|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет социально-экономических и компьютерных наук* |
| Гарифуллин Александр Михайлович  **РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ИЗУЧЕНИЯ АЛГОРИТМОВ**  *Курсовой проект*  студента образовательной программы «Программная инженерия» по направлению подготовки *09.03.04 Программная инженерия*  Руководитель  Преподаватель кафедры ИТБ  А.В. Михайлов |

Пермь, 2024 год

Аннотация

Отчет по курсовой работе на тему «Веб-приложение для изучения алгоритмов».

Авторы работы: Гарифуллин Александр Михайлович.

Во введении обоснована актуальность работы, поставлена цель работы, определены задачи работы, указаны объект и предмет исследования.

Первая глава работы посвящена анализу объекта автоматизации. В ней рассмотрена основная документация предметной области, отобраны и сравнены между собой наиболее популярные существующие решения, выполнен анализ сценариев использования. Результатом анализа является техническое задание на разработку системы.

Во второй главе описаны результаты проектирования информационной системы. Построена модель поведения системы, выполнено проектирование, нормализация и реализация базы данных. Результатами главы являются физическая модель базы данных.

Третья глава содержит описание реализации и тестирования системы. Результатами главы являются созданное приложение базы данных, а также результаты тестирования приложения.

Заключение содержит описание результатов, полученных в ходе выполнения работы.

Работа состоит из Х страниц основного текста и Х страниц приложений, содержит Х иллюстраций, Х таблиц, библиография – Х наименований, Х приложений.

Оглавление

[Аннотация 2](#_Toc159621147)

[Оглавление 3](#_Toc159621148)

[Введение 4](#_Toc159621149)

[Глава 1. Анализ процесса изучения алгоритмов и разработка требований к веб-приложению для изучения алгоритмов 6](#_Toc159621150)

[1.1 Анализ процесса изучения алгоритмов. 6](#_Toc159621151)

[1.2 Анализ наиболее известных существующих решений для изучения алгоритмов. 7](#_Toc159621152)

[1.2.1 Существующие решения с теорией по алгоритмам 8](#_Toc159621153)

[1.2.2 Существующие решения с практикой алгоритмов 8](#_Toc159621154)

[1.2.3 Итоговый обзор существующих решений 10](#_Toc159621155)

[1.3 Анализ требований к системе для изучения алгоритмов. 11](#_Toc159621156)

[1.4 Сценарии использования и бизнес-процессы. 13](#_Toc159621157)

[1.4.1 Процессы читателя. 14](#_Toc159621158)

[1.4.2 Процессы редактора. 16](#_Toc159621159)

[1.4.3 Процессы администратора. 21](#_Toc159621160)

[1.5 Результаты анализа процесса изучения алгоритмов. 23](#_Toc159621161)

[Заключение 24](#_Toc159621162)

[Библиографический список 25](#_Toc159621163)

[ПРИЛОЖЕНИЕ А Техническое задание 26](#_Toc159621164)

Введение

В настоящее время в Российской Федерации наблюдается рост спроса на IT специалистов, в частности, программистов [2]. Чтобы стать квалифицированным разработчикам человек должен эффективно решать поставленные задачи и с максимальными возможностями использовать языки программирования. Для развития этих навыков следует изучать алгоритмы [1].

В открытом доступе в сети «Интернет» существуют различные веб-сайты с теорией по алгоритмам: e-maxx [4], algorithmica [5], neerc.ifmo.wiki [3] и т.д. Но, для того чтобы закрепить знания, необходимо применять алгоритмы на практике. Существуют сервисы, где пользователи могут решать олимпиадные задачи на алгоритмы: Codeforces [7], Яндекс.Контест [8], acmp [6] и т.д.

После прочтения теории на каком-либо веб-сайте человек должен самостоятельно искать задачи уже на другом сервисе, т.к. на данный момент не существует решений, где будет как теория, так и сопутствующая ей практика.

В связи с вышеописанным создание веб-приложения для изучения алгоритмов, включающей в себя и теорию, и практику, является актуальной задачей.

Веб-приложение позволит не тратить время на поиск задач на алгоритмы, поможет изучить и закрепить теорию.

Объектом исследования является процесс изучения алгоритмов. Предметом исследования является веб-приложение для изучения алгоритмов.

Цель работы – разработка веб-приложение для изучения алгоритмов, включающее в себя как теорию, так и практику, позволяющую освоить теорию.

Для выполнения поставленной цели необходимо выполнить следующие задачи:

1. Провести анализ процесса изучения алгоритмов и разработать требования к системе.
2. Выполнить проектирование веб-приложения для изучения алгоритмов.
3. Выполнить реализацию веб-приложения для изучения алгоритмов.
4. Провести тестирование разработанного веб-приложения для изучения алгоритмов.
5. Провести развертывание разработанного веб-приложения для изучения алгоритмов.

Этап анализа процесса изучения алгоритмов и разработки требований к системе должен включать в себя:

1. Анализ основной документации предметной области.
2. Анализ наиболее известных существующих решений.
3. Описание сценариев использования и бизнес-процессов в нотации UML.
4. Выделение функциональных и нефункциональных требований к системе и подготовка технического задания по ГОСТ…

Этап проектирования веб-приложения для изучения алгоритмов:

1. Построение первичного отношения и его нормализация до базы данных.
2. Создание физической модели.
3. Построение и представление модели полученной базы данных в нотации UML.

Практическая значимость результатов заключается в том, что на изучение алгоритмов будет тратиться меньше времени, т.к. и теория, и практика будут в одном месте.

Веб-приложение реализуется в Microsoft Visual Studio на языке программирования C# с использованием фреймворка ASP.NET MVC. Кроме того, требуется провести компонентное тестирование приложения согласно критериям черного ящика.

**Глава 1. Анализ процесса изучения алгоритмов и разработка требований к веб-приложению для изучения алгоритмов**

В ходе анализа процесса изучения алгоритмов необходимо выполнить следующие задачи:

1. Выполнить анализ процесса изучения алгоритмов.
2. Сравнить и оценить существующие решения, которые позволяют изучать алгоритмы.
3. Сформировать список требований к веб-приложению для изучения алгоритмов.
4. Описать сценарии использования и бизнес-процессы в нотации UML.

Результатом анализа должно стать техническое задание, выполненное по ГОСТ 19.201-78.

1. Анализ процесса изучения алгоритмов.

Процесс изучения любого алгоритма можно разделить на 2 части: изучение теории и закрепление изученного на практике. С теорией можно познакомиться в специальных курсах и лекциях, книгах и статьях. Практика при изучении алгоритмов – это написание алгоритма в программе на любом языке программирования. При этом, необходимо не просто выучить алгоритм, знать его реализацию, основную идею и асимптотику, важно также научиться понимать, в каких случаях применять алгоритм, когда возникают такие случаи, и как нужно модифицировать алгоритм под конкретную задачу. Простое перепечатывание алгоритма в программу не поможет, таким способом нельзя изучить алгоритм. Для полного понимания алгоритмов существуют олимпиадные задачи по программированию, где по условиям задачи явно не сказано, как и где применить алгоритмы, какие алгоритмы вообще стоит применить в задаче. В процессе решения таких задач можно научится «видеть» алгоритмы, «придет» большее понимание алгоритмов в целом. После десятка решенных задач можно научиться применять алгоритм.

При этом алгоритмы неразрывно связаны со структурами данных, так как большинство алгоритмов используют какие-либо структуры данных в своей реализации. Также отметим, что часто алгоритмы – это не просто набор команд, приводящих к какому-то результату, это идея, которую можно использовать в различных задачах. При этом эту идею можно применять на разных уровнях: это может быть простая задача, где можно просто перепечатать алгоритм из конспекта, а может быть задача, где заметить и применить идею очень сложно.

Не стоит также забывать про то, что во время изучения алгоритмов человек может пользоваться конспектами. Особенно часто к теоретическому материалу обращаются при изучении сложных алгоритмов, где нелегко с прочтения сразу все понять. Конспекты могут быть как письменными, так и электронными. При этом следуют обращаться к наиболее достоверным и понятным ресурсам, поэтому стоит отдавать предпочтения конспектам, написанными профессионалами и экспертами в области алгоритмов и структур данных.

Важно отметить, нужно изучать разные алгоритмы одного уровня, невозможно изучить на очень высоком уровне только какой-то один вид алгоритма, при этом не зная хотя бы несколько алгоритмов среднего уровня, так как чем сложнее алгоритм, тем больше шанс того, что он будет использовать алгоритмы или идеи алгоритмов более низкого уровня. Зная несколько разных алгоритмов одного уровня, можно легче и быстрее изучить алгоритмы более высокого уровня. Это связано также с тем, что для изучения алгоритмов высокого уровня может просто не хватать умственных способностей, так как мозг будет не сильно развит и ему будет сложнее представлять идеи алгоритма. Но если человек будет изучать разные алгоритмы, то его кругозор расширится, усилится логическое и абстрактное мышление, что поспособствует тому, что изучать новые темы будет намного легче. Поэтому важно постепенно изучать разные алгоритмы одного уровня, а не углубляться сразу в какой-то один вид алгоритмов, игнорируя другие виды алгоритмов. Поэтапно с увеличением числа пройденных алгоритмов на разные темы одного уровня можно переходить к более сложным алгоритмам.

1. Анализ **наиболее известных существующих решений для изучения алгоритмов**.

Для облегчения процесса изучения алгоритмов уже существует несколько решений. Некоторые из них затрагивают только теорию, другие только практику, но есть и те, где можно встретить и теорию, и практику, но в виде исключения (т.е. в основном в системе практика, но по некоторым задачам есть и теория, или наоборот). Разделим обзор решений на две части:

1.2.1 Существующие решения с теорией по алгоритмам

В основном все решения с теорией похожи: они хранят какие-либо статьи по алгоритмам. Самыми популярными сервисами являются: Викиконспекты ИТМО [3], e-mmax[4] и Алгоритмика [5]. При этом просматривать материалы можно на каждом сайте не зарегистрировавшись.

Викиконспекты ИТМО хранят конспекты лекций по дисциплине «Алгоритмы и структуры данных» университета ИТМО. Контексты отсортированы по сложности материала, но он рассчитан для студентов, которые уже разбираются в базовых алгоритмах. Для новичков они могут показаться слишком сложными для понимания.

e-mmax хранит не только статьи по алгоритмам, но и книги по ним. Сервис относительно старый, поэтому статьи могут содержать неточности и ошибки. Также в статьях может быть представлена не самая лучшая реализация алгоритмов, так как эта реализация была написана на старых версиях языка программирования C++. В конце некоторых статей можно встретить пару ссылок на задачи на других сервисах, при этом часть ссылок уже не действительна. Отметим, что статьи распределены по темам, но нет распределения по сложности материала. При этом затрагиваются средние и выше среднего по сложности темы. Сайт рассчитан не для новичков.

Алгоритмика – относительно новый открытый проект с материалами лекций различных курсов по Computer Science, проводимых в Tinkoff Generation. Основная аудитория сайта – школьники, поэтому конспекты написаны на более простом и понятном для новичков языке. Статьи на сайте различного уровня: как для начинающих, так и для победителей международной олимпиады школьников по программированию. При этом статьи разделены по темам, но не разделены по сложности.

1.2.2 Существующие решения с практикой алгоритмов

Самым популярным и общедоступным сервисом для решения задач в Российской Федерации является Codeforces [7]. На сайте зарегистрировано более 140 000 пользователей. При этом большую часть составляют граждане других стран. Сервис предоставляет возможность решения олимпиадных задач по программированию разного уровня (от новичка до победителя международных олимпиад по программированию) на более чем 10 языках программирования и разных компиляторов. Отметим, что для каждой отправленной посылки для решения задачи можно посмотреть максимальное время выполнения программы и объем затраченной памяти в МБ. некоторых случаях можно посмотреть тесты задач. При этом есть специальный раздел обучения, где можно посмотреть лекции по некоторым темам, и пройти практику (решить задачи на эти темы). Но этот раздел посвящен не новичкам, так как там затрагивают лишь сложные темы (Дерево отрезков, система непересекающихся множеств, z-функция и другое). Дополнительно стоит отметить, что сайт предоставляешь возможность создания групп для решения наборов задач и самих наборов задач (мэшапов). Группы для решений могут быть как общедоступными (любой зарегистрированный пользователь может вступить в группу и начать решать задачи), работающие по приглашению пользователей (только приглашенные пользователи могут решать задачи), по подтверждению (любой пользователь может подать заявку на вступление в группу и начать решать задачи, после одобрения заявки). При этом в мэшапах можно использовать уже существующие задачи из общедоступного архива задач (архив содержит около 10 000 задач), задачи из других отрытых мэшапов (около 2000 задач), а также можно самому создать задачи и добавить их в мэшап (для этого стоит воспользоваться системой Polygon.Codeforces) [9]. Дополнительно отметим, что к каждому мэшапу можно прикрепить материалы (ссылка на что-либо, либо файл формата pdf). Помимо возможности хранения мэшапов для решения задач, можно легко отслеживать процесс решения участниками как каждого мэшапа по отдельности, так и всех мэшапов в общем, так так в каждом мэшапе есть свое «положение», где будут указаны все пользователи, решавшие задачи из этого мэшапа, а также можно легко добавить рейтинг пользователей, где перечислить мэшапы для составления рейтинга, после этого можно будет смотреть динамику решения задач по всем мэшапам.

Стоит также отметить Яндекст.Контест [8] и acmp [6], где собрано множество олимпиадных задач, но данные сервисы предоставляют куда меньшие возможности, чем Codeforces[7]. Также отмечу, что просматривать задачи можно в каждом сервисе не зарегистрировавшись, но отправлять посылки на задачи могут лишь зарегистрированные пользователи.

1.2.3 Итоговый обзор существующих решений

Лучшим сервисом с теорией для изучения алгоритмов считаю Алгоритмику [5], так как конспекты сайта относительно других сервисов самые новые, при этом написаны на более простом и понятном язык, и уже использовались в кружках Tinkoff Generation для обучения школьников алгоритмам. Но на сайте нет определенного плана изучения конспектов, поэтому самостоятельное изучение алгоритмов будет затруднительно. Также на сайте нет возможности как-либо испытать знания на практике.

Лучшим сервисом для практики считаю Codeforces, так как он предоставляет большой набор задач для решения, а также возможности для организации тренировок (группы с мэшапами). Но на сайте нет возможности новичку изучить алгоритмы. При этом задачи на одну тему и одной сложности, согласно сервису, могут сильно отличаться по реальной сложности, так как сайт существует уже давно (более 10 лет) и старые задачи заметно проще новых, но при этом рейтинг старых задач не меняется под новые реалии.

Соединив теорию с Алгоритмики и задачи с Codeforces, можно получить сервис, который будет одновременно предоставлять лучшую теорию и лучшую практику. Разрабатываемая система должна будет избавиться от минусов уже существующих решений, поэтому стоит сделать следующее:

1. Конспекты должны быть по сложности на уровне Алгоритмики.
2. Должно быть разделение конспектов по темам.
3. Должно быть разделение конспектов по сложности.
4. Стоит добавить пошаговый план изучения алгоритмов.
5. Должна быть возможность решать задачи, при этом задачи должны быть отсортированы по реальной сложности.

Стоит отметить, что в одиночку реализовать возможности системы для безопасного решения задач хотя бы на самых популярных языках программирования, а также для добавления задач в тренировки невероятно трудно и занимает огромное время, из-за чего работа займет намного больше времени, чем уделено на курсовую работу 3 курса, поэтому вместе с руководителем курсовой работ было принято решение: не разрабатывать такие возможности, а использовать уже готовое, а именно на Codeforces будет создана группа с наборами задач на определенные темы, при этом в разрабатываемой системе в конспектах должны быть ссылки на эти наборы задач. В таком случае пользователь сможет прочитать конспект, после чего перейти по ссылке из этого конспекта на задачу на сайте Codeforces и решить ее. При этом пользователь не тратит время на поиски задач для закрепления теории, так что практическая значимость соблюдена. В дальнейшем можно добавить возможность решать задачи прямо в разрабатываемой системе, или добавить какой-либо анализ информации из группы с Codeforces.

Также стоит разделить пользователей на три вида: редакторы, читатели и администраторы. Читатели могут читать конспекты, проходить план изучения алгоритмов. Им не нужно регистрироваться и входить в систему. Редакторы могут добавлять, редактировать, удалять и читать конспекты, редактировать план обучения. Для работы они должны регистрироваться и авторизовываться в системе. При этом редакторы не могут сами зарегистрироваться, им выдают логин и пароль. Администраторы могут делать все, что могут делать редакторы, при этом они создают новых редакторов и администраторов.

1. Анализ требований к системе для изучения алгоритмов.

Определим функциональные (определяют задачи, которые должны выполняться) и нефункциональные (свойства и ограничения, которые должны быть) требования к системе.

Список функциональных требований к системе:

1. Пользователи должны иметь возможность войти в систему через уникальный логин (имя пользователя) и пароль.
2. В системе должны быть конспекты алгоритмов, представленные в виде статей.
3. В системе должен быть пошаговый план изучения алгоритмов, который состоит из упорядоченного списка конспектов.
4. Редакторы должны иметь возможность добавлять новые конспекты.
5. Редакторы должны иметь возможность редактировать существующие конспекты.
6. Редакторы должны иметь возможность удалять существующие конспекты.
7. Редакторы должны иметь возможность просматривать существующие конспекты.
8. Читатели должны иметь возможность просматривать существующие конспекты.
9. Конспекты должны быть распределены по темам. Каждый конспект может быть привязан только к одной теме. Конспекты не могут существовать без темы.
10. Редакторы должны иметь возможность задавать конспектам темы.
11. Редакторы должны иметь возможность создавать темы для конспектов.
12. Редакторы должны иметь возможность удалять темы для конспектов.
13. Редакторы должны иметь возможность редактировать темы для конспектов.
14. Редакторы должны иметь возможность просматривать темы для конспектов.
15. Редакторы должны иметь возможность добавлять конспекты в пошаговый план для изучения алгоритмов.
16. Редакторы должны иметь возможность удалять конспекты из пошагового плана для изучения алгоритмов.
17. Редакторы должны иметь возможность изменять порядок конспектов в пошаговом плане для изучения алгоритмов.
18. Редакторы должны иметь возможность просматривать конспекты в пошаговом плане для изучения алгоритмов.
19. Читатели должны иметь возможность просматривать конспекты в пошаговом плане для изучения алгоритмов.
20. Администраторы должны иметь возможность создавать новых пользователей в системе.
21. Администраторы должны иметь возможность редактировать пользователей в системе.
22. Администраторы должны иметь возможность удалять пользователей в системе.
23. Администраторы должны иметь возможность просматривать пользователей в системе.
24. Администраторы должны иметь возможность выдавать роли пользователям.
25. Администраторы должны иметь возможность изменять роли пользователей.
26. Администраторы должны иметь доступ ко всем функциями системы: функции редакторов и свои дополнительные привилегии.
27. Добавление конспекта представляет собой загрузку Markdown документа.
28. Редактирование конспекта представляет собой загрузку нового Markdown документа.
29. При удалении темы конспектов удаляются все конспекты, привязанные к этой теме.
30. Добавлять в пошаговый план можно только уже существующие конспекты. При удалении конспектов они удаляются из пошагового плана.

Список нефункциональных требований к системе:

1. База данных системы должна быть нормализована как минимум до третьей нормальной формы.
2. Пароли пользователей должны храниться в зашифрованном виде.
3. Время загрузки конспектов должно быть минимальным для обеспечения удобства пользователей.
4. Запросы на добавление, редактирование и удаление должны обрабатываться быстро и эффективно.
5. Интерфейс должен быть интуитивно понятным и удобным в использовании.
6. Редакторы и администраторы обязаны выполнить вход в систему через логин и пароль для доступа к своим правам.
7. Читатели не должны входить в систему для просмотра контента.
8. Сценарии использования и бизнес-процессы.

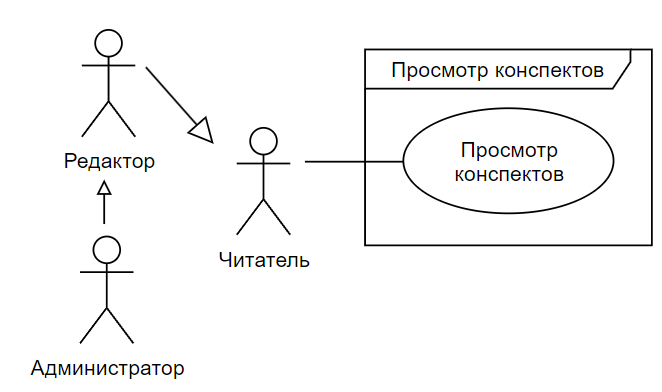
Исходя из описанных требований к системе, у пользователей должны быть такие права:

1. Читатели:
   1. Просмотр конспектов.
   2. Просмотр пошагового плана.
2. Редакторы:
   1. Все права читателей.
   2. Вход в систему.
   3. Редактирование конспектов.
   4. Удаление конспектов.
   5. Добавление конспектов.
   6. Создание тем для конспектов.
   7. Редактирование тем для конспектов.
   8. Просмотр тем для конспектов.
   9. Удаление тем для конспектов.
   10. Присваивание конспектам тем.
   11. Добавление конспекта в пошаговый план.
   12. Удаление конспекта из пошагового плана.
   13. Изменение порядка конспектов в пошаговом плане.
3. Администраторы:
   1. Все права редакторов.
   2. Создание новых пользователей.
   3. Редактирование пользователей.
   4. Удаление пользователей.
   5. Просмотр пользователей.
   6. Выдача ролей пользователем.
   7. Изменение ролей пользователей.

Перейдем к use case и диаграммам активностей на основе описанных требований.

1.4.1 Процессы читателя.

Диаграмма прецедентов для процесса просмотра конспектов представлена на рисунке 1.



***Рисунок 1 – Use case для процесса просмотра конспектов***

Ниже приведены описания прецедентов для процесса просмотра конспектов.

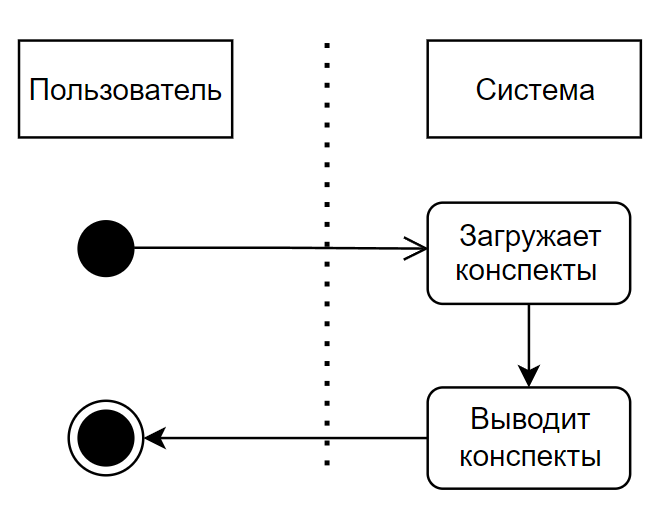
Название: просмотр конспектов.

Предусловие: пользователь заходит в систему.

Основной поток:

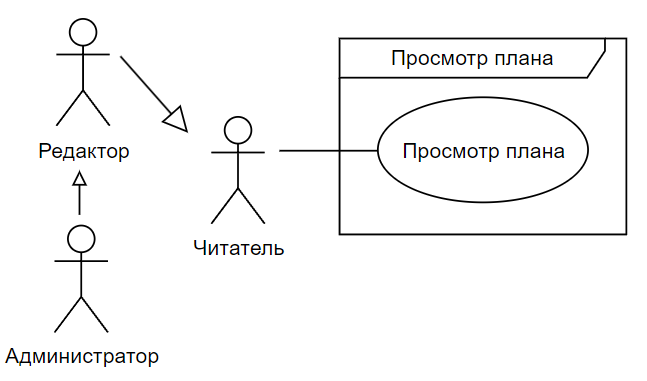
1. Система загружает все конспекты.
2. Система выводит конспекты.

Диаграмма активностей для процесса просмотра конспектов представлена на рисунке 2.



***Рисунок 2 – Диаграмма активности для процесса просмотра конспектов***

Диаграмма прецедентов для процесса просмотра пошагового плана представлена на рисунке 3.



***Рисунок 3 – Use case для процесса просмотра плана***

Ниже приведены описания прецедентов для процесса просмотра пошагового плана конспектов.

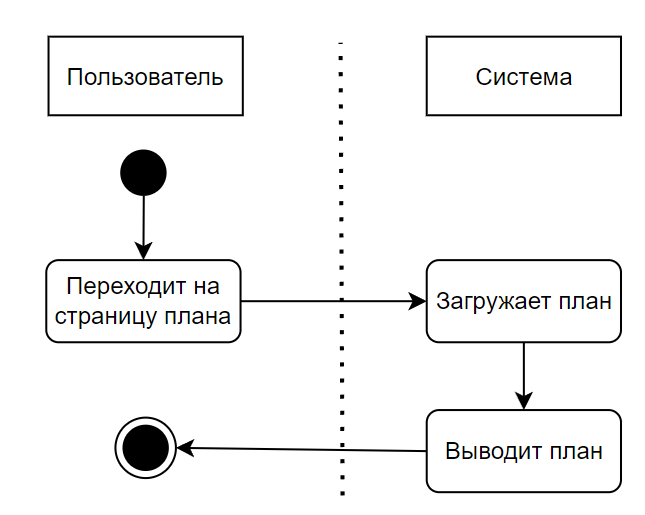
Название: просмотр плана.

Предусловие: пользователь зашел в систему.

Основной поток:

1. Пользователь кнопку, либо выполняет другое действие, вызывающее переход на страницу пошагового плана.
2. Система загружает план.
3. Система отображает план.

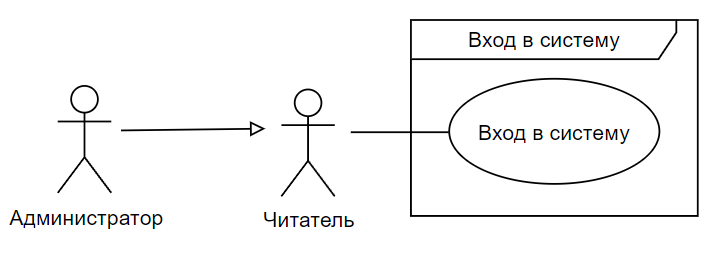
Диаграмма активностей для процесса просмотра пошагового плана представлена на рисунке 4.



***Рисунок 4 – Диаграмма активности для процесса просмотра плана***

1.4.2 Процессы редактора.

Диаграмма прецедентов для процесса входа в систему представлена на рисунке 5.



***Рисунок 5 – Use case для процесса входа в систему***

Ниже приведены описания прецедентов для процесса просмотра пошагового плана конспектов.

Название: вход в систему.

Предусловие: пользователь находится в главном меню системы.

Основной поток:

1. Пользователь нажимает кнопку, либо выполняет другое действие, вызывающее функцию входа в систему.
2. Система открывает окно для входа.
3. Пользователь вводит логин и пароль для входа.
4. Пользователь подтверждает логин и пароль (Б).
5. Система проверяет существование пользователя (А).
6. Система выдает права пользователю.
7. Система закрывает окно для входа.

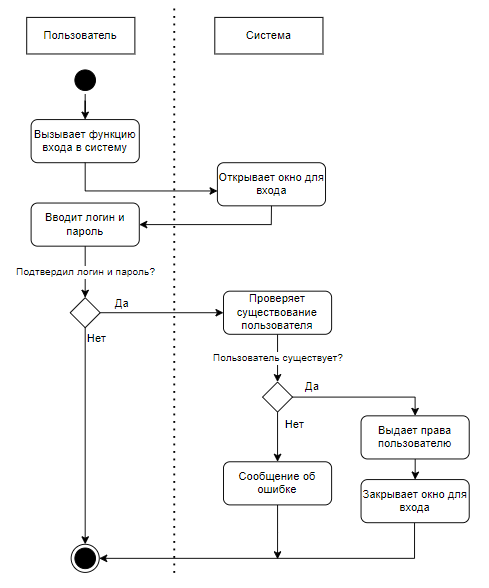
Альтернативный поток А (ошибочный ввод данных):

1. Система проверяет входные данные на соответствие ограничениям ввода и существование пользователя в системе.
2. Система выводит сообщение о том, что данные некорректны. Окно входа не закрывается.

Альтернативный поток Б (выход без подтверждения):

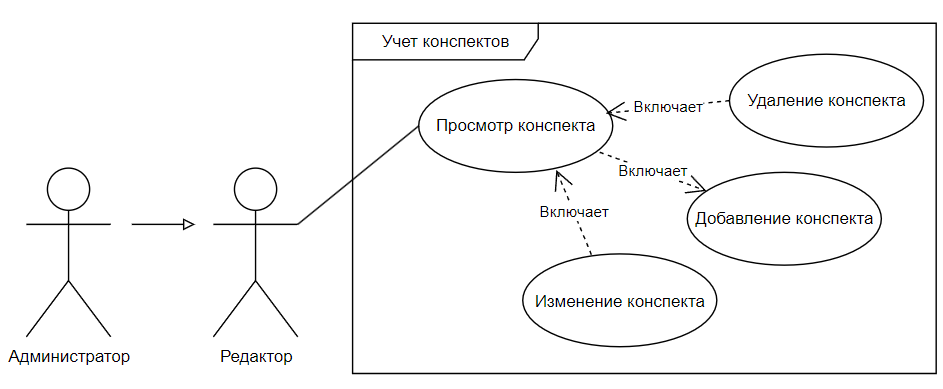
1. Пользователь закрывает окно входа без подтверждения.

Диаграмма активностей для процесса входа в систему представлена на рисунке 6.



***Рисунок 6 – Диаграмма активности для процесса входа в систему***

Диаграмма прецедентов для процесса учета конспектов представлена на рисунке 7.



***Рисунок 7 – Use case для процесса учета конспектов***

Ниже приведены описания прецедентов для процесса учета конспектов.

Название: изменение конспекта.

Предусловие: выполнен прецедент «просмотр конспекта»; пользователь находится в окне просмотра данных.

Основной поток:

1. Пользователь выбирает конспект, который нужно изменить.
2. Система открывает окно редактирования конспекта.
3. Пользователь изменяет данные.
4. Пользователь подтверждает изменения (А).
5. Система проверяет изменения на соответствие ограничениям ввода (Б).
6. Окно редактирования конспектов закрывается.
7. Система отображает изменения в окне просмотра данных.

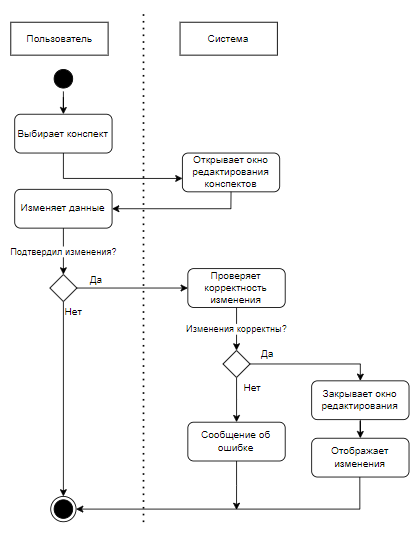
Альтернативный поток А (выход без подтверждения):

1. Пользователь закрывает окно входа без подтверждения.

Альтернативный поток Б (ошибочный ввод данных):

1. Система проверяет входные данные на соответствие ограничениям ввода.
2. Система выводит сообщение о том, что данные некорректны. Окно входа не закрывается.

Диаграмма активностей для процесса редактирования конспектов представлена на рисунке 8.



***Рисунок 8 – Диаграмма активности для процесса редактирования конспектов***

Название: добавление конспекта.

Предусловие: пользователь находится в главном меню системы.

Основной поток:

1. Пользователь нажимает кнопку, либо выполняет другое действие, вызывающее функцию добавления нового конспекта.
2. Система открывает окно добавления конспекта.
3. Пользователь загружает конспект.
4. Пользователь подтверждает изменения (А).
5. Система проверяет добавление на соответствие ограничениям ввода (Б).
6. Окно добавления конспектов закрывается.
7. Система отображает конспект в окне просмотра данных.

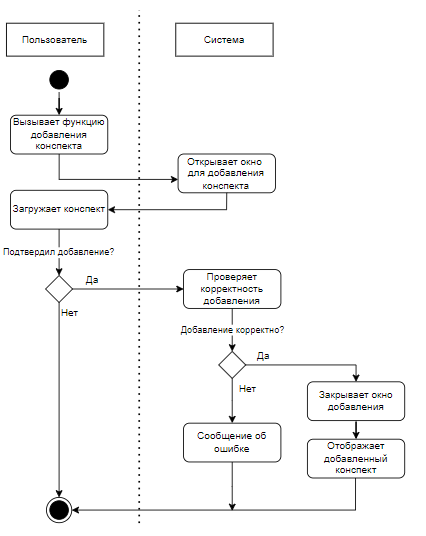
Альтернативный поток А (выход без подтверждения):

1. Пользователь закрывает окно входа без подтверждения.

Альтернативный поток Б (ошибочный ввод данных):

1. Система проверяет входные данные на соответствие ограничениям ввода.
2. Система выводит сообщение о том, что данные некорректны. Окно добавления не закрывается.

Диаграмма активностей для процесса добавления конспектов представлена на рисунке 9.



***Рисунок 9 – Диаграмма активности для процесса добавления конспектов***

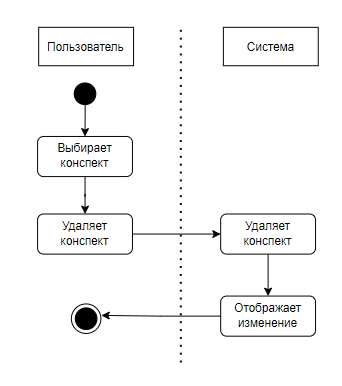
Название: удаление конспекта.

Предусловие: выполнен прецедент «просмотр конспекта»; пользователь находится в окне просмотра данных.

Основной поток:

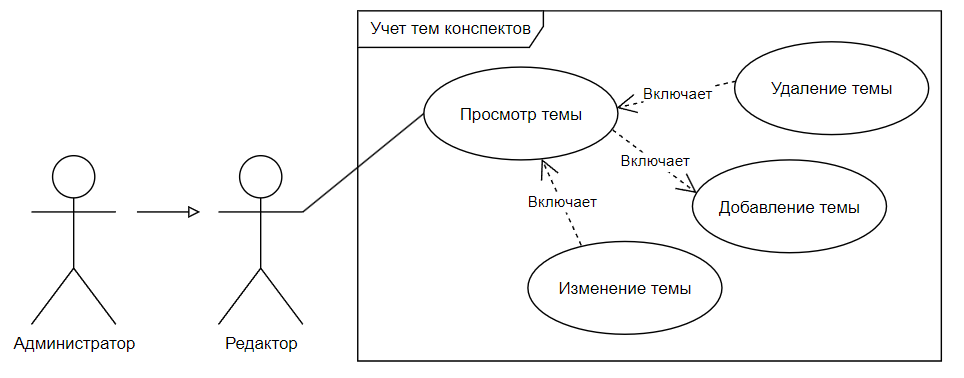
1. Пользователь выбирает конспект для удаления.
2. Пользователь удаляет данные.
3. Система удаляет данные.
4. Система отображает изменение в коне просмотра конспектов.

Диаграмма активностей для процесса удаления конспектов представлена на рисунке 10.



***Рисунок 10 – Диаграмма активности для процесса удаления конспектов***

Диаграмма прецедентов для процесса учета тем для конспектов представлена на рисунке 11.



***Рисунок 11 – Use case для процесса учета тем для конспектов***

Ниже приведены описания прецедентов для процесса учета тем для конспектов.

Название: добавление темы.

Предусловие: пользователь находится в главном меню системы.

Основной поток:

1. Пользователь нажимает кнопку, либо выполняет другое действие, вызывающее функцию добавления темы.
2. Система открывает окно добавления темы.
3. Пользователь добавляет тему.
4. Пользователь подтверждает добавление (А).
5. Система проверяет добавление на соответствие ограничениям ввода (Б).
6. Окно добавления темы закрывается.
7. Система отображает тему в окне просмотра данных.

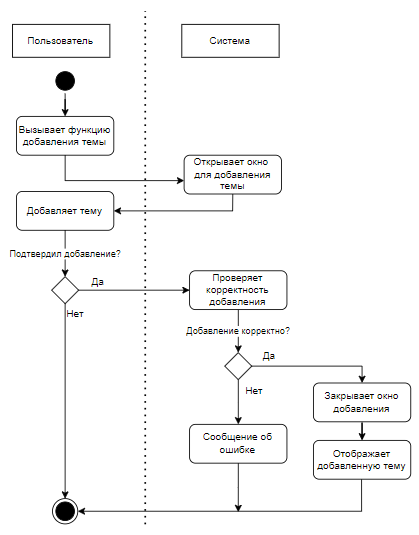
Альтернативный поток А (выход без подтверждения):

1. Пользователь закрывает окно входа без подтверждения.

Альтернативный поток Б (ошибочный ввод данных):

1. Система проверяет входные данные на соответствие ограничениям ввода.
2. Система выводит сообщение о том, что данные некорректны. Окно добавления не закрывается.

Диаграмма активностей для процесса добавления темы для конспекта представлена на рисунке 11.



***Рисунок 12 – Диаграмма активности для процесса добавления темы для конспекта***

Название: изменение темы конспекта.

Предусловие: выполнен прецедент «просмотр темы»; пользователь находится в окне просмотра данных.

Основной поток:

1. Пользователь выбирает тему, которую нужно изменить.
2. Система открывает окно редактирования темы.
3. Пользователь изменяет данные.
4. Пользователь подтверждает изменения (А).
5. Система проверяет изменения на соответствие ограничениям ввода (Б).
6. Окно редактирования темы закрывается.
7. Система отображает изменения в окне просмотра данных.

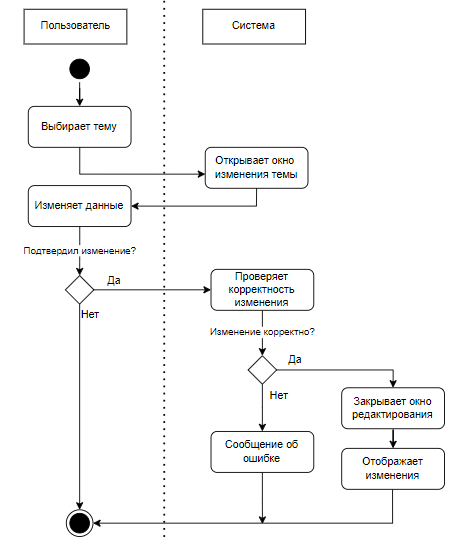
Альтернативный поток А (выход без подтверждения):

1. Пользователь закрывает окно входа без подтверждения.

Альтернативный поток Б (ошибочный ввод данных):

1. Система проверяет входные данные на соответствие ограничениям ввода.
2. Система выводит сообщение о том, что данные некорректны. Окно входа не закрывается.

Диаграмма активностей для процесса редактирования темы для конспектов представлена на рисунке 13.



***Рисунок 13 – Диаграмма активности для процесса редактирования тем для конспектов***

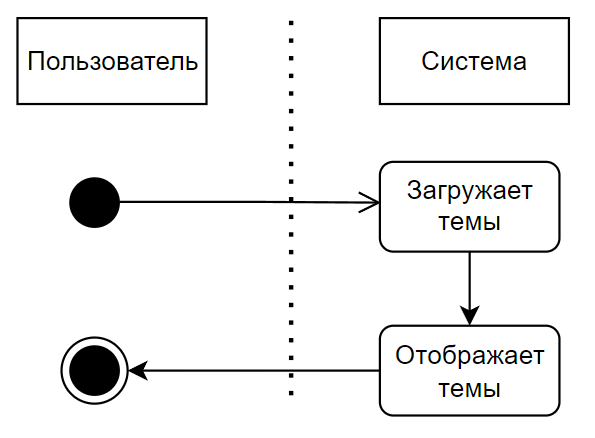
Название: просмотр темы.

Предусловие: пользователь зашел в систему.

Основной поток:

1. Система загружает темы.
2. Система отображает темы.

Диаграмма активностей для процесса просмотра темы для конспектов представлена на рисунке 14.



***Рисунок 14 – Диаграмма активности для процесса просмотра тем для конспектов***

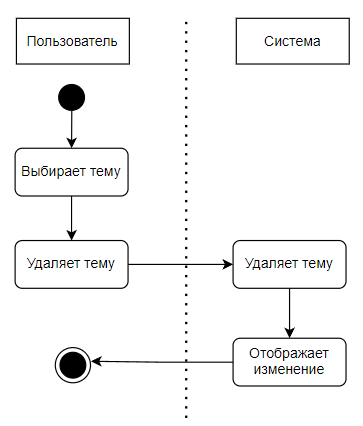
Название: удаление темы.

Предусловие: выполнен прецедент «просмотр темы»; пользователь находится в окне просмотра данных.

Основной поток:

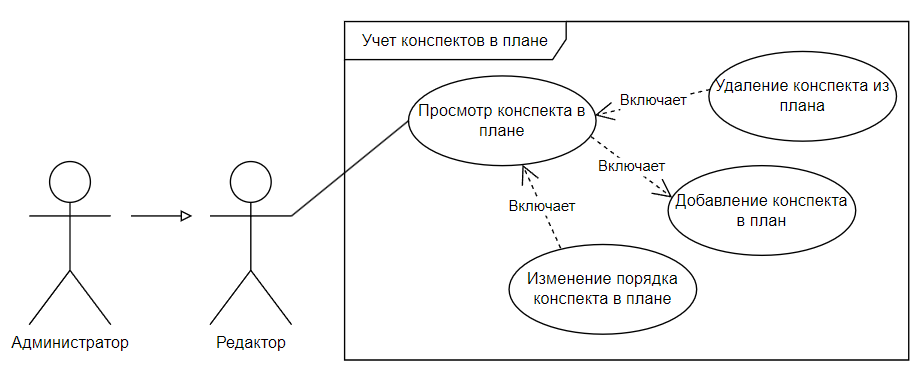
1. Пользователь выбирает тему для удаления.
2. Пользователь удаляет данные.
3. Система удаляет данные.
4. Система отображает изменение в окне просмотра темы.

Диаграмма активностей для процесса удаления темы для конспектов представлена на рисунке 15.



***Рисунок 15 – Диаграмма активности для процесса удаления тем для конспектов***

Диаграмма прецедентов для процесса учета конспектов в плане представлена на рисунке 16.



***Рисунок 16 – Use case для процесса учета конспектов в плане***

Ниже приведены описания прецедентов для процесса учета конспектов в плане.

Название: добавление конспекта в пошаговый план.

Предусловие: пользователь находится в меню пошагового плана.

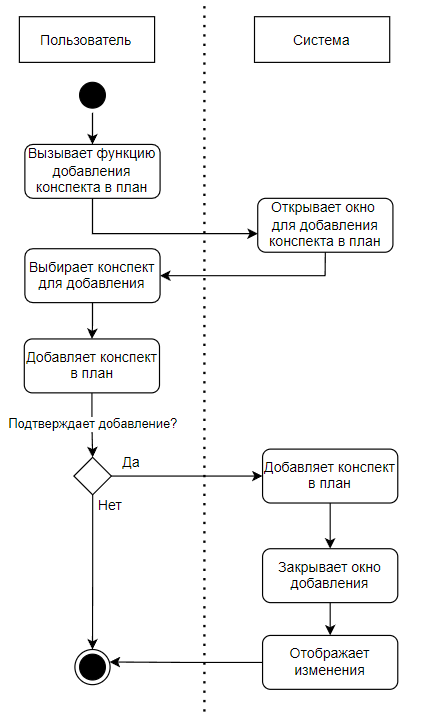
Основной поток:

1. Пользователь нажимает кнопку, либо выполняет другое действие, вызывающее функцию добавления конспекта в план.
2. Система открывает окно добавления конспекта в план
3. Пользователь выбирает конспект для добавления.
4. Пользователь добавляет конспект.
5. Пользователь подтверждает добавление (А).
6. Окно добавления конспекта в план закрывается.
7. Система отображает конспект в окне просмотра данных.

Альтернативный поток А (выход без подтверждения):

1. Пользователь закрывает окно входа без подтверждения.

Диаграмма активностей для процесса добавления конспекта в пошаговый план представлена на рисунке 17.



***Рисунок 17 – Диаграмма активности для процесса добавления конспекта в пошаговый план***

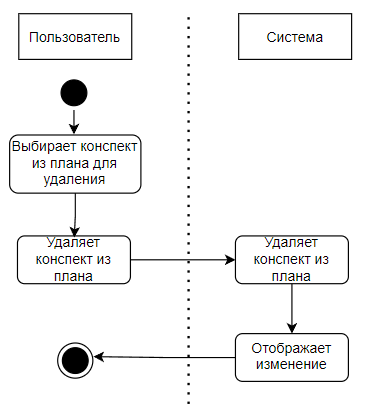
Название: удаление конспекта из пошагового плана.

Предусловие: выполнен прецедент «просмотр пошагового плана»; пользователь находится в окне просмотра данных.

Основной поток:

1. Пользователь выбирает конспект из плана для удаления.
2. Пользователь удаляет данные.
3. Система удаляет данные.
4. Система отображает изменение в окне просмотра темы.

Диаграмма активностей для процесса удаления конспекта из пошагового плана представлена на рисунке 18.



***Рисунок 18 – Диаграмма активности для процесса удаления конспекта из пошагового плана***

Название: изменения порядка конспекта в пошаговом плане.

Предусловие: выполнен прецедент «просмотр пошагового плана»; пользователь находится в окне просмотра данных.

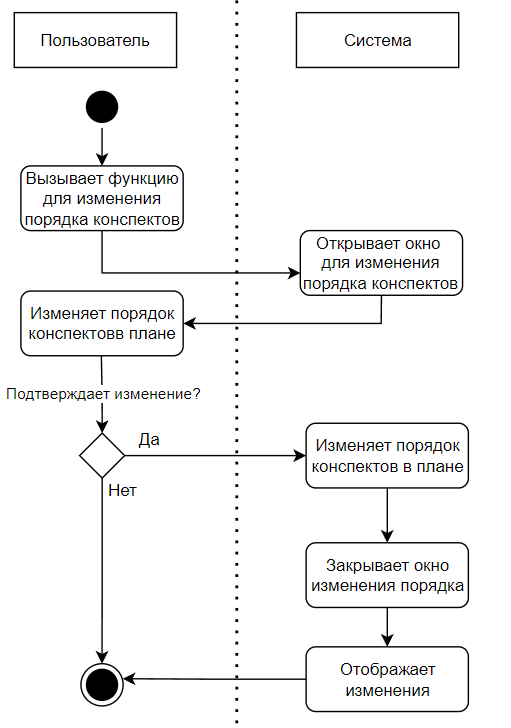
Основной поток:

1. Пользователь нажимает кнопку, либо выполняет другое действие, вызывающее функцию изменения порядка конспектов.
2. Система открывает окно для изменения порядка.
3. Пользователь изменяет порядок конспекта.
4. Пользователь подтверждает изменения (А).
5. Система изменяет порядок конспектов в плане.
6. Окно изменения порядка закрывается.
7. Система отображает изменения в окне просмотра данных.

Альтернативный поток А (выход без подтверждения):

1. Пользователь закрывает окно входа без подтверждения.

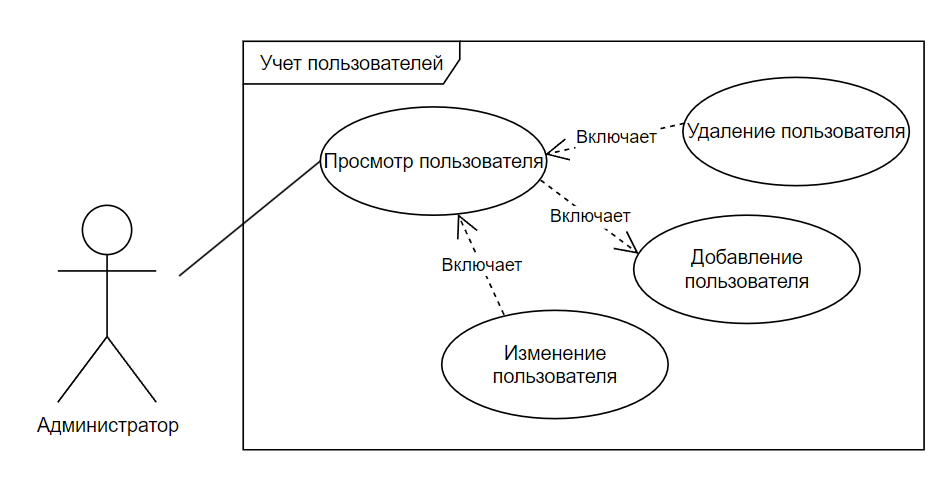
Диаграмма активностей для процесса изменения порядка конспектов в пошаговом плане представлена на рисунке 19.



***Рисунок 19 – Диаграмма активности для процесса изменения порядка конспектов в пошаговом плане***

1.4.3 Процессы администратора.

Диаграмма прецедентов для процесса учета пользователей представлена на рисунке 20.



***Рисунок 20 – Use case для процесса учета пользователей***

Ниже приведены описания прецедентов для процесса учета пользователей.

Название: добавление пользователя.

Предусловие: администратор находится в главном меню системы.

Основной поток:

1. Администратор нажимает кнопку, либо выполняет другое действие, вызывающее функцию добавления пользователя.
2. Система открывает окно добавления пользователя.
3. Администратор добавляет пользователя.
4. Администратор подтверждает добавление (А).
5. Система проверяет добавление на соответствие ограничениям ввода (Б).
6. Окно добавления пользователя закрывается.
7. Система отображает пользователя в окне просмотра данных.

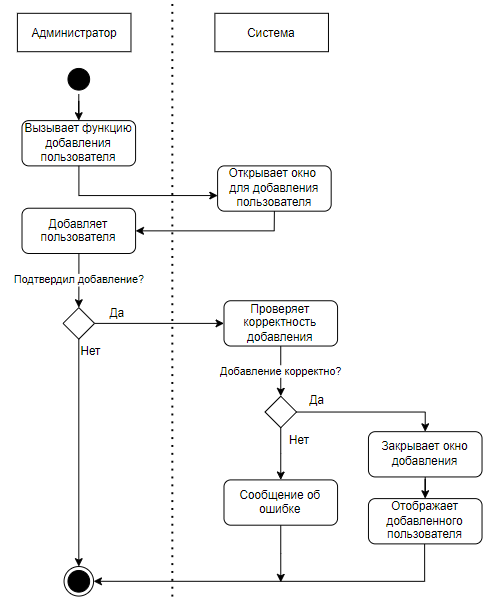
Альтернативный поток А (выход без подтверждения):

1. Администратор закрывает окно входа без подтверждения.

Альтернативный поток Б (ошибочный ввод данных):

1. Система проверяет входные данные на соответствие ограничениям ввода.
2. Система выводит сообщение о том, что данные некорректны. Окно добавления не закрывается.

Диаграмма активностей для процесса добавления пользователей представлена на рисунке 21.



***Рисунок 21 – Диаграмма активности для процесса добавления пользователей***

Название: изменение пользователя.

Предусловие: выполнен прецедент «просмотр пользователя»; администратор находится в окне просмотра данных.

Основной поток:

1. Администратор выбирает пользователя, которого нужно изменить.
2. Система открывает окно редактирования пользователя.
3. Администратор изменяет данные.
4. Администратор подтверждает изменения (А).
5. Система проверяет изменения на соответствие ограничениям ввода (Б).
6. Окно редактирования пользователя закрывается.
7. Система отображает изменения в окне просмотра данных.

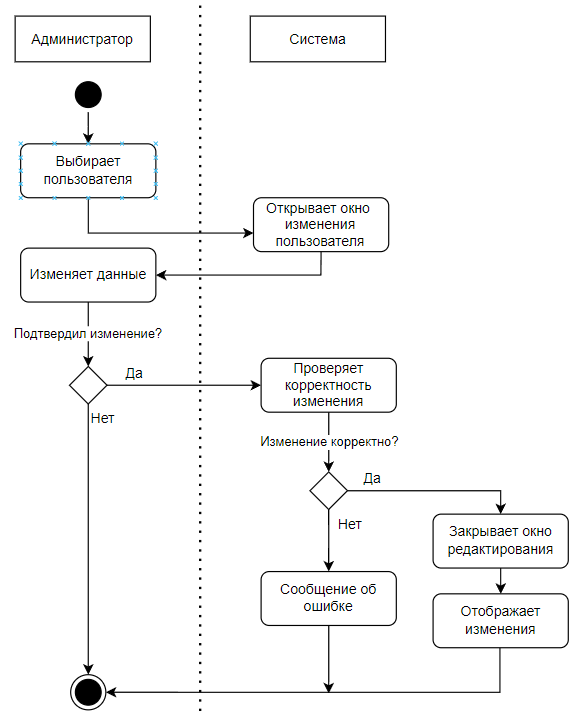
Альтернативный поток А (выход без подтверждения):

1. Пользователь закрывает окно входа без подтверждения.

Альтернативный поток Б (ошибочный ввод данных):

1. Система проверяет входные данные на соответствие ограничениям ввода.
2. Система выводит сообщение о том, что данные некорректны. Окно входа не закрывается.

Диаграмма активностей для процесса редактирования пользователей представлена на рисунке 22.



***Рисунок 22 – Диаграмма активности для процесса редактирования пользователей***

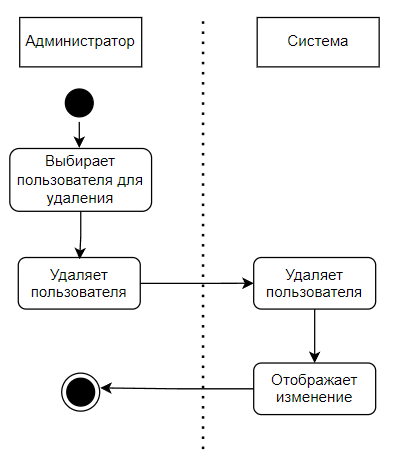
Название: удаление пользователя.

Предусловие: выполнен прецедент «просмотр пользователя»; администратор находится в окне просмотра данных.

Основной поток:

1. Администратор выбирает пользователя для удаления.
2. Администратор удаляет данные.
3. Система удаляет данные.
4. Система отображает изменение в окне просмотра пользователей.

Диаграмма активностей для процесса удаления пользователей представлена на рисунке 23.



***Рисунок 23 – Диаграмма активности для процесса удаления пользователей***

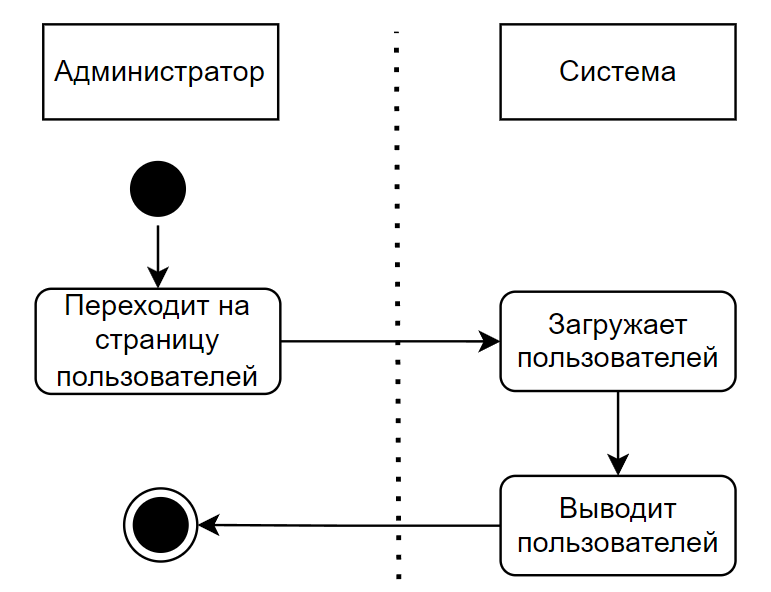
Название: просмотр пользователя.

Предусловие: администратор зашел в систему.

Основной поток:

1. Администратор нажимает кнопку, либо выполняет другое действие, вызывающее переход на страницу пользователей.
2. Система загружает пользователей.
3. Система отображает пользователей.

Диаграмма активностей для процесса просмотра пользователей представлена на рисунке 24.



***Рисунок 24 – Диаграмма активности для процесса просмотра пользователей***

1. Результаты анализа процесса изучения алгоритмов.

Результатам и анализа процесса изучения алгоритмов стали:

1. Рассмотрены и проанализированы понятия и проблемы предметной области.
2. Выявлены и сравнены наиболее популярные решения.
3. Разработаны требования к системе.
4. Определены сценарии работы пользователей, приведено подробное описание прецедентов, построены диаграммы активностей.

На основе результатов анализа составлено техническое задание на разработку веб-приложения (см. приложение А).

Заключение

опр

Библиографический список

1. <https://habr.com/ru/articles/279453/>
2. <https://www.cnews.ru/news/top/2023-07-04_v_rossii_vzryvnoj_rost_sprosa>
3. <https://neerc.ifmo.ru/wiki/index.php?title=Заглавная_страница>
4. <https://e-maxx.ru/algo/>
5. <https://ru.algorithmica.org/>
6. <https://acmp.ru/>
7. <https://codeforces.com/>
8. <https://contest.yandex.ru/>
9. https://polygon.codeforces.com/

Приложение А

ПРИЛОЖЕНИЕ А  
Техническое задание

|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ |
|  |  | Кандидат технических наук, доцент кафедры информационных технологий в бизнесе |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_ Викентьева О.Л. |
|  |  | “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2024 |

**веб-приложение для изучения алгоритмов**

**Техническое задание**

Инв. № под.

Подпиь и дата

Взам. нв. №

Инв. №дубл.

Подпись и Дата

**Лист утверждения**

**Листов Х** **А.В.00001-01 ТЗ 01**

|  |  |  |
| --- | --- | --- |
|  |  | Руководитель разработки |
|  |  | кандидат физико-математических наук, доцент кафедры информационных технологий в бизнесе |
|  |  | \_\_\_\_\_\_\_\_\_\_\_ Михайлов А. В. |
|  |  | “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2024 |
|  |  |  |
|  |  | Исполнитель |
|  |  | Студент 3 курса бакалавриата программы «Программная инженерия» |
|  |  | \_\_\_\_\_\_\_\_\_\_Гарифуллин А.М. |
|  |  | “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2024 |

1 Введение

Наименование программы – «СУАР», система учета арендаторов жилых помещений.

Система предназначена для автоматизации учета арендаторов жилых помещений.

2 Основания для разработки

Программа разрабатывается на основании Положения о проектной, научно-исследовательской деятельности и практиках студентов Национального исследовательского университета «Высшая школа экономики» от 24.06.2016.

3 Назначение разработки

С точки зрения функционального назначения система должна предоставлять функции для реализации операций ввода, просмотра, редактирования и удаления для следующих объектов реального мира, участвующих в процессе учета арендаторов жилья: клиентов (арендаторов и арендодателей), жилых зданий, квартир, договоров аренды, арендных платежей. Также система должна предоставлять данные о задолженностях арендаторов.

Система предназначена для использования сотрудниками компаний, работающих в сфере аренды жилых помещений с целью ведения процесса учета арендаторов жилых помещений.

4 Требования к программе

4.1 Требования к функциональным характеристикам

Программа должна предоставлять следующие функции:

* Добавление, просмотр, изменение и удаление данных о клиентах, жилых помещениях – квартирах и зданиях, в которых расположены квартиры.
* Добавление, просмотр, удаление данных о договорах аренды и платежах.
* Возможность продления срока договоров аренды.
* Возможность смены статуса договора аренды как вручную, так и автоматически (по завершении срока действия договора).
* Возможность фильтрации данных.
* Возможность экспорта данных в MS Excel.
* Предоставление информации о задолженностях арендаторов.

4.2 Требования к надежности

Программа должна представлять собой веб-приложение для работы с базой данных Microsoft SQL Server. Для обеспечения контроля логической целостности данных база данных должна быть приведена как минимум к третьей нормальной форме. При добавлении и редактировании данных программа должна учитывать их формат, а также дополнительные ограничения (диапазон допустимых значений), установленные при проектировании базы данных. Программа должна обеспечивать работу механизма каскадного обновления данных.

Для обеспечения физической целостности базы данных необходимо выполнять резервное копирование базы данных не реже одного раза в сутки. Резервное копирование выполняется вручную.

4.3 Условия эксплуатации

Программа предназначена для эксплуатации на настольных компьютерах. Доступ к программе осуществляется через веб-браузер Google Chrome. База данных информационной системы располагается на удаленном сервере.

Необходимый персонал для эксплуатирования программы состоит из сотрудника, являющегося конечным пользователем программы. Развертыванием базы данных на сервере должен заниматься системный администратор.

Системный администратор должен иметь среднее или высшее профессиональное образование в области информационных технологий. Обязанностями системного администратора являются:

* Развертывание базы данных на сервере;
* поддержание работоспособности системы, устранение неполадок.

Требования к профессиональной квалификации конечных пользователей программы не выставляются. Конечные пользователи программы должны подробно ознакомиться с руководством пользователя.

Климатические условия эксплуатации программы совпадают с климатическими условиями эксплуатации оборудования, на которое будет установлена программа.

4.4 Требования к составу и параметрам технических средств

Для функционирования программы необходим компьютер-сервер, а также как минимум один компьютер с доступом к сети Интернет. Технические требования к серверу и компьютерам конечных пользователей совпадают с техническими требованиями для установки и эксплуатации ОС Microsoft Windows 10 и базы данных Microsoft SQL Server.

4.5 Требования к информационной и программной совместимости

Системные программные средства должны быть представлены лицензионной версией операционной системы Microsoft Windows 10 или выше.

База данных должна работать под управлением СУБД Microsoft SQL Server 2019.

Исходный код программы должен быть реализован на языке программирования C# на основе фреймворка ASP.NET MVC.

Для импортирования или экспортирования данных, формирования отчетов необходимо наличие Microsoft Excel.

4.6 Требования к маркировке и упаковке

Программа представляет собой набор файлов в архиве rar.

4.7 Требования к транспортированию и хранению

Программа хранится в электронном виде в информационной системе ЛМС.

4.8 Специальные требования

Специальные требования к программе не предъявляются.

5 Требования к программной документации

Программная документация должна включать:

* техническое задание на разработку информационной системы, выполненное по ГОСТ 19.201-78;
* руководство пользователя, выполненное по РД 50-34.698-90;

6 Технико-экономические показатели

В рамках данной работы расчет экономической эффективности и предполагаемой годовой потребности не выполняется.

Основным преимуществом разработки является отсутствие на российском рынке информационных систем для учета аренды жилой недвижимости. Система «СУАР» позволяет хранить подробную информацию о местоположении объекта аренды, его инфраструктуре. Система автоматизирует контроль платы за аренду, а также подготовку различной отчетности, что позволит существенно упростить эти операции для сотрудников и сократить время их выполнения.

7 Стадии и этапы разработки

|  |  |
| --- | --- |
| Этап | Дата контрольной точки |
| Анализ предметной области и разработка технического задания | 18 мая 2021 |
| Проектирование базы данных и приложения | 20 мая 2021 |
| Реализация и программной системы | 25 мая 2021 |
| Тестирование и отладка программной системы | 27 мая 2021 |

8 Порядок контроля и приемки

1. Должно быть проведено компонентное тестирование форм ввода и редактирования данных, а также функций фильтрации данных.
2. Должно быть проведено системное тестирование для проверки взаимодействия окон системы.