МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет *компьютерных наук*

Кафедра *информационных систем*

*Онлайн-портал сети библиотек LibOn*

*Курсовой проект*

*по дисциплине*

*Технологии программирования*

09.03.21 *Информационные системы и технологии*

*Информационные системы и сетевые технологии*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*А. Ю. Протодьяконов, 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*М. А. Ломакин, 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*А. Д. Дроботенко, 3 курс, д/о*

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Иванов И. Ю.*

Воронеж 2019

Оглавление

[Оглавление 2](#_Toc10527550)

[Введение 3](#_Toc10527551)

[Используемые определения 4](#_Toc10527552)

[1 Постановка задачи 5](#_Toc10527553)

[2 Анализ предметной области 7](#_Toc10527554)

[2.1 Диаграммы IDEF0 7](#_Toc10527557)

[2.2 Диаграмма вариантов использования 15](#_Toc10527558)

[2.3 Схема базы данных 16](#_Toc10527559)

[2.4 Средства реализации 17](#_Toc10527560)

[3 Тестирование 18](#_Toc10527561)

[3.1 План тестирования 18](#_Toc10527562)

[3.2 Результаты тестирования 18](#_Toc10527563)

[4 Реализация 22](#_Toc10527564)

[4.1 Диаграмма состояний 22](#_Toc10527565)

[4.2 Диаграмма деятельности 23](#_Toc10527566)

[4.3 Диаграмма классов 24](#_Toc10527567)

[4.4 Диаграммы последовательности и взаимодействия 25](#_Toc10527568)

[4.5 Диаграмма развёртывания 29](#_Toc10527569)

[Заключение 30](#_Toc10527570)

# Введение

Сегодня, в эпоху информации, традиционные библиотеки теряют актуальность. Все знания, все развлечения человечества переходят в цифровой формат. И книги - не исключение. Поэтому всё больше библиотек стараются не отставать от прогресса и начинают предоставлять свои услуги онлайн.

Сеть библиотек LibOn – один из примеров подобного современного подхода. Целью сети является предоставить своим читателям возможность ознакомиться с выбором книг, не выходя из дома, а также позволить им читать электронные версии интересующих изданий. Именно получение лицензий на электронные издания является основной причиной запустить онлайн-портал.

Важными факторами в предоставлении онлайн-услуг для LibOn являются простота использования и доступность их сервиса. При этом на данный момент необходимый функционал ограничен внедрением базы доступных изданий и возможность заказа (взять в пользование) электронных версий книг. Поэтому мобильное приложение является избыточным вариантом реализации целей библиотеки. В то же время онлайн-портал – это простое решение, с возможностью долгосрочной поддержки, которое доступно для большинства пользователей сети Интернет.

Целью данного курсового проекта была поставлена разработка web-сайта, который удовлетворит обозначенные цели сети библиотек LibOn.

Для достижения поставленной цели были определены следующие задачи:

* определение технических требований к web-сайту
* проектирование web-сайта средствами языка UML
* разработка базы данных для хранения информации о книгах и пользователях
* разработка программной реализации web-сайта
* тестирование

# Используемые определения

* **«Сайт»** - проект **«Онлайн-портал сети библиотек LibOn»**, являющийся web-сайтом с использование базы данных SQL.
* **«Гость»** - неавторизованный на портале человек, пользующийся ограниченным функционалом сайта.
* **«Пользователь»** - авторизованный на портале человек, пользующийся функционалом сайта. Обычно читатель сети библиотек **LibOn**.
* **«Администратор»** - человек, имеющий доступ к расширенному функционалу сайта, работник сети библиотек **LibOn**.
* **«Услуга»** - возможность, предоставляемая сетью библиотек **LibOn** пользователю онлайн-портала.
* **«Книга»** - продукция, доступ к которой предоставляется пользователям онлайн-портала. Ей выступают книги, альбомы, газеты, журналы и т. д.
* **«Таблица»** - таблица базы данных SQL, используемая для записи или возвращения каких-либо данных.
* **«Заказ», «Взять в пользование»** - возможность получения доступа к выбранной книге на ограниченное время.

# Постановка задачи

Необходимо разработать онлайн-портал для сети библиотек **LibOn**, который позволит читателю ознакомиться с ассортиментом сети. Кроме того, в распоряжении библиотек имеются электронные версии некоторых изданий, которые читатель мог бы получить через онлайн-портал.

Система должна предусматривать возможность добавления или удаления книг в базе администратором.

Система хранит данные о каждой книге. Причём для печатного и электронного издания предполагается разная запись с отличным типом. Книга характеризуется следующими параметрами:

* Уникальный идентификатор
* Название
* Автор
* Время аренды (для электронных изданий)
* Тип издания
* Идентификатор рубрики
* Год издания

Все книги в библиотеке разделены по рубрикам (поэзия, проза, учебные пособия т. д.). Каждая рубрика характеризуется её уникальным идентификатором и названием.

Наконец, в системе хранятся данные о пользователях. Для читателей и администраторов предусмотрены разные записи. Читатель характеризуется следующими параметрами:

* Уникальный идентификатор
* Имя
* Фамилия
* E-mail

В системе должен храниться так же пароль читателя для входа на портал. Параметры администратора совпадают с параметрами читателя.

При заказе электронного издания создаётся специальная запись, характеризующаяся следующими параметрами:

* Уникальный идентификатор заказа
* Дата заказа (книга предоставляется временно)
* Идентификатор книги
* Идентификатор пользователя

Взаимодействие с печатными изданиями на портале не предполагается.

Таким образом, в системе можно выделить три уровня доступа:

* Гость
* Пользователь
* Администратор

Также система предполагает следующие ограничения:

* Пользователь может брать только один экземпляр одной книги
* Книга может принадлежать только одной рубрике
* Пользователь не может заказать на портале печатное издание

# Анализ предметной области



# Диаграммы IDEF0

Чтобы наглядно отобразить бизнес-процессы онлайн-библиотеки LibOn, использована нотация IDEF0.

Назначение данных диаграмм – понять, какие процессы происходят в системе, из каких действий они состоят, какие возможности обеспечивают. Более того, данные диаграммы помогут обеспечить атомарность требований к системе.

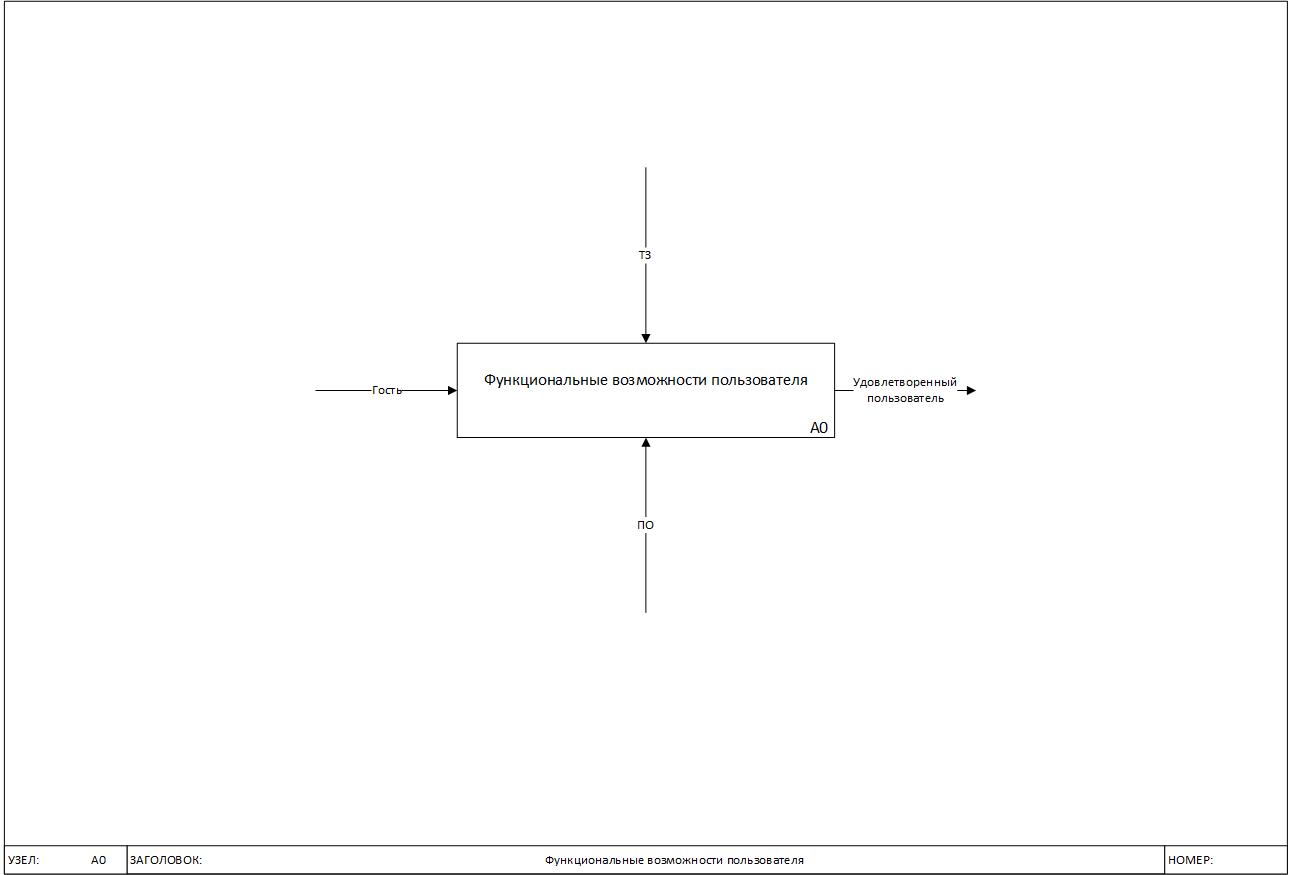
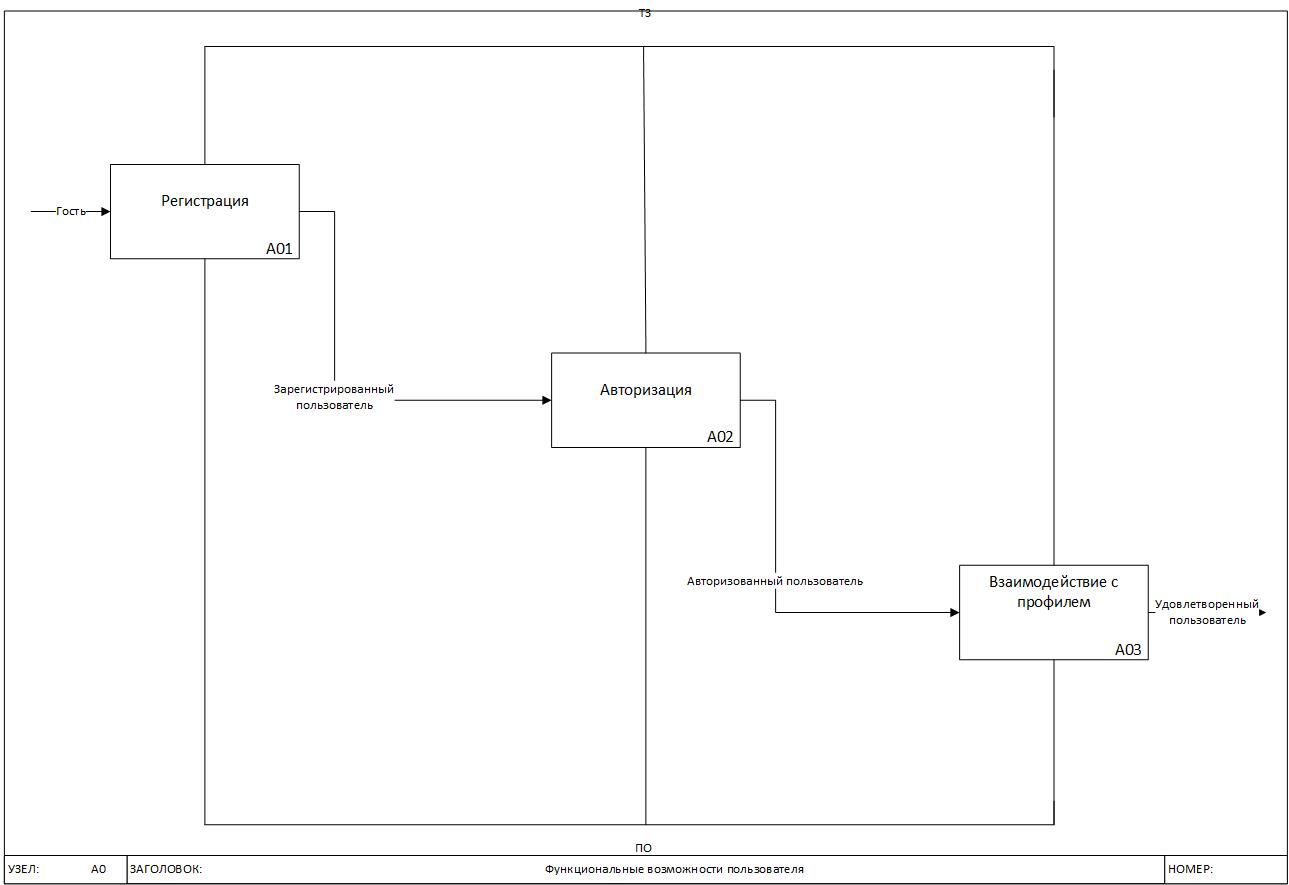


Рисунок 1 – Блок возможностей обычного пользователя

Рисунок 2 – Разбиение блока

Из разбиения видно, что возможности обычного пользователя относительно него самого делятся на три блока: регистрация, авторизация и взаимодействие с профилем. Подробнее рассмотрим каждый из них.

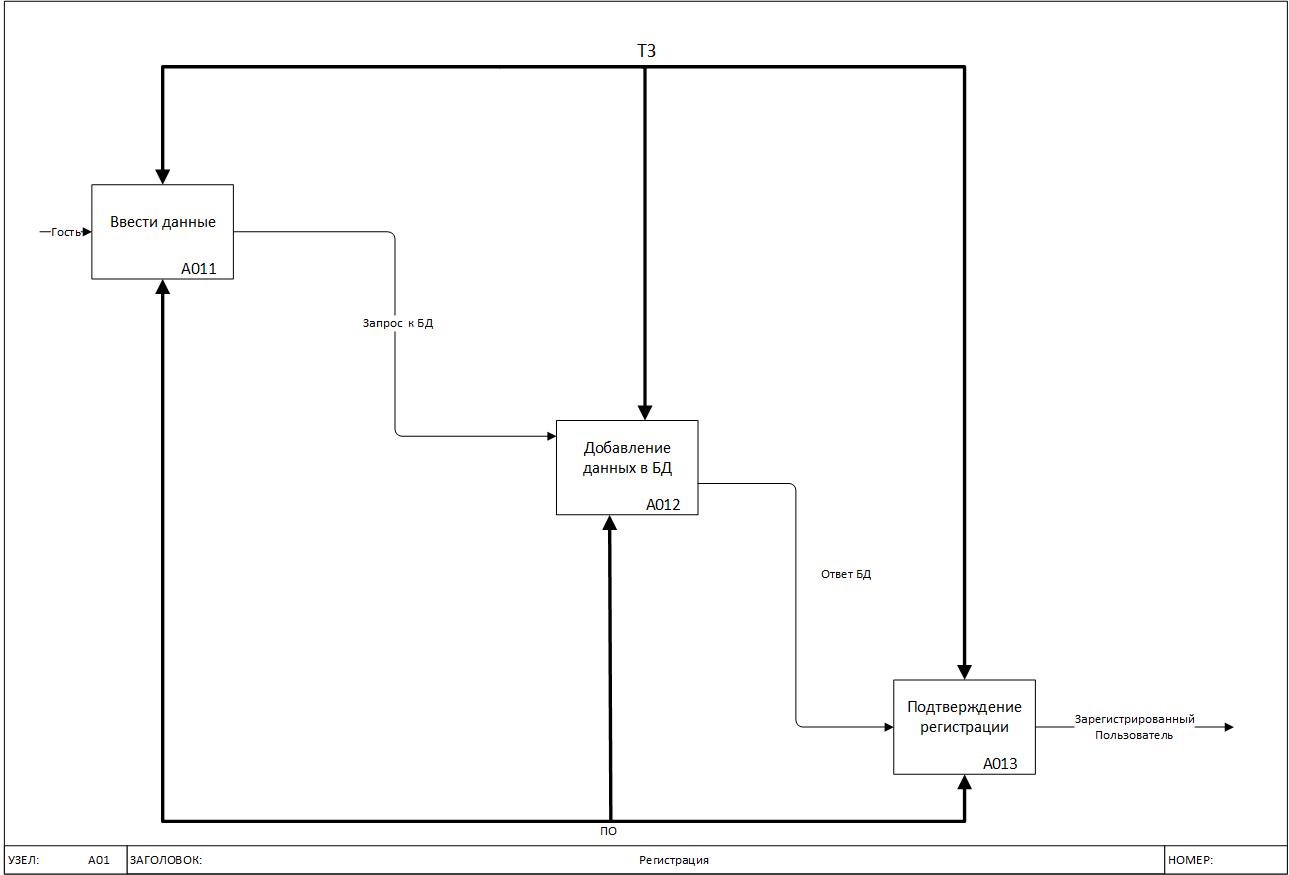


Рисунок 3 – Разбиение регистрации пользователя

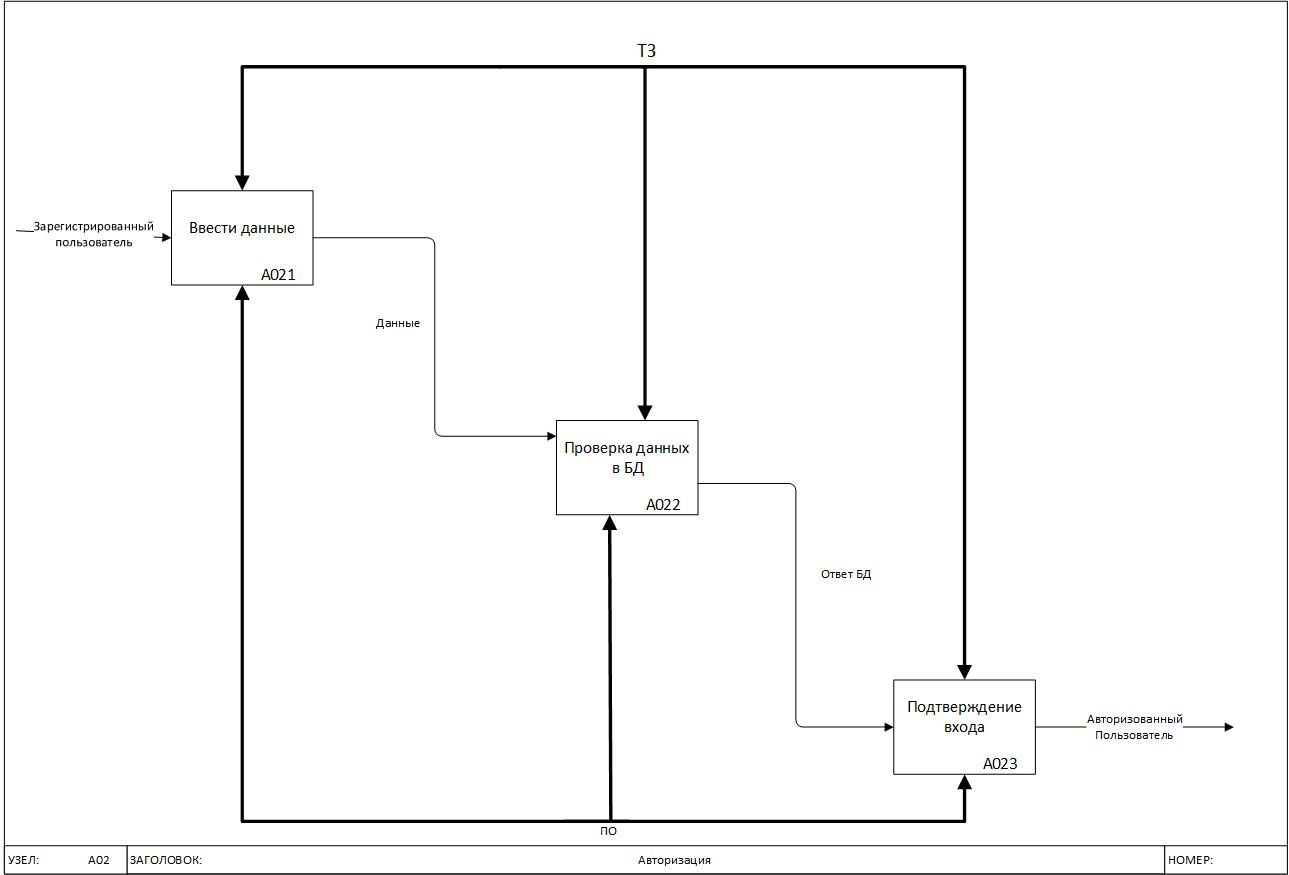


Рисунок 4 – Разбиение авторизации пользователя

По итогу разбиения хорошо видны конкретные действия пользователя и ожидаемая реакция системы. Теперь будет просто описать варианты использования системы и построить модель базы данных, хранящую все необходимые для работы системы данные. Однако перед этим необходимо выполнить разбиение возможностей взаимодействия с библиотекой и возможностей администратора онлайн-портала.

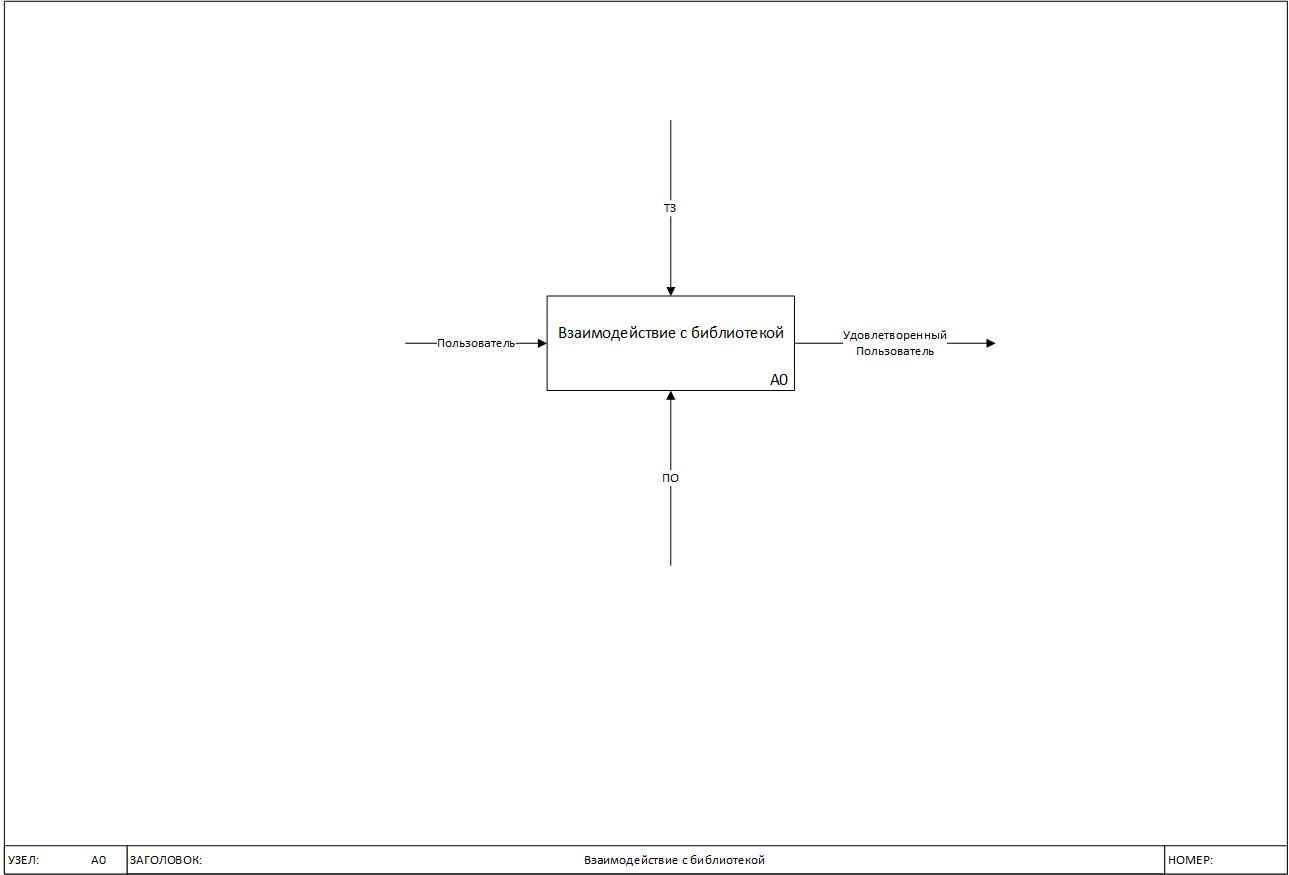


Рисунок 5 – Блок взаимодействия с библиотекой

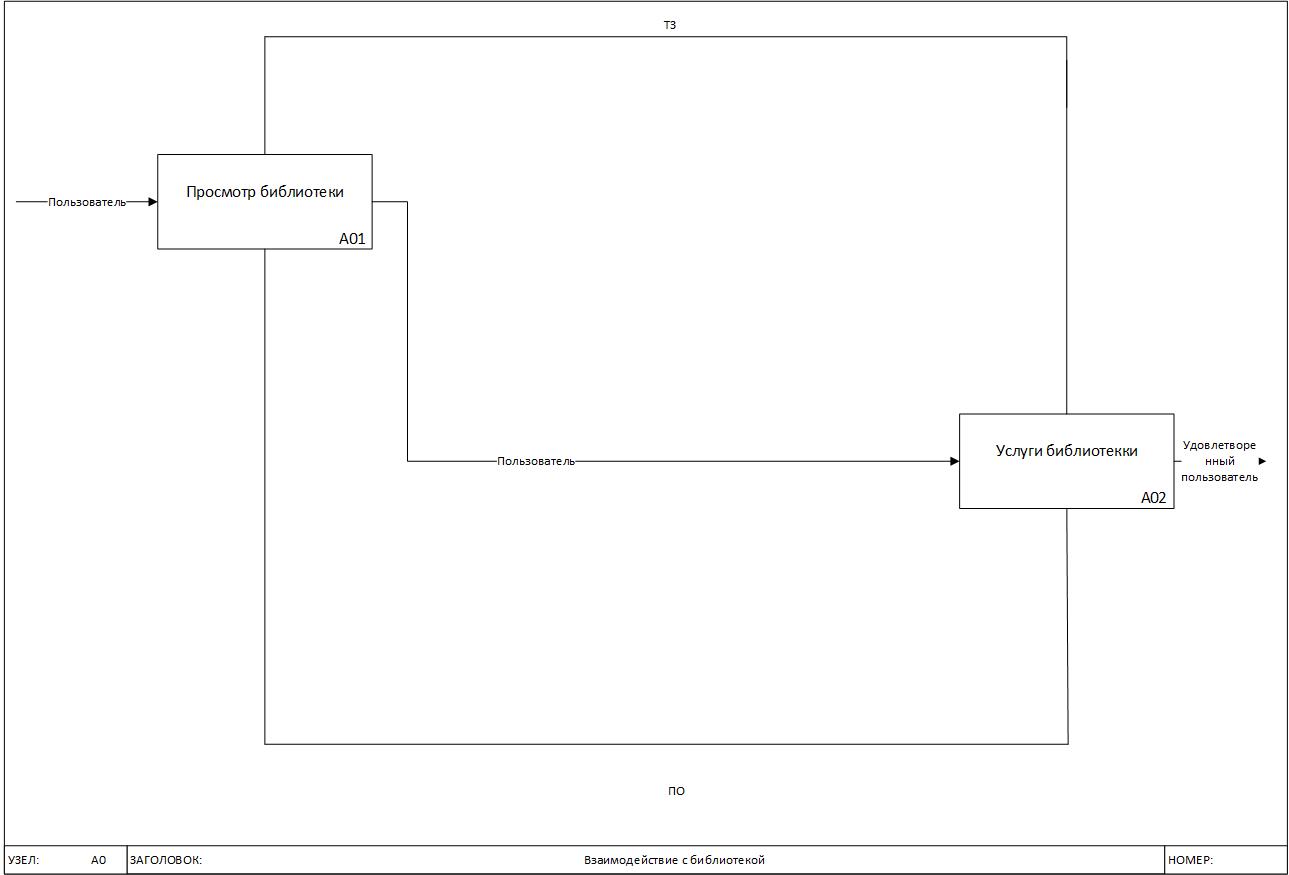


Рисунок 6 – Разбиение блока

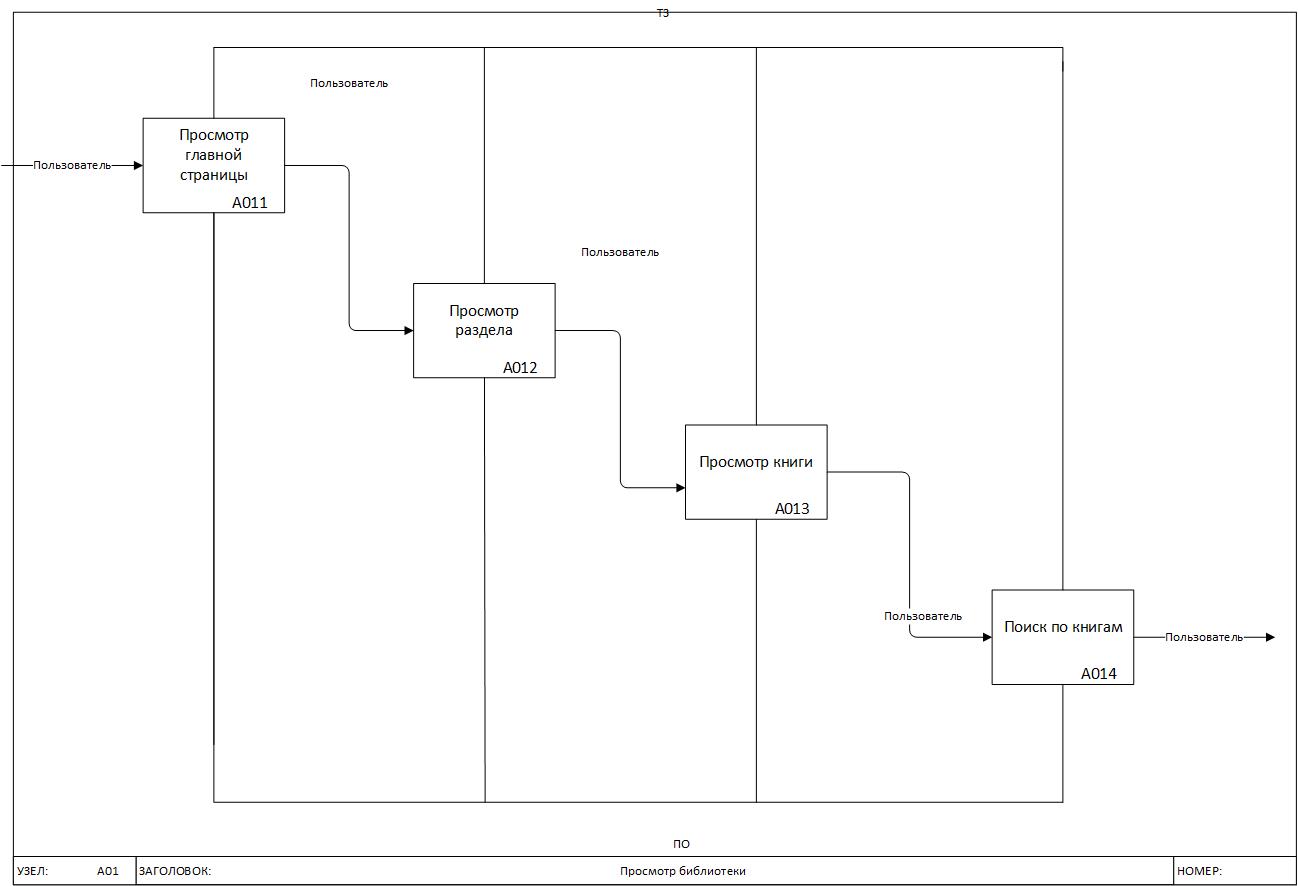


Рисунок 7 – Разбиение просмотра библиотеки

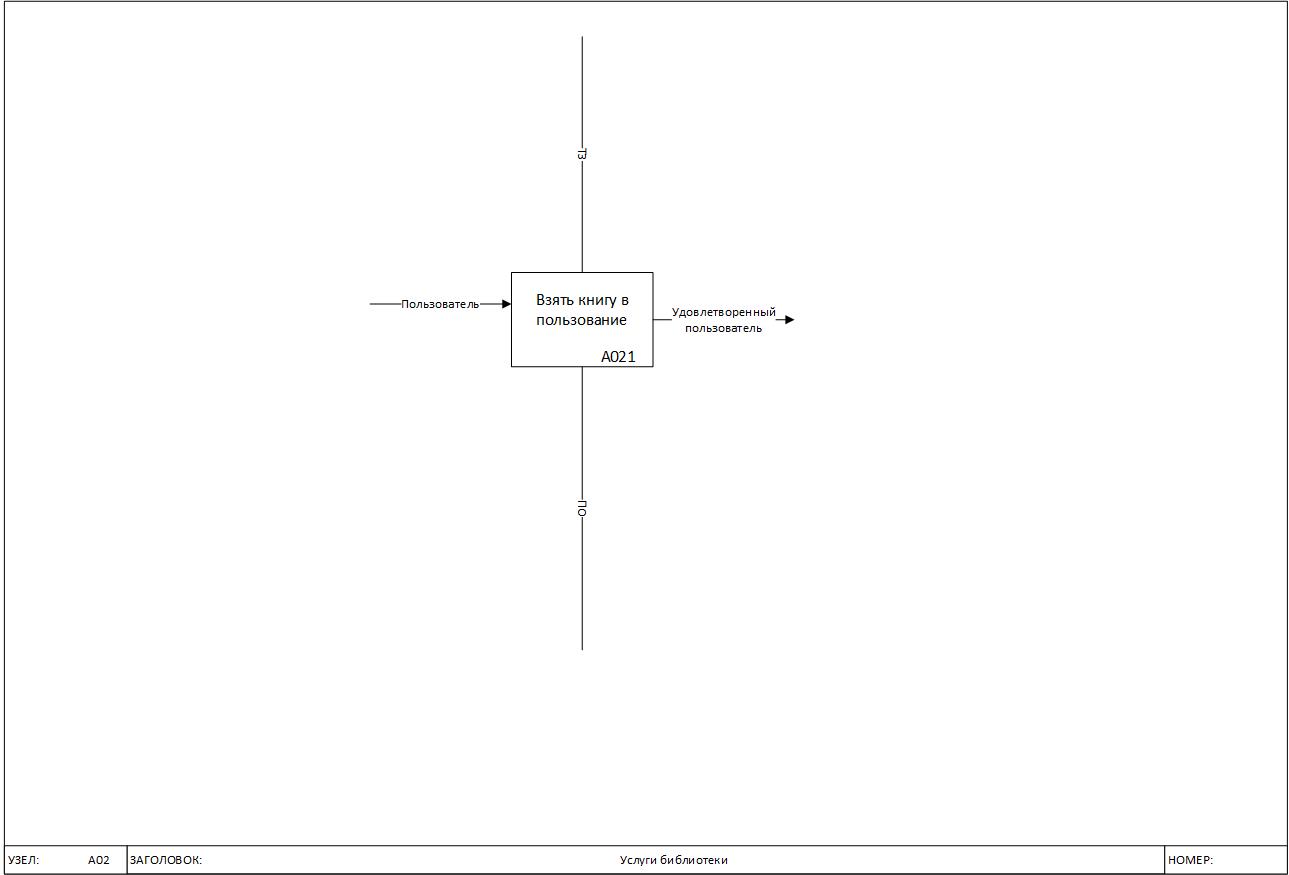
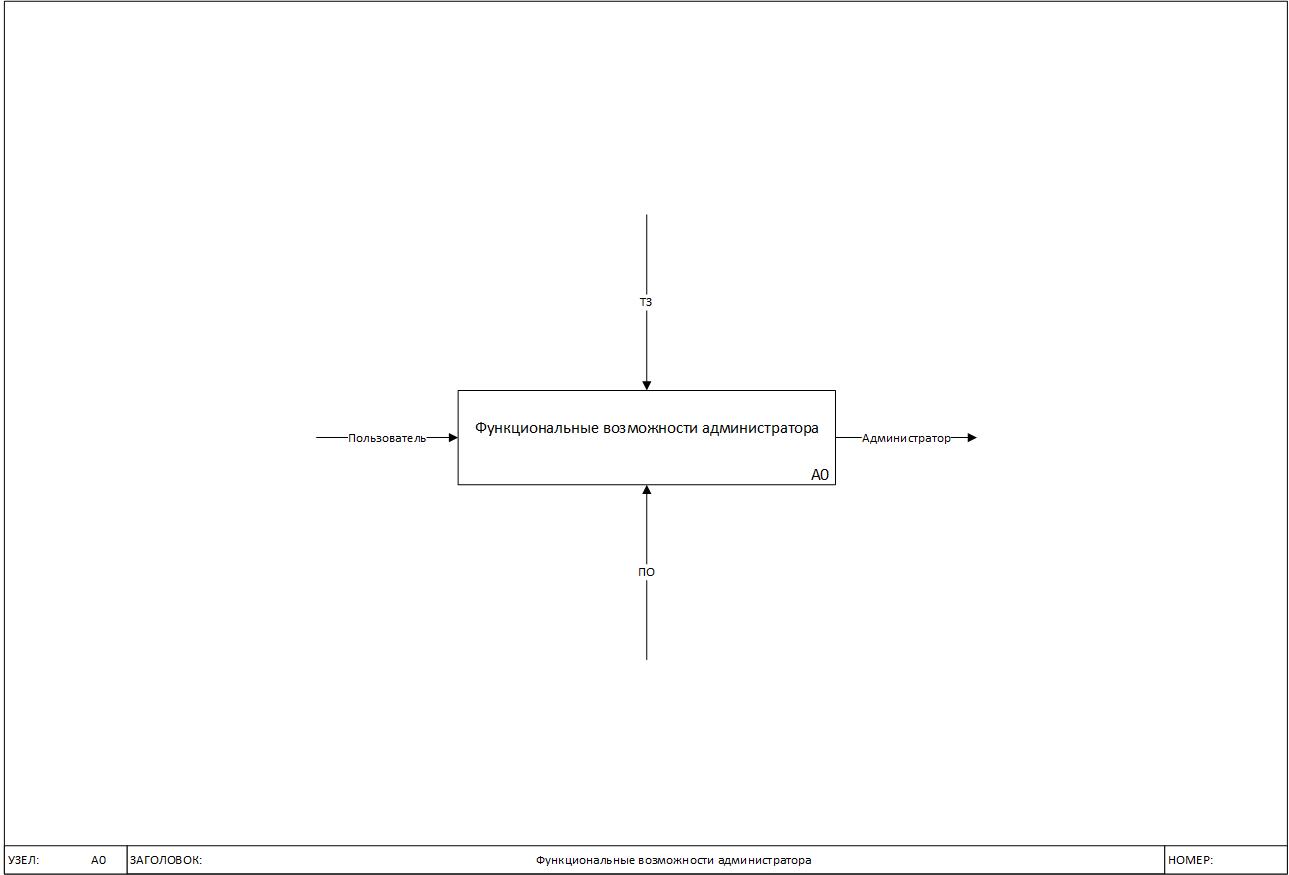
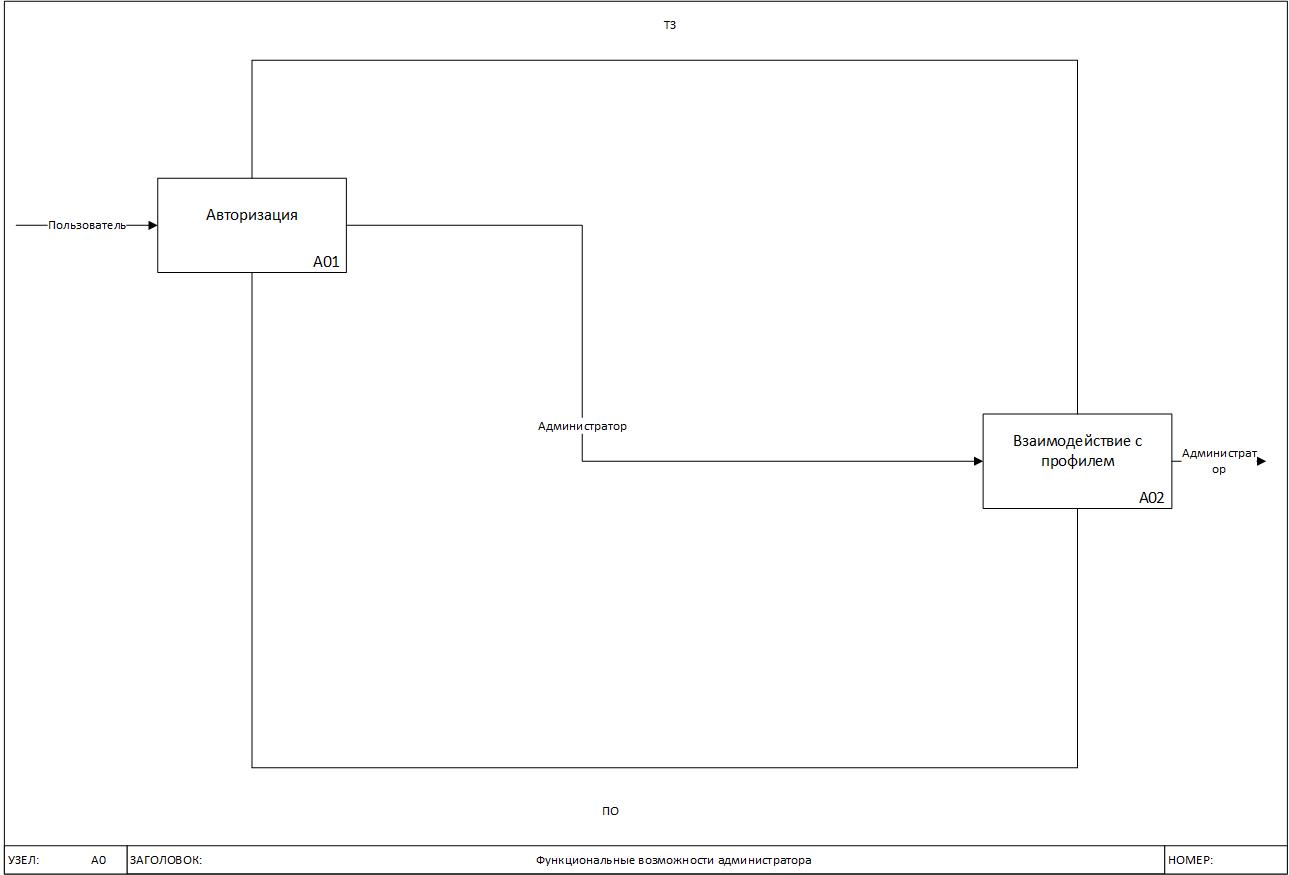


Рисунок 8 – Разбиение услуг библиотеки

Рисунок 9 – Блок возможностей администратора

Рисунок 10 – Разбиение блока

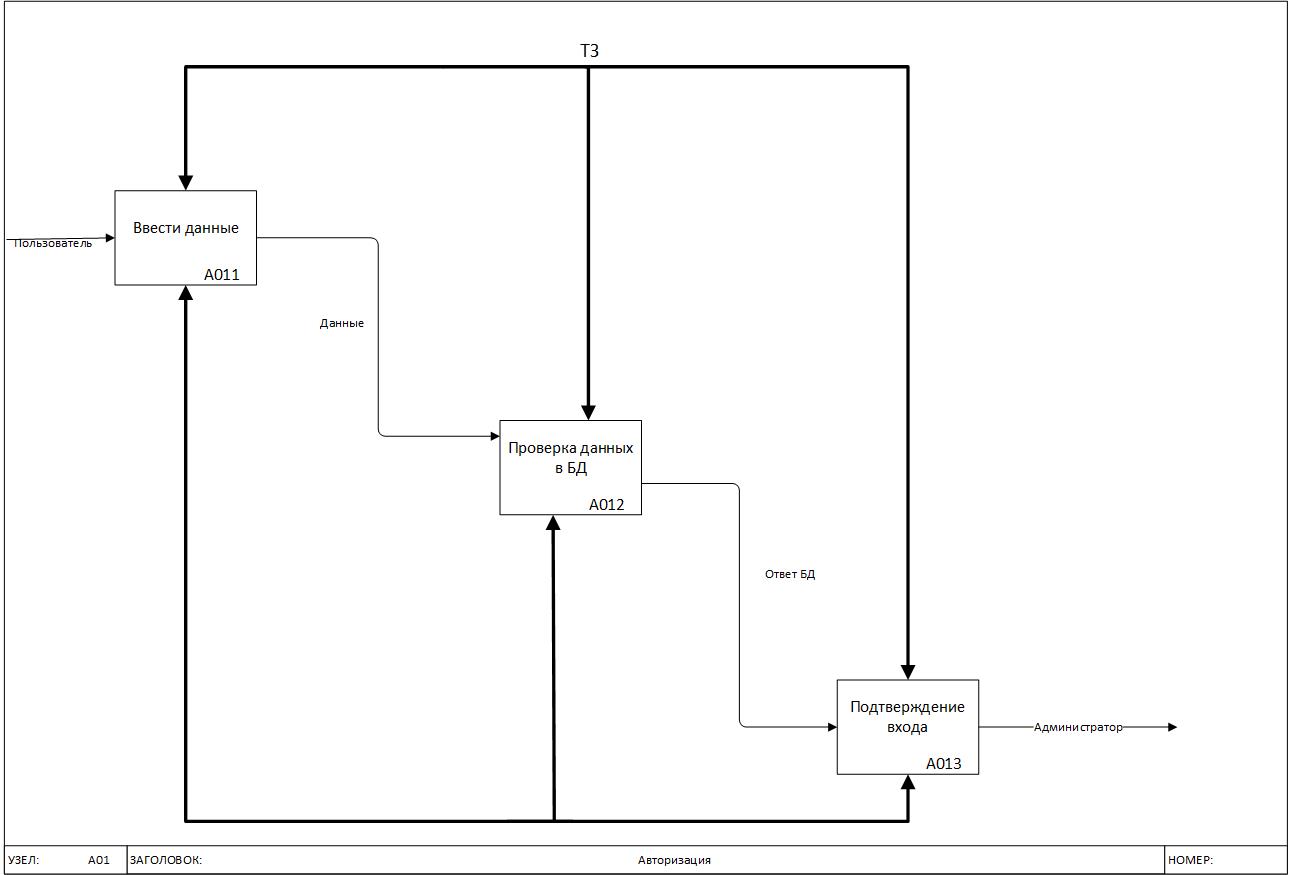


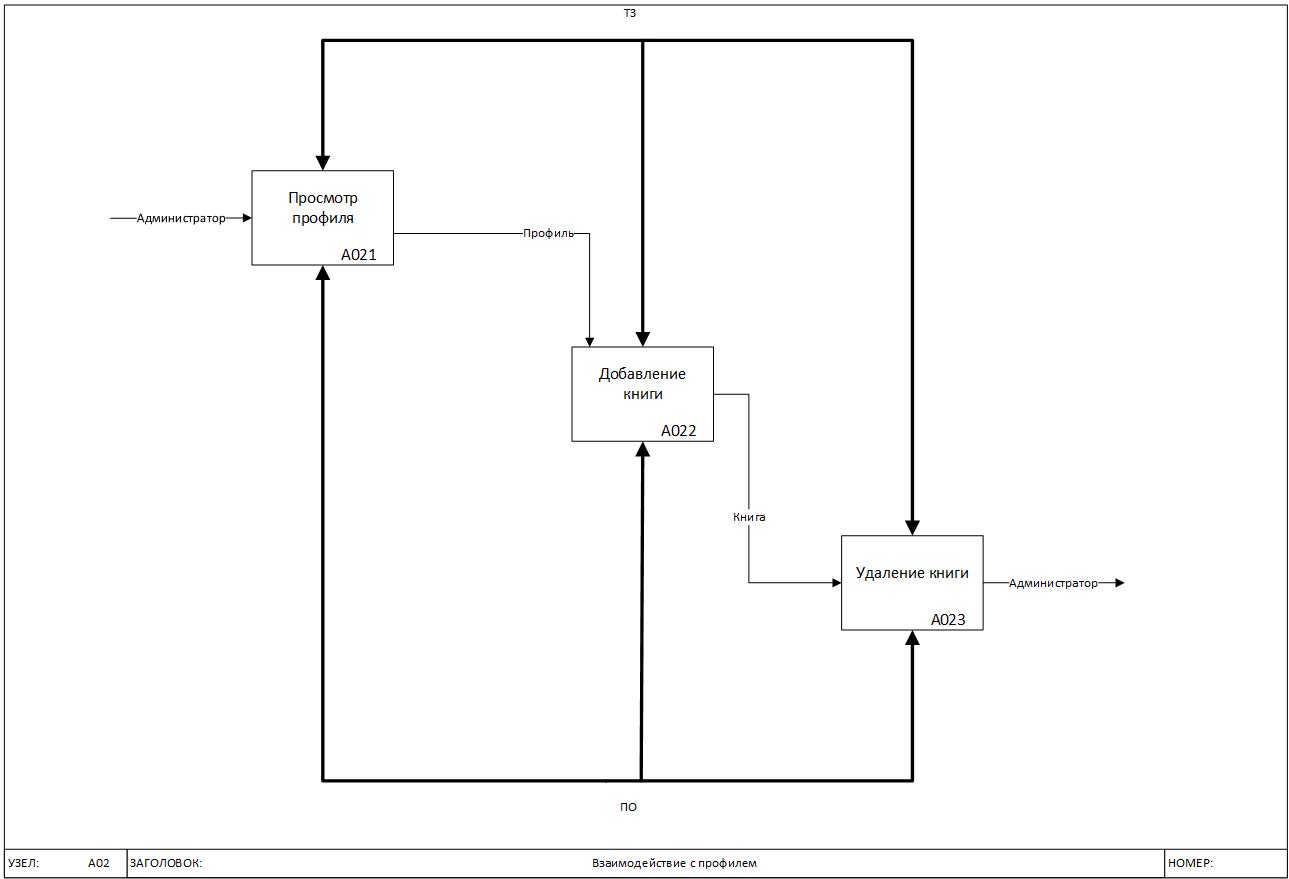
Рисунок 11 – Разбиение авторизации

Рисунок 12 – Разбиение взаимодействия с пользователей

# Диаграмма вариантов использования

Теперь, когда хорошо известны бизнес-процессы, происходящие в системе, необходимо определить варианты её использования. Для этого взяты две авторизованные группы пользователей: обычные читатели и администраторы.

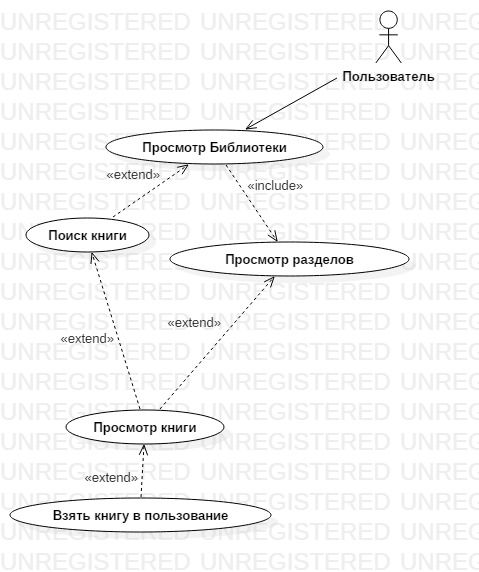


Рисунок 13 – Диаграмма вариантов использования для обычного пользователя

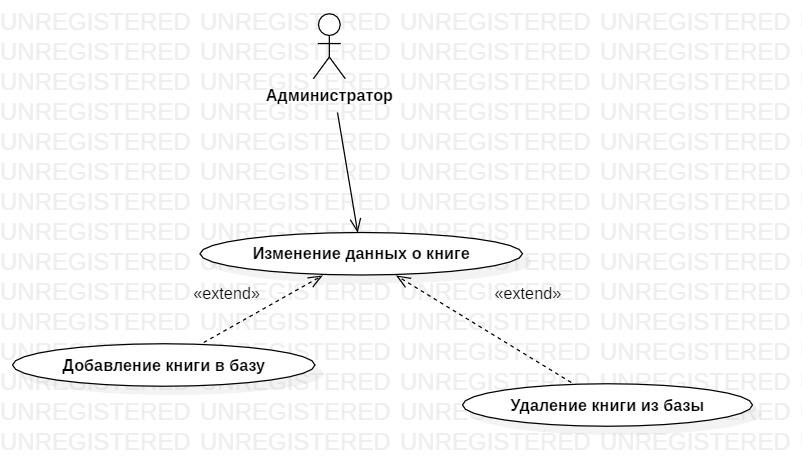
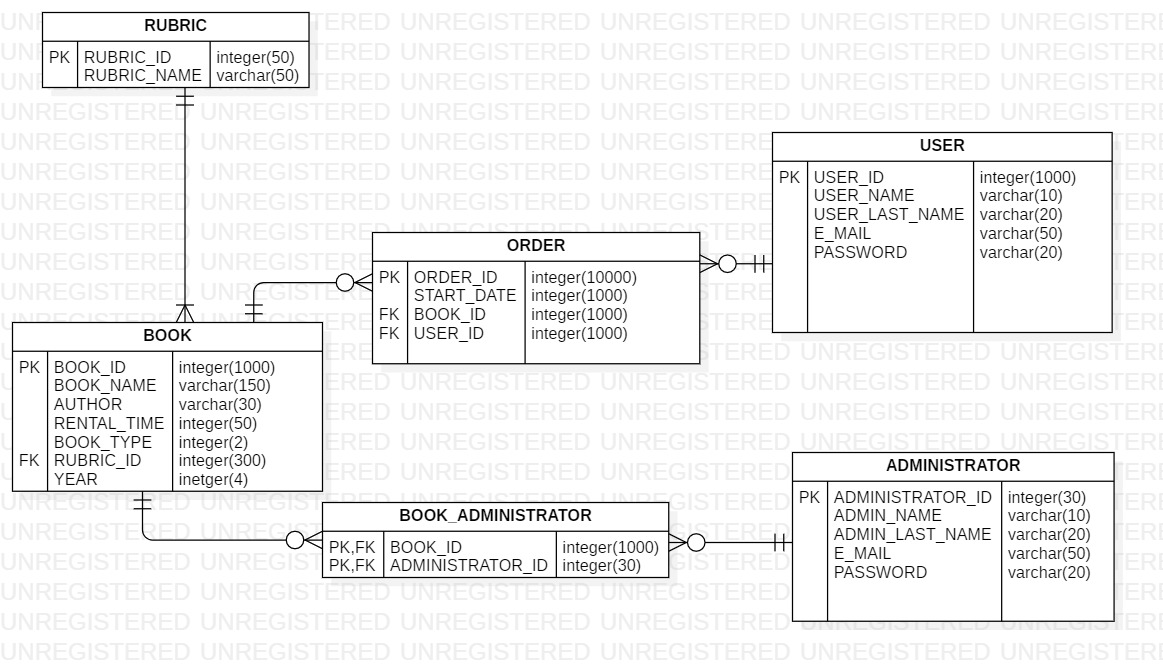


Рисунок 14 – Диаграмма вариантов использования для администратора (только уникальные для данной группы варианты)

# Схема базы данных

При известных процессах, объектах и ролях необходимо определить, какие данные нужно хранить в базе. Схема базы данных определяет таблицы, поля и ограничения целостности. Данная схема представлена на рисунке (15):

Рисунок 15 – Схема базы данных

# Средства реализации

Для реализации данного проекта был выбран язык программирования Python 3.6. Данный язык на сегодняшний день является одним из самых популярных среди web-разработчиков, что упрощает дальнейшую поддержку продукта и его возможную модернизацию.

В качестве фреймворка выбран flask. Данный фреймворк при своей минималистичности обеспечивает все необходимые возможности для реализации проекта и является достаточным инструментом. Также он является одним из наиболее распространённых фреймворков для реализации web-приложений, что упростит дальнейшую поддержку продукта.

Выбранная для проекта СУБД – SQLite3. Данная СУБД является встроенным решением для Python 3.6, а используемые в ней решения уменьшают накладные расходы, время отклика и упрощают программу.

Интерфейс реализован на web-страницах с использованием языков разметки HTML и CSS.

Для работы сайта подходит любой браузер на базе Chromium, операционная система: Windows 7 и выше. Браузеры на базе Chromium являются одними из самых распространённых в мире, что обеспечит крупную базу пользователей.

# Тестирование

# План тестирования

По ходу разработки проекта было проведено три типа тестирования:

* Модульное – поиск дефектов в отдельных элементах приложения (проверялась корректность отправки запросов в БД, разметка HTML-страниц)
* Интеграционное – было проверено взаимодействие между объектами (корректное взаимодействие классов и БД)
* Системное – представлено на демо-видео к проекту

# Результаты тестирования

На рисунках (16), (17), (18) и (19) представлены отчёты о найденных дефектах. На рисунке (20) представлен итоговый отчёт о тестировании.

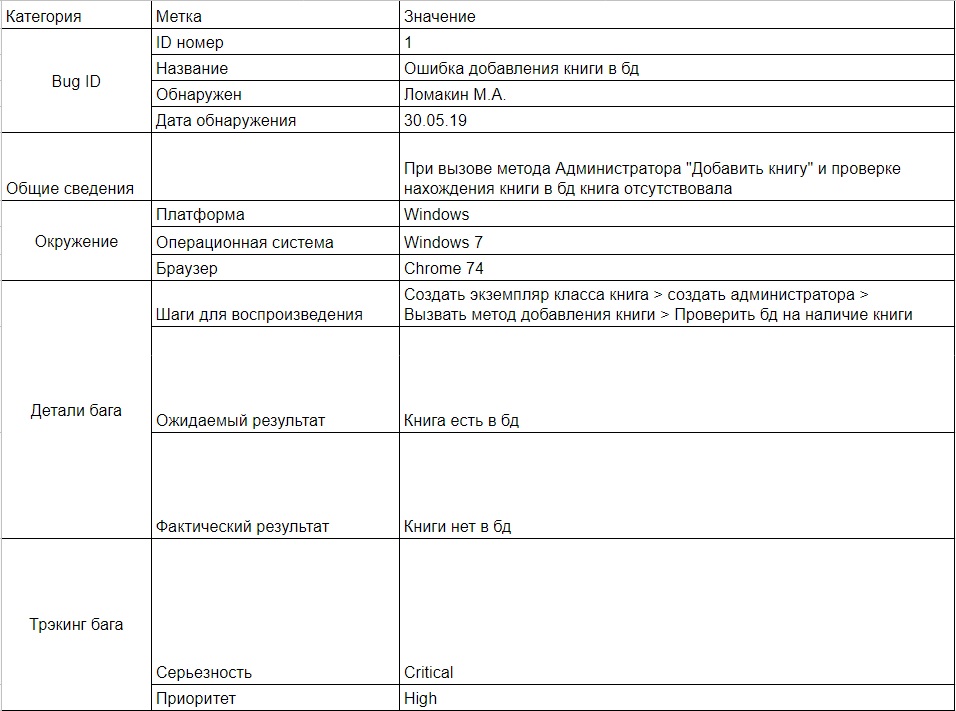
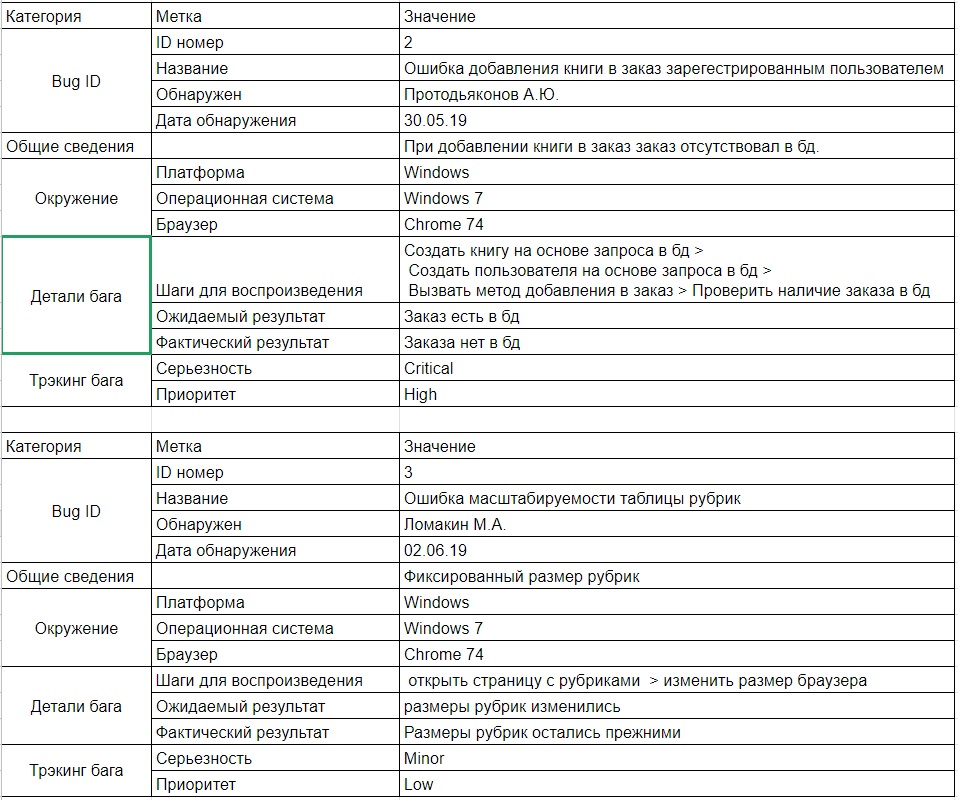
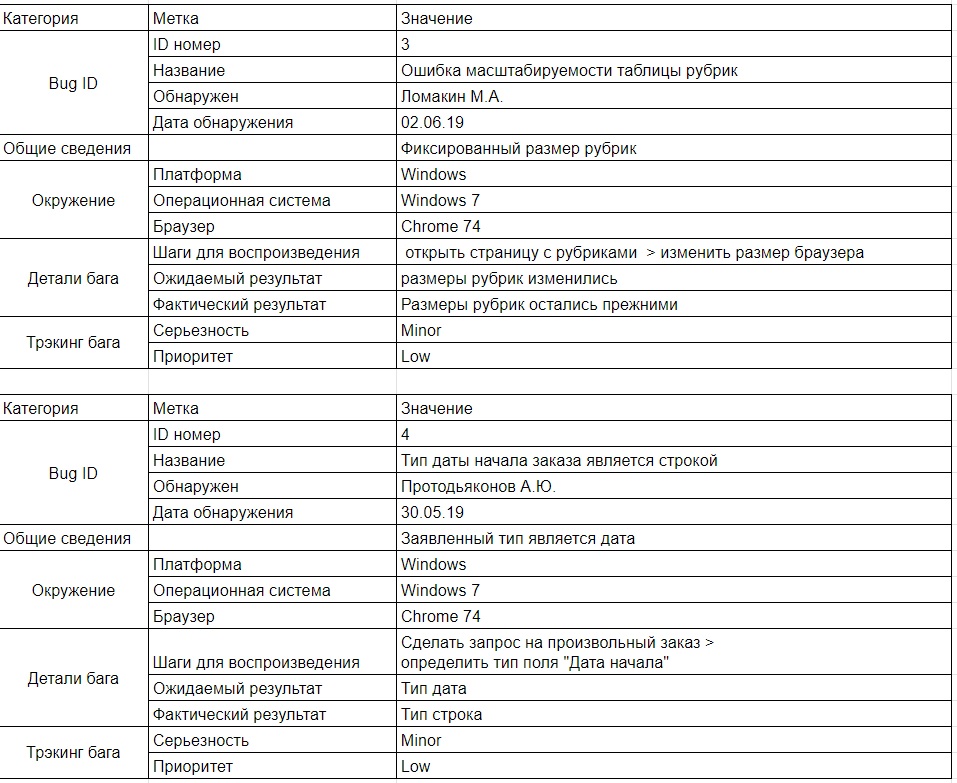
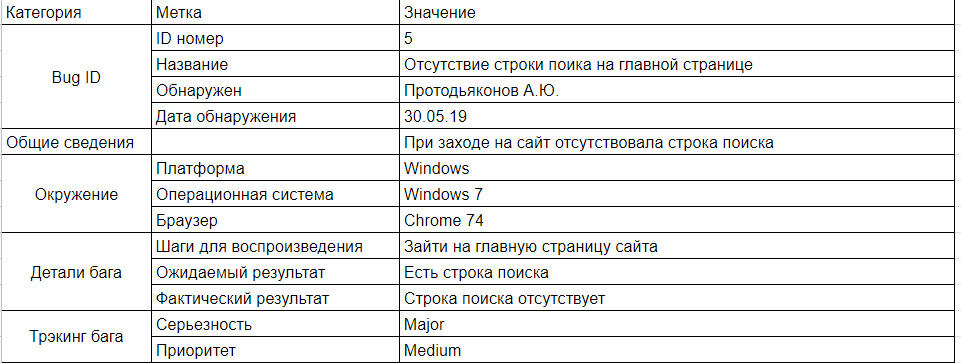
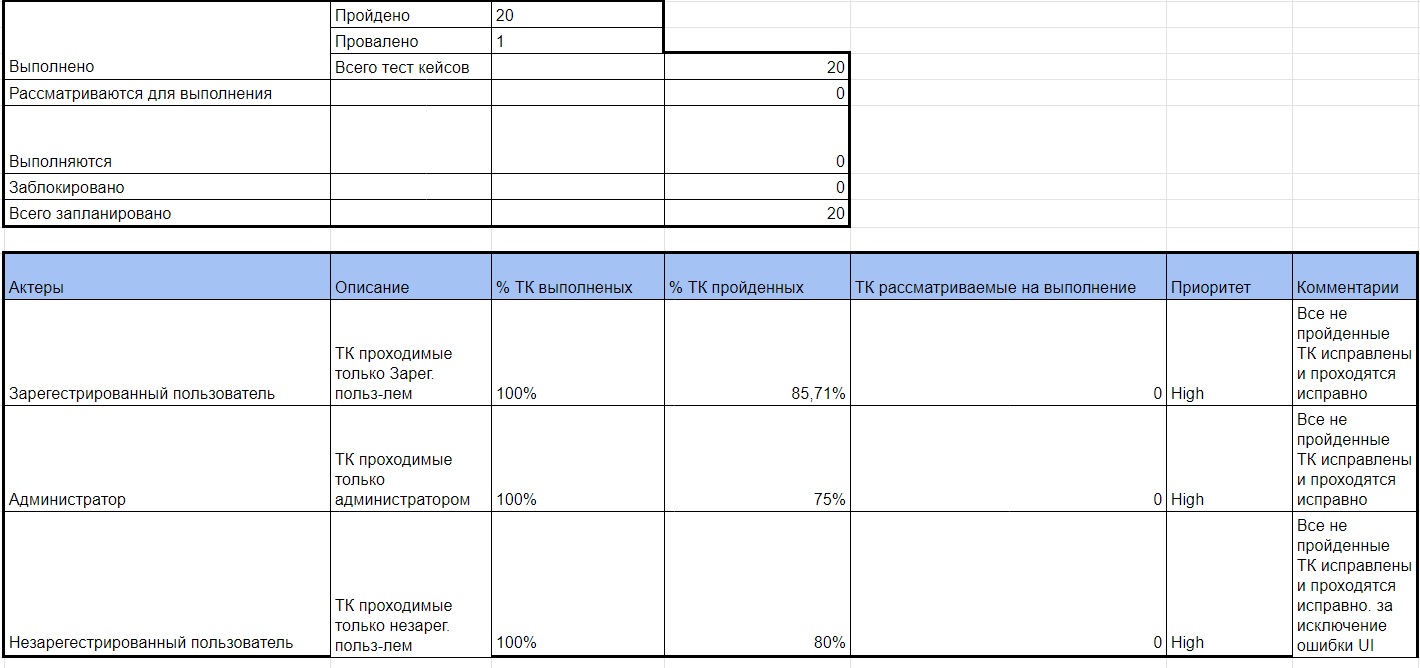


Рисунок 16

Рисунок 17Рисунок 18Рисунок 19

Рисунок 20 – Результат тестирования

# Реализация

# Диаграмма состояний

Диаграмма состояний показывает все возможные состояния, в которых может находиться объект. В данном случае это профиль пользователя.

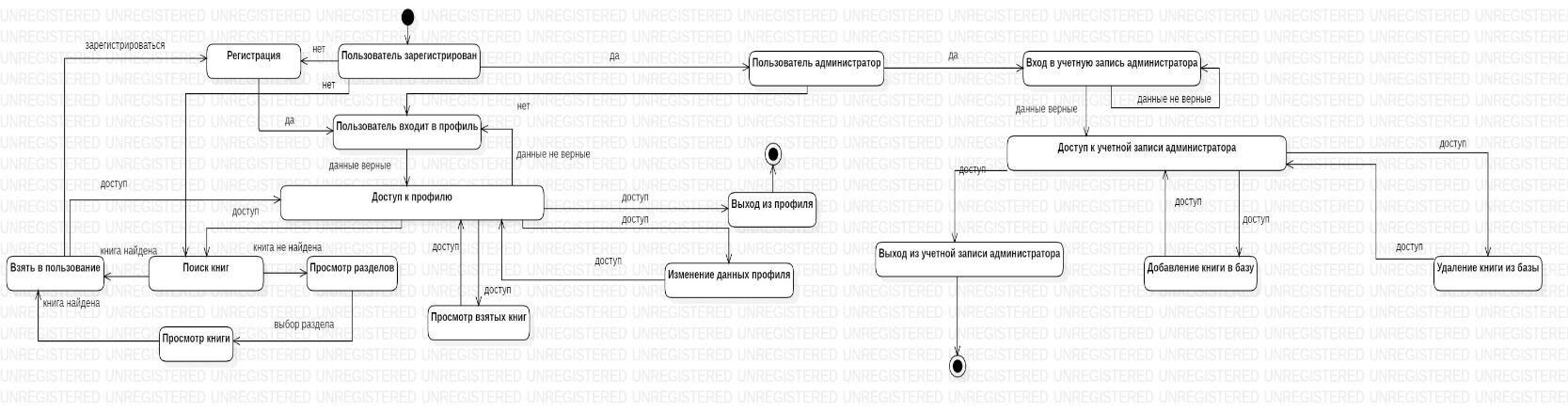
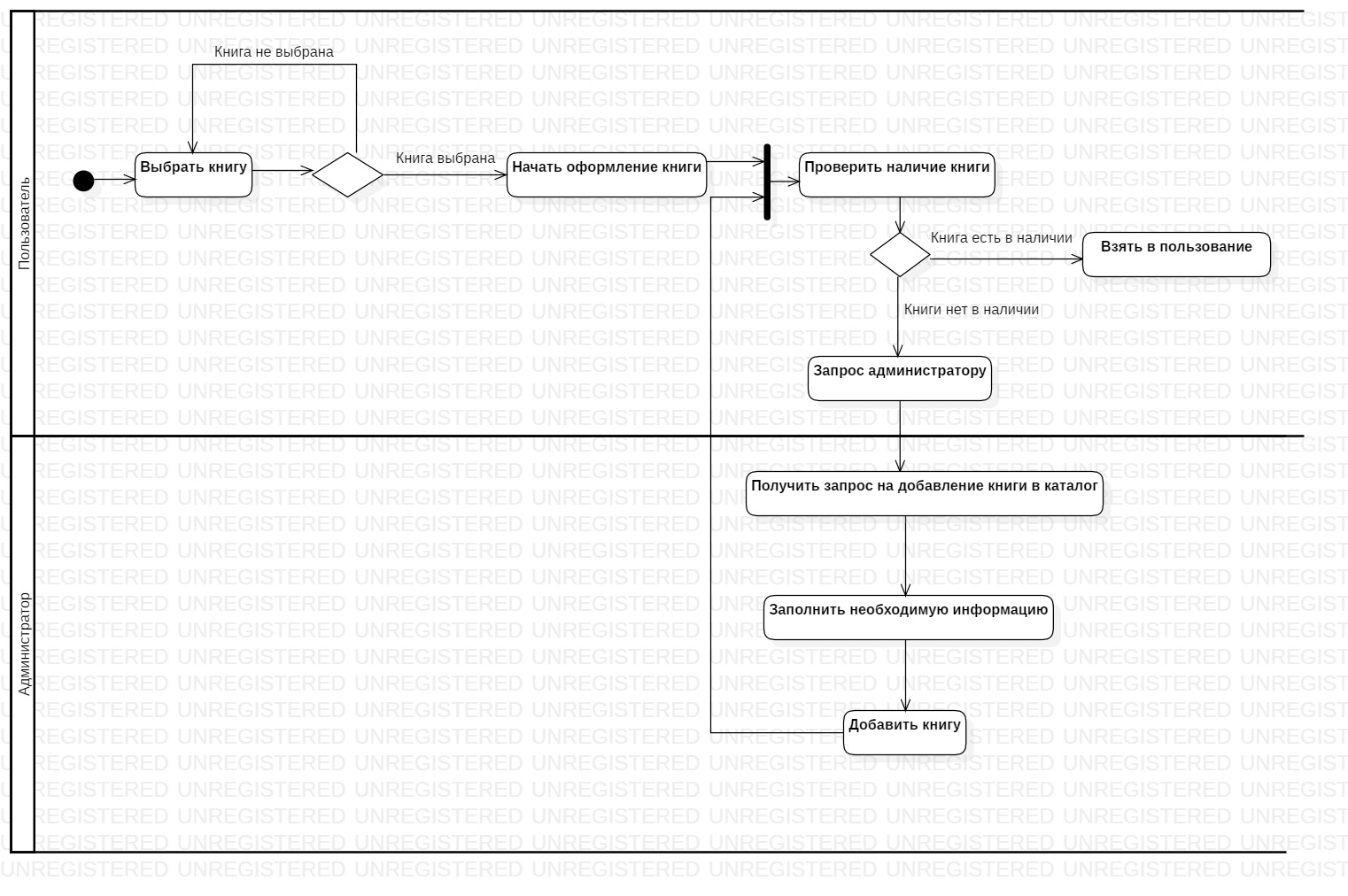


Рисунок 21 – Диаграмма состояний разрабатываемого приложения

# Диаграмма деятельности

Диаграмма деятельности показывает действия, состояния которых описано на диаграмме состояний.

Рисунок 22 – диаграмма деятельности

# Диаграмма классов

Диаграмма классов показывает классы системы, их атрибуты, методы и взаимодействия.

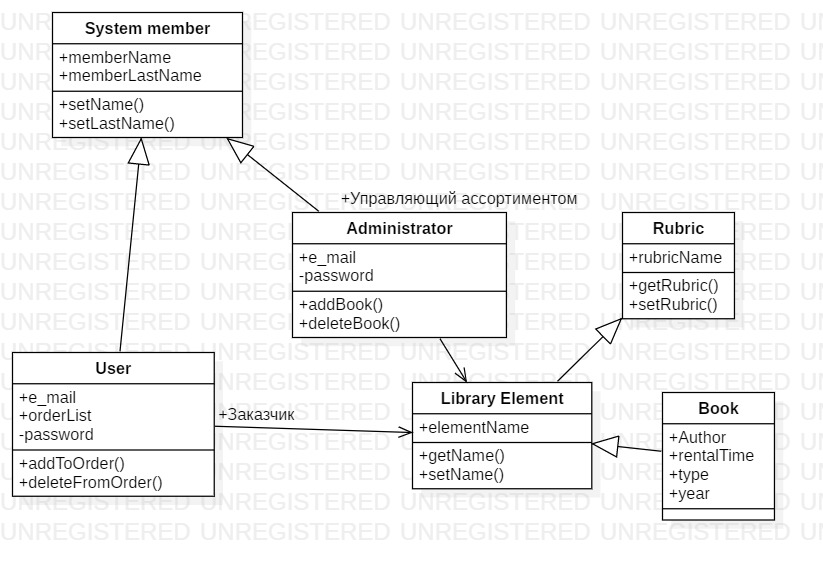


Рисунок 23 – Диаграмма классов

# Диаграммы последовательности и взаимодействия

Необходимо теперь определить, какие способы взаимодействия с системой есть в распоряжении пользователя. Кроме того, для каждого взаимодействия нужно определить последовательность запросов в системе.



Рисунок 24 – Диаграмма взаимодействия для регистрации

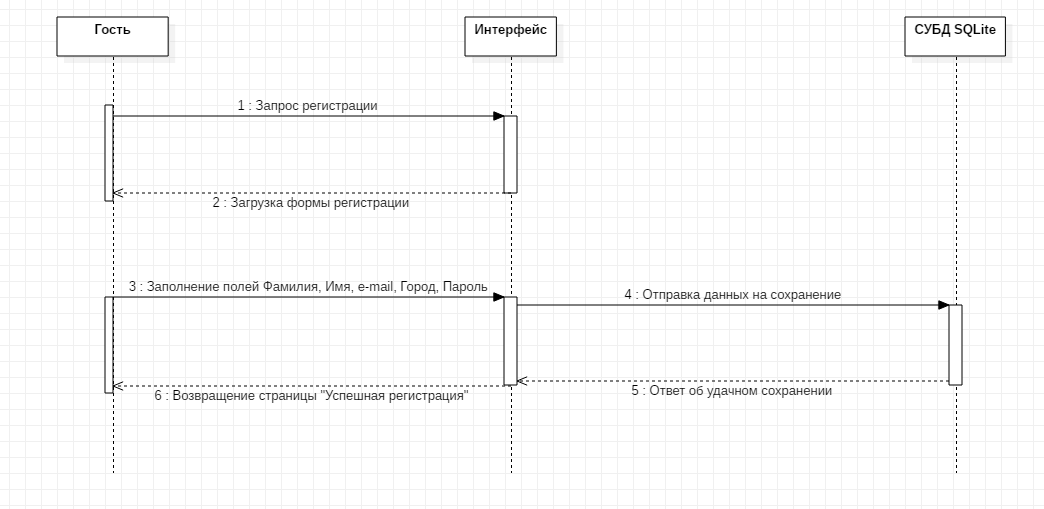


Рисунок 25 – Диаграмма последовательности для регистрации



Рисунок 26 - Диаграмма взаимодействия для входа

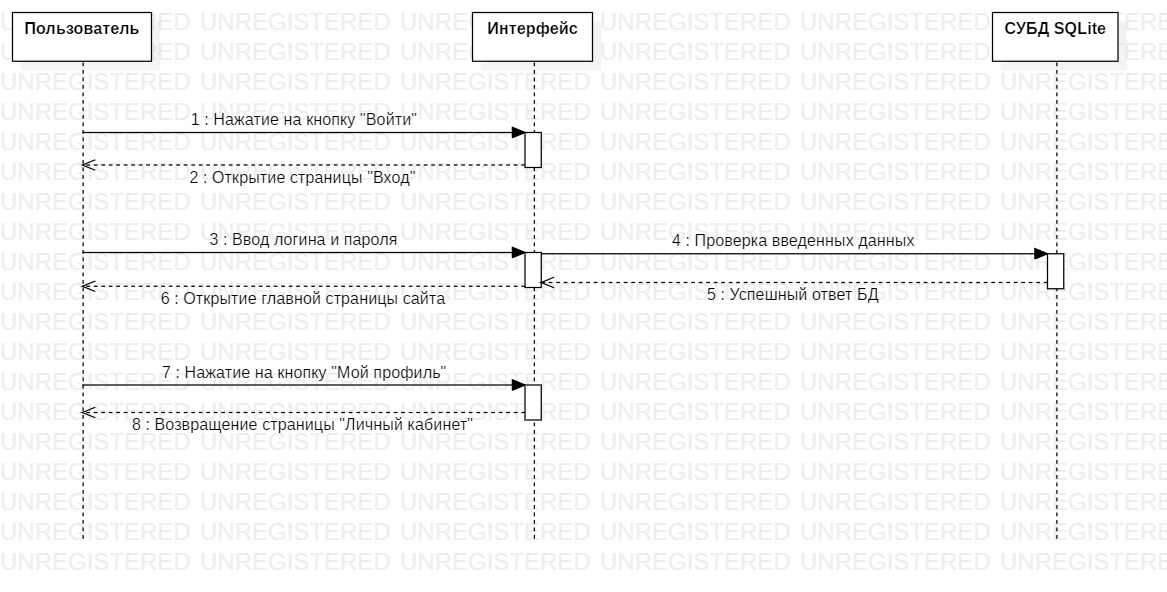


Рисунок 27 - Диаграмма последовательности для входа

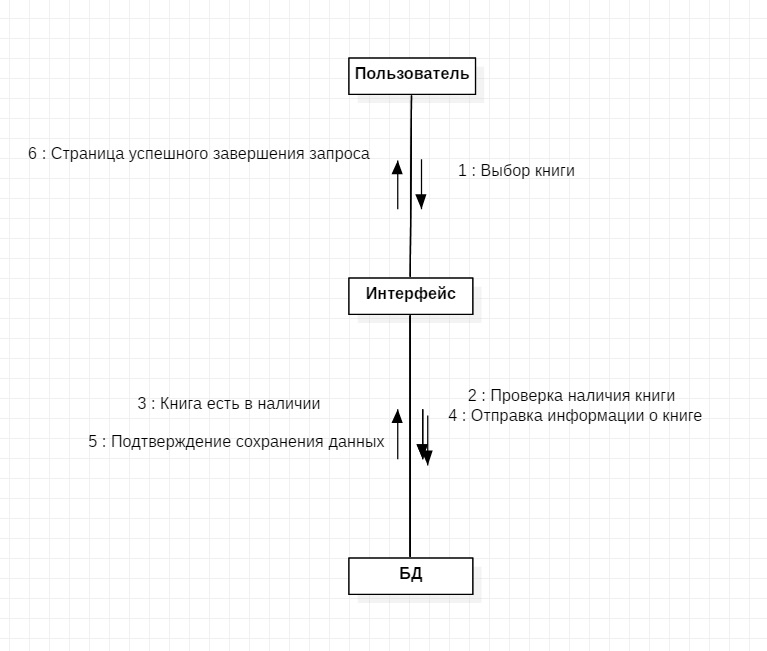


Рисунок 28 - Диаграмма взаимодействия для заказа книги

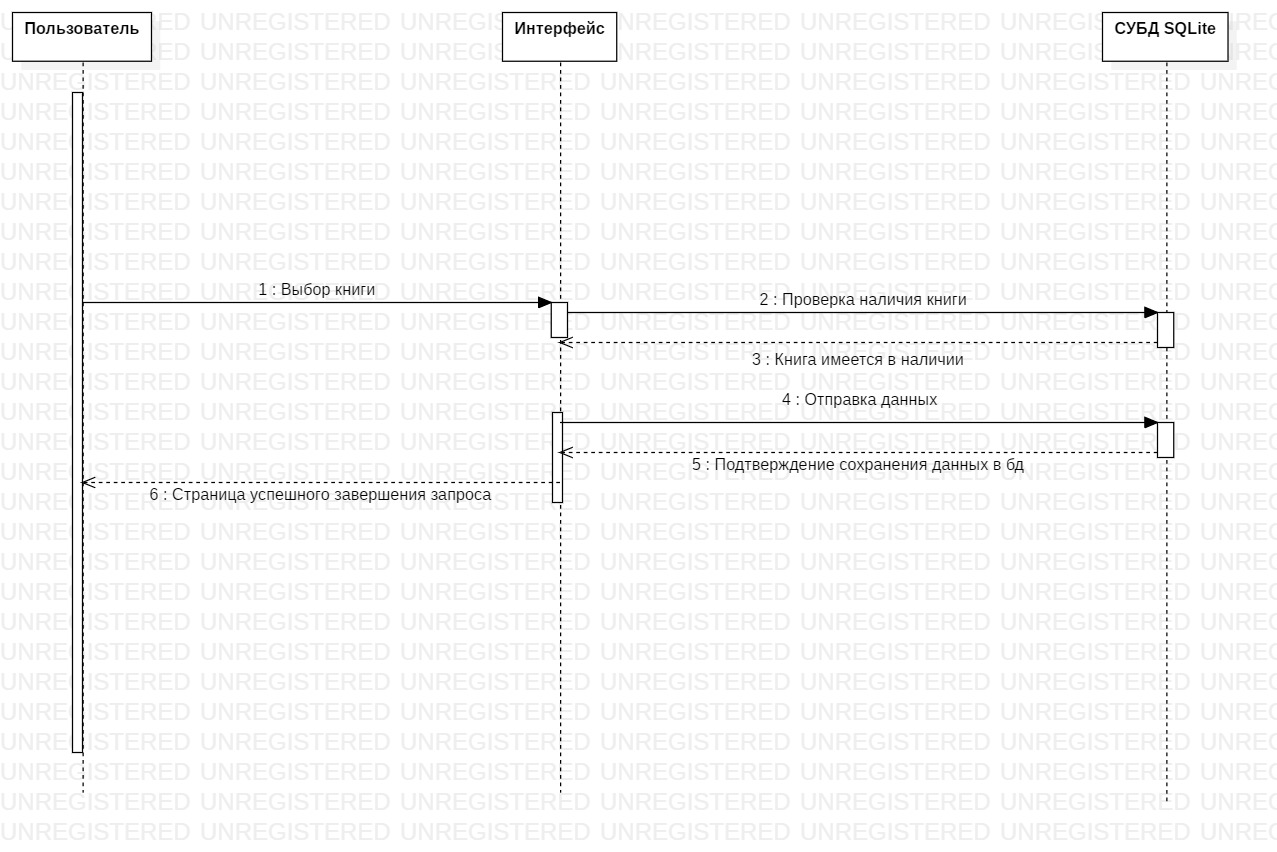


Рисунок 29 - Диаграмма последовательности для заказа книги



Рисунок 30 - Диаграмма взаимодействия для добавления книги

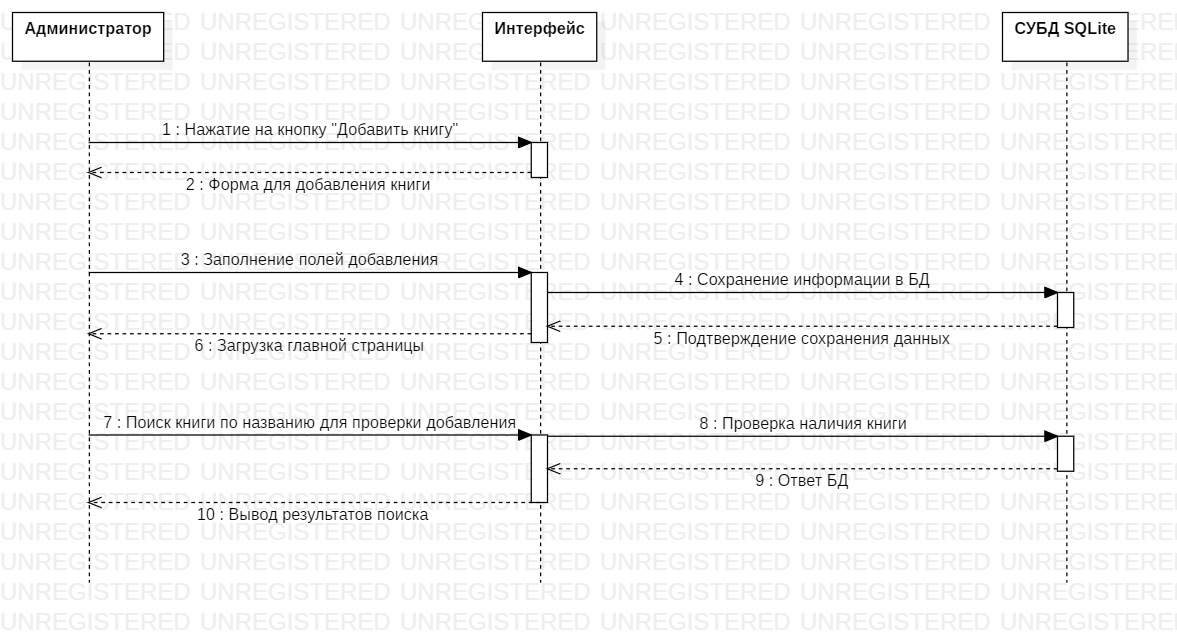


Рисунок 31 - Диаграмма последовательности для добавления книги

# Диаграмма развёртывания

Диаграмма развертывания должна показывать, какие аппаратные компоненты существуют, какие программные компоненты работают на каждом узле, и как различные части этого комплекса соединяются друг с другом.

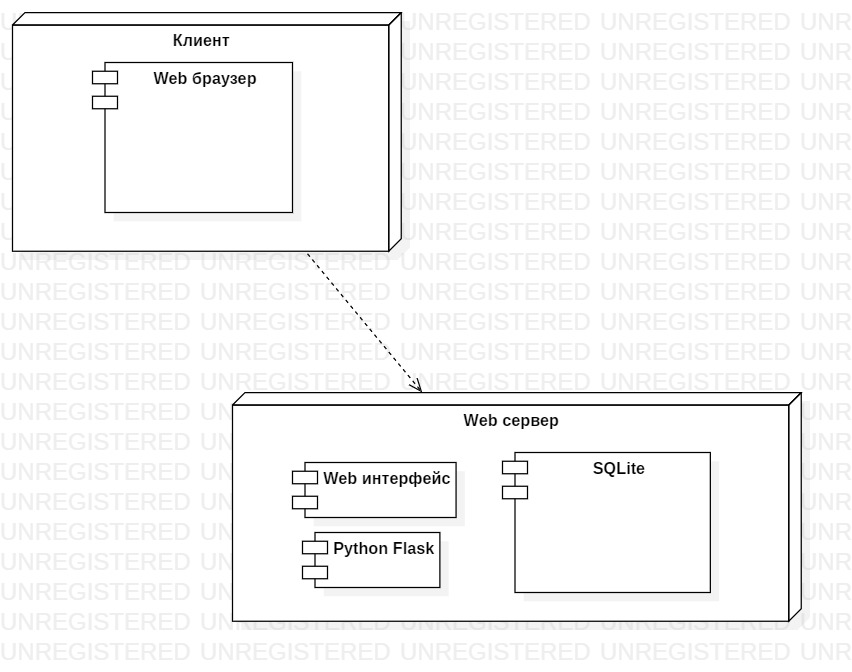


Рисунок 32 – Диаграмма развёртывания

# Заключение

В ходе данной курсовой работы была произведена разработка онлайн-портала сети библиотек LibOn. Серверная часть приложения была разработана на языке Python с использование фреймворка flask. Клиентская часть написана на языках HTML и CSS. Информация о книгах и пользователях хранится в базе данных SQLite.

Перед разработкой были произведены:

* определение технических требований к web-сайту
* проектирование web-сайта средствами языка UML

После разработки было произведено тестирование и запись демо-видео.

Отчёт по ролям

1. **Техническое Задание**Работали: Дроботенко А. Д., Ломакин М. А., Протодьяконов А. Ю.  
   Ответственный: Протодьяконов А. Ю.
2. **Функциональные Требования**Работали: Протодьяконов А. Ю.  
   Ответственный: Протодьяконов А. Ю.
3. **Макеты сайта**Работали: Дроботенко А. Д.  
   Ответственный: Дроботенко А. Д.
4. **Описание предметной области**Работали: Протодьяконов А. Ю.  
   Ответственный: Протодьяконов А. Ю.
5. **Отчётный документ**Работали: Дроботенко А. Д., Ломакин М. А., Протодьяконов А. Ю.  
   Ответственный: Протодьяконов А. Ю.
6. **Диаграммы вариантов использования**Работали: Ломакин М. А., Протодьяконов А. Ю.  
   Ответственный: Ломакин М. А.
7. **Диаграмма активности**Работали: Дроботенко А. Д.  
   Ответственный: Дроботенко А. Д.
8. **Схема базы данных**Работали: Ломакин М. А.  
   Ответственный: Ломакин М. А.
9. **Диаграммы взаимодействия и последовательностей**Работали: Дроботенко А. Д.  
   Ответственный: Дроботенко А. Д.
10. **Диаграмма классов**Работали: Ломакин М. А., Протодьяконов А. Ю.  
    Ответственный: Ломакин М. А.
11. **Диаграмма развёртывания**Работали: Ломакин М. А.  
    Ответственный: Ломакин М. А.
12. **Диаграмма состояний**Работали: Дроботенко А. Д.  
    Ответственный: Дроботенко А. Д.
13. **Диаграммы IDEF0**Работали: Дроботенко А. Д.  
    Ответственный: Дроботенко А. Д.
14. **Интерфейсная реализация классов**Работали: Дроботенко А. Д., Ломакин М. А.  
    Ответственный: Ломакин М. А.