Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

Допускаю к защите

|  |  |
| --- | --- |
| Руководитель |  |
|  | подпись |
|  | З.А. Бахвалова |
|  | И.О. Фамилия |

|  |
| --- |
| Проектирование и разработка прикладного ПО |
| «Онлайн-регистрация авиабилетов» |
| наименование темы |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по дисциплине

|  |
| --- |
| Информатика |

|  |  |  |
| --- | --- | --- |
| 1.001.00.00 – ПЗ | | |
| обозначение документа | | |
| Выполнил студент |  | ИСТб-20-3 | |  |  | |  | Д.Э. Афанасьев |
|  |  | шифр | |  | подпись | |  | И.О. Фамилия |
| Нормоконтроль |  |  | |  |  | |  | З.А. Бахвалова |
|  |  |  | |  | подпись | |  | И.О. Фамилия |
| Курсовая работа защищена с оценкой | | | | |  | | | |

Иркутск 2023 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **По курсу** | Технологии разработки программных комплексов | | | | | | | | | | | | | | | | |
| **Студенту** | Афанасьеву Д.Э. | | | | | | | | | | | | | | | | |
|  | (фамилия, инициалы] | | | | | | | | | | | | | | | | |
| **Тема работы** | | Проектирование и разработка прикладного ПО. | | | | | | | | | | | | | | | |
| **Исходные данные** | | Книжный магазин. Проанализировать предметную область и реализовать систему в соответствии с жизненный циклом ПО. | | | | | | | | | | | | | | | |
| **Рекомендуемая литература** | | | | | | |  | | | | | | | | | | |
| 1. СТО 005-2015 СИСТЕМА МЕНЕДЖМЕНТА КАЧЕСТВА. Учебно-методическая деятельность. Оформление курсовых проектов (работ) и выпускных квалификационных работ технических специальностей (http://www.istu.edu/structure/57/2506/) | | | | | | | | | | | | | | | | | |
|  | | |  | | | | | | | | | | | | | | |
| Дата выдачи задания | | | « | 16 | » | февраля | | | | | 2023 г. | | | | |
|  | | | | | | | | | | | | | | | | | |
| Задание получил | | | | | | | |  | | | | |  | | Д.Э. Афанасьев | | |
|  | | | | | | | | подпись | | | | |  | | И.О. Фамилия | | |
|  | | | | | | | | |  |  | |  | |  | | |  | |
| Дата представления работы руководителю | | | | | | | | | « | 24 | | » | | мая | | | 2023г. |
|  | | | | | | | | | | | | | | | | | | |
| Руководитель курсовой работы | | | | | | | |  | | | | |  | | З.А. Бахвалова | | |
|  | | | | | | | | подпись | | | | |  | | И.О. Фамилия | | |

**Содержание**

[1 Формирование требований 5](#_Toc134953269)

[1.1 Описание предметной области 5](#_Toc134953270)

[1.2 Описание проблемы 5](#_Toc134953271)

[1.3 Цель 5](#_Toc134953272)

[1.4 Словари 6](#_Toc134953273)

[1.5 Постановка задачи 9](#_Toc134953274)

[1.6 Функциональные требования ПО 10](#_Toc134953275)

[2 Анализ функциональных требований 11](#_Toc134953276)

[2.1 Модель предметной области (IDEF0) 11](#_Toc134953277)

[2.1 Диаграмма вариантов использования Use Case 13](#_Toc134953278)

[2.2 DFD - диаграмма потоков данных 14](#_Toc134953279)

[2.3 Концептуальная модель хранилища данных 15](#_Toc134953280)

[2.4 Описание вариантов использования 16](#_Toc134953281)

[2.5 Диаграммы состояний 20](#_Toc134953282)

[2.6 Описание интерфейса 21](#_Toc134953283)

[3 Проектирование 23](#_Toc134953284)

[3.1 Инструменты 23](#_Toc134953285)

[3.2 Реализация 27](#_Toc134953286)

[3.3 Описание алгоритма 28](#_Toc134953287)

[3.4 Интерфейс 29](#_Toc134953288)

[4 Тестирование 31](#_Toc134953289)

[4.1 Исходный код авто тестов 31](#_Toc134953290)

[5 Документация пользователя 32](#_Toc134953291)

[Заключение 33](#_Toc134953292)

[Список использованных источников 34](#_Toc134953293)

**Введение**

Современные технологии продолжают вносить существенные изменения во все сферы нашей жизни, и розничная торговля не исключение. Вместе с ростом популярности электронных книг и онлайн-платформ для чтения, казалось бы, традиционные книжные магазины должны уступить место виртуальным аналогам. Однако, они все еще имеют свое место в нашем обществе, предлагая не только доступ к печатным изданиям, но и уникальный опыт ощущения и запаха книг.

Один из главных вызовов, с которым сталкиваются книжные магазины, заключается в обеспечении эффективного и удобного обслуживания своих клиентов. Посетители магазинов часто испытывают трудности в поиске нужной книги, что приводит к необходимости обращаться к кассирам с вопросами о наличии и расположении конкретного издания. Это отнимает время и отвлекает персонал магазина от основной задачи – обслуживания покупателей.

Для решения этой проблемы предлагается создание инновационного приложения для поиска информации о наличии и расположении книги в книжном магазине. Данное приложение будет работать локально внутри магазина и позволит клиентам самостоятельно находить интересующие их книги, не обращаясь к кассирам. Это сократит время, затрачиваемое на поиск и улучшит общее качество обслуживания в магазине.

Целью данной курсовой работы является исследование и разработка данного решения. Мы будем рассматривать технические аспекты создания приложения, такие как выбор платформы и инструментов разработки, а также функциональные возможности, необходимые для удобного использования приложения клиентами магазина.

# 1 Формирование требований

## 1.1 Описание предметной области

Заказчик – директор небольшого книжного магазина, занимающегося продажей печатных изданий. Магазин занимается продажей печатных изданий, которые характеризуются названием, автором, описанием, изображением обложки, жанром, годом выпуска, стоимостью, количеством копий в наличии, расположением в магазине (либо один из отделов, либо склад). Издания с одинаковым названием различают по автору. Необходимо автоматизировать процесс получения посетителем информации о печатных изданиях и исключить или минимизировать участие продавца в этом процессе. В магазине используется платформа «1С: Предприятие» для учета товаров и взаимодействия с клиентами. В магазине работает всего несколько сотрудников: директор магазина, занимающийся заказами поставок, два продавца, работающих посменно, уборщица и внештатный специалист по «1С: Предприятие».

## 1.2 Описание проблемы

Посетители магазина часто ищут конкретные печатные издания. Для того, чтобы узнать, есть ли это издание в наличии в данном магазине, посетители обращаются к кассиру. В магазине существует внутренняя система учета товаров и кассир может произвести поиск информации о запрашиваемом издании. Однако, кассир не всегда имеет возможность выполнять такой поиск, так как может быть занят обслуживанием покупателей в очереди. При этом возникает большая вероятность, что человек, который ищет издание покинет магазин. Это приводит к потере потенциальной прибыли и лояльности посетителя. Кроме того, директор магазина хочет узнавать, какие отсутствующие издания посетители хотели бы купить, чтобы повысить продажи.

Предлагаемое решение – разработка приложения, позволяющего посетителям производить поиск имеющихся в магазине печатных изданий и оставлять заявки на закупку отсутствующих изданий.

## 1.3 Цель

Избежать потери прибыли и повысить лояльность клиентов за счет частичной автоматизации процесса получения информации об изданиях.

## 1.4 Постановка задачи

Программа должна представлять собой часть распределенной информационной системы магазина (включающей «1С Предприятие») и будет установлена на находящийся в торговом зале компьютер с ОС Windows.

Программа должна работать с данными, содержащимися в уже имеющейся системой учета издания, созданной на основе «1С: Предприятия». Посетители магазина будут производить поиск издания по названию, программа на экране будет отображать список изданий (даже если нашлось всего одно издание), либо показывать сообщение об отсутствии данных об издании в системе.

В списке пользователь должен будет выбрать один из предложенных вариантов, после чего откроется страница выбранного издания.

На странице издания должна отображаться информация об издании: название, автора, год издания, изображение обложки (если есть в системе), жанр, описание, стоимость, количество в наличии и расположение в магазине. У пользователя должна быть возможность формировать список из выбранных им изданий, чтобы сохранять результаты всех поисков в одном месте. Кроме того, должна быть возможность вернуться к окну поиска.

Если издания нет в наличии, пользователь должен иметь возможность подать заявку на поставку этого издания. Для подачи заявки пользователь должен предоставить свои контакты (телефон и/или почта), также должна быть возможность оставить свое имя. Приложение сохранит данные о заявке в системе, включая издание, контакты и имя пользователя, а также дату создания заявки.

Полученные заявки будут использоваться 1С специалистом магазина при заказе поставок изданий в магазин. Контакты будут собираться для дальнейшего оповещения людей, оставляющих заявки для рассылки оповещений о поступлении изданий. Такая рассылка происходит с использованием уже существующего модуля на базе платформы «1С: Предприятие», поэтому разрабатываемая программа должна использовать базу данных этой платформы. Автоматизацией процессов переноса данных для поставок и рассылки оповещений заниматься не нужно (в штат магазина входит 1С специалист, занимающийся этим).

## 1.5 Словари

**Словарь предметной области**

|  |  |  |
| --- | --- | --- |
| **№** | **Сущность** | **Описание** |
| 1 | Издание/печатное издание | Печатная продукция, продающаяся в магазине. |
| 2 | Заявка | Зафиксированное в системе желание посетителя купить конкретное отсутствующее издание. |
| 3 | Жанр | Категория, к которой относится издание |
| 4 | Стоимость | Цена печатного издания, выраженная в рублях |
| 5 | Расположение в магазине | Конкретные стеллаж и полка, на которых расположено издание в магазине. Записываются в специальном 4-х символьном формате, где 2 первых символа – буквы, идентификатор стеллажа, последние 2 символа – цифры, номер полки на стеллаже (напр. АА01) |
| 6 | Список интересующих изданий | Формируемый посетителем список изданий с информацией об их расположении в магазине. Список нужен для сохранения результатов сбора информации об изданиях в едином месте. Подразумевается, что пользователь сможет сфотографировать его или записать при необходимости. |
| 7 | Стеллаж | Стеллаж в магазине, имеющий идентификатор из 2 букв |
| 8 | Полка | Одна из полок стеллажа. Обладает числовым номером. |

**Словарь по Абботу**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Сущность** | **Глагол** | **Пояснение глагола** |
| 1 | Издание/печатное издание | Искать, отображать |  |
| 2 | Заявка | Формировать, сохранять |  |
| 3 | Жанр | Отображать |  |
| 4 | Стоимость | Отображать |  |
| 5 | Расположение в магазине | Отображать |  |
| 6 | Список изданий | Формировать, распечатывать |  |
| 7 | Стеллаж | - |  |
| 8 | Полка | - |  |

**Объектно-ориентированный словарь**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Сущность** | **Данные** | **Методы** |
| 1 | Издание/печатное издание | Название, автор, жанр, цена, наличие в магазине, расположение в магазине | Искать, отображать |
| 2 | Заявка | Издание, имя, номер телефона, электронная почта, дата | Сохранять |
| 3 | Жанр | Может быть:   1. Детектив 2. Роман 3. Драма 4. Комедия 5. Фантастика 6. Классика 7. Бизнес 8. Наука и техника 9. Философия 10. и т. д. | Отображать |
| 4 | Стоимость | Цена в рублях | Отображать |
| 5 | Расположение в магазине | Идентификатор стеллажа и номер полки | Отображать |
| 6 | Список изданий | Название издания, расположение в магазине | Формировать, распечатывать |
| 7 | Стеллаж | Идентификатор |  |
| 8 | Полка | Числовой номер |  |

## 1.6 Функциональность ПО

**Проверка наличия издания в магазине**

Пользователь должен иметь возможность проверить, есть ли в наличии то или иное издание. Для этого он должен открыть информационное окно издания, произведя поиск по названию. После открытия окна, информация должна быть отображена в информационной строке «Наличие».

**Определение расположения изданий в магазине**

В случае, если издание есть в наличии, пользователь должен иметь возможность узнать на какой полке это издание находится. Эта информация, как и информация о наличии должна находится в окне издания, в информационной строке «Расположение».

**Формирование списка интересующих посетителя изданий, имеющихся в магазине**

Пользователь должен иметь возможность формировать список нужных ему изданий по мере поиска нужной ему информации о наличии и расположении изданий, чтобы в удобной форме получить доступ к собранной информации. Издания можно добавлять в список только если они есть в наличии.

**Оставление посетителем заявки на поставку изданий, которые он хочет купить**

Пользователь в информационном окне издания должен иметь возможность перейти к окну формирования заявки на поставку издания в случае, если издание отсутствует в наличии. При этом, он оставляет свои контактные данные (почта и/или email) и имя. Заявка должна сохраняться в системе.

# 2 Анализ функциональных требований

# 2.1 Модель предметной области (IDEF0)

Модель, представленная на рисунке 1 описывает процесс покупки книг в книжном магазине.

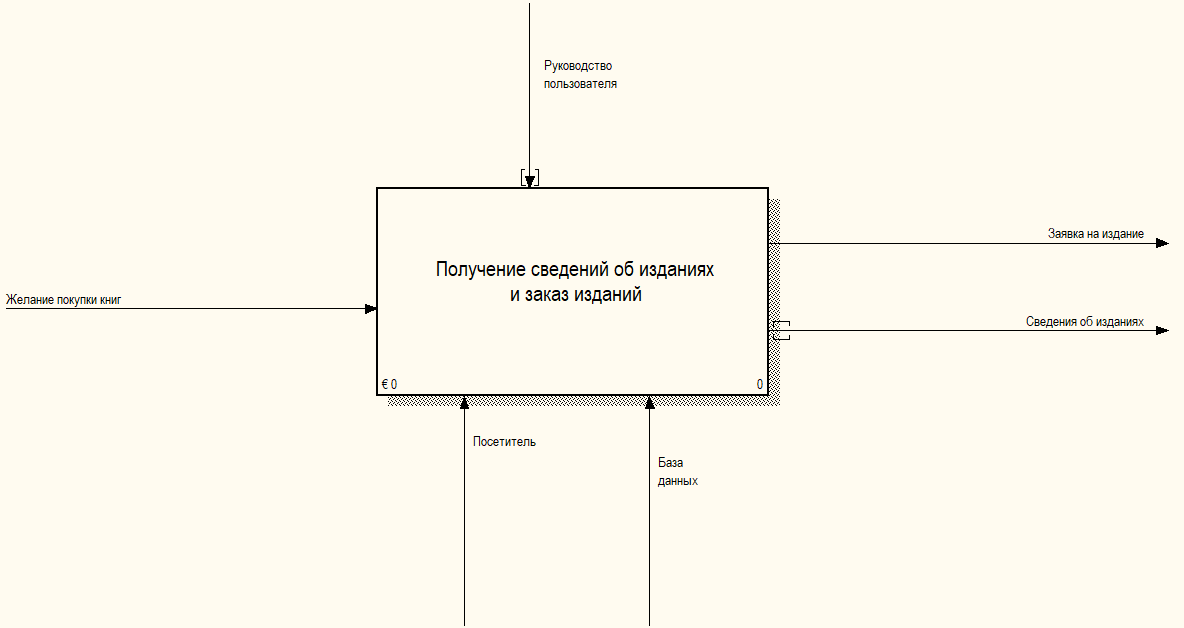


Рисунок 1 – Основной процесс приложения

Целевыми сведениями, получаемыми в ходе выполнения процесса, является информация о наличии и расположении изданий в магазине. Заявка на издание может быть сделана только для тех изданий, которые отсутствуют в магазине во время выполнения процесса, то есть этот результат процесса может быть получен не всегда.

На рисунке 2 представлено описание декомпозиции модели первого уровня.

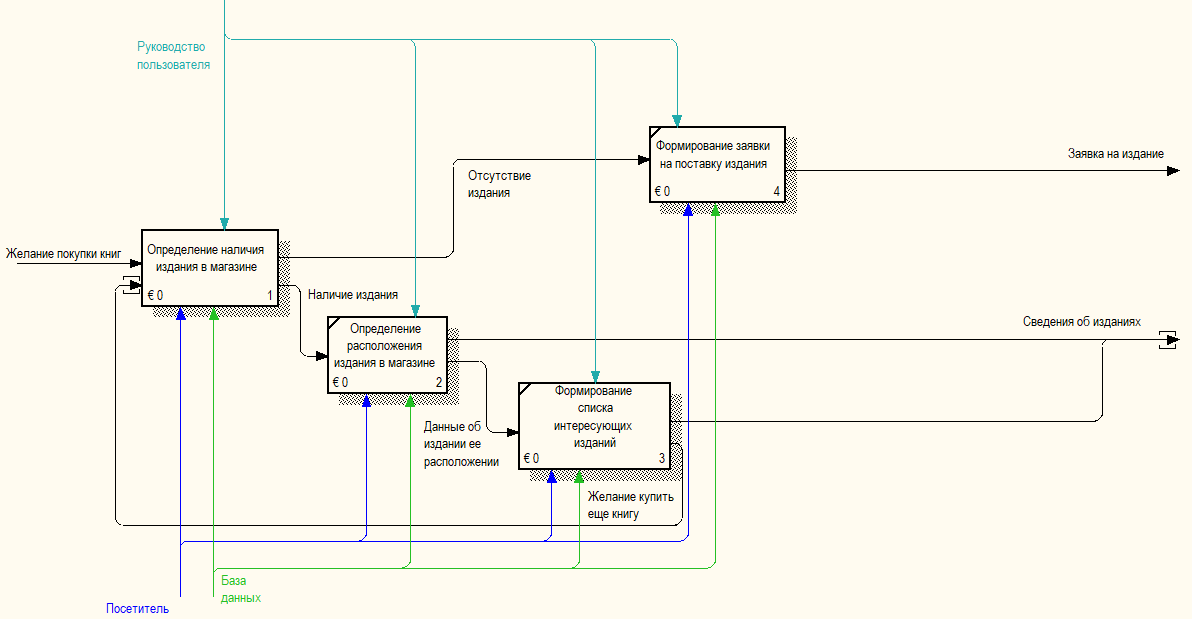


Рисунок 2 – Декомпозиция процесса покупки книг

Процесс покупки книг в магазине включает 4 подпроцесса. При желании купить конкретное издание, посетитель, используя программу, проверяет наличие издания в магазине. Далее есть два варианта развития событий. Если издания в наличии нет, посетитель может сформировать в программе заявку на поставку издания, чтобы приобрести его в будущем. Второй вариант, если издание есть в наличии, предусматривает определение расположения издания в магазине с помощью программы, при этом, если посетителю необходимо купить несколько печатных изданий, для каждого нового издания процесс покупки повторяется с самого начала. Из всех необходимых посетителю изданий он может сформировать список в той же программе, содержащий информацию о расположении изданий и используемый далее при поиске изданий на полках.

Подразумевается, что собранные с помощью программы сведения, помогут посетителю быстрее найти и собрать издания с полок и при этом не отвлекать кассира от обслуживания других покупателей.

На рисунке 3 описывается декомпозиция процесса определения наличия издания в магазине.

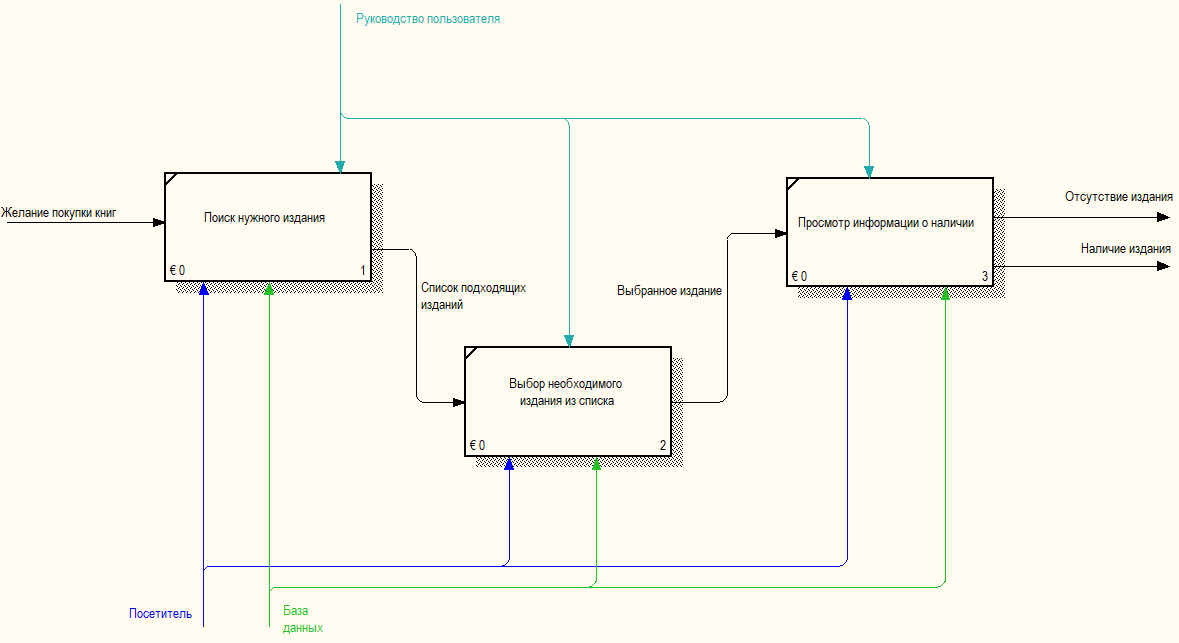


Рисунок 3 - Декомпозиция процесса определения наличия издания

Чтобы у компьютера не скапливались очереди, а посетители могли быстро производить все необходимые действия с программой, все этапы должны быть не загружены действиями. Таким образом, поиск будет производится по самому минимальному требуемому набору характеристик, а уже из списка будет выбираться необходимое издание. Например, поиск по названию и быстрый просмотр нескольких вариантов будет гораздо быстрее, чем заполнение целой формы, включающей название, авторов, жанры, тип обложки и других возможных характеристик.

# 2.2 Описание вариантов использования

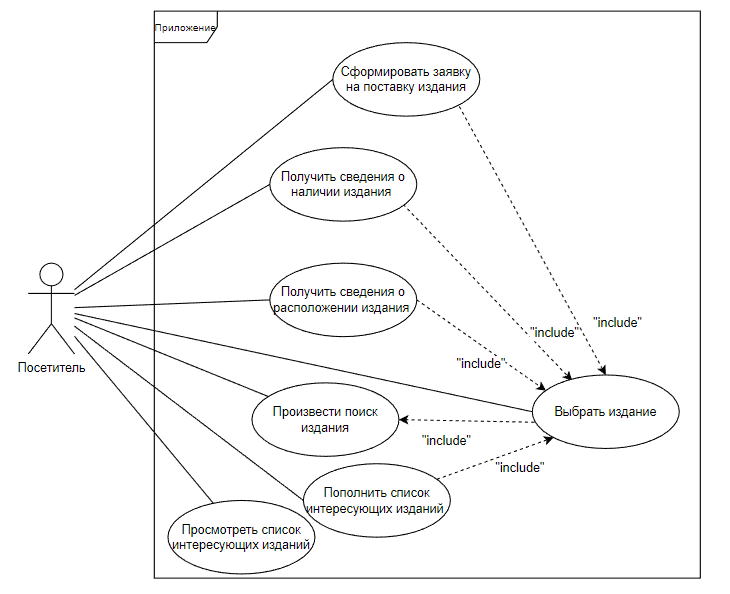


Рисунок 4 – Диаграмма, описывающая функции посетителя

Функции посетителя магазина соответствуют процессам, описанным в модели предметной области.

Поскольку все функции приложения тесно связаны с изданиями, выбор издания – первоочередная функция посетителя, ведь именно с выбранным изданием он и будет работать в приложении.

Функции продавца и администратора системы не описаны, так как они выполняются в рамках других процессов, отличных от тех, что рассматриваются здесь. Кроме того, они слабо связаны с разрабатываемым приложением.

# 2.3 Описание потоков данных

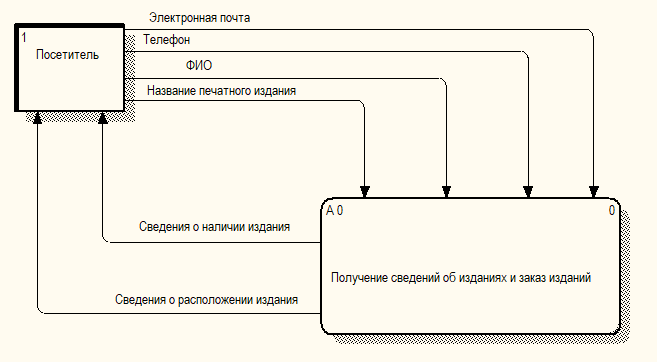


Рисунок 5 – Диаграмма, описывающая потоки данных. Уровень A0

Помимо посетителя, внешней сущностью является система учета на основе «1С: Предприятие», но, поскольку части этой системы учета используются внутри процесса, то обмен данными с ней будет рассмотрен при декомпозиции.

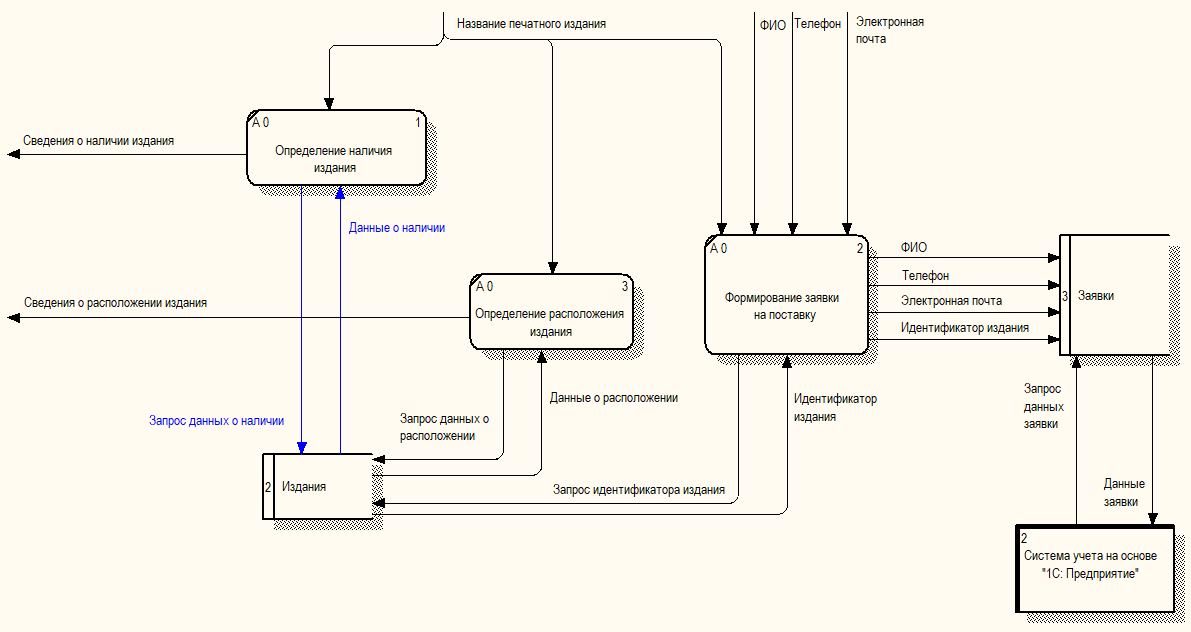


Рисунок 6 - Диаграмма, описывающая потоки данных. Уровень A1

Как и было упомянуто ранее, добавилась дополнительная внешняя сущность, которая получает заявки для дальнейшей рассылки уведомлений.

# 2.4 Концептуальная модель хранилища данных

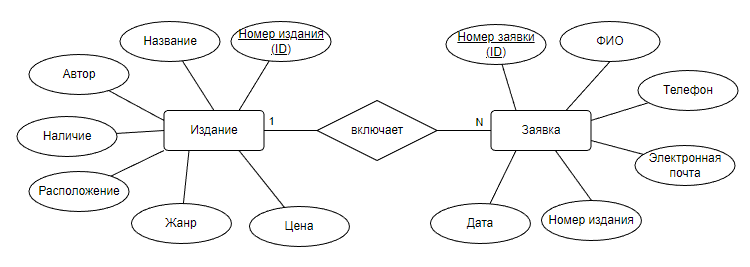


Рисунок 7 – ER-модель

Поскольку хранилище данных уже существует в виде реляционной базы данных, на диаграмме показана только та ее часть, которая будет использоваться приложением.

Каждая заявка включает одно издание, но каждое издание может быть включено во множество заявок.

# 2.5 Описание вариантов использования

## Вариант использования «Получение сведений о наличии»

Спецификация варианта использования «Получить сведения о наличии издания»

**Цель:** Узнать, в наличии ли издание

**Активные субъекты:** Посетитель

**Краткое описание:** Активный субъект может получить информацию о наличии интересующего его издания в данном магазине

**Предусловия:** нет.

**Постусловия:** нет.

**Основной поток событий:**

1. Посетитель производит поиск издания по названию
2. Система формирует список соответствующих изданий
3. Посетитель выбирает нужное издание
4. Система отображает сведения о наличии издания
5. Посетитель просматривает информацию о наличии

**Альтернативные потоки событий:** нет.

**Сценарий обработки ошибок:** нет.

**Специальные требования:** нет.

**Дополнительные замечания:** нет.

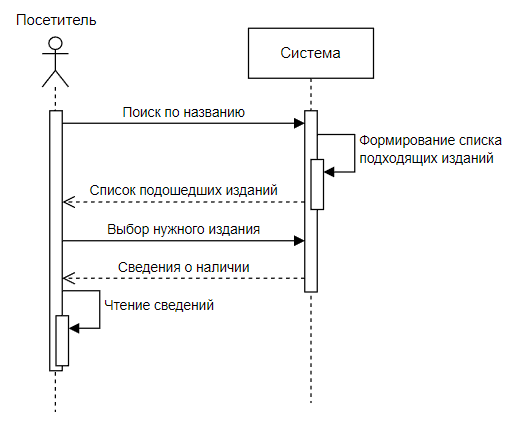


Рисунок 8 - Диаграмма последовательностей варианта использований *«Получить сведения о наличии издания»*

## Вариант использования «Получение сведений о расположении»

Спецификация варианта использования «Получить сведения о расположении издания»

**Цель:** Узнать в каком месте можно найти издание

**Активные субъекты:** Посетитель

**Краткое описание:** Активный субъект может получить информацию о расположении интересующего его издания в данном магазине

**Предусловия:** нет.

**Постусловия:** нет.

**Основной поток событий:**

1. Посетитель производит поиск издания по названию
2. Система формирует список соответствующих изданий
3. Посетитель выбирает нужное издание
4. Система отображает сведения о расположении издания
5. Посетитель просматривает информацию о расположении

**Альтернативные потоки событий:** нет.

**Сценарий обработки ошибок:** нет.

**Специальные требования:** нет.

**Дополнительные замечания:** нет.

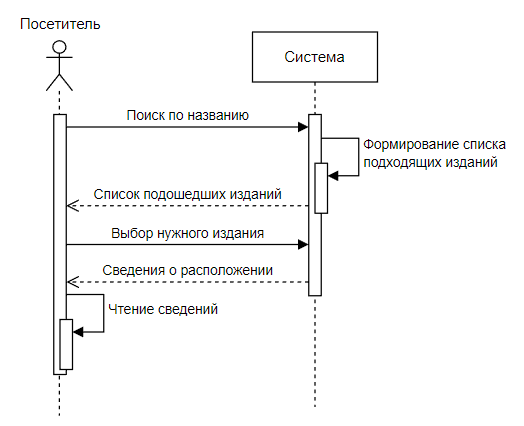


Рисунок 9 - Диаграмма последовательностей варианта использований *«Получить сведения о расположении издания»*

## Вариант использования «Добавить в список интересующих изданий»

Спецификация варианта использования «Добавить в список интересующих изданий»

**Цель:** Сохранить результат поиска сведений об издании

**Активные субъекты:** Посетитель

**Краткое описание:** Активный субъект может добавлять издания в список, чтобы сохранять результаты получения сведений о наличии и расположении.

**Предусловия:** должен быть выполнен вариант использования «Получить сведения о наличии издания» или «Получить сведения о расположении издания».

**Постусловия:** список должен быть очищен.

**Основной поток событий:**

1. Посетитель добавляет выбранное издание в список

**Альтернативные потоки событий:** нет.

**Сценарий обработки ошибок:** нет.

**Специальные требования:** нет.

**Дополнительные замечания:** нет.

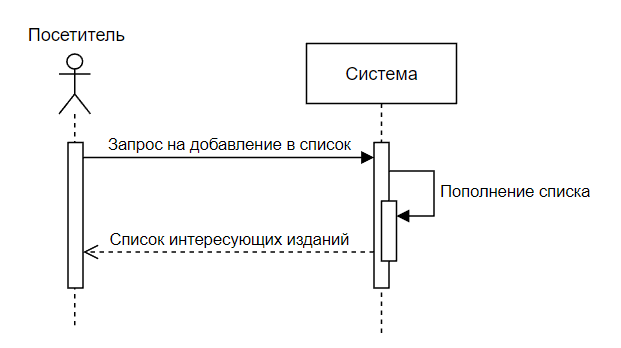


Рисунок 10 - Диаграмма последовательностей варианта использований *«Добавить в список интересующих изданий»*

## Вариант использования «Формирование заявки на поставку»

Спецификация варианта использования «Сформировать заявку на поставку издания»

**Цель:** запросить у магазина заказать определенную книгу и уведомить о ее поступлении.

**Активные субъекты:** Посетитель

**Краткое описание:** Активный субъект запросить поставку издания и уведомить себя, оставив свои контактные данные.

**Предусловия:** должен быть выполнен вариант использования «Получить сведения о наличии издания» (результат – издание отсутствует).

**Постусловия:** нет.

**Основной поток событий:**

1. Посетитель открывает форму создания заявки на выбранную книгу
2. Посетитель заполняет контактную информацию
3. Посетитель отправляет заявку
4. Система сохраняет заявку

**Альтернативные потоки событий:** нет.

**Сценарий обработки ошибок:** нет.

**Специальные требования:** нет.

**Дополнительные замечания:** нет.

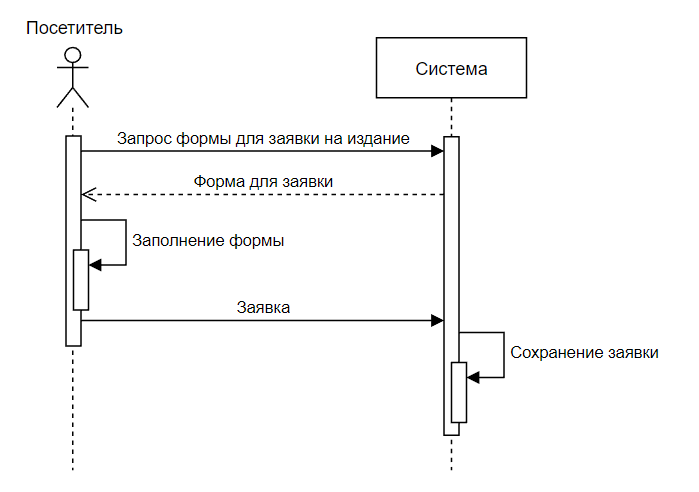


Рисунок 11 - Диаграмма последовательностей варианта использований *«Сформировать заявку на поставку издания»*

# 2.6 Диаграммы состояний

# 2.7 Описание интерфейса

Таблица 4 – Описание элементов интерфейсов

|  |  |  |  |
| --- | --- | --- | --- |
| **Окно** | **Функция** | **Элемент интерфейса** | **Описание элемента** |
| Окно поиска | Получение сведений об издании | Поле ввода названия | Поле, в которое пассажир вводит название интересующей книги |
| Получение сведений об издании | Кнопка поиска | После ввода названия запускается поиск по этому названию с помощью кнопки |
| Получение сведений об издании | Таблица результатов поиска | Таблица, в которой отображается список найденных изданий (название, автор, жанр). Каждый элемент списка позволяет перейти в окно издания |
| Сохранение интересующих изданий | Таблица сохраненных изданий | Таблица, в которой отображается список выбранных пользователем изданий (название, наличие, расположение). Каждый элемент списка позволяет перейти в окно издания |
| Сохранение интересующих изданий | Кнопка обнуления списка | Кнопка, очищающая таблицу выбранных пользователем изданий |
| Получение сведений об издании | Текстовое поле основных данных | Поле, содержащее название и автора издания |
| Получение сведений о наличии издания | Текстовое поле наличия | Поле, содержащее информацию о наличии издания в магазине |
| Получение сведений о расположении издания | Текстовое поле расположения | Поле, содержащее информацию о расположении издания в магазине |
| Создание заявки на поставку издания | Кнопка создания заявки | Кнопка, отображающаяся только при отсутствии издания и открывающая модальное окно с формой заявки |
| Модальное окно заявки | Создание заявки на поставку издания | Форма заявки | Форма с полями для контактных данных и кнопкой отправки заявки |
| Создание заявки на поставку издания | Кнопка закрытия формы | Закрывающая окно заявки кнопка |



Рисунок 13 – Диаграмма интерфейсных классов

На рисунке 13 изображена диаграмма интерфейсных классов, где отображены главные окна и их наполнение и функции, выполняемые на этих окнах. Есть 2 обычных окна (окно поиска, окно издание) и 1 модальное окно, открывающееся в окне издания (модальное окно создания заявки).

## 3 Проектирование

## 3.1 Инструменты

ПО будет является десктоп-приложением, но работать на основе веб технологий. Будет использован инструментарий программных средств:

1. **Electron** - это фреймворк для разработки кросс-платформенных приложений с использованием веб-технологий, таких как HTML, CSS и JavaScript. Позволяет максимально быстро создать десктоп-приложения, имея опыт работы с веб стеком.
2. **Bootstrap** - фреймворк для создания эстетически приятного и отзывчивого пользовательского интерфейса без написания собственных стилей CSS.
3. **jQuery** - библиотека JavaScript, заметно упрощающая манипуляции с элементами HTML и создания интерактивности.
4. **Node.js** - платформа для выполнения JavaScript-кода вне браузера, которая обеспечивает возможность написания ядра приложения.
5. **Express.js** - гибкий фреймворк на платформе Node.js. Значительно ускорит разработку ядра приложения за счет уже созданной логики.
6. **Selenium** – библиотека, позволяющая провести тестирование интерфейса. Бесплатная, открытый исходный код.

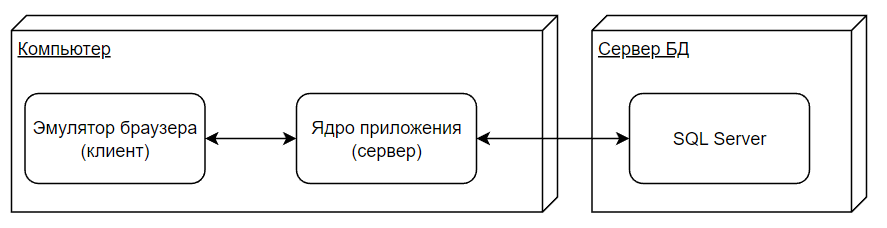


Рисунок 14 – Диаграмма развертывания приложения

Среда работы приложения – компьютер, расположенный в книжном магазине. Electron будет запускать эмулятор браузера в окне Windows, через этот интерфейс пользователь будет взаимодействовать с ядром приложения, являющимся аналогом серверной части в классических веб-архитектурах. Само ядро будет общаться с СУБД для обмена данными, база данных расположена на отдельном компьютере в локальной распределенной сети магазина.

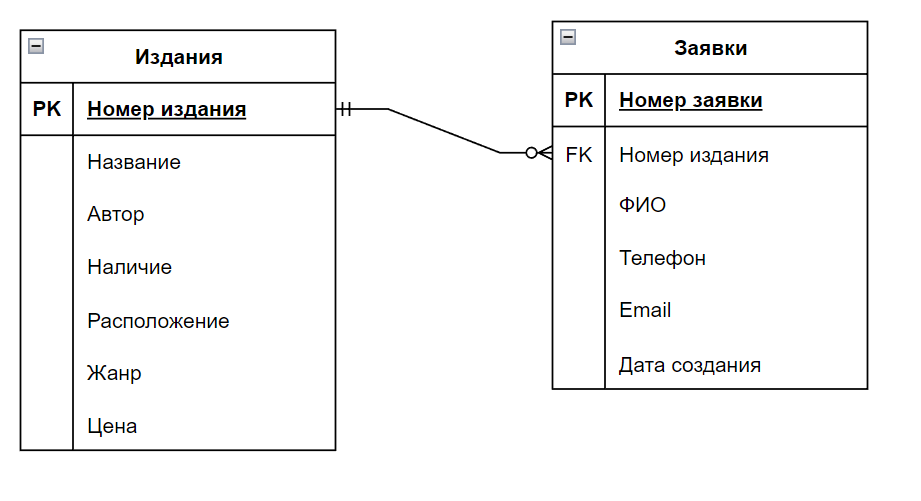


Рисунок 15 – Логическая модель базы данных

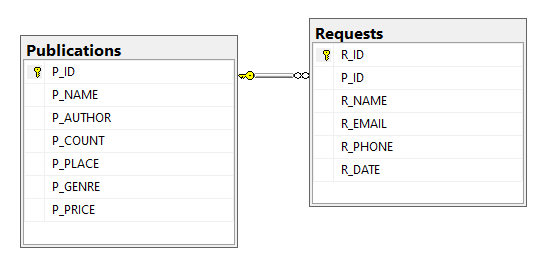


Рисунок 16 – Физическая модель базы данных

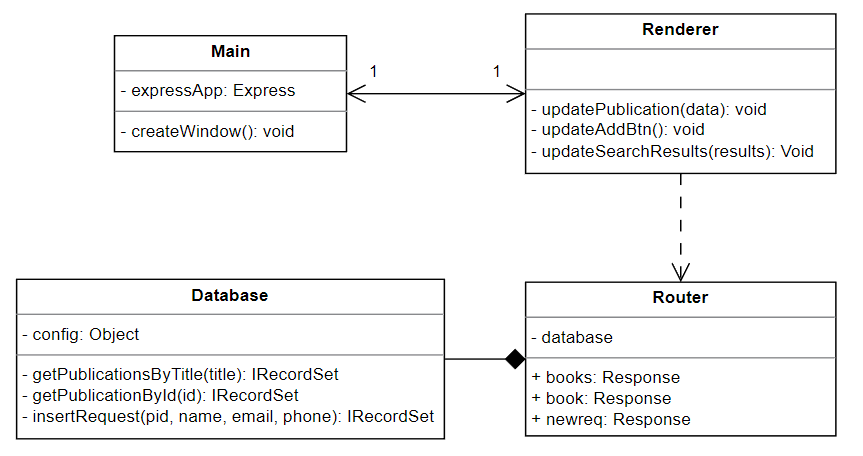


Рисунок 17 – Взаимодействие классов приложения

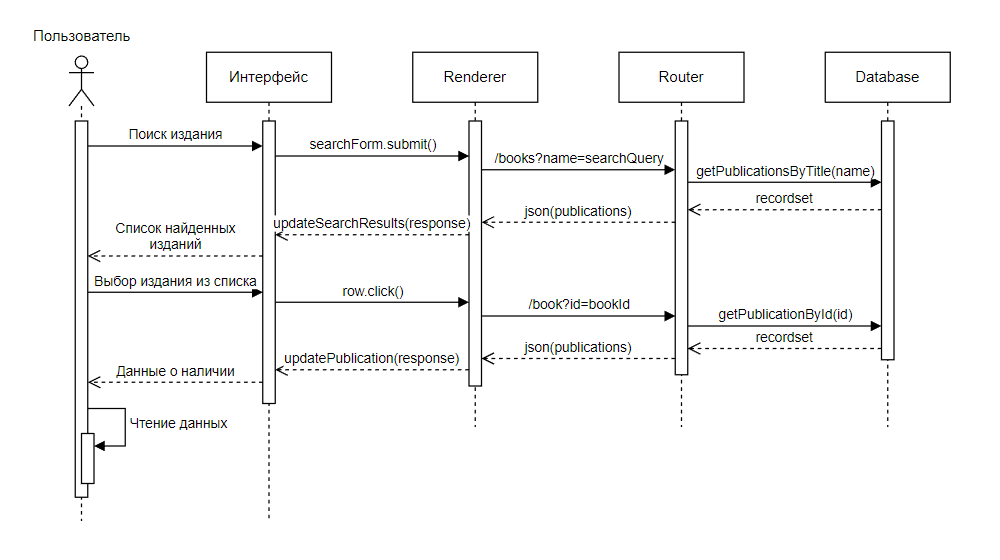


Рисунок 18 – Диаграмма, описывающая последовательность действий при выполнении функции «Получение сведений о наличии»

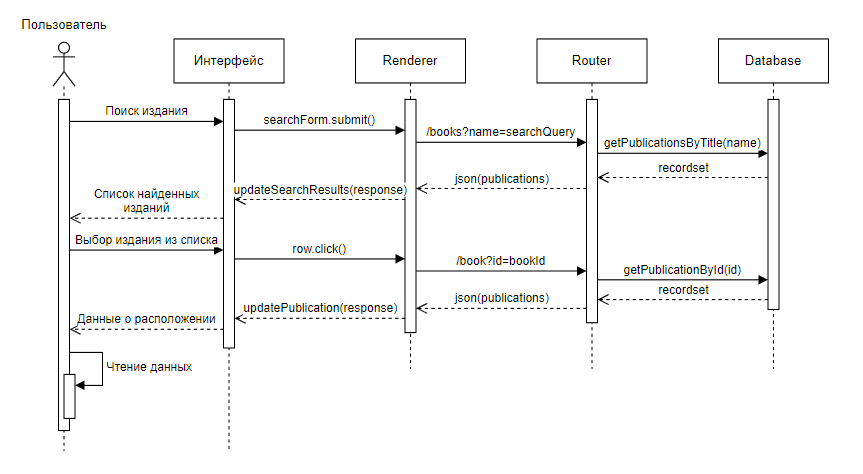


Рисунок 19 – Диаграмма, описывающая последовательность действий при выполнении функции «Получение сведений о расположении»

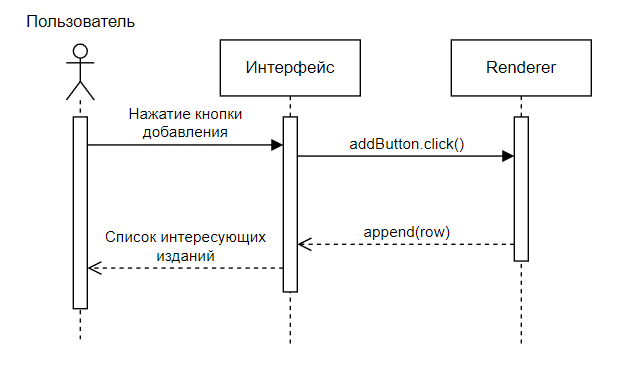


Рисунок 20 – Диаграмма, описывающая последовательность действий при выполнении функции «Формирование списка интересующих изданий»

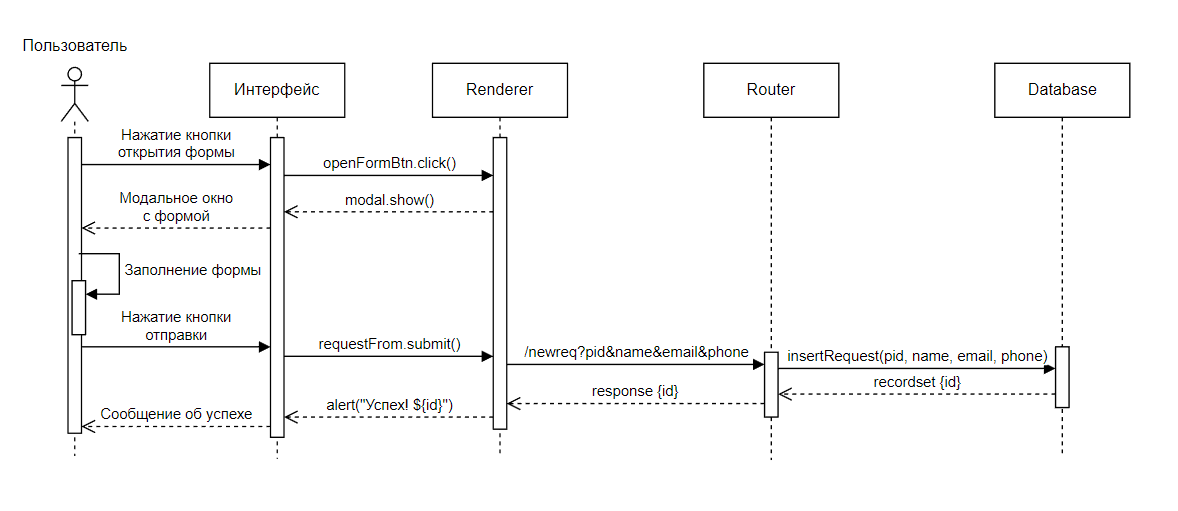


Рисунок 21 – Диаграмма, описывающая последовательность действий при выполнении функции «Формирование списка интересующих изданий»

# 3.2 Реализация

**Main** – точка входа в приложение, отвечает за создание окна приложения и запуск ядра приложения.

**Renderer** – контроллер, отслеживает и изменяет интерфейс, обменивается данными с ядром приложения.

**Database** содержит методы для взаимодействия с базой данных SQL Server. Является частью ядра приложения.

**Router** – маршрутизатор запросов, отправляемых из контроллера. Маршруты представлены в таблице 5.

Таблица 5 – Спецификация функций-обработчиков ядра приложения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Маршрут | Метод | Параметры | Описание |
| 1 | /books | GET | name – название издания | Получение списка изданий, полностью или частично совпадающих с name |
| 2 | /book | GET | id – идентификатор издания | Получение полной информации о конкретном издании, включая наличие и расположение |
| 3 | /newreq | POST | pid – идентификатор издания  name – имя пользователя  email – эл. почта пользователя  phone – телефон пользователя | Сохранение заявки на поставку издания, созданной пользователем |

## 3.3 Описание алгоритма

Приложение условно разделено на 2 части. Это разделение похоже на стандартное деление в веб проектах: frontend и backend (ядро приложения). Пользователь взаимодействует с интерфейсом, в ответ на это взаимодействие, контроллер совершает тот или иной запрос в ядро приложения для получения или сохранения данных. После получения ответа от ядра, контроллер изменяет содержимое интерфейса. Ответы приходят в формате JSON.

Таким образом производятся 3 основные функции: получение сведений о наличии, получение сведений о расположении, формирование заявки на поставку издания.

Протокол общения между frontend и ядром – HTTP.

Для выполнения функции формирования списка интересующих изданий, ядро приложения не задействуется, список формируется только на стороне frontend путем простого добавления строк в таблицу.

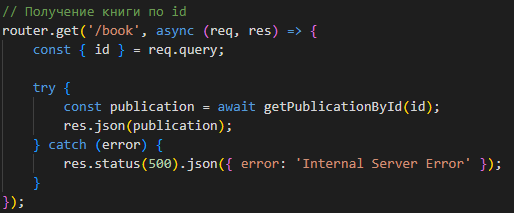


Рисунок 22 – Функция-обработчик, получающая данные из БД

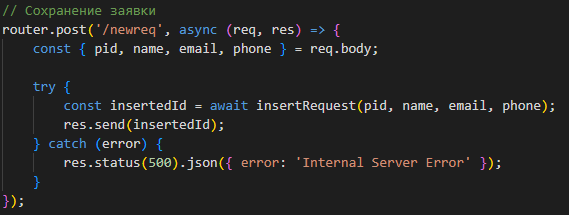


Рисунок 23 – Функция-обработчик, создающая запись в БД



Рисунок 24 – Обработчик, совершающий вставку в список выбранных изданий

## 3.4 Интерфейс

Макет интерфейса были созданы в программе Figma

Ссылка на макет интерфейса: (<https://www.figma.com/file/ywfkrHjxoXuHWRZwvo0rXI/trpk-coursework?type=design&t=qFDDErqvfVOF3Sq1-6>)



Рисунок 25 – Основное окно

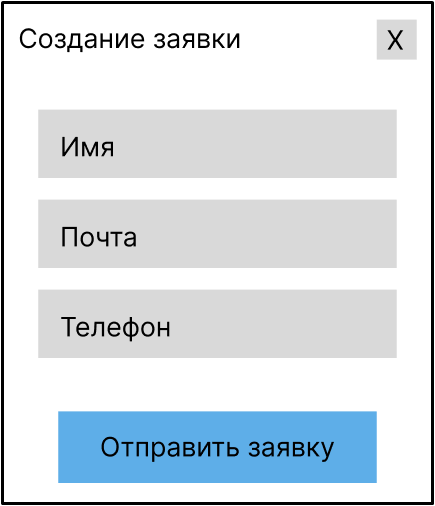


Рисунок 26 – Модально окно с формой заявки

## 4 Тестирование

## 4.1 Исходный код авто тестов

## 5 Документация пользователя

## Заключение

В ходе данной работы было проведено проектирование системы приложения для поиска информации о наличии и расположении книг в книжном магазине. Этот процесс включал определение основных функциональных и нефункциональных требований, анализ архитектуры приложения, разработку базы данных и пользовательского интерфейса.

Анализ требований и проектирование приложения помогли лучше понять потребности пользователей и определить необходимые компоненты системы. В ходе работы были построены различные диаграммы, относящиеся к самым важным аспектам приложения.

Благодаря использованию Selenium был получен опыт в автоматизации веб-браузеров и извлечении данных с веб-страниц, а также автоматического тестирования интерфейса.

Использование Bootstrap для разработки пользовательского интерфейса позволило создать эстетически приятный и отзывчивый дизайн.

В целом, проведенное проектирование системы, работа с выбранными технологиями и разработка приложения позволили расширить знания и навыки в различных областях, связанных с разработкой приложений, баз данных, десктопных приложения с использованием веб технологий.

## Список использованных источников

GitHub репозиторий проекта:

(<https://github.com/Hodache/trpk>)