc.day, sc.start_time;
            """
        cur.execute(query, vars: (student_id,))
        results = cur.fetchall()

        print(f"\nРасписание студента ID={student_id}:\n")
        for row in results:
            print(f"{row[0]} ({row[1]}): {row[2]} - {row[3]}-{row[4]}")

    finally:
        cur.close()
```

Рис 7 - вывод расписания студента

На Рис 7, представлена функция `schedule_get`, которая подключается к базе данных и при помощи в sql запроса выводит расписание студента, id которого мы укажем.

```
def schedule_update(student_id, course_id, day, new_start_time=None, new_end_time=None):
    valid_days = {'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'}

    if day not in valid_days:
        print(f"Ошибка: день '{day}' некорректен. Допустимые дни: {' '.join(valid_days)}.")
        return

    def time_to_minutes(t):
        hour, min = map(int, t.split(":"))
        return hour * 60 + min

    if new_start_time is not None:
        if time_to_minutes(new_start_time) < 540:
            print("Ошибка: Время начала должно быть не ранее 09:00.")
            return

    if new_end_time is not None:
        if time_to_minutes(new_end_time) > 1260:
            print("Ошибка: Время окончания должно быть не позже 21:00.")
            return

    if new_start_time and new_end_time:
        if time_to_minutes(new_start_time) >= time_to_minutes(new_end_time):
            print("Ошибка: Время начала должно быть раньше времени окончания.")
            return

    with conn.cursor() as cur:
        cur.execute("""
            SELECT start_time, end_time
            FROM system.schedule
            WHERE students_id = %s
              AND day = %s
              AND course_id != %s
            """, (student_id, day, course_id))

        existing_times = cur.fetchall()

        new_start = time_to_minutes(new_start_time) if new_start_time else None
        new_end = time_to_minutes(new_end_time) if new_end_time else None

        for start, end in existing_times:
            existing_start = time_to_minutes(start)
            existing_end = time_to_minutes(end)

            if new_start is not None and new_end is not None:
                if not (new_end <= existing_start or new_start >= existing_end):
                    print(f"Ошибка: Время {new_start_time}-{new_end_time} пересекается с другим занятием {start}-{end}.")
                    return

        updates = []
        params = []
```

```

if new_start_time is not None:
    updates.append("start_time = %s")
    params.append(new_start_time)
if new_end_time is not None:
    updates.append("end_time = %s")
    params.append(new_end_time)

if not updates:
    print("Нет данных для обновления.")
    return

update_query = f"""
UPDATE system.schedule
SET {', '.join(updates)}
WHERE students_id = %s
    AND course_id = %s
    AND day = %s
"""

params.extend([student_id, course_id, day])

with conn.cursor() as cur:
    cur.execute(update_query, tuple(params))
    conn.commit()

if cur.rowcount == 0:
    print("Ничего не обновлено – проверьте день и время.")
else:
    print(f"Обновлено расписание: студент {student_id}, день '{day}'.")

schedule_get(3)
schedule_update(student_id=1, day='Monday', new_start_time="11:00", new_end_time="11:30", course_id=2 )

```

Рис 8 - 9 -Изменение расписания

На Рис 8- 9, представлена функция `schedule_updates` , которая обновляет расписание студента. На вход она получает ID студента, день в который он хочет поменять расписание, ID курса для которого он хочет это изменить, а также новый временной промежуток. Ниже на Рис 10 - Рис 12 представлены результаты работы данных модулей. В приложении С будет представлена реакция программы на различные сценарии

```

Anna Semenova (Математика): Wednesday – 13:00:00-14:00:00
Anna Semenova (Физика): Wednesday – 15:00:00-16:00:00
Обновлено расписание: студент 1, день 'Monday'.

```

Рис 10 - Вывод работы функций

	123 students_id	123 course_id	start_time	end_time	A-Z day	
1	1	1	12:00:00	15:00:00	Monday	
2	1	2	16:00:00	18:00:00	Monday	
3	3	1	13:00:00	14:00:00	Wednesday	
4	3	2	15:00:00	16:00:00	Wednesday	

Рис 11 - Таблица Schedule до изменения

1	1	12:00:00	15:00:00	Monday
1	2	11:00:00	11:30:00	Monday

Рис 12 - Таблица Schedule после изменения

3.2 Макеты информационной системы

Проектирование макета информационной системы — это этап разработки, который критически важен для достижения успеха всего проекта. Основные причины для проектирования макета информационной системы включают:

1. **Определение функционала:** При проектировании макета, мы можем четко выделить все необходимые функции, которые должны быть реализованы в системе. В это обычно включают, то как пользователь взаимодействует с системой, поддержку клиентов и так далее.
2. **Уточнение требований заказчика:** Разрабатывая макет, мы можем уточнить чего хочет заказчик, а также убедиться, что понимание того, как должна выглядеть система соответствует ожиданиям. Таким образом в дальнейшем это может предотвратить ненужные конфликты и непонимание с заказчиком.
3. **Оптимизация пользовательского опыта:** Проектирование макета позволяет создать удобный и понятный интерфейс пользователя. Это помогает минимизировать ошибки, совершаемые пользователем и повышает их довольство при использовании системы.
4. **Планирование архитектуры системы:** Макет системы включает в себя планирование архитектуры и структуры системы. Это включает в себя выбор технологий, определение базы данных,

распределение функционала по модулям и другие архитектурные решения.

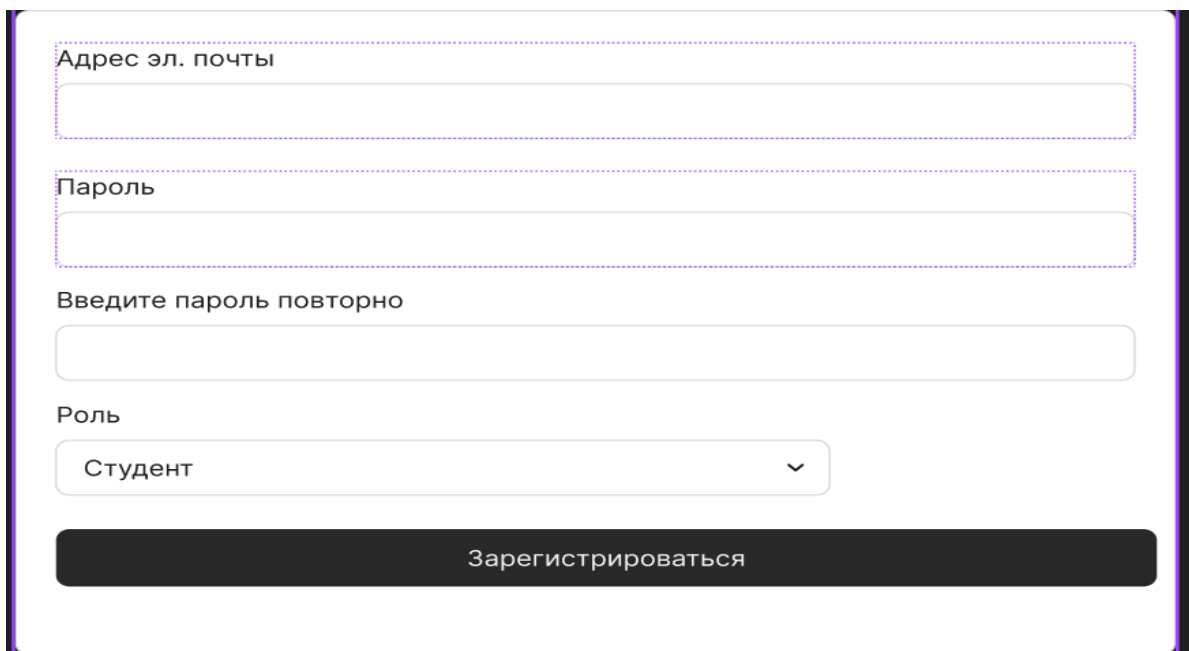
5. Визуализация: Макет позволяет визуализировать интерфейсы системы, что в дальнейшем позволит в обсуждении с заказчиками, пользователями и разработчиками. Это помогает вносить правки и уточнять предпочтение клиентов до начала активной разработки

6. Экономия времени и ресурсов: Проектирование макета помогает предотвратить возможные ошибки на ранних стадиях разработки, что экономит время и ресурсы команды разработчиков.

Все эти факторы делают проектирование макета информационной системы важным этапом, который способствует успешной и эффективной реализации проекта с минимальными рисками и максимальной пользой для пользователей.

В данном пункте будут представлены макеты главных интерфейсов, разрабатываемой системы.

Первым будет представлен макет окна регистрации (см Рис 13).



The image shows a registration form with the following elements:

- A text input field labeled "Адрес эл. почты" (Email address).
- A text input field labeled "Пароль" (Password).
- A text input field labeled "Введите пароль повторно" (Repeat password).
- A dropdown menu labeled "Роль" (Role) with "Студент" (Student) selected.
- A dark button labeled "Зарегистрироваться" (Register).

Рис 13 - Регистрация

На данном макете представлены поля для ввода электронной почты, пароля и его повторного ввода для проверки. Также представлен выбор роли: Студент или Преподаватель.

Далее представлен интерфейс студента, связанные с выбором курсов и настройкой расписания (см Рис 14 и Рис 15).

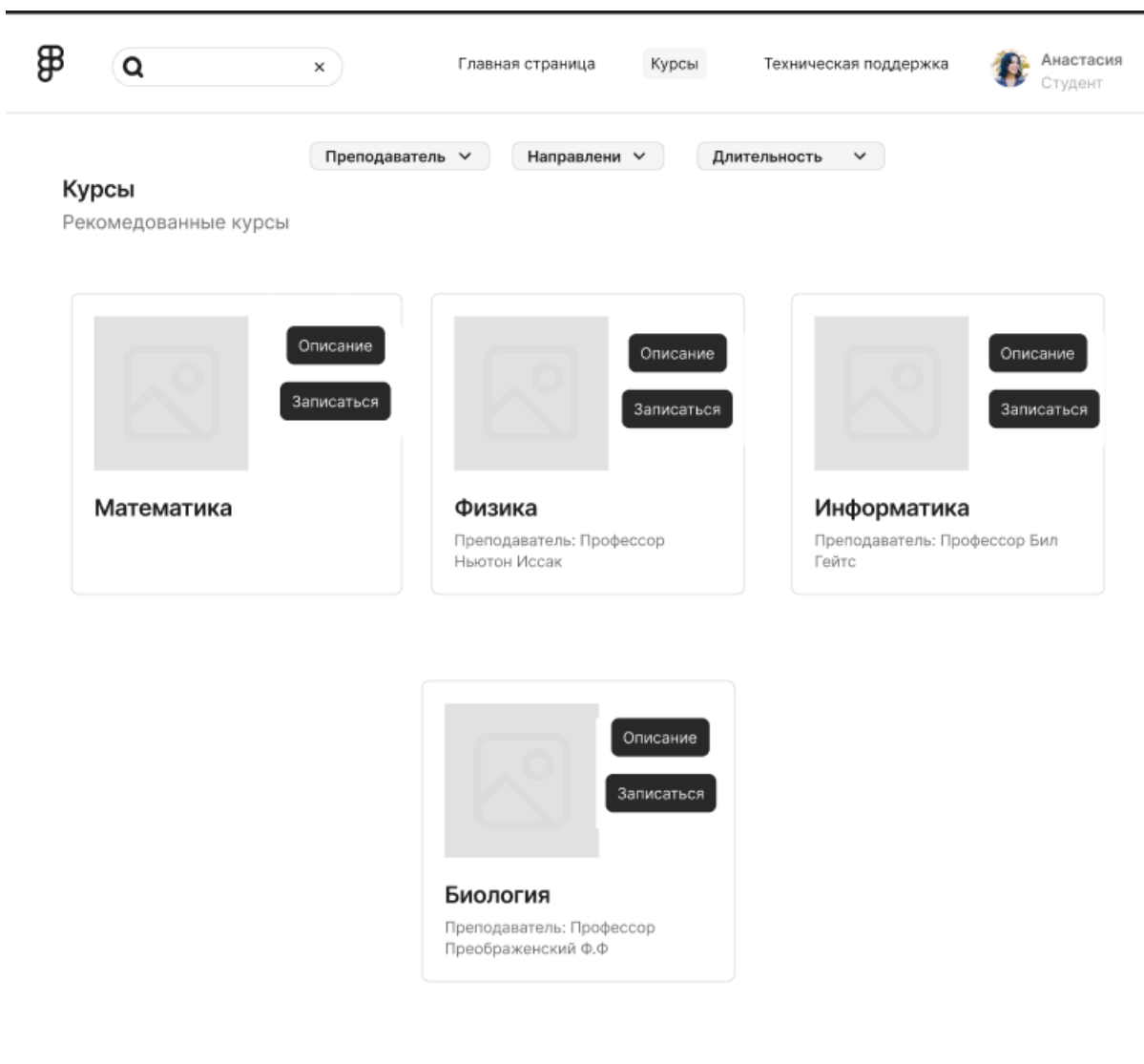


Рис 14 - Интерфейс студента

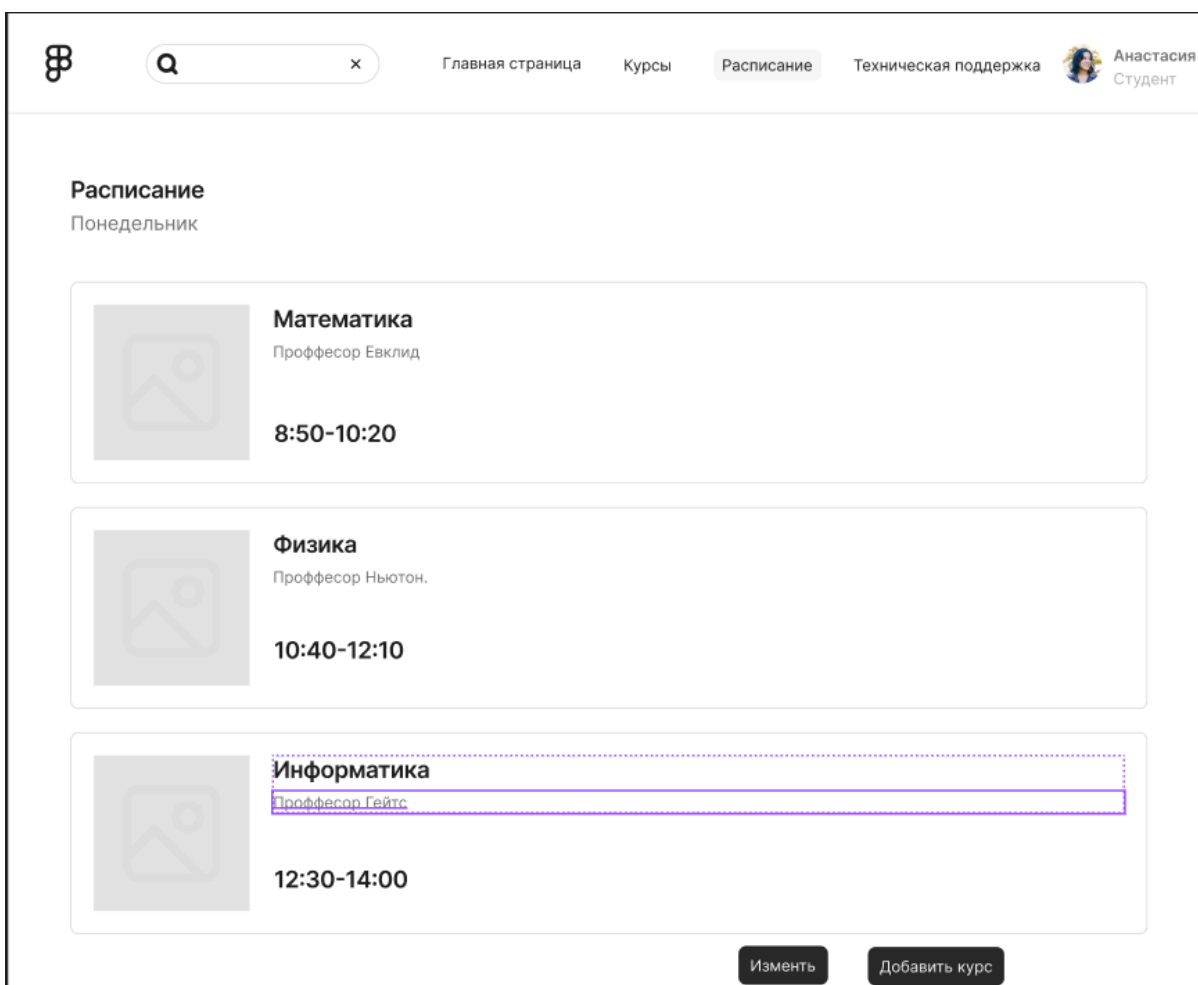


Рис 15 - Расписание студента

На данных макетах представлен основной функционал студента. На Рис 14 показана возможность записи на различные курсы, возможность просмотра описания, а также поиск курсов по различным фильтрам. На Рис 15, представлено окно “Расписание” в котором у студента есть возможность изменить свое расписание, добавить новый курс и посмотреть какие предметы он поставил на тот или иной день.

Далее представлен макет интерфейса Преподавателя(см Рис 16). На нем можно наблюдать аналог электронного журнала, в котором видно кто из студентов, записанных на курс был или не был на занятии.

Посещаемость








<p>Был</p>  <div>Евгений Description</div>	<p>Был</p>  <div>Олег Description</div>	<p>Был</p>  <div>Максим Description</div>
<p>Был</p>  <div>Анастасия Description</div>	<p>Не был</p>  <div>Елизавета Description</div>	<p>Не был</p>  <div>Ирина Description</div>

Рис 16 - Интерфейс посещаемости

Последний интерфейс (см Рис 17) является макетом загрузки курса. В нем можно задать длину курса, название, описание и загрузить фотографию



Название курса

Длина курса (в часах)

Описание

Рис 17 - Интерфейс загрузки курса

3.3 Оценка экономической эффективности

В данном пункте будет проведена оценка экономической эффективности разрабатываемой системы. Основная цель – определить насколько выгодно создание данного приложения с точки зрения затрат.

№	Этап	Дата начал а	Длительнос ть(дни)	Дата оконча ние	Ответств енный	Зарпл ата(в день)	Итого ва сумма
1 . 0	Предпроектный анализ	01.06. 2025	45	21.07.2 025			13500 0.00 руб
1 . 1	Концепция	01.06. 2025	10	11.06.2 025	Сис. аналитик	3000.0 0 руб.	30000 .00 руб.
1 . 2	Маркетинговое исследование	12.06. 2025	7	19.06.2 025.	Сис. аналитик	3000.0 0 руб.	21000 .00 руб
1 . 3	Анализ систем конкурентов	20.06. 2025	7	27.06.2 025	Сис. аналитик	3000.0 0 руб.	21000 .00 руб
1 . 4	Исследование предметной области	28.06. 2025	7	05.07.2 025	Сис. аналитик	3000.0 0 руб.	21000 .00 руб

1 . 5	Архитектура системы	06.07. 2025	7	13.06.2 025	Сис. аналитик	3000.0 0 руб.	21000 .00 руб
1 . 6	Создание техническ ого задания	14.07. 2025	7	21.07.2 025	Сис. аналитик	3000.0 0 руб.	21000 .00 руб
2 . 0	Проектны й анализ	21.07. 2025	23	14.08.2 025			69000 .00 руб
2 . 1	Анализ функ. блоков	21.07. 2025	15	05.08.2 025	Сис. аналитик	3000.0 0 руб.	45000 .00 руб
2 . 2	Анализ способов реализаци й	06.08. 2025	8	14.08.2 025	Сис. аналитик	3000.0 0 руб.	24000 .00 руб
3 . 0	Проектиро вание	14.08. 2025	53	11.10.2 025			13200 0.00 руб.
3 . 1	Разработка UML диаграмм	14.08. 2025	4	18.08.2 025	Сис. аналитик	3000.0 0 руб.	12000 .00 руб.

3 . 2	Разработка DFD диаграмм	19.08. 2025	6	25.08.2 025	Сис. аналитик	3000.0 0 руб.	18000 .00 руб.
3 . 3	Создание ER диаграмм ы	26.08. 2025	6	01.09.2 025	Сис. аналитик	3000.0 0 руб.	18000 .00 руб.
3 . 4	Создание IDEF0/IDE F3 диаграмм ы	02.09. 2025	10	12.09.2 025	Сис. аналитик	3000.0 0 руб.	30000 .00 руб.
3 . 5	разработка общего дизайна, логотипа	13.09. 2025	20	03.10.2 025	Дизайнер	6000.0 0 руб.	12000 0.00 руб.
3 . 6	Создание макетов информац ионной системы	04.10. 2025	7	11.10.2 025	Дизайнер	6000.0 0 руб.	42000 .00 руб.

4. 0	Реализация	12.10.2 025	1 4 0	05.03.2 025			740000. 00 руб.
---------	------------	----------------	-------------	----------------	--	--	--------------------

4.1	Backend разработка	12.10.2025	30	12.11.2025	Backend разработчик	7000.00 руб	210000.00 руб
4.2	Frontend разработка	12.11.2025	30	12.12.2025	Frontend разработчик	7000.00 руб	210000.00 руб
4.3	Создание и интеграция базы данных	12.12.2025	20	01.01.2026	Data инженер	6500 руб.	130000.00 руб.
4.4	Дизайн основных блоков ИС	23.01.2026	20	12.02.2026	Дизайнер	6000.руб.	120000.00 руб.
4.5	Интеграция UI	13.02.2026	20	05.03.2026	Frontend разработчик	3500 руб.	70000.00 руб.
5.0	Тестирование	06.03.2026	62	10.05.2026			590000.00 руб.
5.1	Проведение функционального тестирования	06.03.2026	14	20.03.2026	Тестировщик	5000.00 руб.	70000.00 руб.

5.2	Проверка безопасности данных	21.03.2026	14	04.04.2026	Тестировщик	5000.00 руб.	70000.00 руб.
5.3	Тестирование производительности	05.04.2026	14	19.04.2026	Тестировщик	5000.00 руб.	70000.00 руб.
5.4	Исправление найденных неполадок	20.04.2026	20	10.05.2026	frontend, backend разработчики	14000.00 руб.	280000.00 руб.
6.0	Отладка	11.05.2025	10	21.05.2025			50000.руб.
6.1	Отладка работы системы	11.05.2026	10	21.05.2026	Тестировщик	5000.00 руб.	50000.руб.
7.0	Оформление документации	22.05.2026	60	22.08.2026			255000.00 руб.
7.1	Оформление нормативной документации	22.05.202	30	22.06.2026	Юристы	4500.00 руб.	135000.00 руб.
7.2	Создание технической документации	22.06.2026	30	22.08.2026	Тех.писарь	4000.00 руб.	120000.00 руб.

8. 0	Итого	01.06.2 025	3 9 3	22.08.2 026			197100 0.00 руб.
---------	-------	----------------	-------------	----------------	--	--	------------------------

Табл 12 - Общие затраты на разработку

Из данной таблицы мы видим, общие затраты на разработку составят чуть меньше 2млн рублей. Далее была составлена Canvas модель, в которой можно увидеть основные источники трат и прибыли для данной информационной системы(см Рис 18). Canvas модель - это инструмент, позволяющий быстро и легко определить и донести бизнес-идею[15]



Рис 18 - Canvas модель

Из этой модели видны основные источники доходов и расходов для разрабатываемой системы. Исходя из этого можно сказать, что вся прибыль данного приложения будет идти от размещения рекламы внутри приложения и продажи платных подписок для студентов, которые расширяют функционал и будут убирать ее внутри приложения. Данная услуга будет приносить немало денег, т.к многие не заходят видеть раздражающую рекламу, а сама подписка будет стоить примерно 150-200 руб в месяц, что может позволить себе каждый студент. Также

можно будет продавать лицензию для того, чтобы университеты официально использовали данное приложение с своими системами. Исходя из всего этого рассчитаем сроки окупаемости проекта. В случае, разрабатываемой ИС, наш срок окупаемости является простым.[23] Это означает, что если инвестиции были только на старте, при этом доходы поступают либо одинаковыми суммами, либо одной суммой в конце периода инвестиций. Для данного срока формула расчета выглядит как: $\text{Срок окупаемости} = \text{размер капитальных вложений} / \text{планируемая прибыль за год}$.[24]. При цене подписки в 200 руб/мес, кол-во пользователей в 1000 ед., что является нормальным числом для нового проекта, продажи рекламы внутри приложения за 50000 руб/мес и при условии, что два вуза купили лицензию за 100000 руб/год. Получаем, что за год прибыль будет 3.1 млн. Далее рассчитаем капиталовложения за год. С учетом вложений на разработку, затрат на зарплаты команде, лицензии, маркетинговое продвижение, безопасность и т.д получаем, что в среднем сумма всех капиталовложений за год составит, 4,971 млн рублей. Теперь можем рассчитать срок окупаемости: $4,971/3,1 = 1,6$ год

Из всего этого, можно сказать, что приложение имеет неплохую экономическую эффективность, т.к продажа лицензий и подписок при большом количестве пользователей будут приносить прибыль, что позволит достаточно быстро окупить все затраты.

Заключение

В данной работе была рассмотрена актуальная проблема индивидуализации обучения в условиях цифрового образования. В рамках работы был произведен анализ предметной области, а также проектирование и реализация информационной системы, способная формировать индивидуальное расписание для студентов.

Проведенный анализ показал, что существующие решения на рынке, не предлагают достаточную гибкость в выстраивании расписания под себя или не имеют аналитические функции и учет посещаемости.

В ходе работы были произведены различные виды анализов, такие как SWOT и PEST анализы, для оценки рисков и внешних факторов. Определены функциональные и нефункциональные требования к системе. Созданы различные диаграммы, а также описаны различные

пользовательские сценарии. Также были произведены проектирования интерфейсов и базы данных.

Также хочется обратить внимание, на то, что можно улучшить и реализовать в дальнейшем.

1. Реализация мобильного приложения, что позволит сделать систему более доступной.
2. Разработка внутренних чатов для более удобной коммуникации между преподавателями и студентами
3. Реализация уведомлений и напоминаний, синхронизация с календарем
4. Реализация модуля голосований для возможности выбора удобного времени для большого кол-ва студентов

Данная система демонстрирует перспективность применения информационных технологий для повышения эффективности и персонализации учебного процесса. При правильном развитии, это приложение может внести огромный вклад в цифровизацию образования.

Приложение А

Главный раздел сценария выполнения варианта использования Выбор интересующих курсов/предметов”

Вариант использования	Выбор интересующих курсов/предметов
Актеры	Обучающийся
Краткое описание	Обучающийся выбирает курс

Цель	Обучающийся успешно записывается на курс
Тип	Базовый
Ссылки на другие варианты использования	Расширяет: Ознакомление с описанием курса/преподавателем

Табл 13 - Главный раздел сценария выполнения варианта использования
“Выбор интересующих курсов/предметов”

Типичный ход событий сценария выполнения варианта использования
«Выбор интересующих курсов/предметов»

Действия актеров	Отклик системы
1.Пользователь заходит в приложение	
2. Пользователь нажимает на кнопку “Курсы”	
	3.Переход к окну “Курсы” в котором есть выбор фильтров для выбора курса подходящего студенту ,
4.Пользователь выбирает подходящие для него фильтры	
5.Пользователь нажимает кнопку “Найти”	
	6.Система отправляет запрос к базе данных для проверки наличия курсах с заданными фильтрами

	7.База данных проверяет наличие таких курсов
	Исключение 1: удовлетворяющих курсов нету в базе данных
	8. База данных подтверждает наличие таких курсов у себя
	9. Система выводит список курсов
10. Пользователь выбирает курс и нажимает на кнопку “Записаться”	
	11.Система отправляет запрос в базу данных на добавление пользователя
	Исключение 2: ошибка базы данных
	12.Система добавляет пользователя в курс
13. Пользователь успешно записан на курс	

Табл 14 - Типичный ход событий сценария выполнения варианта использования «Выбор интересующих курсов/предметов»

Исключение	сценария
выполнения	варианта
использования	«Выбор
интересующих	
курсов/предметов»	

Исключение 1: удовлетворяющих курсов нету в базе данных

Действия актеров	Отклик системы
	7.1 База данных отправляет системе ответ, о том что таких курсов нет
	7.2 Система выдает сообщение о том, что такие курсы не найдены и просит сменить фильтры

Табл 15 - Исключение 1: удовлетворяющих курсов нету в базе данных

Исключение 2: ошибка базы данных

Действия актеров	Отклик системы
	11.1 Система не получает ответ от базы данных
	11.2 Система выдает ошибку и просит повторить попытку позже

Табл 16 - Исключение 2: ошибка базы данных

Главный раздел сценария выполнения варианта использования “Выбор предложенных курсов”

Вариант использования	Выбор предложенных курсов
Актеры	Обучающийся
Краткое описание	Выбор предложенных курсов на основе ваших предпочтения
Цель	Пользователь успешно выбирает предложенный курс
Тип	Базовый
Ссылки на другие варианты использования	Расширяется: Ознакомление с описанием курса/преподавателем

Табл 17 - Главный раздел сценария выполнения варианта использования “Выбор предложенных курсов”

Типичный ход событий сценария выполнения варианта использования “Выбор предложенных курсов”

Действия актеров	Отклик системы
1. Пользователь заходит в вкладку “Курсы”	
	2. Система на основе курсов или их популярности(если пользователь новый) на которые записан пользователь отправляет запрос в базу данных для нахождения курсов с похожими направлениями,

	преподавателями и т.д
	3. База данных обрабатывает данный запрос
	Исключения 1: ошибка базы данных
	4. Система получает список и выводит его на экран
5. Пользователь выбирает интересующий курс	

Табл 18 - Типичный ход событий сценария выполнения варианта использования “Выбор предложенных курсов”

Исключение сценария выполнения варианта использования “Выбор предложенных курсов”

Исключение 1: ошибка базы данных

Действия актеров	Отклик системы
	3.1 Система не получает ответ от базы данных
	3.2 Система выдает ошибку и просит повторить попытку позже

Табл 19 - Исключение 1: ошибка базы данных

<p>Главный раздел сценария выполнения варианта использования</p> <p>“Ознакомление с описанием курса/преподавателя”</p>
--

Вариант использования	Ознакомление с описанием курса/преподавателя
Актеры	Обучающийся
Краткое описание	Просмотр описания курса
Цель	Пользователь успешно просматривает описание курса
Тип	Базовый
Ссылки на другие варианты использования	Расширяет: “Выбор предложенных курсов” и «Выбор интересующих курсов/предметов»

Табл 20 -Главный раздел сценария выполнения варианта использования “Ознакомление с описанием курса/преподавателя”

<p>Типичный ход событий сценария выполнения варианта использования</p> <p>“Ознакомление с описанием курса/преподавателя”</p>
--

Действия актеров	Отклик системы
1. Пользователь заходит в вкладку “Курсы”	
2. Пользователь нажимает на	

кнопку “Описание”	
	3. Система отправляет запрос в базу данных на получение описания курса
	4. База данных обрабатывает запрос
	Исключения 1: Ошибка базы данных
	5. Система получает описание выбранного курса и выводит его
6. Пользователь видит описание выбранного курса	

Табл 21 - Типичный ход событий сценария выполнения варианта использования “Ознакомление с описанием курса/преподавателя”

Исключение сценария выполнения варианта использования
“Ознакомление с описанием курса/преподавателя”

Исключение 1: ошибка базы данных

Действия актеров	Отклик системы
	4.1 Система не получает ответ от базы данных

	4.2 Система выдает ошибку и просит повторить попытку позже
--	--

Табл 22 -Исключение 1: ошибка базы данных

Главный раздел сценария выполнения варианта использования
“Создание расписания”

Вариант использования	Создание расписание
Актеры	Обучающийся
Краткое описание	Создание расписания
Цель	Пользователь успешно создание расписание
Тип	Базовый
Ссылки на другие варианты использования	Включает: Изменение расписания

Табл 23 - Главный раздел сценария выполнения варианта использования
“Создание расписания”

Типичный ход событий сценария выполнения варианта использования
“Создание расписания”

Действия актеров	Отклик системы
1.Пользователь заходит в вкладку расписание и выбирает	

определенный день	
2. Нажимает кнопку добавить курс	
	3. Система запрашивает у базы данных список курсов на которые записан пользователь
	4. База данных обрабатывает запрос
	Исключения 1: Ошибка базы данных
	Исключение 2: Пользователь не записан на курсы
	5. Система выгружает список курсов
6. Пользователь добавляет курс в выбранный день, в выбранное время	
	7. Система отправляет запрос в базу данных на добавление курса в таблицу в расписание на определенный день
	8. База данных обрабатывает запрос
	Исключение 3: выбранное время не подходит
	9. Система успешно добавляет курс
10. Пользователь видит сообщение “Курс успешно добавлен”	
11. Пользователь нажимает кнопку сохранить	

Табл 24 - Типичный ход событий сценария выполнения варианта использования “Создание расписания”

Исключение сценария выполнения варианта использования
“Создание расписания”

Исключение 1: ошибка базы данных

Действия актеров	Отклик системы
	4.1 Система не получает ответ от базы данных
	4.2 Система выдает ошибку и просит повторить попытку позже

Табл 25 - Исключение 1: ошибка базы данных

Исключение 2: Пользователь не записан на курсы

Действия актеров	Отклик системы
	4.1 Система выдает ошибку и просит пользователя записаться на курсы

Табл 26 - Исключение 2: Пользователь не записан на курсы

Исключение 3: выбранное время не подходит

Действия актеров	Отклик системы
	8.1 Система выдает ошибку и просит выбрать другой временной промежуток т.к оно уже занято или не является учебным временем
9 Пользователь выбирает новый временной промежуток	

Табл 27 - Исключение 3: выбранное время не подходит

Главный раздел сценария выполнения варианта использования “Загрузка информации о курсе”

Вариант использования	Загрузка информации о курса
Актеры	Преподаватель
Краткое описание	Загрузка информации о курсе
Цель	Пользователь успешно загружает информацию о новом курсе
Тип	Базовый
Ссылки на другие варианты использования	

Табл 28 - Главный раздел сценария выполнения варианта использования “Загрузка информации о курсе”

Типичный ход событий сценария выполнения варианта использования
“Загрузка информации о курса”

Действия актеров	Отклик системы
1.Пользователь нажимает кнопку “Добавить курс”	
2.Пользователь добавляет описание и название для курса и нажимает кнопку сохранить	
	3.Система отправляет запрос в базу данных для добавления нового курса
	4.База данных обрабатывает данный запрос
	Исключение 1: база данных не отвечает
	5.Система успешно добавляет курс в базу данных
6. Пользователь видит сообщение, что курс успешно добавлен	

Табл 29 - Типичный ход событий сценария выполнения варианта использования “Загрузка информации о курсе”

Исключение сценария выполнения варианта использования
“Загрузка информации о курсе”

Исключение 1: база данных не отвечает

	4.1 Система не получает ответ от базы данных
	4.2 Система выдает ошибку и просит повторить попытку позже

Табл 30 - Исключение 1: база данных не отвечает

Главный раздел сценария выполнения варианта использования
“Ознакомление с отчетами о курсах”

Вариант использования	Ознакомление с отчетами о курсах
Актеры	Преподаватель
Краткое описание	Просмотр отчетов о курсах
Цель	Пользователь успешно просматривает отчеты о курсах
Тип	Базовый
Ссылки на другие варианты использования	Включает: “Отчет о посещаемости” и “Отчет о кол-ве записавшихся на курс”

Табл 31 - Главный раздел сценария выполнения варианта использования
“Ознакомление с отчетами о курсах”

Типичный ход событий сценария выполнения варианта использования
“Ознакомление с отчетами о курсах”

Действие актеров	Отклик системы
1.Пользователь заходит на страницу собственного курса	
2.Нажимает кнопку отчет и выбирает период	
	3.Система отправляет запрос в базу данных для получения информации о курсах
	4.База данных обрабатывает запрос
	Исключение 1: на данном промежутке времени нету статистики
	Исключение 2: ошибка базы данных
	5.Система формирует вкладки с различными таблицами
6.Пользователь выбирает какую таблицу посмотреть	

Табл 32 - Типичный ход событий сценария выполнения варианта использования “Ознакомление с отчетами о курсах”

Исключение сценария выполнения варианта использования
“Ознакомление с отчетами о курсах”

Исключение 1: на данном промежутке времени нету статистики

	4.1 Система выдает ошибку и просит изменить промежуток
5.Пользователь меняет временной промежуток	

Табл 33 - Исключение 1: на данном промежутке времени нету статистики

Исключение 2: база данных не отвечает

	4.1 Система не получает ответ от базы данных
	4.2 Система выдает ошибку и просит повторить попытку позже

Табл 34 - Исключение 2: база данных не отвечает

Главный раздел сценария выполнения варианта использования “Отчет о посещаемости”

Вариант использования	Ознакомление с отчетам о посещаемости
Актеры	Преподаватель
Краткое описание	Просмотр отчетов о посещаемости
Цель	Пользователь успешно просматривает отчеты о посещаемости
Тип	Зависимый
Ссылки на другие варианты использования	Включено в : “Отчет о посещаемости”

Табл 35 - Главный раздел сценария выполнения варианта использования “Отчет о посещаемости”

Типичный ход событий сценария выполнения варианта использования “Отчет о посещаемости”

Действие актеров	Отклик системы
1.Пользователь выбирает таблицу “Посещаемость”	
	2.Система выводит таблицу
	Исключение 1: данных за данный период нету

3.Пользователь получает доступ к информации о посещаемости	
--	--

Табл 36 - Типичный ход событий сценария выполнения варианта использования “Отчет о посещаемости”

Исключение сценария выполнения варианта использования “Отчет о посещаемости”
--

Исключение 1: данных за данный период нету
--

	2.1 Система выдает ошибку и просит изменить промежуток
3.Пользователь меняет временной промежуток	

Табл 37 - Исключение 1: данных за данный период нету

Главный раздел сценария выполнения варианта использования “Отчет о кол-ве записавшихся на курс”

Вариант использования	Ознакомление с отчетам о записавшихся на курс
Актеры	Преподаватель

Краткое описание	Просмотр отчетов о записанных на курс
Цель	Пользователь успешно просматривает отчеты о записанных на курс
Тип	Зависимый
Ссылки на другие варианты использования	Включено в : “Отчет о посещаемости”

Табл 38 - Главный раздел сценария выполнения варианта использования “Отчет о кол-ве записавшихся на курс”

Типичный ход событий сценария выполнения варианта использования “Отчет о кол-ве записавшихся на курс”

Действие актеров	Отклик системы
1.Пользователь выбирает таблицу “Записанные”	
	2.Система выводит таблицу
	Исключение 1: данных за данный период нету
3.Пользователь получает доступ к информации о записанных студентах	

Табл 39 - Типичный ход событий сценария выполнения варианта использования “Отчет о кол-ве записавшихся на курс”

Исключение сценария выполнения варианта использования
“Отчет о кол-ве записавшихся на курс”

Исключение 1: данных за данный период нету

	2.1 Система выдает ошибку и просит изменить промежуток
3.Пользователь меняет временной промежуток	

Табл 40 - Исключение 1: данных за данный период нету

Главный раздел сценария выполнения варианта использования
“Изменение расписания”

Вариант использования	Изменение собственного расписания
Актеры	Студент
Краткое описание	Изменения расписания студентом
Цель	Студент успешно меняет расписания под себя
Тип	Зависимый
Ссылки на другие варианты использования	Включено в : Создание расписания

--	--

Табл 41 - Главный раздел сценария выполнения варианта использования
“Изменение расписания”

Типичный ход событий сценария выполнения варианта использования “Изменение расписания”

Действие актеров	Отклик системы
1.Пользователь нажимает кнопку “Изменить” в вкладке расписание	
2.Пользователь выбирает день, курс и промежуток времени который он хочет задать.	
3.Пользователь нажимает на кнопку “Сохранить” для подтверждения изменений	
	4.Система отправляет запрос в базу данных для изменения данных
	5.База данных обрабатывает запрос
	Исключение 1: Выбраны неподходящие данные
	Исключение 2: База данных не отвечает
	6.Система выводит измененное расписание

Табл 42 - Типичный ход событий сценария выполнения варианта использования “Изменение расписания”

Под неподходящими данными в таблице 42 имеется ввиду, то что временной промежуток или день не подходят для учебы(нерабочее время или выходной)

Исключение сценария выполнения варианта использования “Изменение расписания”

Исключение 1: Выбраны неподходящие данные

Действие актеров	Отклик системы
	5.1 Данные которые выбраны пользователем не подходят(день, временной промежуток)
	5.2 Система просит проверить данные и повторить еще раз

Табл 43 - Исключение 1: Выбраны неподходящие данные

Исключение 2: База данных не отвечает

Действие актеров	Отклик системы
	5.1 Система не получает ответ от базы данных

	5.2 Система выдает ошибку и просит повторить попытку позже
--	--

Табл 44 - Исключение 2: Выбраны неверные данные

Приложение В

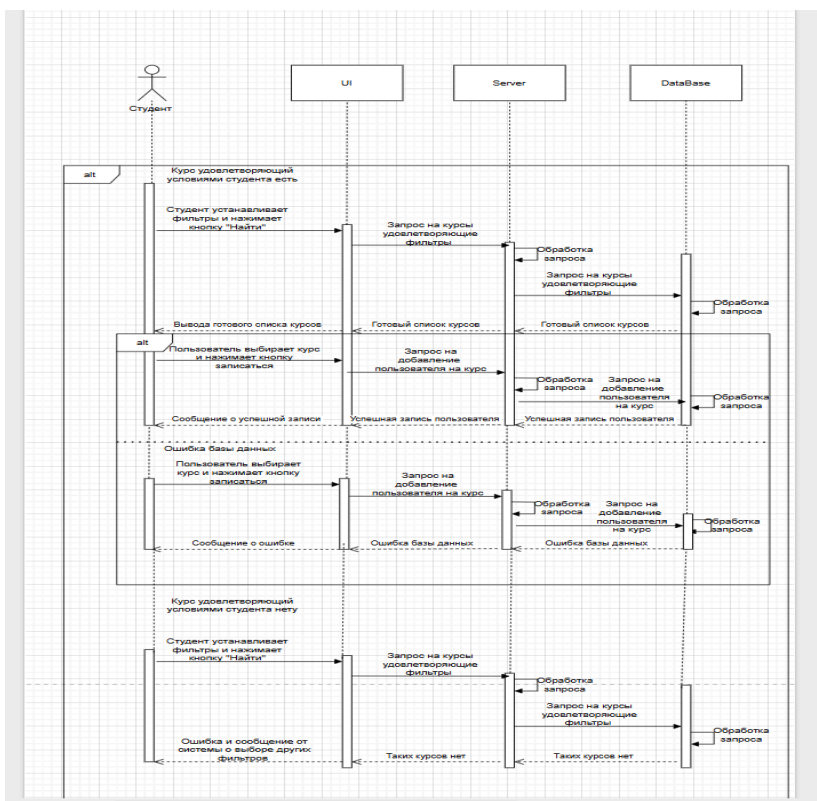


Рис 18 - Диаграмма последовательности для блока “Выбор интересующих курсов/предметов”

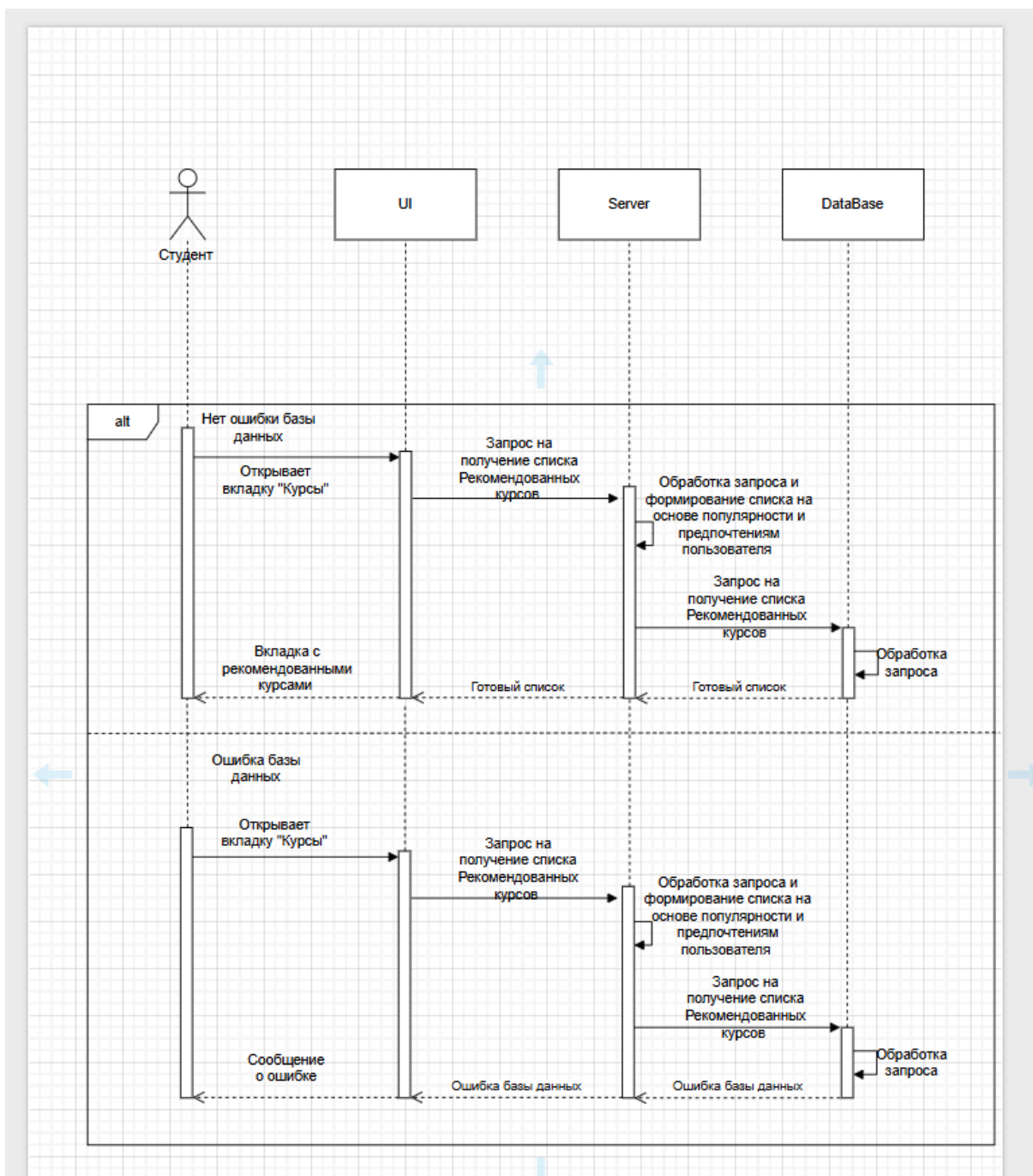


Рис 19 - Диаграмма последовательности для блока “Выбор предложенных курсов”

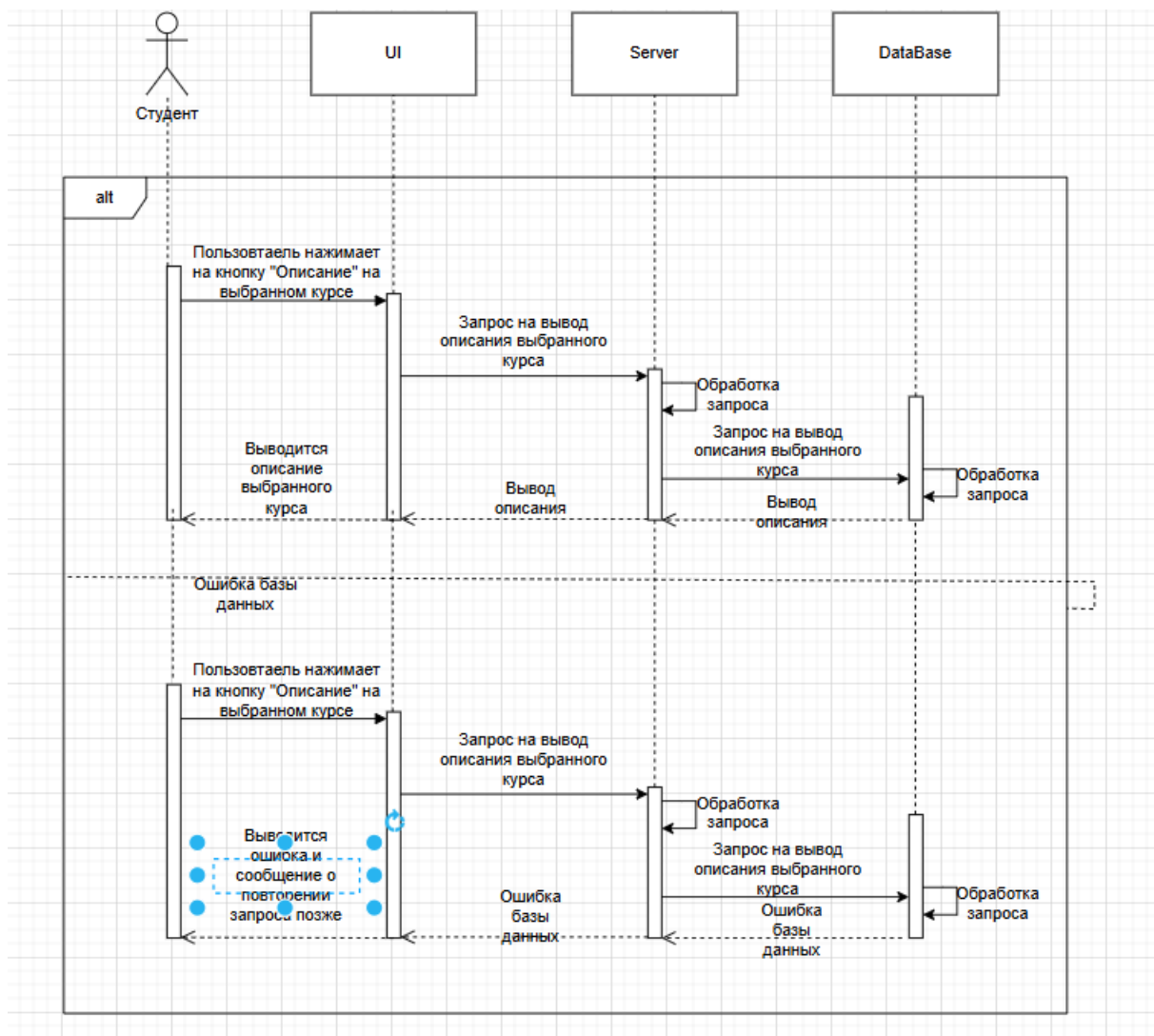


Рис 20 - Диаграмма последовательности для блока “Ознакомление с описанием курса/преподавателя”

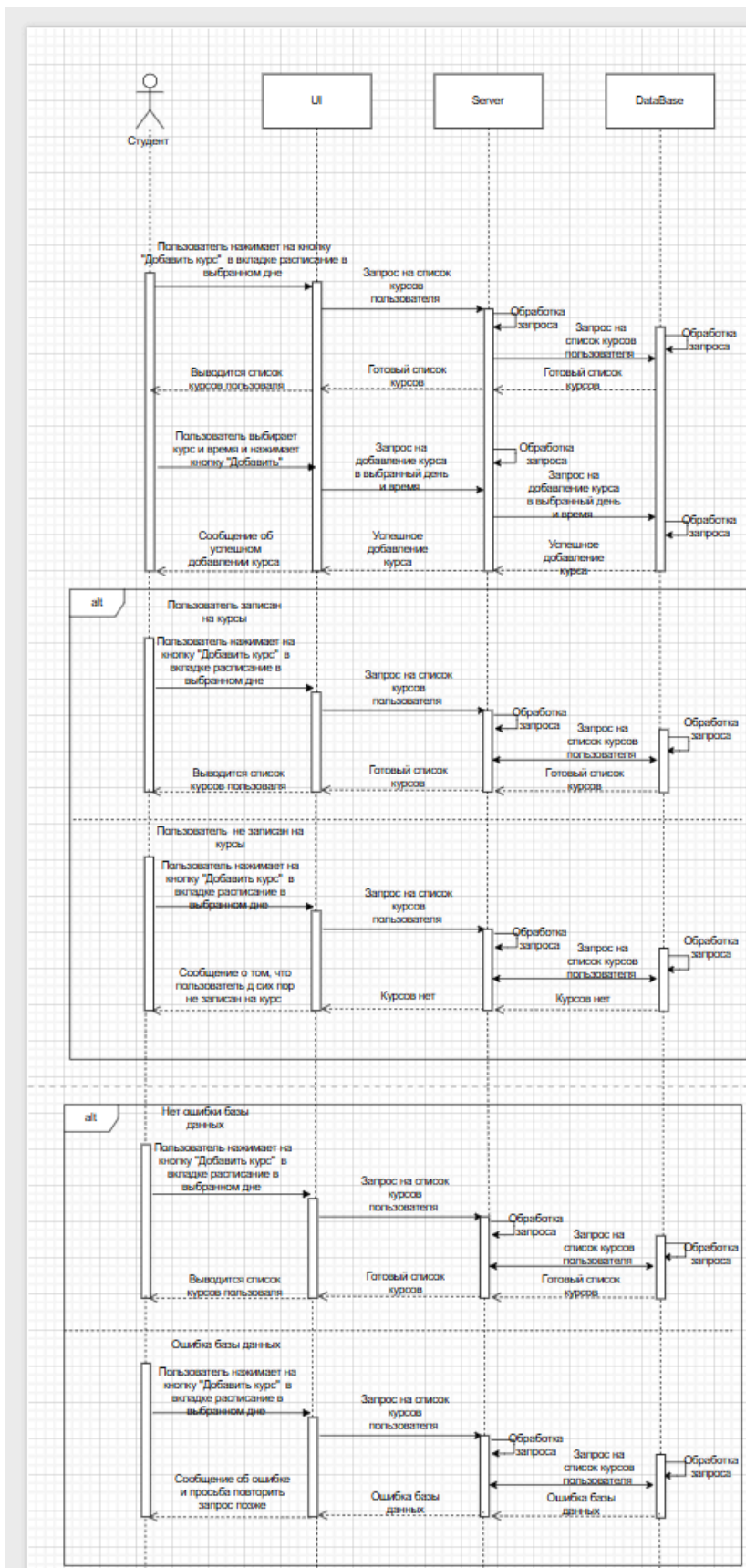




Рис 21-22 - Диаграмма последовательности для блока "Создание расписания"

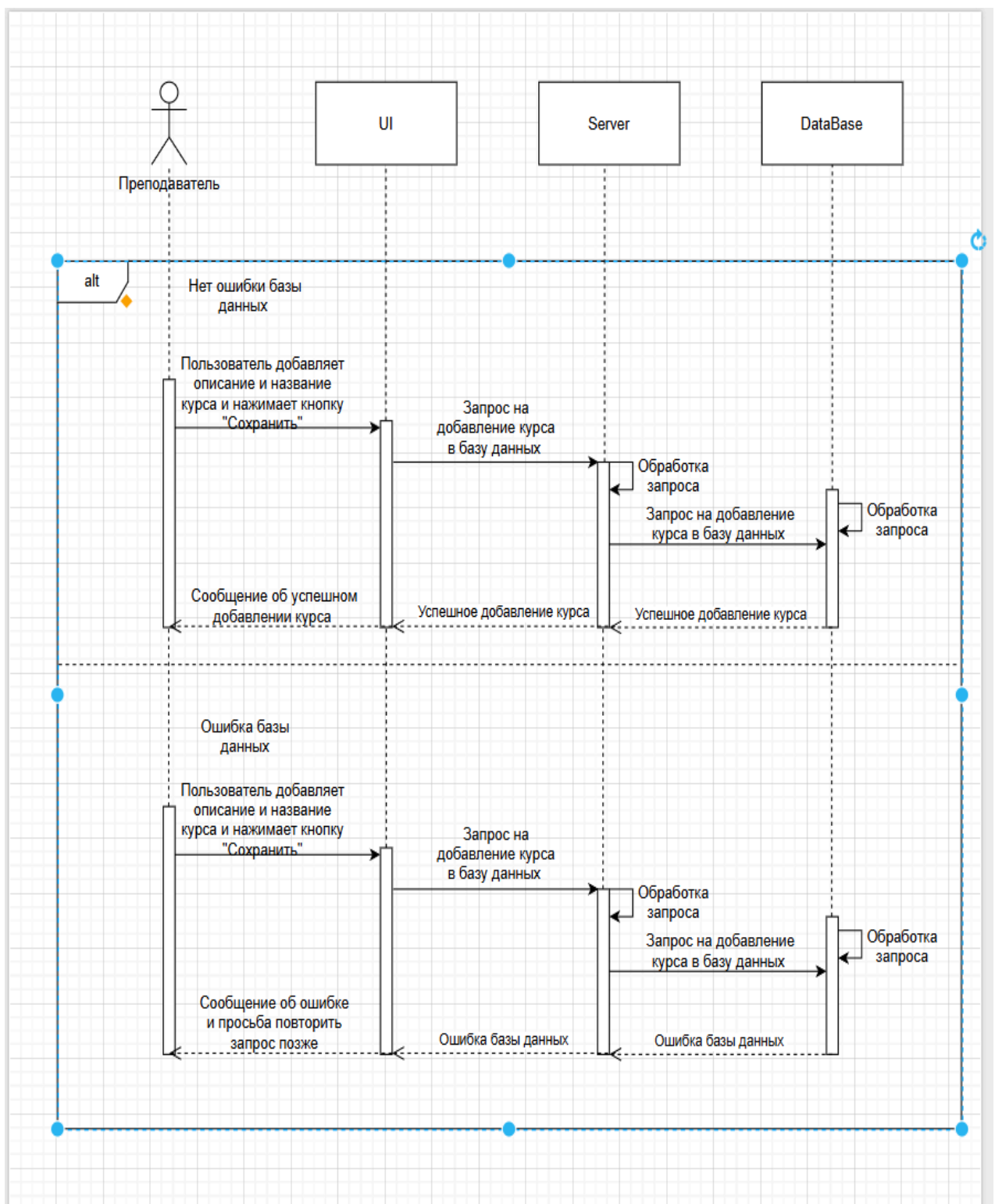


Рис 23 - Диаграмма последовательности для блока “Загрузка информации о курсе”

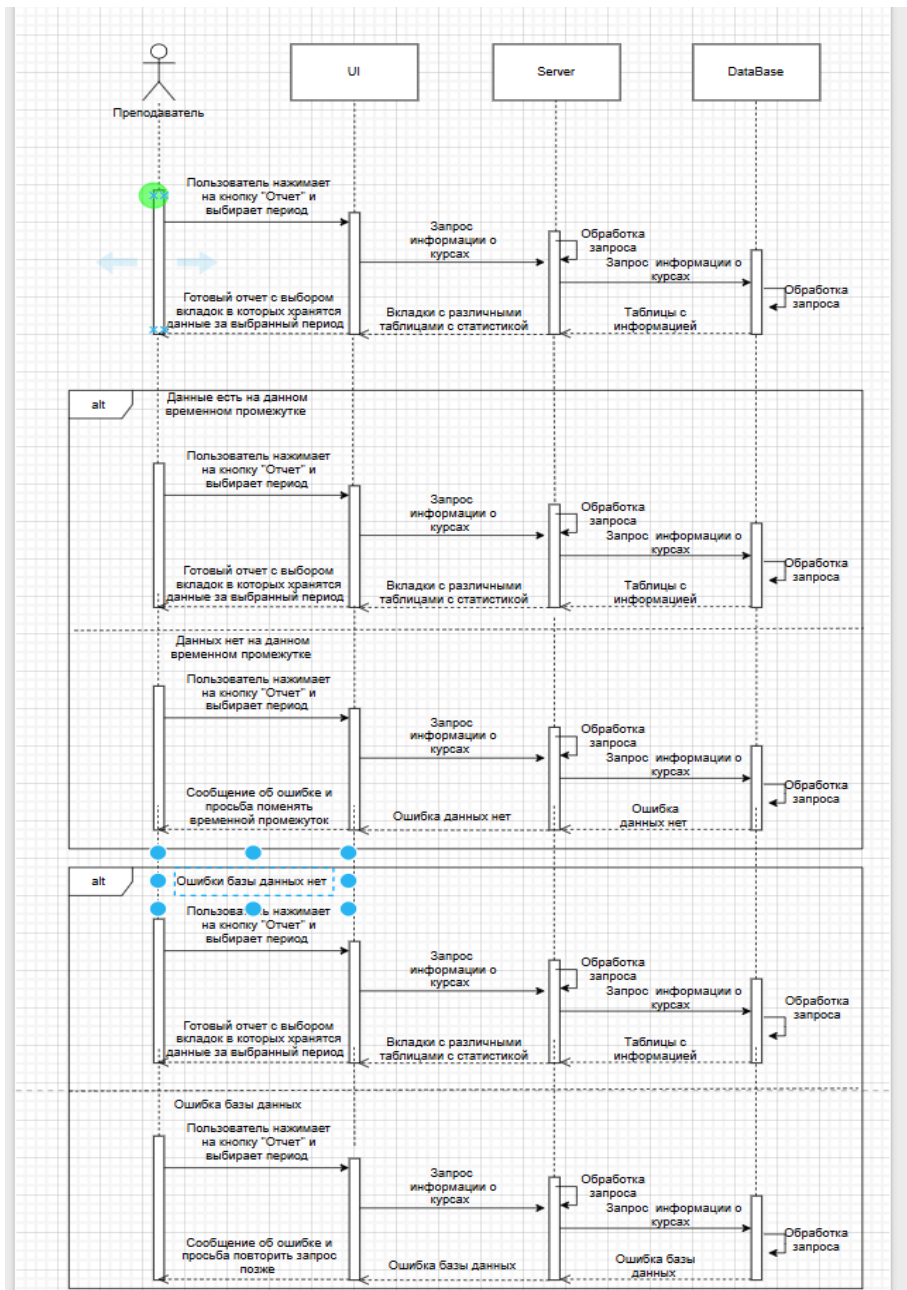


Рис 24 - Диаграмма последовательности для блока “Ознакомление с отчетами о курсах”

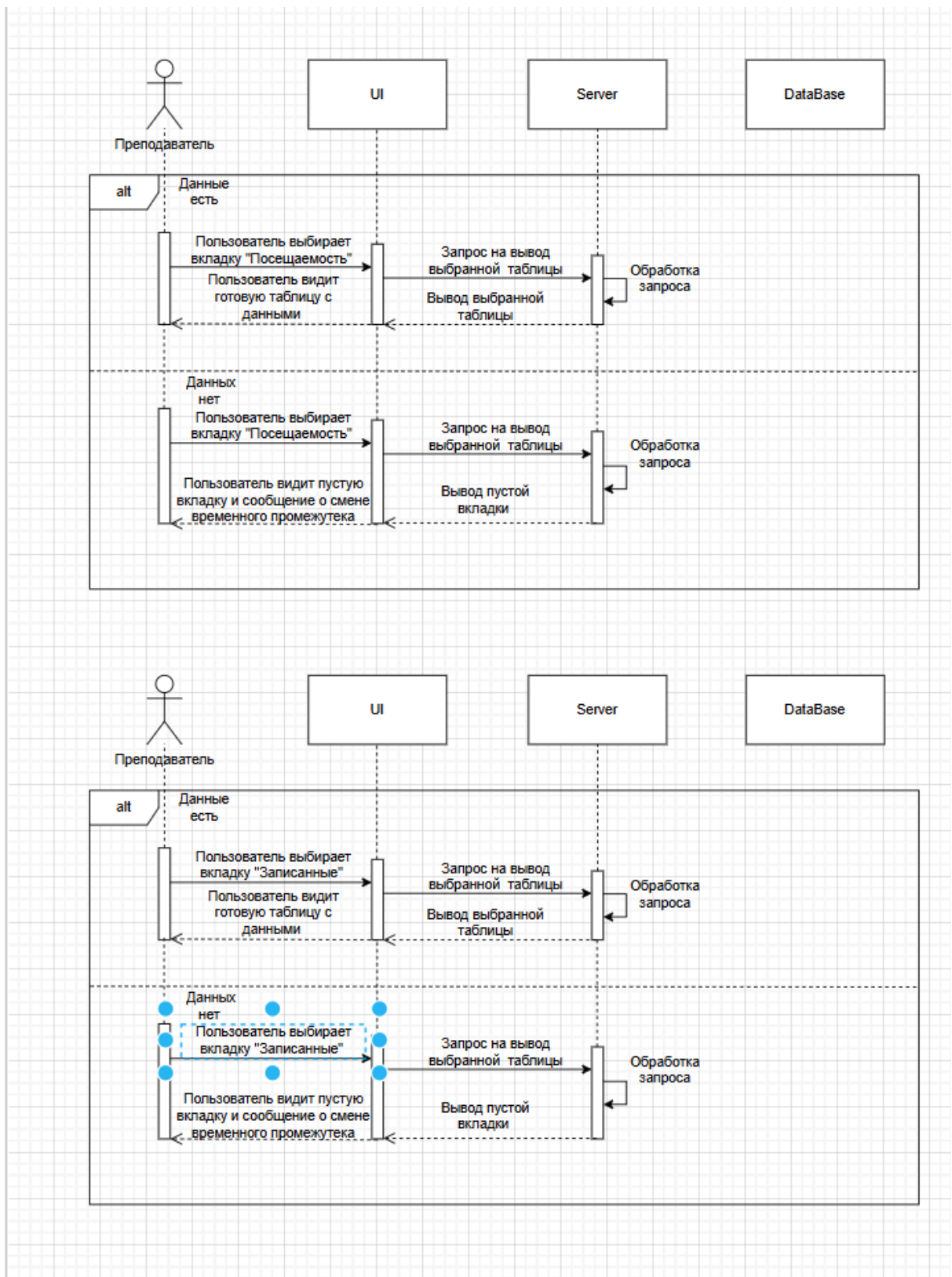


Рис 25 - Диаграмма последовательности для блоков “Отчет о посещаемости” и “Отчет о кол-ве записавшихся на курс”

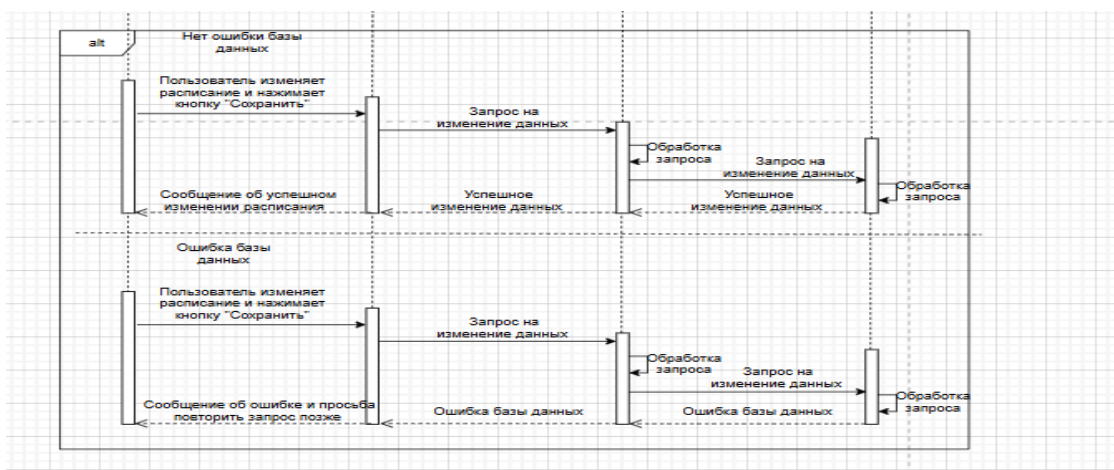
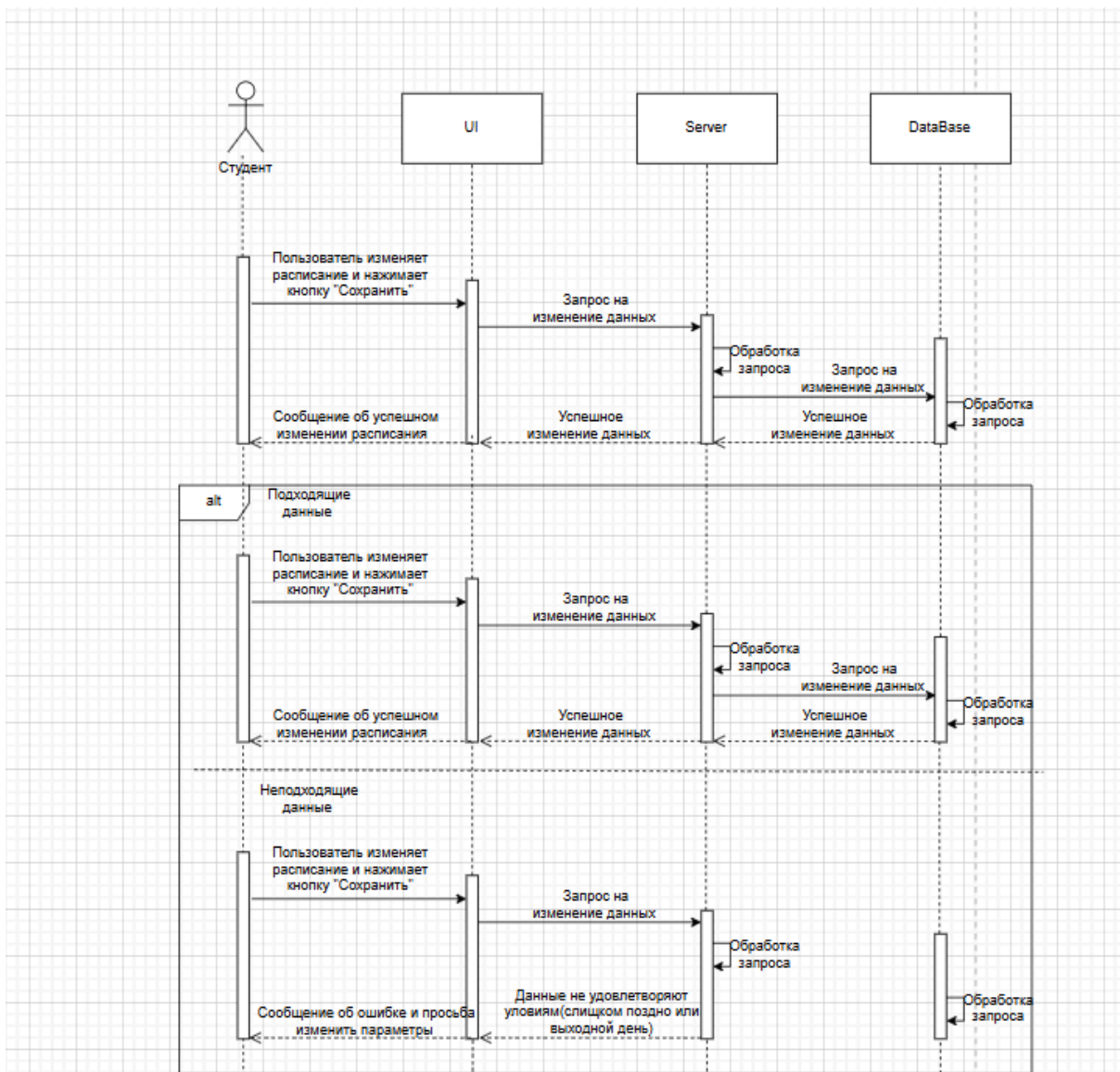


Рис 26-27 -Диаграмма последовательности для блоков “Изменение расписания”

Приложение С

```
schedule_get(3)
schedule_update( student_id: 1, day = 'Sunday', new_start_time="12:00", new_end_time = "17:00", course_id=2 )
```

Ошибка: Время 12:00-17:00 пересекается с другим занятием 12:00-15:00.

Process finished with exit code 0

Рис 28-29- Реакция на пересечение с другими курсами

```
schedule_update( student_id: 1, day = 'Monday', new_start_time="21:00", new_end_time = "22:00", course_id=2 )
```

Ошибка: Время окончания должно быть не позже 21:00.

Process finished with exit code 0

Рис 30-31 - Реакция на выбор позднего времени

```
schedule_get(3)
schedule_update( student_id: 1, day = 'Sunday', new_start_time="12:00", new_end_time = "17:00", course_id=2 )
```

Ошибка: день 'Sunday' некорректен. Допустимые дни: Saturday, Monday, Friday, Thursday, Tuesday, Wednesday.

Рис 32-33 - Реакция на некорректный выбор дня

Источники:

[1] CASE // Википедия. URL: <https://ru.wikipedia.org/wiki/CASE> .(дата обращения 05.04.2025)

[2]Что такое SWOT-анализ // Яндекс.Практикум. URL: <https://practicum.yandex.ru/blog/cto-takoe-swot-analyz/> .(дата обращения 05.04.2025)

[3]PEST-анализ // Библиотека фасилитатора. URL: <http://www.facilab.pro/cards/23> .(дата обращения 05.04.2025)

[4]Коцюба И.Ю., Чунаев А.В., Шиков А.Н. Основы проектирования информационных систем: учебное пособие. – СПб.: Университет ИТМО, 2017. – 48 с. – URL: <https://books.ifmo.ru/file/pdf/1705.pdf> .(дата обращения 10.04.2025)

[5]Ананченко И.В., Войтюк Т.Е., Марченко Е.В. Архитектура информационных систем: учебное пособие. – СПб.: Университет ИТМО, 2021. – 6 с. – URL: <https://books.ifmo.ru/file/pdf/3361.pdf> .(дата обращения 10.04.2025)

[6]Database Design Fundamentals// Itonboard. URL: <https://www.geeksforgeeks.org/database-design-fundamentals/> .(дата обращения 10.04.2025)

[7]Информационные системы. Проектирование информационных систем //ПедагогРФ. URL: <https://pedagog-rf.ru/journal/868-informacionnye-sistemy/> .
вариантов (дата обращения 10.04.2025)

[8]Использование диаграммы вариантов использования UML при проектировании ПО // Хабр. URL: <https://habr.com/ru/articles/566218/> .
(дата обращения 11.04.2025)

[9]Al-Fedaghi S. UML Sequence Diagram: An Alternative Model // ResearchGate. URL: https://www.researchgate.net/publication/352016825_UML_Sequence_Diagram_An_Alternative_Model . (дата обращения 11.04.2025)

[10]Александров Д.В., Грачев И.В., Фадин Д.Н. CASE-технологии: учебное пособие. – URL: <https://studfile.net/preview/7244123/page:4/> .

[11]Диаграмма последовательности (sequence-диаграмма) // Хабр. URL: <https://habr.com/ru/articles/814769/> . (дата обращения 11.04.2025)

[12]UML: обзор основных типов диаграмм. Диаграмма компонентов. Ч.2 // Хабр. URL: <https://habr.com/ru/articles/756552/> . (дата обращения 11.04.2025)

[13]ГОСТ Р ИСО/МЭК ТО 10032—2007. Эталонная модель управления данными. – URL: <https://ohranatruda.ru/upload/iblock/075/4293832367.pdf> ((дата обращения 12.04.2025)

[14]Система управления базами данных // Большая российская энциклопедия. URL: <https://bigenc.ru/c/sistema-upravleniia-bazami-dannykh-98ed12> . (дата обращения 12.04.2025)

[15]How To: Business Model Canvas Explained // Medium. URL: <https://medium.com/seed-digital/how-to-business-model-canvas-explained-ad3676b6fe4a> . (25.04.2025)

[16]Учебное пособие: Диаграмма развертывания // Creately. URL: <https://creately.com/blog/ru/uncategorized-ru/учебное-пособие-по-диаграмме-развёрт/> . (дата обращения 11.04.2025)

[17]Что такое use case: примеры и шаблоны // Sky.pro. URL: <https://sky.pro/wiki/javascript/chto-takoe-use-case-primery-i-shablony/> . (дата обращения 11.04.2025)

[18]Человеко-машинный интерфейс. 2-й курс. ПрЧМИ. Справочный материал. – URL: <https://studfile.net/preview/6796113/> . (13.04.2025)

[19]Диаграмма вариантов использования (прецедентов) (use case diagram) //imlearning.ru. URL: https://imlearning.ru/netcat_files/file/FSIS/ФСИС_семинар-2_Диаграммы-вариантов-использования.pdf . (дата обращения 16.04.2025)

[20]Astratova G.V., Rutkauskas T.K., Parushina N.V., Suchkova N.A. New trends and competition in the market of higher education services in the context of digitalization // SHS Web of Conferences. 2021. – Vol. 112. – URL: https://www.shs-conferences.org/articles/shsconf/pdf/2021/25/shsconf_ntsscem2021_01024.pdf . (дата обращения 05.04.2025)

[21]Порядок создания технического задания для разработки информационной системы // Хабр. URL: <https://habr.com/ru/articles/769460/> . (дата обращения 11.04.2025)

[22]Диаграмма прецедентов (use case диаграмма) // TestEngineer.ru. URL: <https://testengineer.ru/use-case-diagram/#h--definition> . (дата обращения 13.04.2025)

[23]Срок окупаемости проекта // Сбербанк. URL: https://www.sberbank.com/ru/s_m_business/pro_business/srok-okupaemosti-proekta#kakoy-srok-okupaemosti-schitat-horoshim . (дата обращения 10.05.2025)

[24]Срок окупаемости инвестиций // Банки.ру. URL:
https://www.banki.ru/wikibank/srok_okupaemosti_investitsiy/ . (дата
обращения 10.05.2025)

[25]Язык UML: преимущества и описание // evmservice.ru. URL:
<https://evmservice.ru/blog/yazyk-uml/#Преимущества>. (дата обращения
10.04.2025)

[26]Фаулер М. UML: основы. Краткое руководство по стандартному
языку объектного моделирования. — URL:
<https://csharpcooking.github.io/theory/Osnovy-UML-Ru.pdf>. (дата
обращения 10.04.2025)

[27]Макаров Н.С. UML: поддержка проектирования и инструментальные
среды // Cyberleninka. URL:
[https://cyberleninka.ru/article/n/uml-podderzhka-proektirovaniya-i-instrument
alnye-sredy/viewer](https://cyberleninka.ru/article/n/uml-podderzhka-proektirovaniya-i-instrumentalnye-sredy/viewer) .(дата обращения 10.04.2025)