

Министерство науки и образования Украины
Одесский Национальный Политехнический Университет
Институт компьютерных систем
Кафедра информационных систем

КУРСОВАЯ РАБОТА

по дисциплине “Технологии создания программных продуктов”

по теме

“Разработка информационного продукта для обучения ALLearn”

Выполнили:

студент 3-го курса

группы НАИ-185

Бектимиров Р.З.

Шаргородский В.А.

Гараба Я.В.

Проверил:

Блажко О.А.

Одесса 2020

Аннотация

В курсовой работе описывается процесс создания программного продукта.

Результат работы размещен на github-репозитории по адресу:

<https://github.com/Kramzin/ALLearn>

Содержание

1 Требования к программному продукту	6
1.1 Определение потребностей потребителя	6
1.1.1 Иерархия потребностей потребителя	6
1.1.2 Детализация материальной потребности	7
1.2 Бизнес-требования к программному продукту	8
1.2.1 Описание проблемы потребителя	8
1.2.1.1 Концептуальное описание проблемы потребителя	8
1.2.1.2 Метрическое описание проблемы потребителя	8
1.2.2 Цель создания программного управления	9
1.2.2.1 Проблемный анализ существующих программных продуктов	9
1.2.2.2 Цель создания программного продукта	11
1.2.3 Название программного продукта	11
1.2.3.1 Лозунг программного продукта	11
1.2.3.2 Логотип программного продукта	12
1.3 Требования пользователя к программному продукту	12
1.3.1 История пользователя программного продукта	12
1.3.2 Диаграмма прецедентов программного продукта	13
1.3.3 Сценарии исполнения прецедентов программного продукта	14
1.4 Функциональные требования к программному продукту	19
1.4.1. Многоуровневая классификация функциональных требований	19
1.4.2 Функциональный анализ существующих продуктов	21
1.5 Нефункциональные требования к программному продукту	22
1.5.1 Описание внешних интерфейсов	22
1.5.1.1 Описание интерфейса пользователя	22
1.5.1.1.1 Описание INPUT-интерфейса пользователя	22

1.5.1.1.2 Описание OUTPUT-интерфейса пользователя	23
1.5.1.2 Описание интерфейса с внешними устройствами (оборудованием)	25
1.5.1.3 Описание программных интерфейсов	25
1.5.1.4 Описание интерфейсов передачи информации	26
1.5.1.5 Описание атрибутов продуктивности	26
2 Планирование процесса разработки программного продукта	26
2.1 Планирование итераций разработки программного продукта	26
2.2 Концептуальное описание архитектуры программного продукта	28
2.3 План разработки программного продукта	28
2.3.1 Оценка трудоемкости разработки программного продукта	28
2.3.2 Определение дерева работ по разработке программного продукта	32
2.3.3 График работ по разработке программного продукта	33
2.3.3.1 Таблица с графиком работ	33
2.3.3.2 Диаграмма Ганта	34
3 Проектирование программного продукта	35
3.1 Концептуальное и логическое проектирование структур данных программного продукта	35
3.1.1 Концептуальное проектирование на основе UML-диаграммы концептуальных классов	35
3.1.2 Логическое проектирование структур данных	35
3.2 Проектирование программных классов	37
3.3 Проектирование алгоритмов работы методов программных классов	37
3.4 Проектирование тестовых наборов методов программных классов	44
4 Конструирование программного продукта	45
4.1 Особенности конструирования структур данных	45
4.1.1 Особенности инсталляции и работы с СУБД	45
4.1.2 Особенности создания структур данных	47
4.2 Особенности конструирования программных модулей	49

4.2.1 Особенности работы с интегрированной средой разработки	49
4.2.4 Особенности разработки алгоритмов методов программных классов или процедур функций	51
4.3 Модульное тестирование программных классов	52
5 Развертывание и валидация программного продукта	56
5.1 Инструкция по установке программного продукта	56
5.2 Инструкция по использованию программного продукта	56
5.3 Результат валидации программного продукта	57
Выводы	58

Дополнение А Структура разделов, подразделов, пунктов и подпунктов пояснительной записки к курсовой работе

1 Требования к программному продукту

1.1 Определение потребностей потребителя

1.1.1 Иерархия потребностей потребителя

Известно, что в теории маркетинга потребности человека могут быть представлены в виде пирамиды потребностей американского психолога Абрахама Маслоу, которая содержит такие ступени:

1. Физиологические потребности
2. Потребность в безопасности
3. Потребность в любви/Принадлежность к чему-либо
4. Потребность в уважении
5. Потребность в познании
6. Эстетические потребности
7. Потребность в самоактуализации

Наш продукт призван помочь человеку в его стремлении к познанию. Мы предлагаем ему удобную организацию его учебных материалов, также он с легкостью может найти новые, интересные ему материалы для изучения с помощью функций нашей программы. Ему будут представлены различные курсы, книги и материалы с форума, и все это совместно с удобной навигацией и поиском.

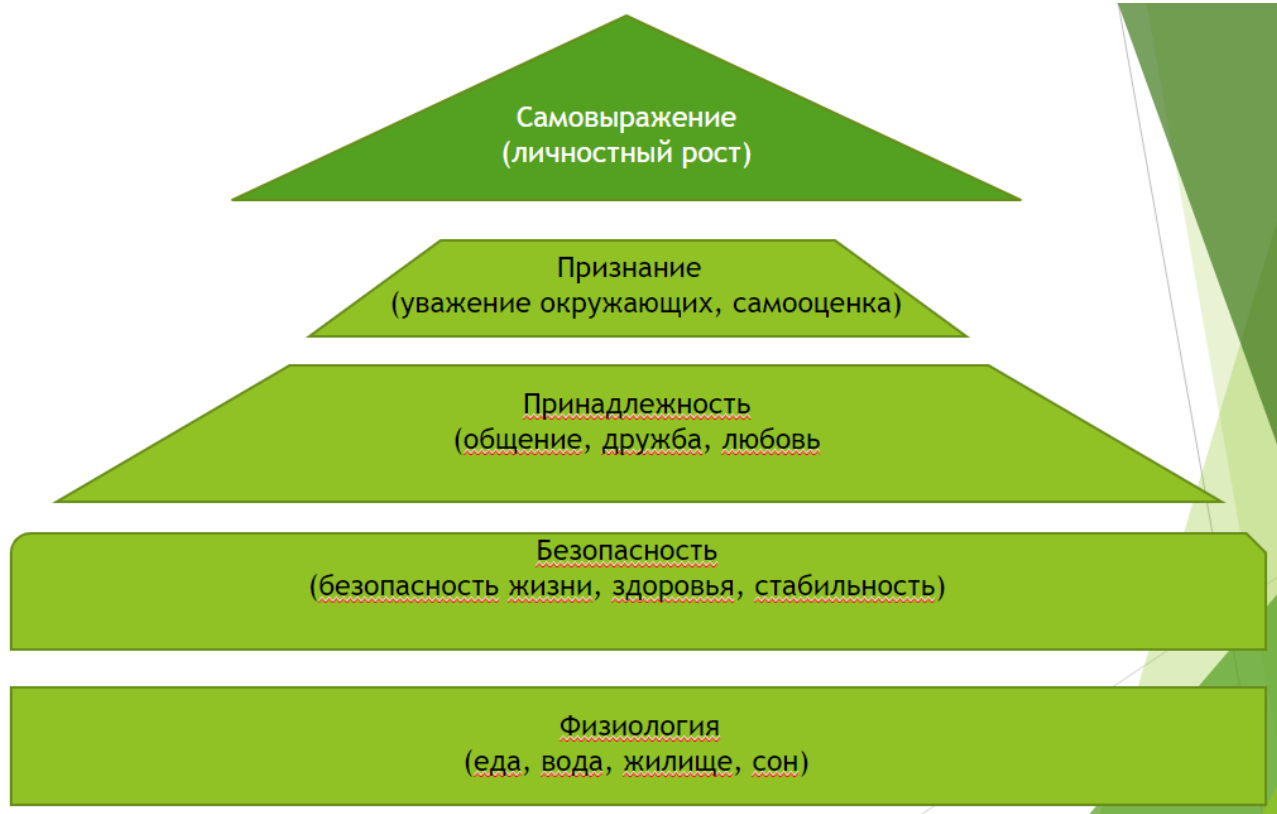


Рисунок 1.1 - пример иерархии нужд потребителя

1.1.2 Детализация материальной потребности



Рисунок 1.1 - MindMap

1.2 Бизнес-требования к программному продукту

1.2.1 Описание проблемы потребителя

1.2.1.1 Концептуальное описание проблемы потребителя

У каждого человека, от школьника до пенсионера есть потребность в познании, изучении чего-то нового, будь то непосредственная потребность или просто желание изучить что-то интересное.

Критерии, благодаря которым наш программный продукт помогает вышеуказанной категории людей восполнить их потребности, или, по крайней мере, облегчить их исполнение:

- 1.) Доступность. Наш продукт предлагает как бесплатные материалы, так и платные, для того, чтобы пользоваться преимуществами нашего ресурса нужна всего лишь регистрация.
- 2.) Продукт графически построен интуитивно понятно и легок в навигации даже не для продвинутого пользователя.
- 3.) Полезность. Наш продукт имеет множество преимуществ и организованную базу данных знаний, чтобы пользователь мог найти то, что его интересует.
- 4.) Актуальность. Наш продукт имеет актуальные знания и может прибавлять их в режиме “живого времени”, т.е. база знаний нашего продукта постоянно пополняется и актуальна.

1.2.1.2 Метрическое описание проблемы потребителя

Общее описание проблемы:

Количество информации, которое находится во всемирной паутине, может ответить на любой возникший у вас вопрос, но эти ответы искать достаточно непросто

Метрические показатели неудовлетворения пользователя:

Количество информации, подтвержденное сотрудниками-администраторами и самими пользователями, как информация, которая действительно решает стоящую перед нами задачу

Уровень полезности для решаемой задачи $NU = NA / N$,

Где NA - количество сообщений(форумы) и обучающих файлов(курсы) которые пользователи отметили как верные и пользуются спросом у пользователей A N - общее количество информации сообщений(форумы) и обучающих файлов(курсы) которое есть на нашем сайте

1.2.2 Цель создания программного управления

1.2.2.1 Проблемный анализ существующих программных продуктов

№	Название ресурса	Стоимость	Уровень готовности	Недостатки ресурса, в сравнении с предполагаемым нашим функционалом
1	Coursera	Бесплатно/Платно	2	Отсутствие форума, где можно было бы задать свой вопрос, когда что-то не получается
2	Stackoverflow	Бесплатно	2	Отсутствие возможности прикреплять

				<p>файлы начиная от средних размеров, и это как плюс, но отсутствие похожих форумов в других отраслях</p>
3	Prometheus	Бесплатно/Платно	3	<p>Отличная платформа, от нашей отличается разве-что наличием собственной библиотеки (в нашей)</p>
4	<p>Электронные библиотеки и платформы для видеозвонков и чатов</p>	Бесплатно/Платно	1	

1.2.2.2 Цель создания программного продукта

Повысить уровень полезности информации на нашем сайте, чтоб во всемирной паутине было место, где вся информация упорядочена и действительно полезна. Уровень полезности информации разного рода на нашем сайте должен стремиться к единице, т.е. все сообщения(форумы) и обучающие файлы(курсы) были полезны и востребованы

1.2.3 Название программного продукта

1.2.3.1 Лозунг программного продукта

«Платформа для обучения - ALLearn».

Это приложение предназначено для того, чтобы каждый пользователь смог найти свое „цифровое золото“, а именно — интересующую его информацию. Обучение еще никогда не было таким затягивающим!

1.2.3.2 Логотип программного продукта

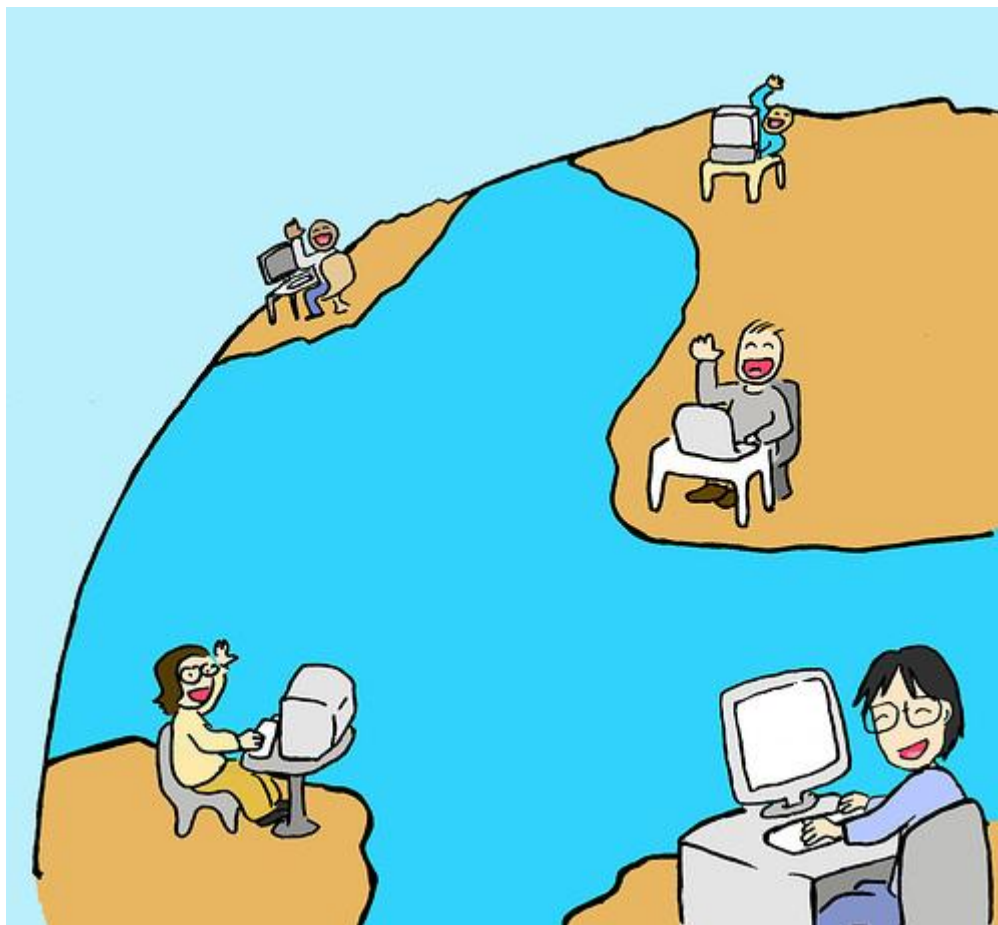


Рисунок 1.2.3 - логотип

Наш логотип отображает, что наша платформа из всей информации всемирной паутины, вмещает в себе лишь самую полезную, объединяя таким образом пользователей общей целью.

1.3 Требования пользователя к программному продукту

1.3.1 История пользователя программного продукта

Создание User Story потребителя ПП

Описание требований по работе ПП для автоматизации работы приложения:

1. Админ(приложение) предоставляет возможности приложения
2. Пользователь получает меню(возможности) приложения
3. Пользователь желает задать вопрос на форуме

4. Сотрудник получает заданный вопрос и отправляет П возможный ответ
5. П получает ответ и остается довольным
6. П хочет получить доступ к платным\бесплатным курсам
7. С принимает просьбу от П и передает ее М
8. М передает информацию А(приложение) и передает купленную информацию П
9. П получает свои купленные материалы

1.3.2 Диаграмма прецедентов программного продукта

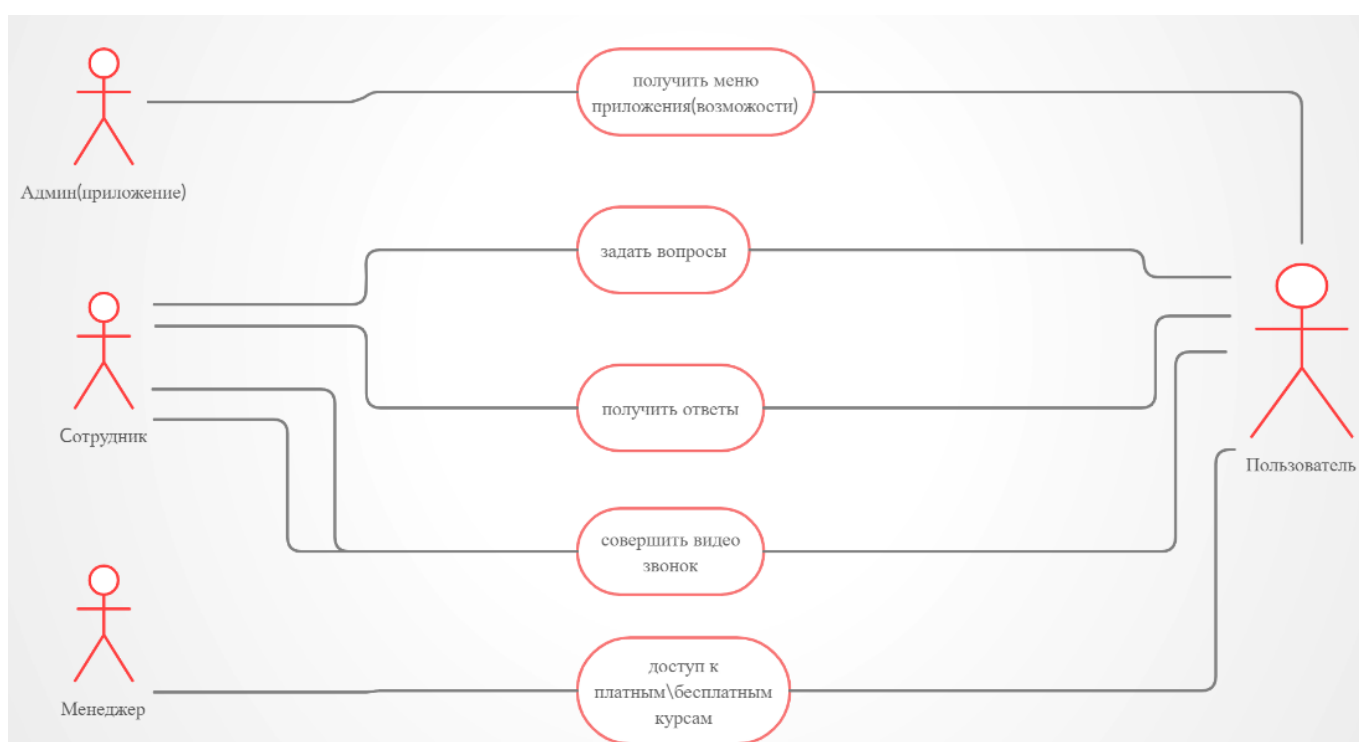


Рисунок 1.3.2 - диаграмма прецедентов

Пример диаграммы прецедентов с учетом работы будущего ПП

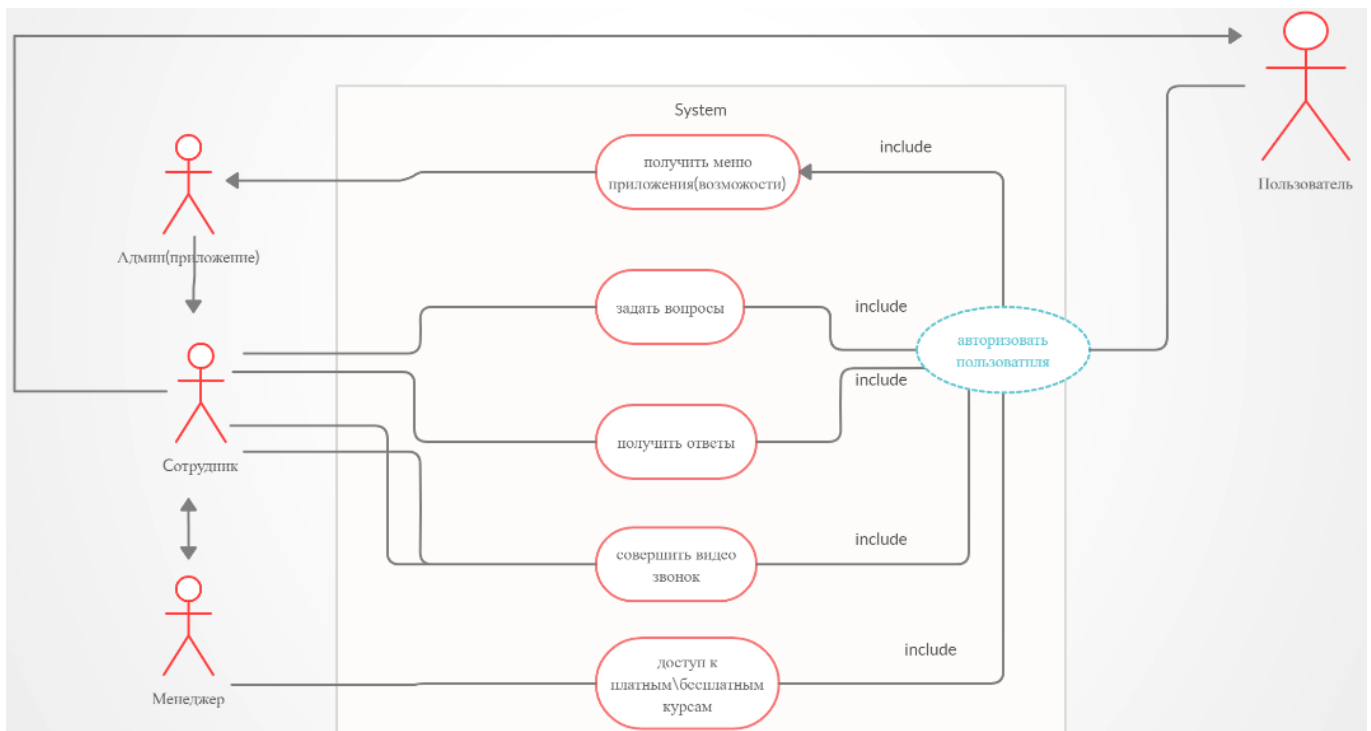


Рисунок 1.3.3 -диаграмма прецедентов

1.3.3 Сценарии исполнения прецедентов программного продукта

Описание сценария прецедента ПП №1

Список сокращений: Программный Продукт(ПП),Пользователь(П).

Пользовательская активность подразумевает под собой таких явлений, как пополнение форума вопросами/ответами/комментариями и тд, отзывы о курсах, книгах, выявление недостатков ПП, обращения в службу поддержки по поводу этих проблем, что позволяет ПП совершенствоваться и тд.

Прецедент «Регистрация»

П заинтересован возможностями, предоставляемыми ПП. П заинтересован в этом, так как посредством регистрации он получит искомый им доступ к средства ПП, а ПП, в свою очередь, получает нового П. П инициирует прецедент регистрации, обращаясь к ПП. В случае успешного выполнения сценария, П получит полноценный доступ к ПП, а ПП получит нового пользователя.

Пример основного успешного сценария прецедента «Регистрация»:

1. ПП запрашивает у пользователя параметры для регистрации (Идентификаторы/Аутентификаторы. иные данные).
2. Пользователь передает ПП свои ИА-параметры и другие данные, требуемые для регистрации.
3. ПП предоставляет пользователю полноценный доступ к другим прецедентам ПП.

Пример альтернативного сценария для предыдущего примера основного успешного сценария прецедента «Регистрация»:

3.1 ПП обнаруживает, что пользователь передал ему некорректные значения ИА-параметров или недопустимые иные данные.

3.а.1 ПП выдает сообщение об ошибке и переходит к шагу 1 основного успешного сценария.

Описание сценария прецедента ПП №2

Прецедент «Авторизация пользователя»

П требуется доступ к ПП. П заинтересован получить нужную ему информацию из ПП. П инициирует обращение к ПП. В случае, если авторизация пройдет успешно, то П получит доступ к необходимым ему ресурсам, а ПП получит взаимосвязь с П.

Пример основного успешного сценария прецедента «Авторизация пользователя»:

1. ПП запрашивает у пользователя параметры авторизации (Идентификаторы/Аутентификаторы).
2. Пользователь передает ПП свои ИА-параметры.
3. ПП предоставляет пользователю доступ к другим прецедентам ПП.

Пример альтернативного сценария для предыдущего примера основного успешного сценария прецедента «Авторизовать пользователя»:

3.1 ПП обнаруживает, что пользователь передал ей неправильные значения ИА-параметров.

3.а.1 ПП выдает сообщение об ошибке и переходит к шагу 1 основного успешного сценария.

Описание сценария прецедента ПП №3

Прецедент «Работа с форумом»

Авторизованный П проявляет интерес к инструментам форума. П заинтересован в получении нужной ему информации с помощью форума, ПП получает пользовательскую активность. П инициирует обращение к инструментам форума у ПП. В случае, если взаимодействие с форумом произойдет успешно, П получит то, что он искал, а ПП получит дополнение к объему информации ресурса.

Пример основного успешного сценария прецедента «Работа с форумом»:

1. ПП запрашивает у пользователя информацию о его вопросе, если П задает его, или ответе, если П отвечает на чей-то вопрос.
2. П передает ПП определенную информацию.
3. ПП публикует информацию П на форуме, что дает П доступ к остальным прецедентам.

Пример альтернативного сценария для предыдущего примера основного успешного сценария прецедента «Работа с форумом»:

3.1 ПП обнаруживает, что П отправил некорректную или недопустимую информацию.

3.а.1 ПП выдает сообщение об ошибке и переходит к шагу 1 основного успешного сценария.

Описание сценария прецедента ПП №4

Прецедент «Получение доступа к курсам(платным/бесплатным)»

Авторизованный П проявляет интерес к курсам. П заинтересован в информации курсов. ПП получает пользовательскую активность и/или денежные средства за покупку курса. П инициирует обращение к курсам у ПП. В случае, если взаимодействие(доступ/покупка) курсов произойдет успешно, П получит то, что ему нужно в виде информации, а ПП получит П активность и/или денежные средства.

Пример основного успешного сценария прецедента «Получение доступа к курсам(платным/бесплатным)»:

1. ПП запрашивает у пользователя интересующие его курсы.
2. ПП предоставляет пользователю список курсов и их условия.
3. П, выбрав нужный ему курс, запрашивает доступ к нему у ПП.
4. ПП предоставляет доступ к курсу П.

Пример альтернативного сценария для предыдущего примера основного успешного сценария прецедента «Получение доступа к курсам(платным/бесплатным)».

- 1.1 П запрашивает несуществующий курс или вводит некорректные данные.
 - 1.а.1 ПП выдает сообщение об ошибке и переходит к шагу 1 основного успешного сценария.
- 3.1 Если П выбрал платный курс, к которому у него еще нет доступа, ПП отправляет инструкции П для оплаты.
 - 3.а.1 П производит оплату.
 - 3.а.2 ПП предоставляет доступ к курсу.

Описание сценария прецедента ПП №5

Прецедент «Получение доступа к электронной библиотеке»

Авторизованный П проявляет интерес к электронной библиотеке. П заинтересован в информации электронной библиотеки. ПП получает пользовательскую активность. П инициирует обращение к электронной библиотеке у ПП. В случае, если действие пройдет успешно, П получит нужную ему информацию в электронной библиотеке, а ПП получит П активность.

Пример основного успешного сценария прецедента «Получение доступа к электронной библиотеке»:

1. ПП запрашивает у пользователя интересующую его литературу.
2. ПП предоставляет пользователю найденную по запросу литературу П.
3. П, выбрав нужную ему литературу, запрашивает ее у ПП.
4. П получает от ПП литературу в электронном виде.

Пример альтернативного сценария для предыдущего примера основного успешного сценария прецедента «Получения доступа к электронной библиотеке».

- 1.1 П запрашивает несуществующую литературу или вводит некорректные данные.
- 1.а.1 ПП выдает сообщение об ошибке и переходит к шагу 1 основного успешного сценария.

Описание сценария прецедента ПП №6

Прецедент «Совершение видео-звонков»

Авторизованный П проявляет интерес в совершении видеозвонков. П заинтересован в звонках через видео-связь. ПП получает пользовательскую активность. П инициирует запрос к ПП для установления видео-связь с интересующим его П. В случае, если действие пройдет успешно, П получит успешную видео-связь с другим П, а ПП получит П активность.

Пример основного успешного сценария прецедента «Совершение видео-звонков»:

1. П отправляет запрос с ПП с просьбой установить связь с другим П.

2. ПП налаживает связь между двумя П.

Пример альтернативного сценария для предыдущего примера основного успешного сценария прецедента «Совершение видео-звонков».

1.1. П получает уведомление от ПП, что он получает входящий звонок.

1.a.1. ПП налаживает связь между двумя П, если оба согласны на то.

1.a.2. ПП не налаживает связь между П, в случае, если кто либо из них не отклонил.

2.1 ПП не удалось установить связь, П получает сообщение об ошибке и переходит к шагу 1 основного успешного сценария.

1.4 Функциональные требования к программному продукту

1.4.1. Многоуровневая классификация функциональных требований

Таблица 1 - описание функций с присвоением уникальных иерархических идентификаторов

Идентификатор функции	Название функции
FR1	Регистрация пользователя
FR1.1	Создание запроса к пользователю для получения его параметров для идентификации и аутентификации
FR1.2	Передача от пользователя его ИА-параметров
FR1.3	Передача пользователю доступа
FR1.3.1	Вывод информации об ошибке, возвращение его к первому шагу
FR2	Авторизация пользователя
FR2.1	Создание запроса к пользователю для получения его

	ИА-параметров для авторизации
FR2.2	Передача от пользователя его ИА-параметров
FR2.2.1	Вывод информации об ошибке, возвращение его к первому шагу
FR3	Работа с форумом
FR3.1	Создание запроса к пользователю для получения информации о вопросе или ответе
FR3.2	Передача от пользователя запрашиваемой его информации
FR3.3	Передача данных пользователя на форум
FR3.3.1	Вывод информации об ошибке, возвращение его к первому шагу
FR4	Получение доступа к курсам(платным/бесплатным)
FR4.1	Создание запроса к пользователю для получения его предпочтений
FR4.1.1	Передача от пользователя его предпочтений
FR4.2	Вывод информации об ошибке, возвращение его к первому шагу
FR4.3	Передача пользователю информации на основании его предпочтений
FR4.3.1	Создание запроса от пользователя на получение информации
FR4.3.2	Передача пользователю инструкций и создание запроса на оплату
FR4.3.3	Передача от пользователя оплаты
FR4.3.4	Передача данных пользователю
FR5	Получение доступа к электронной библиотеке
FR5.1	Создание запроса к пользователю для спецификации его потребностей
FR5.2	Передача от пользователя данных
FR5.3	Передача пользователю информации на основании его данных
FR5.4	Создание запроса от пользователя на получение информации

FR5.5	Передача данных пользователю
FR6	Совершение видео-звонков
FR6.1	Пользователь создает запрос на установление связи
FR6.1.1	Выделение канала связи для пользователя
FR6.2	Вывод информации об ошибке, возвращение его к первому шагу

Описание иерархической WBS-структуры многоуровневой классификации функциональных потребностей.

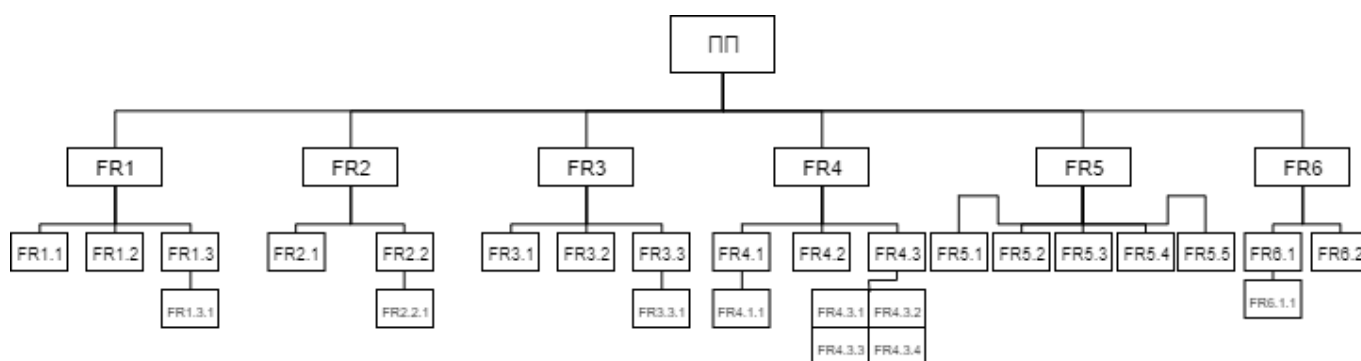


Рисунок 1.4.1 - WBS-структура многоуровневой классификации функциональных потребностей

1.4.2 Функциональный анализ существующих продуктов

Идентификатор функции	iSpring	GetCourse	We Study	Описание
FR1	+	+	+	На всех ресурсах доступна регистрация
FR2	+	+	+	На всех ресурсах доступна регистрация
FR3	+	-	-	Работа с

				форумом доступна только на iSpring
FR4	++/-	+/-	+/-	Курсы существуют на всех приведенных платформах, в основном на них находятся платные курсы и лишь несколько бесплатных
FR5	-	-	-	Электронной библиотеке нет ни на одном из приведенных ресурсов
FR6	-	+	+	Совершение видеозвонков недоступно только на первом ресурсе

1.5 Нефункциональные требования к программному продукту

1.5.1 Описание внешних интерфейсов

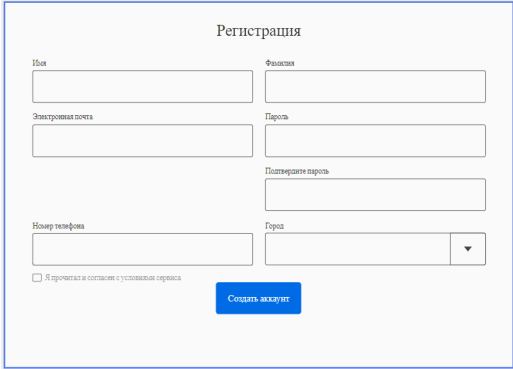
1.5.1.1 Описание интерфейса пользователя

1.5.1.1.1 Описание INPUT-интерфейса пользователя

Идентификатор функции	Способ INPUT-потока	Особенности исполнения
FR1	стандартная компьютерная	

(регистрация пользователя)	клавиатура / 2/3-кнопочный манипулятор типа "мышь"	
FR2 (авторизация пользователя)	стандартная компьютерная клавиатура / 2/3-кнопочный манипулятор типа "мышь"	
FR3 (работа с форумом)	стандартная компьютерная клавиатура / 2/3-кнопочный манипулятор типа "мышь"	
FR4 (Получение доступа к курсам(платным/бесплатным))	2/3-кнопочный манипулятор типа "мышь" / 2/3-кнопочный манипулятор типа "мышь"	
FR5 (Получение доступа к электронной библиотеке)	2/3-кнопочный манипулятор типа "мышь" / 2/3-кнопочный манипулятор типа "мышь"	
FR6 (Совершение видео-звонков)	2/3-кнопочный манипулятор типа "мышь" / 2/3-кнопочный манипулятор типа "мышь"	

1.5.1.1.2 Описание OUTPUT-интерфейса пользователя

Идентификатор функции	Способ OUTPUT-потока	Особенности исполнения
FR1 (регистрация пользователя)	Графический интерфейс	

FR1.2	Графический интерфейс	<div> <div>Регистрация</div> <div> <div>Имя</div> <div>Петр</div> </div> <div> <div>Фамилия</div> <div>Чайковский</div> </div> <div> <div>Электронная почта</div> <div>tchaikovsky@gmail.com</div> </div> <div> <div>Пароль</div> <div>Ptichka28</div> </div> <div> <div>Подтвердите пароль</div> <div>Ptichka28</div> </div> <div> <div>Номер телефона</div> <div>+3806688362</div> </div> <div> <div>Город</div> <div>Алушта</div> </div> <div> <input type="checkbox"/> Я прочитал и согласен с условиями сервиса </div> <div>Создать аккаунт</div> </div>
-------	-----------------------	---

1.5.1.2 Описание интерфейса с внешними устройствами (оборудованием)

Идентификатор функции	Внешнее устройство
FR1	Desktop-персональный компьютер/Notebook/смартфон/приборы печати;
FR2	
FR3	
FR4	
FR5	
FR6	

1.5.1.3 Описание программных интерфейсов

Версии операционных систем и библиотек, которые понадобятся для реализации большинства функций ПП:

- Windows 10/7/Vista/XP
- Mac OS
- Android
- IOS
- Linux
- Windows Phone

1.5.1.4 Описание интерфейсов передачи информации

Интерфейсы передачи информации, которые понадобятся при реализации большинства функций ПП.

- Internet/Wi-fi

1.5.1.5 Описание атрибутов продуктивности

Идентификатор функции	Максимальное время реакции ПП на действия пользователя, секунды
FR1	2
FR2	2
FR3	2
FR4	2
FR5	2
FR6	2

2 Планирование процесса разработки программного продукта

2.1 Планирование итераций разработки программного продукта

Таблица 2 – описание функциональных приоритетов

Идентификатор функции	Функциональные зависимости	Влияние на достижение цели, %	Приоритет функции
FR1	-	0	М
FR1.1	-	0	М
FR1.2	-	0	М

FR1.3	-	0	M
FR1.3.1	-	0	M
FR2	FR1	0	M
FR2.1	FR2	0	M
FR2.2	FR2	0	M
FR2.2.1	FR2	0	M
FR3	FR2	5	W
FR3.1	FR3	5	W
FR3.2	FR3	5	W
FR3.3	FR3	5	W
FR3.3.1	FR3	5	W
FR4	FR2	3	S
FR4.1	FR4	3	S
FR4.1.1	FR4	3	S
FR4.2	FR4	3	S
FR4.3	FR4	3	S
FR4.3.1	FR4	3	S
FR4.3.2	FR4	3	S
FR4.3.3	FR4	3	S
FR4.3.4	FR4	3	S
FR5	FR2	5	M
FR5.1	FR5	5	M
FR5.2	FR5	5	M
FR5.3	FR5	5	M
FR5.4	FR5	5	M

FR5.5	FR5	5	M
FR6	FR2	9	W
FR6.1	FR6	9	W

2.2 Концептуальное описание архитектуры программного продукта

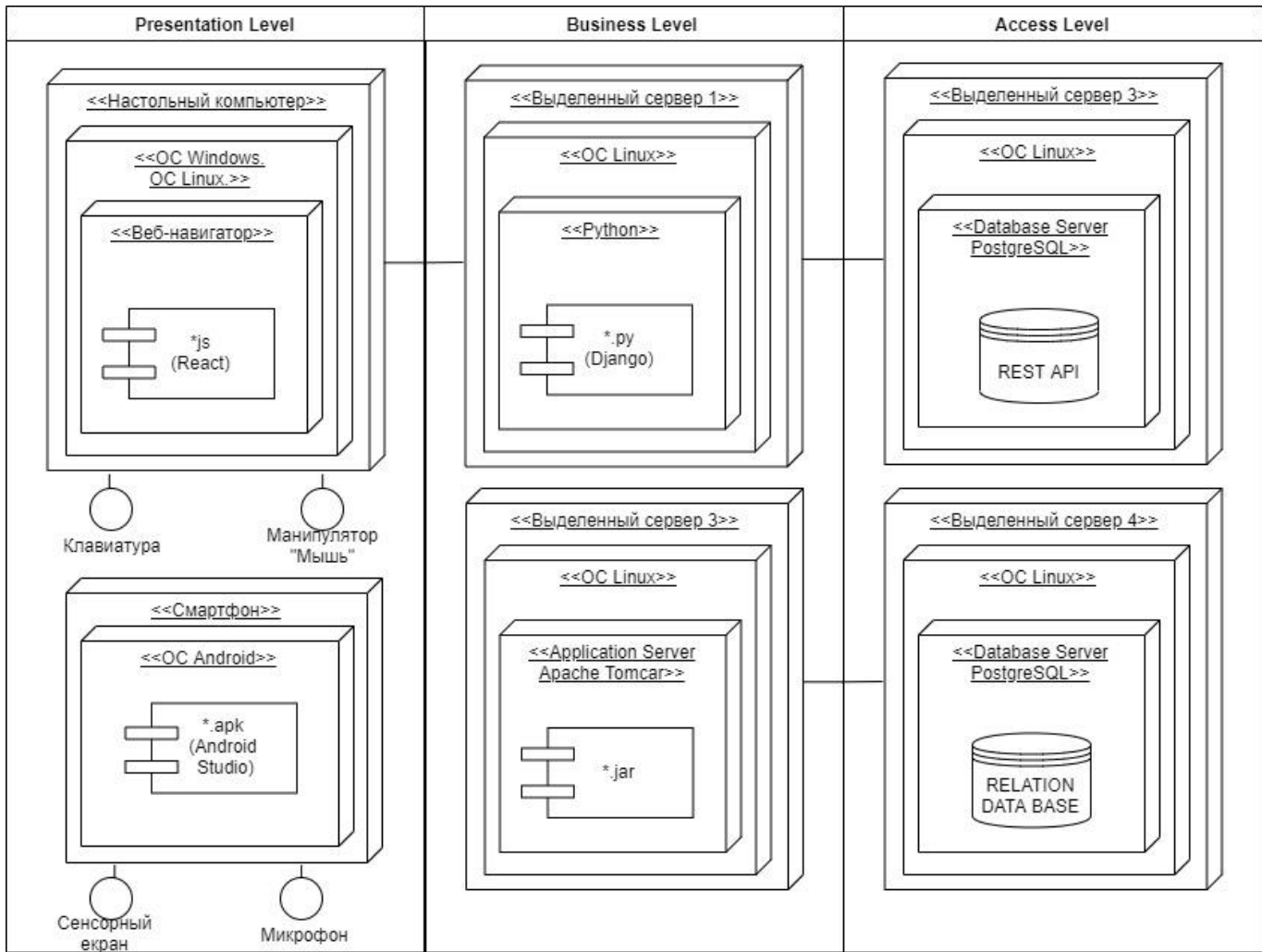


Рисунок 2.2 - UML-диаграмма развертывания ПП

2.3 План разработки программного продукта

2.3.1 Оценка трудоемкости разработки программного продукта

Все прецеденты делятся на три типа; простые, средние и сложные в зависимости от количества шагов успешных сценариев (основных и альтернативных). Общее количество прецедентов каждого типа умножается на соответствующий весовой коэффициент, затем вычисляется общий весовой показатель

Весовые коэффициенты прецедентов

Сложность прецедента	Количество транзакций	Фактор веса
Простой	≤ 3	5
Средний	4-7	10
Сложный	≥ 7	15

$$UUCP=A+UC=70+9=79$$

Все акторы делятся на три типа: простые, средние и сложные. Простой актор представляет внешнюю систему с четко определенным программным интерфейсом. Средний актор представляет или внешнюю систему, взаимодействующую с ПП с помощью сетевых протоколов, или личность, пользуется текстовым интерфейсом (например, алфавитно-цифровым терминалом). Сложный актор представляет личность, которая пользуется графическим интерфейсом. Общее количество акторов каждого типа умножается на соответствующий весовой коэффициент, затем вычисляется общий весовой показатель.

Весовые коэффициенты акторов

Тип актора	Весовой коэффициент
Простой	1
Средний	2

Сложный	3
---------	---

Техническая сложность проекта (TCF - Technical Complexity Factor) вычисляется с учетом показателей технической сложности. Каждому показателю присваивается значение ST_i в диапазоне от 0 до 5: 0 означает отсутствие значимости показателя для данного проекта, 5 - высокую значимость). Значение TCF вычисляется по формуле - $TCF = 0,6 + (0,01 * (ST_i * Вага_i))$

TCF-таблица

Показатель	Описание показателя	Вес
T1	Распределенная система	2
T2	Высокая производительность (пропускная способность)	1
T3	Работа конечных пользователей в режиме он-лайн	1
T4	Сложная обработка данных	-1
T5	Повторное использование кода	1
T6	Простота установки	0.5
T7	Простота использования	0.5
T8	Переносимость	2
T9	Простота внесения изменений	1
T10	Параллелизм	1
T11	Специальные требования к безопасности	1
T12	Непосредственный доступ к системе со стороны внешних пользователей	1
T13	Специальные требования к обучению пользователей	1

$TCF = 0.98$

Определение уровня квалификации работников

Уровень квалификации разработчиков (EF - Environmental Factor)

Показатель квалификации разработчиков

Показатель	Описание показателя	Вес
F1	Знакомая технология	1.5
F2	Опыт разработки приложений	0.5
F3	Опыт использования объектно-ориентированного подхода	1
F4	Наличие ведущего аналитика	0.5
F5	Мотивация	1
F6	Стабильность требований	2
F7	Частичная занятость	-1
F8	Сложные языки программирования	-1

$EF = 1.04$ $UCP = 80.50$

Значения компонентов для определения общего количества баллов вариантов использования

Фактор	Описание	Вес
UUCP	Unadjusted Use Case	79
TCF	Technical Complexity Factor	0.98
EF	Environmental Factor	1.04
UCP	Use Case numberInputs	80.50

2.3.2 Определение дерева работ по разработке программного продукта

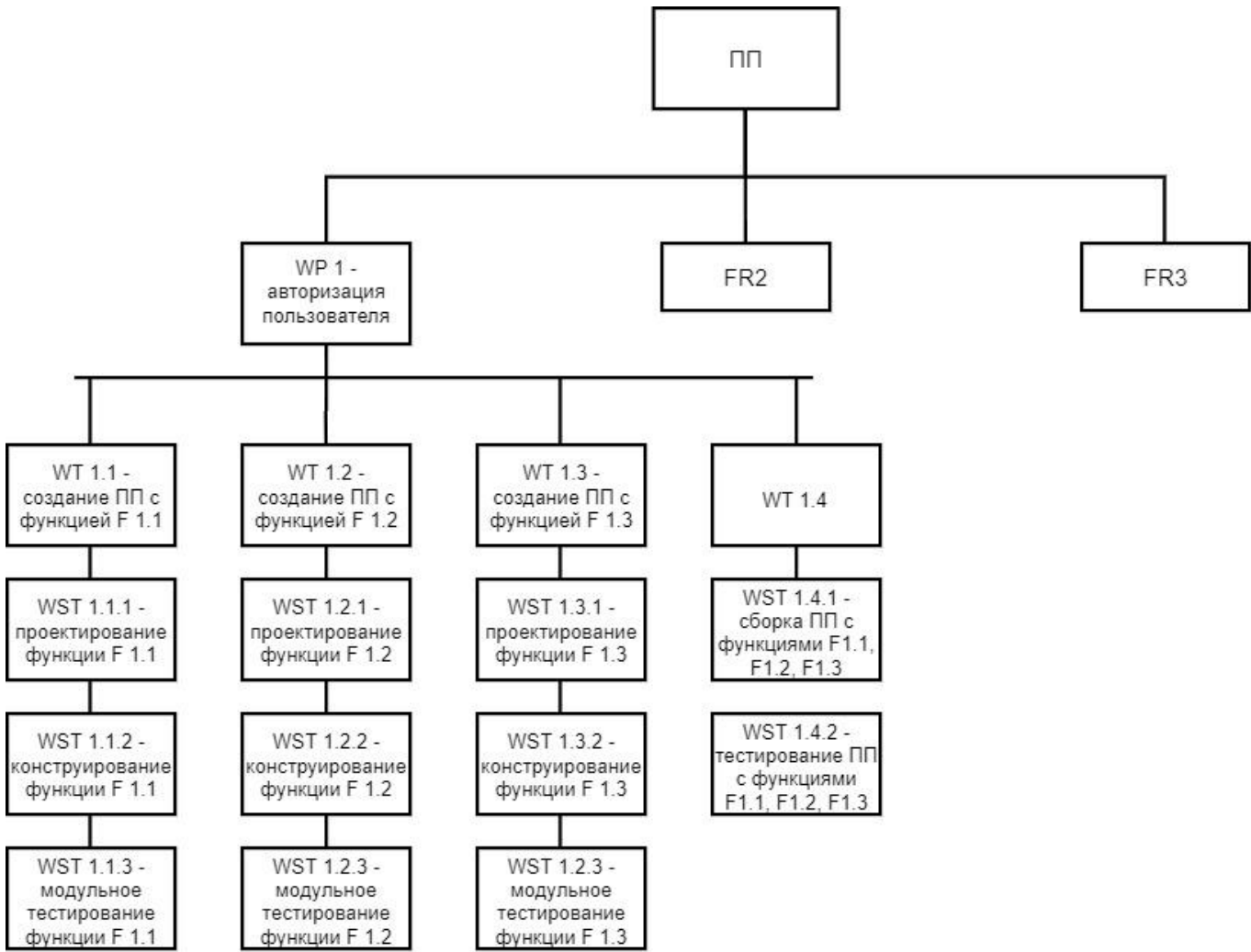


Рисунок 2.3.2 - дерево работ WBS

Таблица 5 пример описания подзадач с прикреплением исполнителей

Подзадача	Исполнитель
WST1.1.1	Бектимиров Р.З.
WST1.1.2	Бектимиров Р.З.
WST1.1.3	Бектимиров Р.З.

WST1.2.1	Шаргородский В.А.
WST1.2.2	Шаргородский В.А.
WST1.2.3	Шаргородский В.А.
WST1.3.1	Гараба Я.
WST1.3.2	Гараба Я.
WST1.3.3	Гараба Я.
WST1.4.1	Бектимиров Р.З.
WST1.4.2	Бектимиров Р.З.

2.3.3 График работ по разработке программного продукта

2.3.3.1 Таблица с графиком работ

Таблица 6 график работ

WST	Дата начала	Дни	Дата завершения	Исполнитель
1.1.1	01.10.2020	1	01.10.2020	Бектимиров Р.З.
1.1.2	02.10.2020	2	03.10.2020	Бектимиров Р.З.
1.1.3	04.10.2020	1	04.10.2020	Бектимиров Р.З.
1.2.1	01.10.2020	1	01.10.2020	Шаргородский В.А.
1.2.2	02.10.2020	2	03.10.2020	Шаргородский В.А.
1.2.3	04.10.2020	1	04.10.2020	Шаргородский В.А.
1.3.1	01.10.2020	1	01.10.2020	Гараба Я.
1.3.2	02.10.2020	2	03.10.2020	Гараба Я.
1.3.3	04.10.2020	1	04.10.2020	Гараба Я.
1.4.1	01.10.2020	1	05.10.2020	Бектимиров Р.З.
1.4.2	02.10.2020	1	06.10.2020	Бектимиров Р.З.

2.3.3.2 Диаграмма Ганта

	A	B	C	D	E	F	G	H	I	J	K
1	WST	Дата начала	Дни	Дата завершени	Исполнитель	01.10.2020	02.10.2020	03.10.2020	04.10.2020	05.10.2020	06.10.2020
2	1.1.1	01.10.2020	1	01.10.2020	Бектимиров Р.З.						
3	1.1.2	02.10.2020	2	03.10.2020	Бектимиров Р.З.						
4	1.1.3	04.10.2020	1	04.10.2020	Бектимиров Р.З.						
5	1.2.1	01.10.2020	1	01.10.2020	Шаргородский В.А.						
6	1.2.2	02.10.2020	2	03.10.2020	Шаргородский В.А.						
7	1.2.3	04.10.2020	1	04.10.2020	Шаргородский В.А.						
8	1.3.1	01.10.2020	1	01.10.2020	Гараба Я.						
9	1.3.2	02.10.2020	2	03.10.2020	Гараба Я.						
10	1.3.3	04.10.2020	1	04.10.2020	Гараба Я.						
11	1.4.1	01.10.2020	1	01.10.2020	Максимова А.Е.						
12	1.4.2	02.10.2020	1	02.10.2020	Максимова А.Е.						

Рисунок 2.3.3 - Диаграмма Ганта

3 Проектирование программного продукта

3.1 Концептуальное и логическое проектирование структур данных программного продукта

3.1.1 Концептуальное проектирование на основе UML-диаграммы концептуальных классов

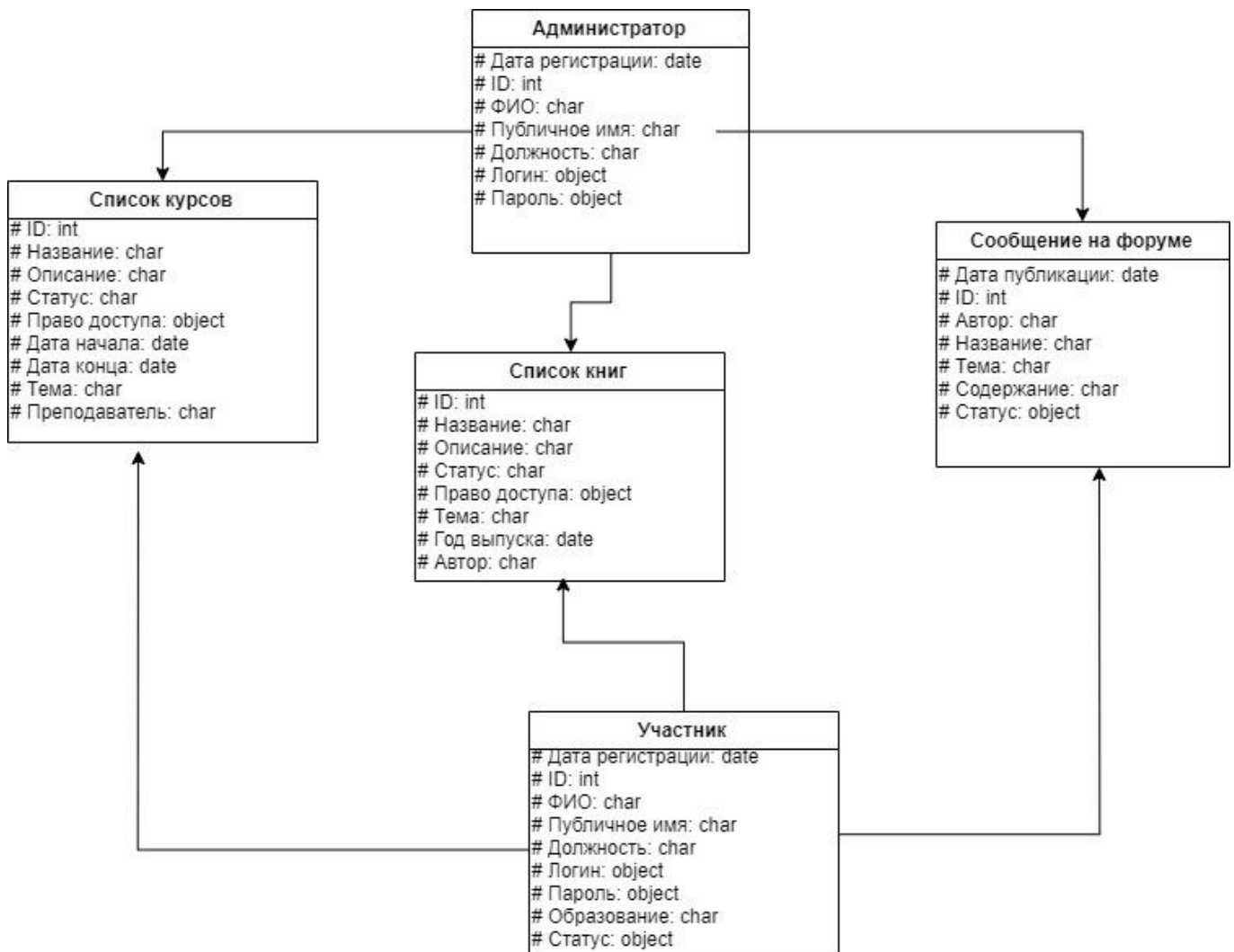


Рисунок 3.1.1 - UML-диаграмма концептуальных классов

3.1.2 Логическое проектирование структур данных

Auth		
ID	INTEGER	(PK)
Login	CHARACTER(40)	
Password	CHAR(40)	

User_Role		
ID	INTEGER	(PK)
Role_Name	CHAR(40)	
Acces_Granted	INTEGER	

Library		
ID	INTEGER	(PK)
Name	CHAR(40)	
Description	CHAR(40)	
Status	CHAR(40)	
Subject	CHAR(40)	
Publish_Date	DATE	
Author	CHAR(40)	

Courses		
ID	INTEGER	(PK)
Name	CHAR(40)	
Description	CHAR(40)	
Status	CHAR(40)	
Subject	CHAR(40)	
Publish_Date	DATE	
Author	CHAR(40)	
Teacher	CHAR(40)	
Acces_rule	CHAR(40)	

User		
ID	INTEGER	(PK)
Public_Name	CHARACTER(40)	
Login_Name	CHAR(40)	
Password	CHAR(40)	
Email	CHAR(40)	
Role_Given	CHAR(40)	

Forum_Message		
ID	INTEGER	(PK)
Author	CHAR(40)	
Name	CHAR(40)	
Topic	CHAR(40)	
Subject	CHAR(40)	
Content	CHAR(4000)	
Status	CHAR(40)	
Publish_date	DATE	

Рисунок 3.1.2 - UML-диаграмма структурных классов

3.2 Проектирование программных классов

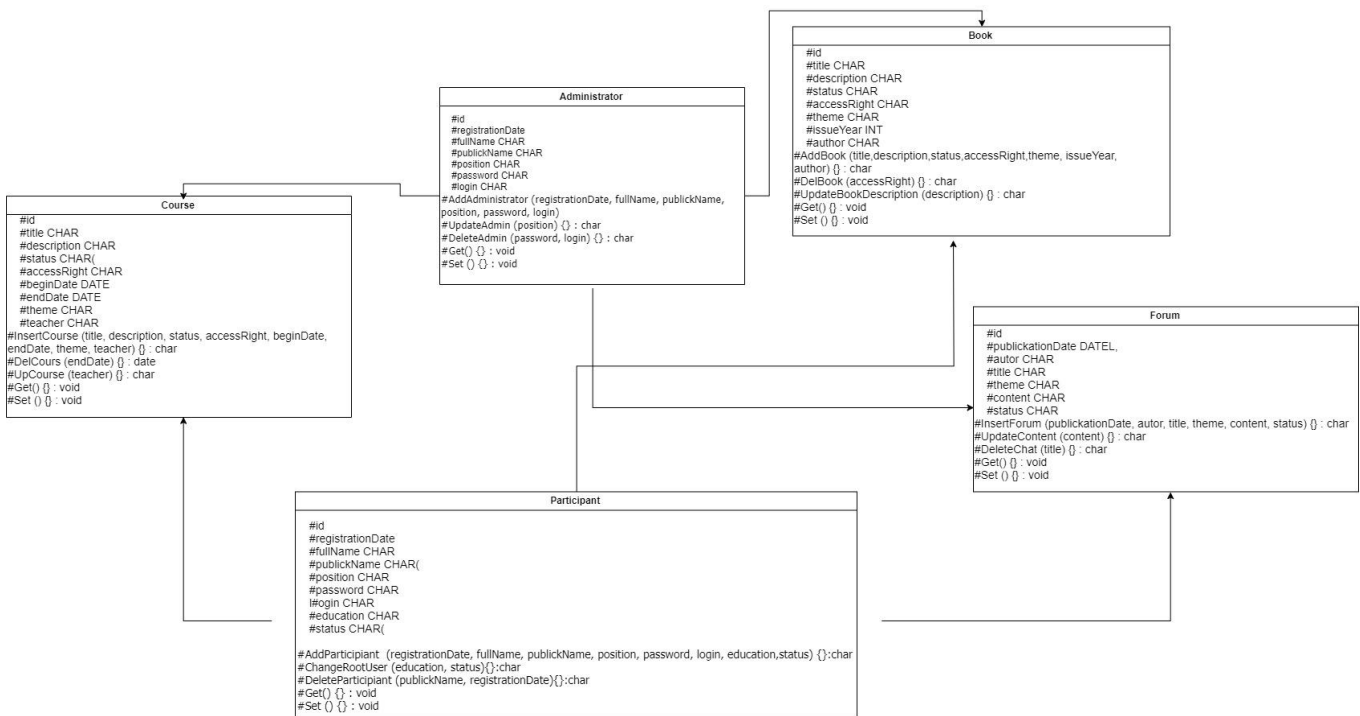


Рисунок 3.2 - Диаграмма программных классов

3.3 Проектирование алгоритмов работы методов программных классов

Алгоритм Search_All_Accessible_Courses изображён на рисунке 3.3.1

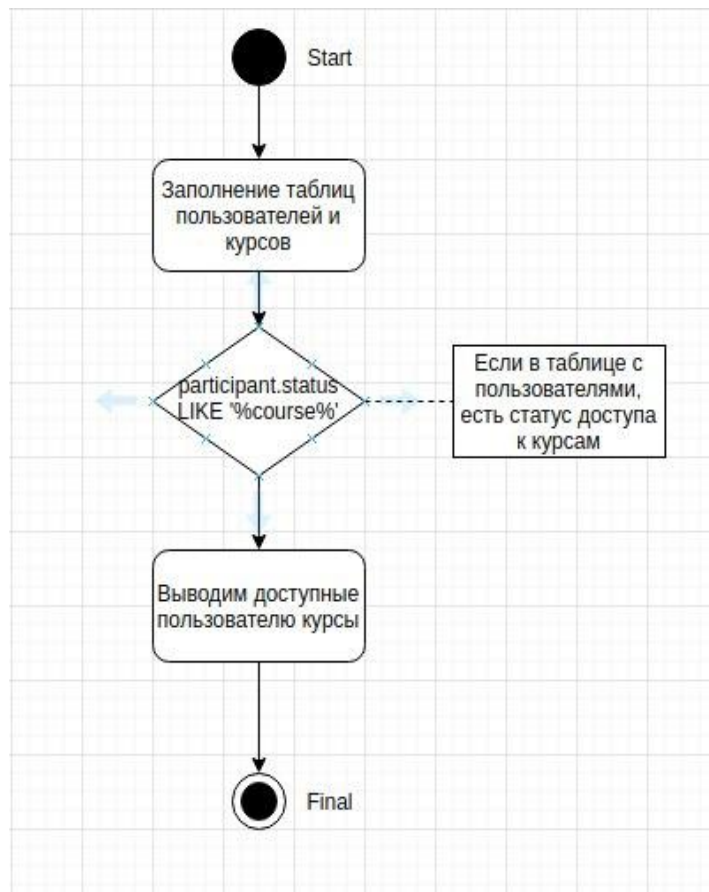


Рисунок 3.3.1 - Алгоритм класса Search_All_Accessible_Courses

Алгоритм Most_Recent_Message изображен на рисунке 3.3.2



Рисунок 3.3.2 - Алгоритм класса `Most_Recent_Message`

Алгоритм `Forgot_Login` изображен на рисунке 3.3.3

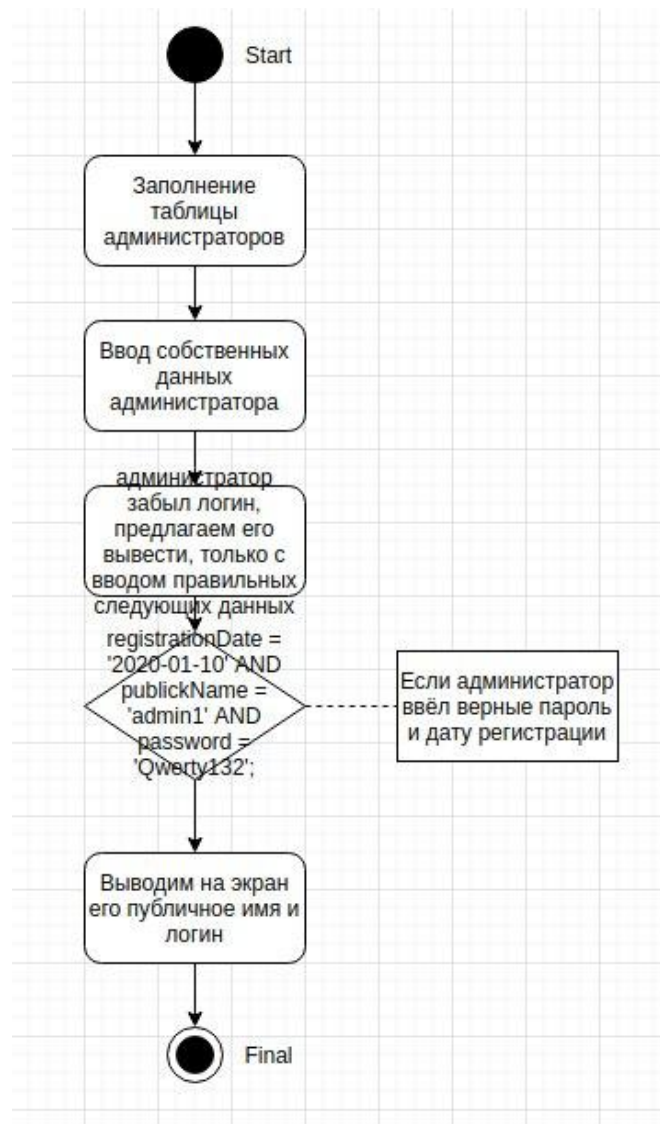


Рисунок 3.3.3 - Алгоритм класса Forgot_Login

Алгоритм Search_Book_By_Author изображен на рисунке 3.3.4

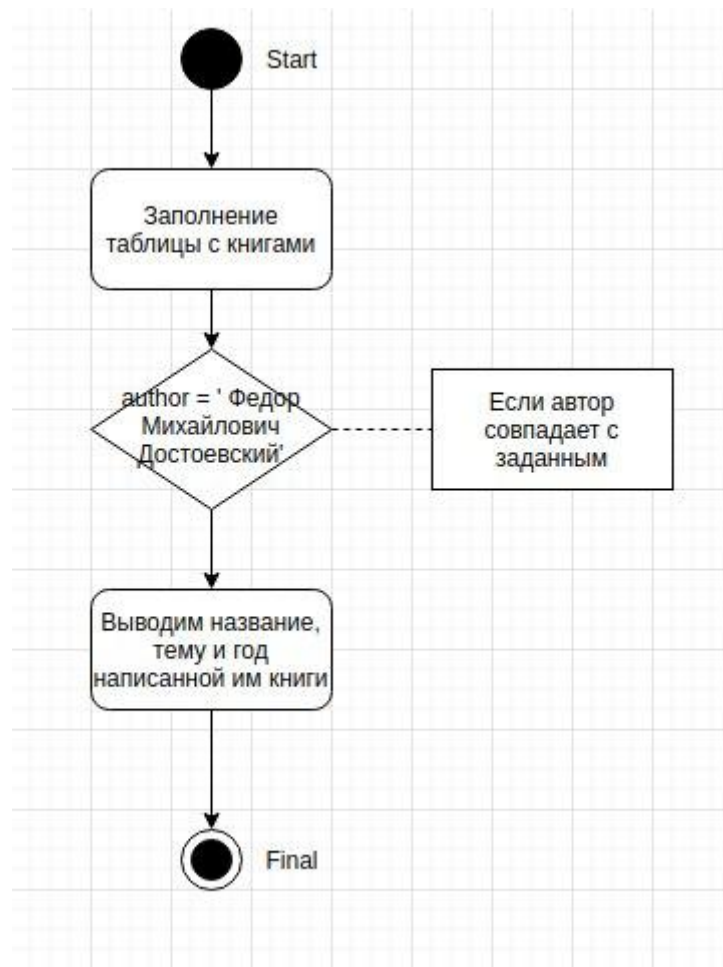


Рисунок 3.3.4 - Алгоритм класса Search_Book_By_Author

Алгоритм Find_Book_Description изображен на рисунке 3.3.5

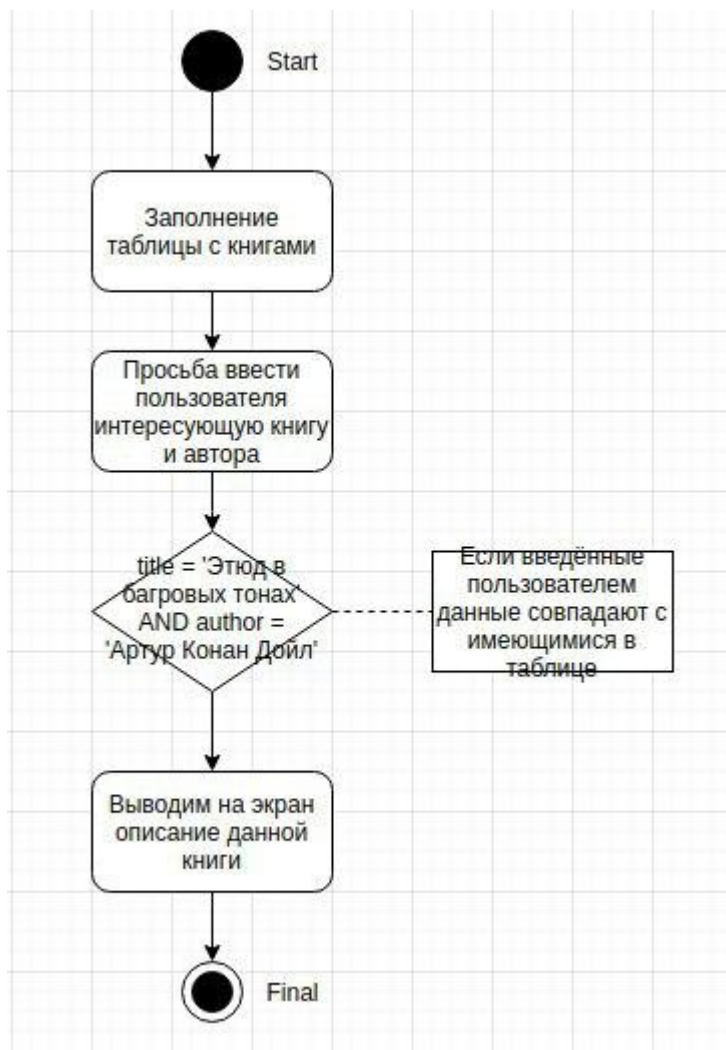


Рисунок 3.3.5 - Алгоритм класса Find_Book_Description

Алгоритм Give_Privilige изображён на рисунке 3.3.6

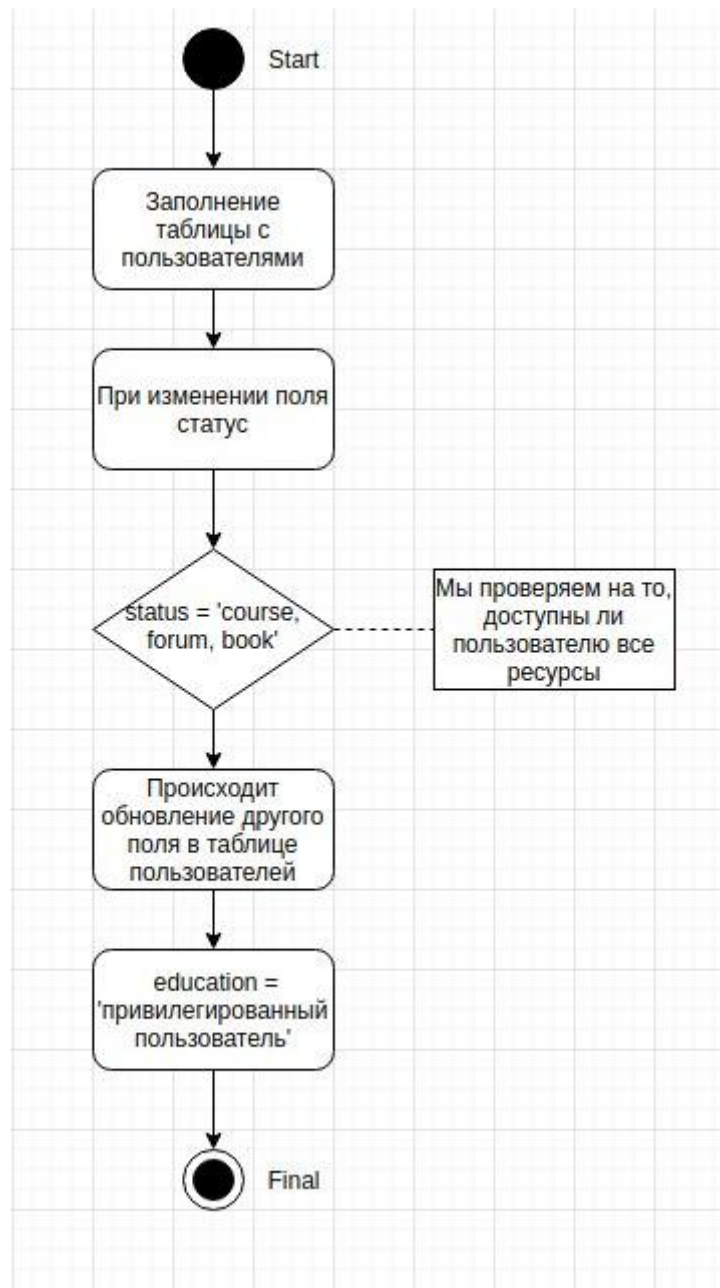


Рисунок 3.3.6 - Алгоритм класса Give_Privilege

Алгоритм Delete_User изображен на рисунке 3.3.7

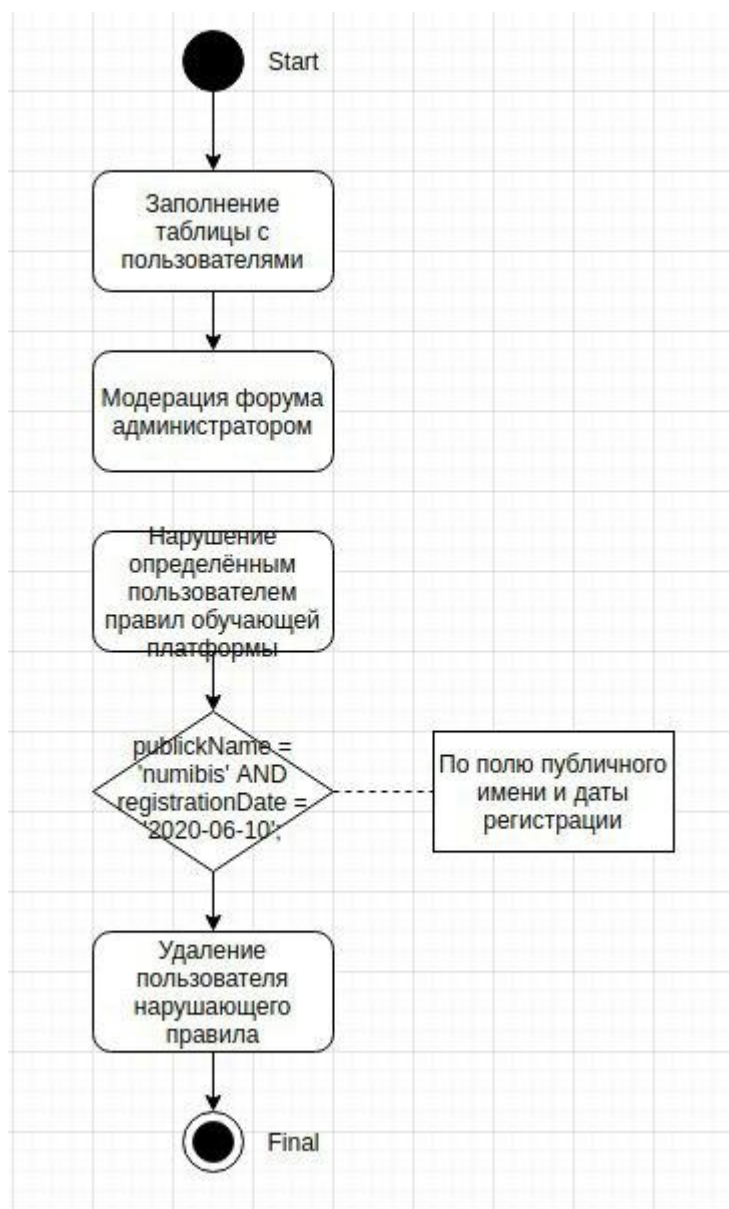


Рисунок 3.3.6 - Алгоритм класса Delete_User

3.4 Проектирование тестовых наборов методов программных классов

Таблица 3.4.1 – Наборы тестирования.

Название функции	№ Теста	Описание входных значений	Описание ожидаемых значений результата
------------------	---------	---------------------------	--

Авторизация1	1	Логин - Kasra Пароль - пусто	Ошибка
Авторизация2	2	Логин - пусто Пароль - Kasra	Ошибка
Авторизация3	3	Логин - Kasra Пароль - Kasra	Переход на интерфейс для авторизации администратора
Отображение информации	4	Запрос к БД, нажатие на вкладку с нужной информацией	Вывод информации на экран
Добавление информации в БД	5	Заполнение всех полей и обращение к БД	Успешная запись в БД
Удаление из БД1	6	Идентификатор удаляемого объекта	Успешное удаление из БД
Удаление из БД2	7	-	Ошибка
Изменение информации в БД	8	Внесение изменений в необходимые поля	Успешное внесение изменений в БД

4 Конструирование программного продукта

4.1 Особенности конструирования структур данных

4.1.1 Особенности инсталляции и работы с СУБД

PostgreSQL — свободная объектно-реляционная система управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

Для отслеживания процесса разработки была использована платформа PgAdmin4. pgAdmin – это открытая платформа администрирования и разработки для PostgreSQL и связанных с ней систем управления базами данных. Написанная на Python и jQuery, эта платформа поддерживает все функции PostgreSQL. Вы можете использовать pgAdmin для выполнения любых задач: от написания базовых

SQL-запросов до мониторинга ваших баз данных и настройки расширенных архитектур БД.

Для установки СКБД PostgreSQL необходимо перейти по ссылке:

<https://www.postgresql.org/download/>

Установка осуществляется почти на любую операционную систему с официально сайта [postgresql.org](https://www.postgresql.org) , так же есть возможность установки с помощью snap-пакетов, проект также можно клонировать с git branch: git.postgresql.org.

Downloads

PostgreSQL Downloads

PostgreSQL is available for download as ready-to-use packages or installers for various platforms, as well as a source code archive if you want to build it yourself.

Packages and Installers

Select your operating system family:



Select your Linux distribution:



Source code

The source code can be found in the main [file browser](#) or you can access the source control repository directly at git.postgresql.org. Instructions for building from source can be found in the [documentation](#).

Beta/RC Releases and development snapshots (unstable)

There are source code and binary **packages** of beta and release candidates, and of the current development code available for testing and evaluation of new features. Note that these builds should be used **for testing purposes only**, and not for production systems.

Рисунок 4.1.1.1 - Вид установки

4.1.2 Особенности создания структур данных

Dashboard Properties SQL Statistics Dependencies Dependents

```
1 -- Table: public.book
2
3 -- DROP TABLE public.book;
4
5 CREATE TABLE public.book
6 (
7     id integer NOT NULL DEFAULT nextval('book_id_seq'::regclass),
8     title character(40) COLLATE pg_catalog."default" NOT NULL,
9     description character(1000) COLLATE pg_catalog."default",
10    status character(20) COLLATE pg_catalog."default" NOT NULL,
11    accessright character(30) COLLATE pg_catalog."default" NOT NULL,
12    theme character(700) COLLATE pg_catalog."default",
13    issueyear integer NOT NULL,
14    author character(50) COLLATE pg_catalog."default" NOT NULL,
15    CONSTRAINT book_pkey PRIMARY KEY (id)
16 )
17
18 TABLESPACE pg_default;
19
20 ALTER TABLE public.book
21     OWNER to space;
```

Рисунок 4.1.2.1 - Создание таблицы SQL-запросом

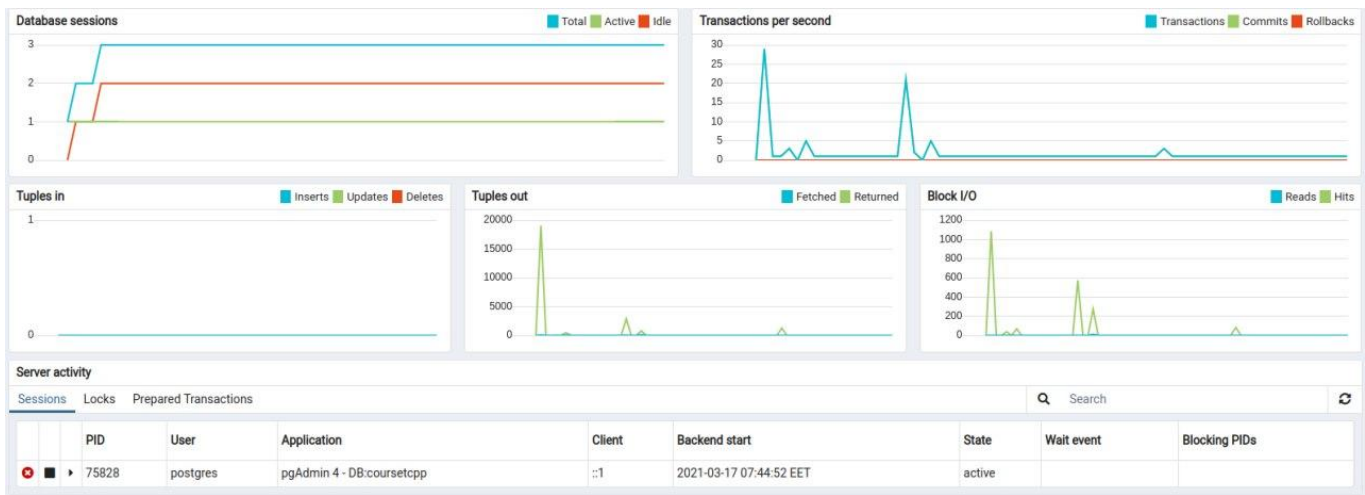
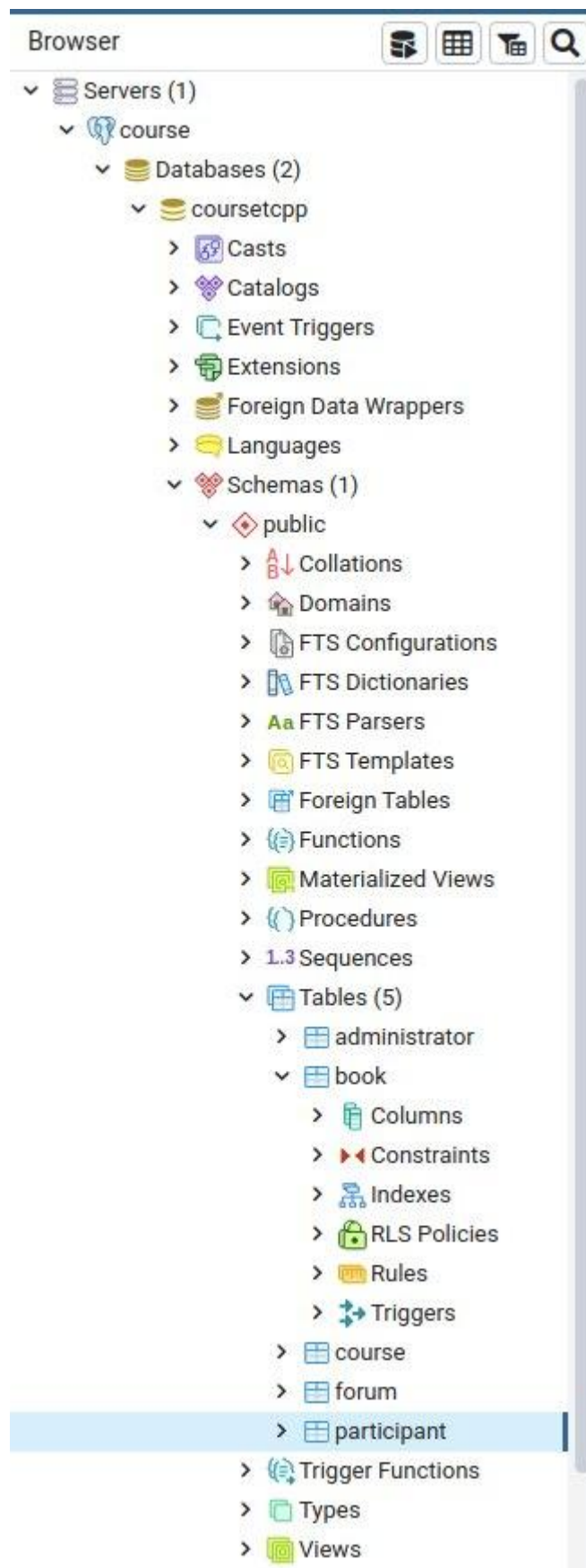


Рисунок 4.1.2.2 - Отслеживание нагрузки на базу данных



4.1.2.3 - Панель инструментов

4.2 Особенности конструирования программных модулей

4.2.1 Особенности работы с интегрированной средой разработки

Реализация алгоритма Search_All_Accessible_Courses

Query Editor Query History

```
1  SELECT course.title, participant.fullname
2  FROM course, participant
3  WHERE (
4      participant.status LIKE '%course%'
5  )
```

Рисунок 4.2.1.1 - Search_All_Accessible_Courses

	Data Output	Explain	Messages	Notifications
	<div><div>title</div><div>character (40)</div></div>	<div><div></div><div></div></div>	<div><div>fullname</div><div>character (50)</div></div>	<div><div></div><div></div></div>
1	Python (продвинут...		Калыш К.С.	
2	Курс менеджмент...		Калыш К.С.	
3	Курс ораторского ...		Калыш К.С.	
4	Python (продвинут...		Мудрик В.С.	
5	Курс менеджмент...		Мудрик В.С.	
6	Курс ораторского ...		Мудрик В.С.	
7	Python (продвинут...		Волков С.В.	
8	Курс менеджмент...		Волков С.В.	
9	Курс ораторского ...		Волков С.В.	

Рисунок 4.2.1.2 - Результат работы алгоритма Search_All_Accessible_Courses

Реализация алгоритма Most_Recent_Message

Query Editor Query History

```

1  SELECT DISTINCT ON (1)
2  title,
3  autor,
4  theme,
5  publicationDate
6  FROM forum
7  ORDER BY 1, publicationDate DESC

```

Рисунок 4.2.1.3 - Most_Recent_Message

Data Output	Explain	Messages	Notifications
<div>▲</div> title character (40)	<div>🔒</div> autor character (50)	<div>🔒</div> theme character (100)	<div>🔒</div> publicationdate date
1 allforum	...	Aleksandr-HH	...
2 forumOdd	...	Richard	...
3 kiberforum	...	McLarenPlay	...

Рисунок 4.2.1.4 - Результат работы алгоритма Most_Recent_Message

Реализация алгоритма Search_Book_By_Author

Query Editor Query History

```

1  SELECT title, theme, issueYear
2  FROM book
3  WHERE author = ' Федор Михайлович Достоевский '

```

Рисунок 4.2.1.5 - Search_Book_By_Author

Data Output			Explain	Messages	Notifications
	title character (40)		theme character (700)		issueyear integer
1	Этюд в багровых ...		Лгущий себе самом...		1879

Рисунок 4.2.1.6 - Результат работы алгоритма Search_Book_By_Author

4.2.4 Особенности разработки алгоритмов методов программных классов или процедур функций

UPDATE participant

SET education = 'привилегированный пользователь'

WHERE status = 'course, forum, book'

Query Editor	Query History
1	UPDATE participant
2	SET education = 'привилегированный пользователь'
3	WHERE status = 'course, forum, book'

Рисунок 4.2.4.1 - Алгоритм Give_Privilege

UPDATE 1
Query returned successfully in 73 msec.

Рисунок 4.2.4.2 - Результат работы алгоритма Give_Privilege

DELETE FROM participant

WHERE publickName = 'numibis' AND registrationDate = '2020-06-10';

```
Query Editor  Query History
1  DELETE FROM participant
2  WHERE publicName = 'numibis' AND registrationDate = '2020-06-10';
```

Рисунок 4.2.4.3 - Алгоритм Delete_User

```
Data Output  Explain  Messages  Notifications
DELETE 1
Query returned successfully in 87 msec.
```

Рисунок 4.2.4.4 - Результат работы алгоритма Delete_User

4.3 Модульное тестирование программных классов

1. Авторизация 1

AllLearn - Authoriization

Введите ваш логин

Kasra

Войти

Введите ваш пароль

Регистрация

Рисунок 4.3.1 - Окно авторизации1

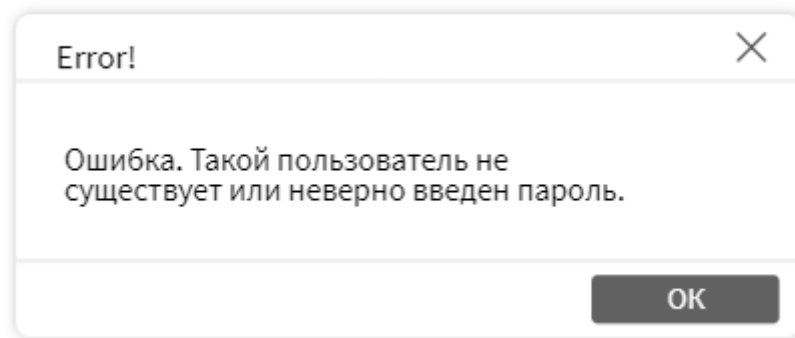


Рисунок 4.3.2 - Результат авторизации 1

2. Авторизация2

A window titled "AllLearn - Authoriization" with standard window controls (minimize, maximize, close) in the top right. The window contains two input fields and two buttons. The first input field is labeled "Введите ваш логин" (Enter your login) and is empty. The second input field is labeled "Введите ваш пароль" (Enter your password) and contains the text "Kasra". To the right of the first input field is a blue button labeled "Войти" (Login). To the right of the second input field is a blue button labeled "Регистрация" (Registration).

Рисунок 4.3.3 - Окно авторизации2

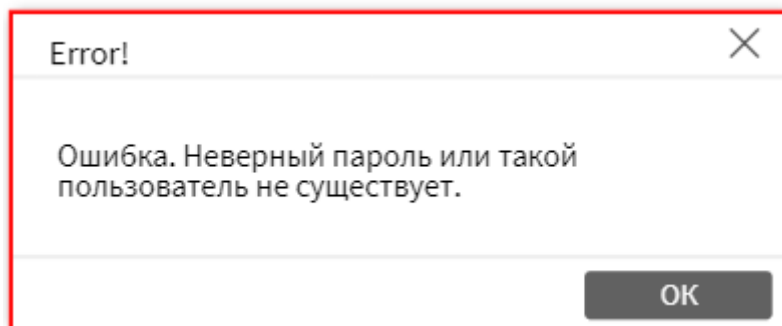


Рисунок 4.3.4 - Результат авторизации2

3. Авторизация3

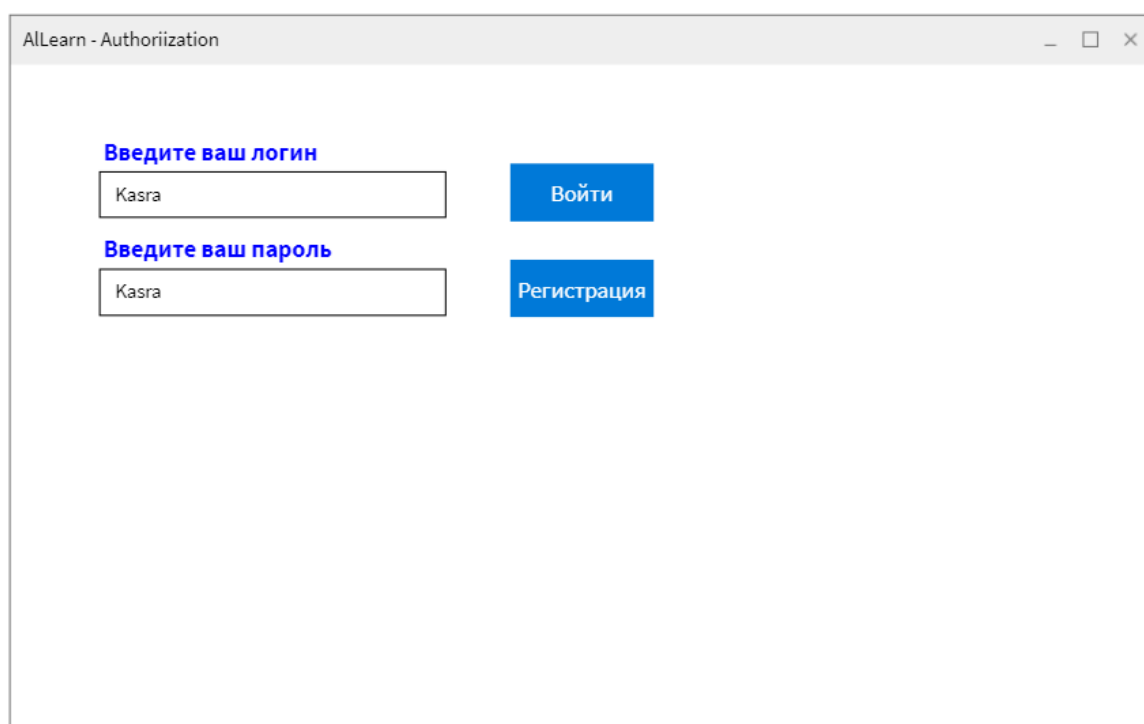


Рисунок 4.3.5 - Окно авторизации3

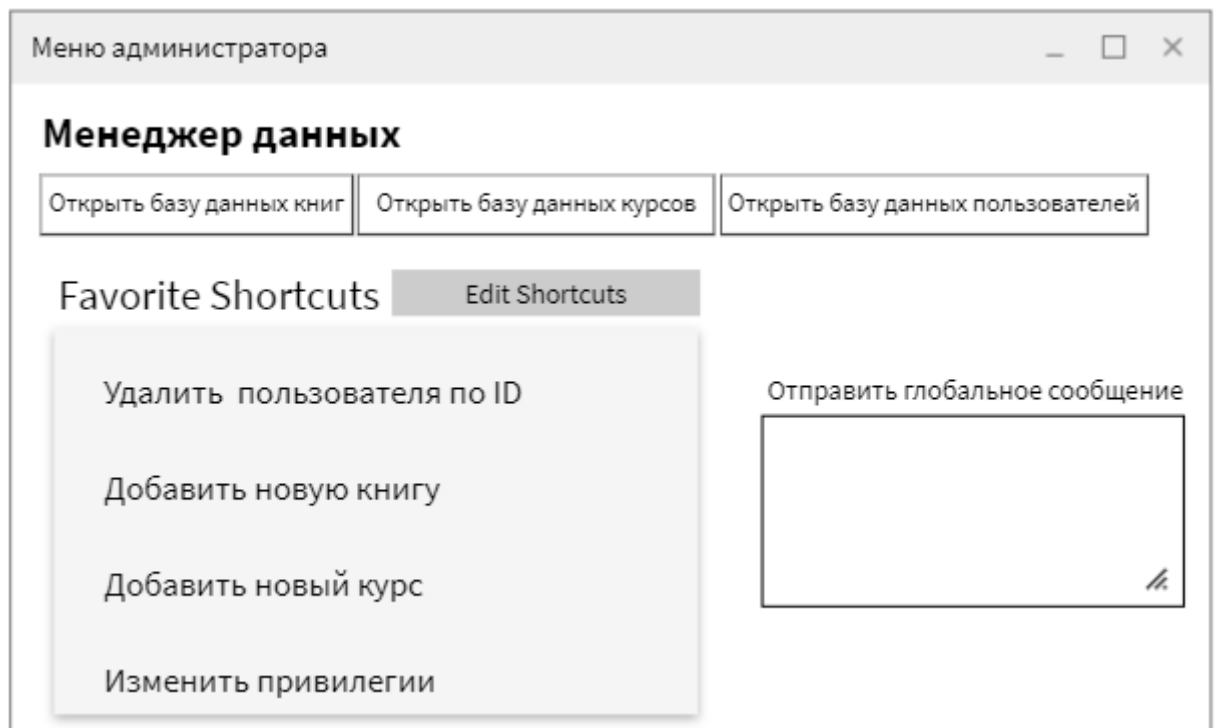


Рисунок 4.3.6 - Результат авторизации3

4. Отображение информации

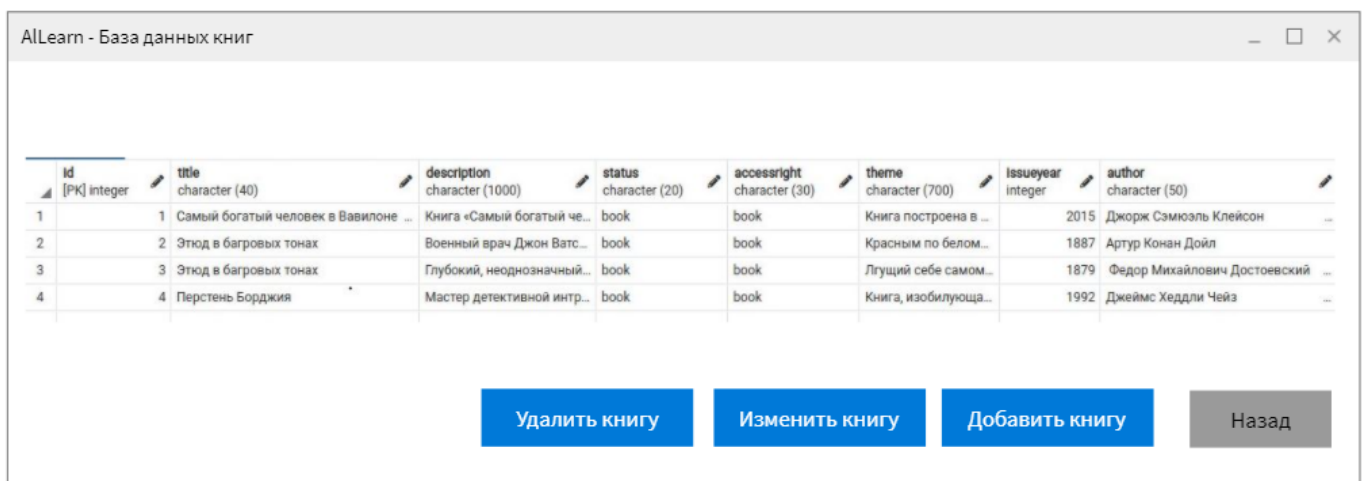


Рисунок 4.3.7 - Результат отображения информации

5. Добавление информации

Label	Value
Title	Условные рефлексы
Description	A guide for anybody...
Status	Free
Accessright	Everyone
Theme	Physiology
Issueyear	1926
Author	Ivan Pavlov

Рисунок 4.3.8 - Окно заполнения для новой книги

5 Развертывание и валидация программного продукта

5.1 Инструкция по установке программного продукта

Для запуска ПП необходимо скачать его github репозиторий и базу данных проекта. В папке запустить exe файл.

5.2 Инструкция по использованию программного продукта

Интерфейс ПП очень понятный, новый пользователь заходя в окно авторизации должен авторизоваться, дальше происходит перенаправление пользователя на интерфейс для пользования всеми функциями ПП.

5.3 Результат валидации программного продукта

Результатом работы ПП является программа с визуальным интерфейсом. С ней можно ознакомиться в github-репозитории.

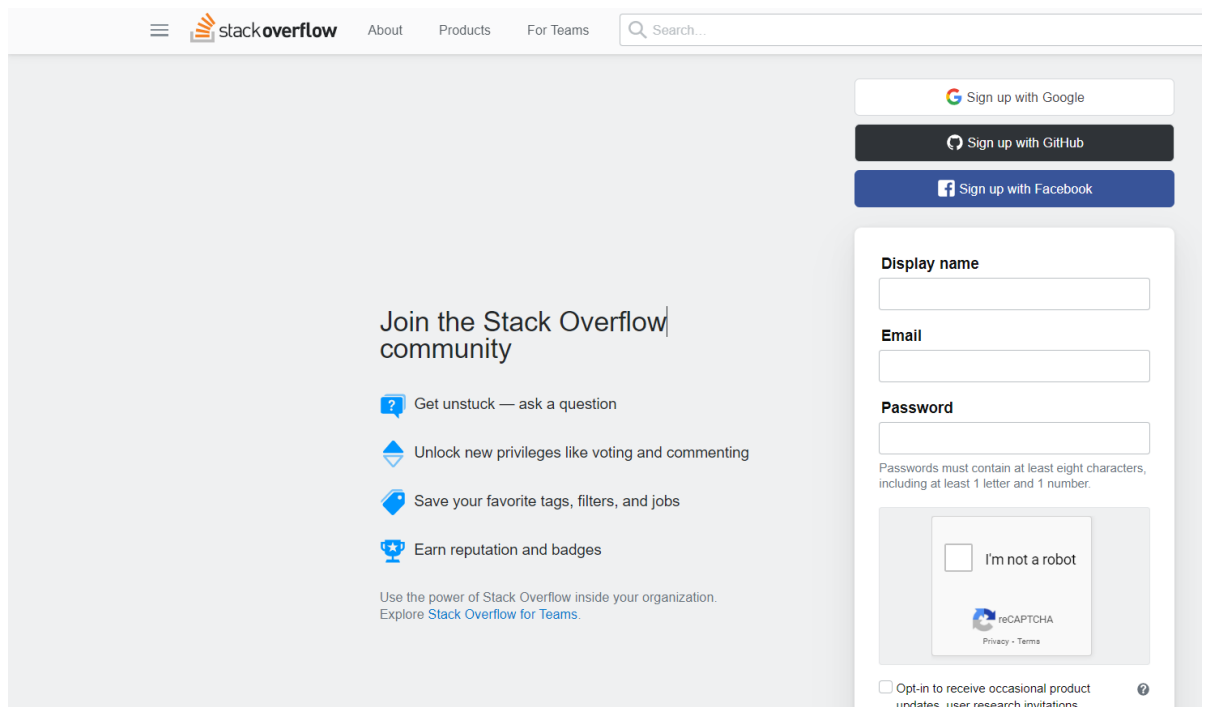


Рисунок 5.3.1 - Окно авторизации\регистрации аналога

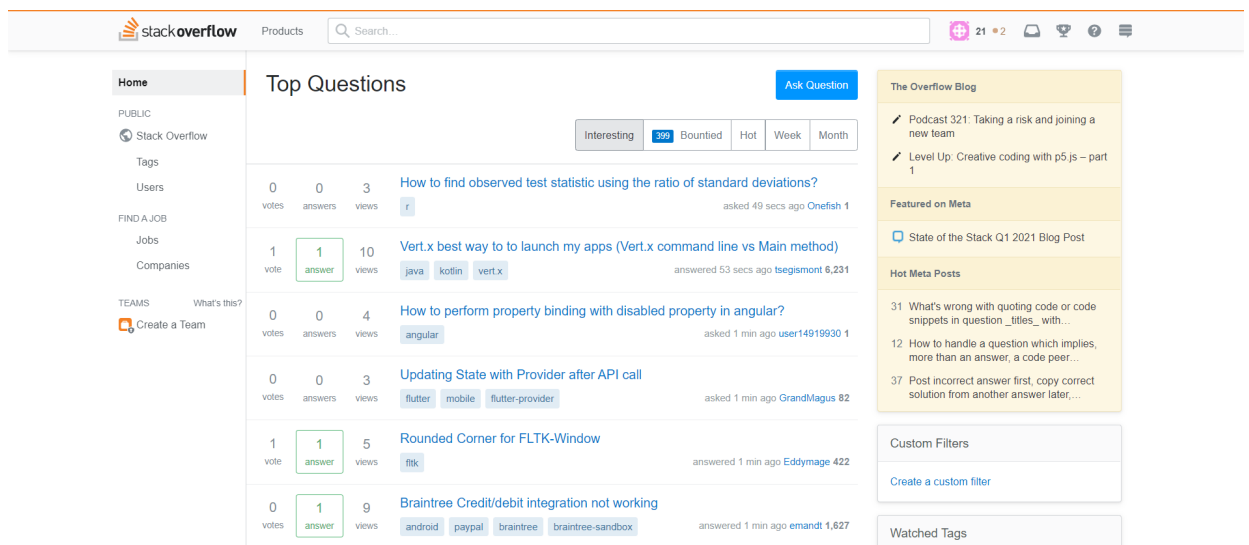


Рисунок 5.3.2 - Программный интерфейс аналога

Выводы

В результате создания программного продукта была достигнута следующая цель “1.2.2.2 Цель создания программного продукта”.

Доказательством является следующее:

1. Графическое представление реализовано
2. Реализованы связи в БД
3. Доступность для нового пользователя

В процессе создания программного продукта возникли следующие трудности: организационные проблемы, проблемы отсутствия специфических знаний, проблема отсутствия опыта, нужного на различных этапах.

1.) Отсутствие опыта, знаний в разработке серверной части

Из-за ограниченного времени на создание программного продукта не была реализована работа всех функций программы.