Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Кубанский государственный университет»

Кафедра информационных технологий

**ОТЧЕТ**

о выполнении лабораторной работы № 8

по дисциплине «Технологии проектирования программного обеспечения»

Выполнил: ст. гр. 35/2

Исупов М. В.

Проверил: доц. каф. ИТ

Полетайкин А.Н.

Краснодар,

2021

Индивидуальное задание «Библиотека»

Лабораторная работа №8

Тема: Документирование и развертывание ИС

Цель: Освоение методики документирования ПС.

Задание:

1. Собрать документированные материалы, разработанные при выполнении лабораторных работ №1–7 в единый документ с разбивкой на разделы.
2. Описать назначение, технические характеристики, принцип работы и меры безопасности при эксплуатации ПС.
3. Составить руководство пользователя.

Ход работы

1. **Лабораторная работа 1.**

Студенческая библиотека — учреждение, собирающее и осуществляющее хранение произведений печати и письменности для общественного пользования, а также ведущее справочно-библиографическую работу.

Студент – физическое лицо, нуждающееся выдаче ему определенных, нужных ему книг. Студент является основным потребителем услуг студенческой библиотеки.

При первом посещении на каждого студента создается карточка студента, в которых содержатся данные о студенте, данные о книгах, которые он когда-либо получал в этой библиотеке, даты выдачи и приёма этих книг обратно в библиотеку.

При поступлении новой книги в библиотеку библиотекарь добавляет сведения о ней в информационную систему.

Перед выдачей книги студенту библиотекарь добавляет в его карточку сведения об этой книге, а также дату выдачи этой книги. Студент обязан вернуть эту книгу обратно библиотеке согласно регламенту, который заранее был установлен библиотекой. Если он не уложится в срок, то система выпишет штраф студенту, который тот будет обязан заплатить.

Таким образом, можно выделить следующие задачи:

* Создание карточки студента / внесение данных о студенте в систему.
* Внесение сведений о новых книгах, поступивших в библиотеку.
* Ведение учёта книг, которые выдали / обратно приняли в библиотеку.
* Просмотр всех книг, которые были выданы / получены.
* Выписывание штрафа лицам, не успевшим вернуть книгу в библиотеку в срок.

Бизнес-процессом в данной лабораторной работе является управление выдачей/приёмом библиотечных книг.

Основным действующим лицом является сам библиотекарь, который вносит все данные в систему.

Рассмотрим схему “Черный ящик” данного процесса:

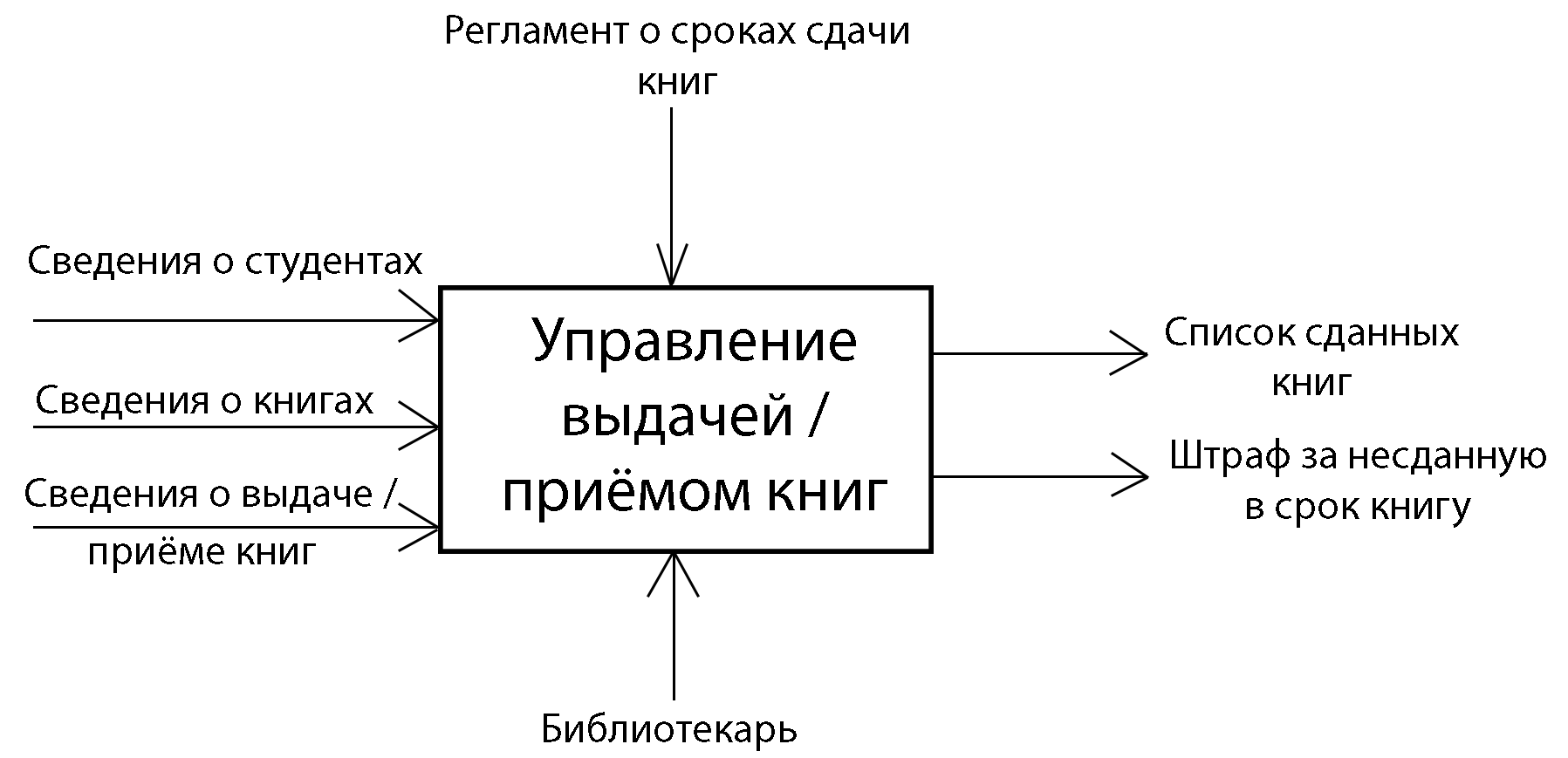


Рисунок 1 – схема бизнес-процесса

Здесь сведения о выдаче / приёме книги представляют собой даты, когда эта книга была соответственно выдана библиотекой или же принята обратно.

Штраф за несданную в срок книгу определяется согласно регламенту о сроках сдачи книг.

Теперь рассмотрим схему вариантов использования бизнес-процесса:

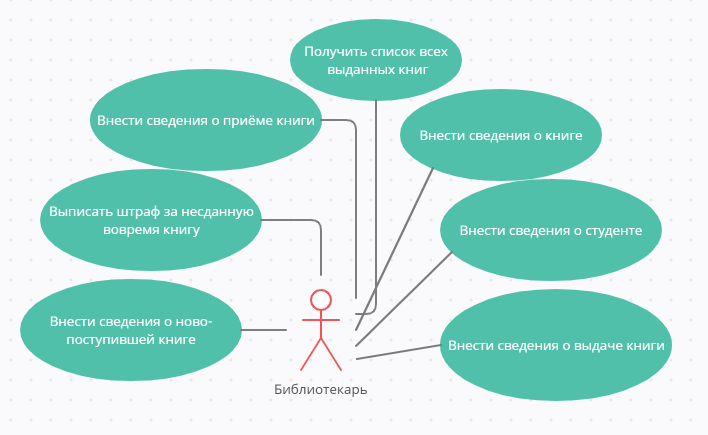


Рисунок 2 – Диаграмма вариантов использования

Опишем внешние информационные процессы в виде таблиц 1 и 2.

Таблица 1. Реестр входных информационных потоков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование  и назначение потока | Форма представ-ления | Обработ-чик | Корреспон-  дент | Характеристики  обработки | |
| Трудозат-  раты, чел-ч | Периодич-ность, регламент |
| 1 | Сведения о студентах | Речь | Библио-текарь | Студент | 0.1 | 1 час |
| 2 | Сведения о книгах | Текст | Библио-текарь | Книга | 0.2 | 8 часов |
| 3 | Сведения о приёме / выдаче книги | Речь | Библио-текарь | Библио-текарь | 0.1 | 20 минут |

Таблица 2. Реестр выходных информационных потоков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование  и назначение потока | Форма представ-ления | Обработ-чик | Корреспон-  дент | Характеристики  обработки | |
| Трудозат-  раты, чел-ч | Периодич-ность, регламент |
| 1 | Список сданных книг | Текст | ПС | Библио-текарь | 0.1 | 30 минут |
| 2 | Выписка штрафа за невовремя сданную книгу | Текст | ПС | Студент | 0.1 | 8 часов |

Внедрение автоматизированной системы управления выдачей/приёмом книг в библиотеку позволит облегчить процесс выдачи/приёма книг библиотекой, облегчит надзор за возвращением книг в положенный срок, позволит обслуживать большее количество посетителей.

1. **Лабораторная работа 2.**

Описание существующих программ для учёта книг в библиотеке

Учёт книг (Windows)

Учет книг - программа для учета и каталогизации книг, журналов, документов, учет читателей, отслеживание должников, учет издательств, авторов, произведений и других объектов. "Учет книг" - удобное решение для автоматизации как небольших, домашних так и больших, промышленных библиотек. Программа поможет Вам быстро осуществлять поиск книги, отслеживать историю выдачи книги, своевременно узнавать о должниках, хранить необходимую информацию о читателях, вести читательский счет, контролировать даты выдачи и возврата книги. Ряд стандартных отчетов предоставит всю необходимую статистическую информацию.

Основные функции:

* Ведение базы книг, журналов. Каталогизация

В базе данных содержится информация о книгах, журналах. Предусмотрены такие поля как – название, авторы, категория, тип, издательство, серия, формат, год издания, количество страниц, тираж, обложка, ISBN, УДК, № шкафа, № полки, блок, подблок, время добавления и т.д.  
Для каждой книги показываются все ее читатели (которые читали эту книгу ранее и читают сейчас).

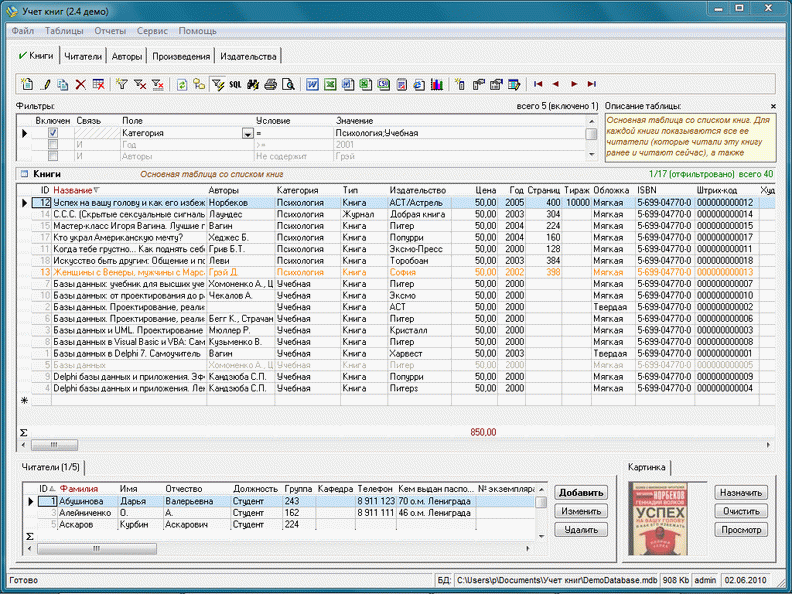
Предусмотрены удобные способы сортировки и фильтрации данных, что позволяет быстро найти нужные книги.  
Любую таблицу базы можно распечатать, экспортировать в MS Word, MS Excel или текстовый формат CSV. Имеется импорт из других источников данных в формате CSV.

* Учет должников по возврату

Система фиксирует информацию о читателях – ФИО, контактная информация, выданные книги, даты выдачи и возврата книг.

Контролируя значение поля с датой возврата книги можно легко вести учет должников.

Таблица “На руках” показывает список всех выданных на руки книг и журналов.



**Рисунок 1** – Интерфейс программы Учёт книг

1С: Библиотека (MacOS, Windows, Linux)

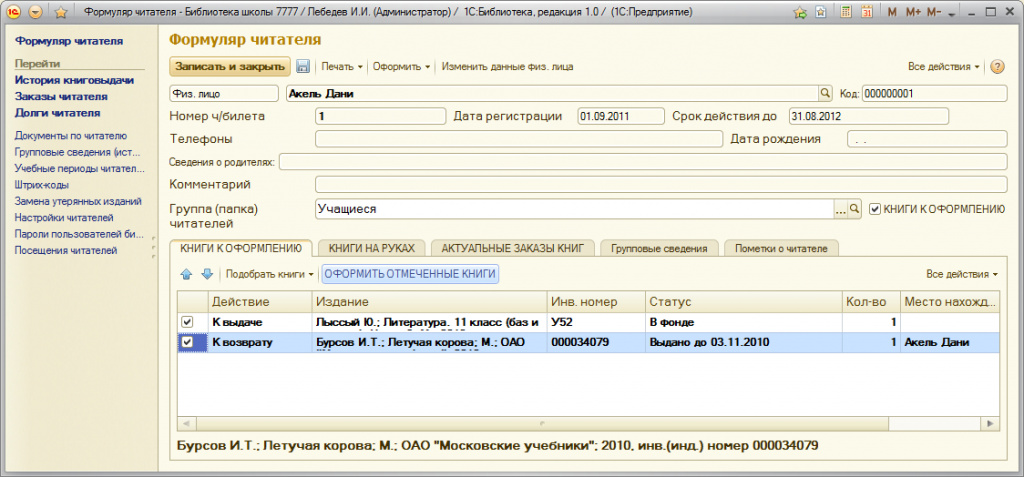
Данный продукт Продукт позволяет автоматизировать рабочие процессы библиотеки, в зависимости от ее назначения, типа, состава фондов, может быть интегрирован с другими типовыми решениями фирмы "1С".

Имеются поиск, формирование и оформление заказов в издательства/книготоргующие организации, контроль выполнения заказов, автоматическая сверка на дублетность.

Присутствуют описание любых видов изданий, включая документы в электронном формате, аудио-, видео-, нотные, картографические, изоиздания и пр.; полная поддержка национального коммуникативного формата RUSMARC, обеспечение контроля вводимых данных на их соответствие требованиям стандартов, автоматическая индексация, на основе которой реализуется быстрый поиск по любым элементам библиографического описания и их сочетаниям, возможность использовать любые графические материалы (изображения титульных листов, артефактов, страниц текста и т.д.), экспорт/импорт данных в MARC-форматах, хранение неограниченного количества библиографических записей.

Также присутствует виртуальный кабинет читателя, куда входят:

* Удаленный доступ читателя к электронному каталогу и поиск изданий по любым элементам библиографического описания.
* Формирование электронной заявки по результатам поиска.
* Удаленный доступ к данным о наличии свободных физических экземпляров изданий в фондах библиотеки.
* Удаленный доступ читателя к своему формуляру.
* Контроль читателя за процессом исполнения сформированных заказов.
* Оповещение читателя о выполненных заказах и его задолженности.



**Рисунок 2** – Интерфейс программы 1С: Библиотека

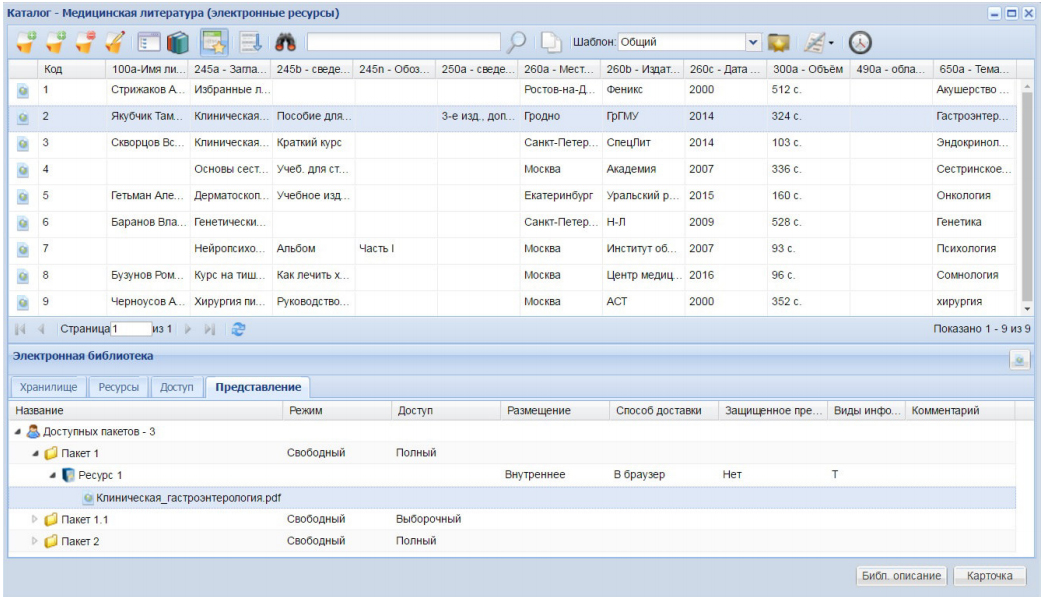
АБИС «Марк» (Windows, MacOS, Linux)

Функциональное назначение - комплексная автоматизация библиотечной деятельности в школе на основе новых информационных технологий.

Создание и ведение электронного каталога учебников, учебных пособий, книг, формирование и печать полного комплекта стандартных выходных форм (каталожных карточек, формуляров, читательских требований, бюллетеней и др.), анализ обеспеченности учебного процесса учебниками, получение статистической информации о читательском спросе, анализ круга чтения учащихся, обслуживание читателей, выдача литературы партиями, получение информации о задолженностях по литературе.

Принципиальные особенности:

* дружественный графический интерфейс,
* создание различных видов библиографических записей,
* режим автоматического определения авторского знака по таблице Хавкиной,
* создание полнотекстовых и мультимедийных ресурсов,
* возможности поиска по любым элементам библиографического описания и его сочетаниям,
* получение результатов поиска в удобной для пользователя форме,
* создание различных видов отчетных документов,
* поддержка штрихкодовой технологии,
* импорт/экспорт в основные коммуникативные форматы RUSMARC, MARC21, UNIMARC,



**Рисунок 3** – Интерфейс программы АБИС «Марк»

Сравнение

Таблица 1. Сравнение библиотечных информационных систем

|  |  |  |  |
| --- | --- | --- | --- |
| Показатель | Учёт книг | 1С: Библиотека | АБИС Марк |
| Эффективность | Есть (Простота в использовании, быстродействие) | Есть (большое количество библиотек, поддерживаемых форматов) | Средняя |
| Гибкость | Нет | Нет | Есть (можно гибко настроить и расширить функциональные возможности благодаря использованию встроенных скриптов (JavaScript и API)). |
| Долговечность | - | 2 года | 18 месяце |
| Стоимость и тех.  поддержка | Низкая (3000 рублей), невысокие требования к системе | Средняя (стоимость варьируется от 10000 до 20000 рублей) | Высокая (сравнительно высокие системные требования, стоимость от 30000 до 70000). |
| Удобство пользователь-ского интерфейса | Высокое (интерфейс не перегружен, большинство функций интуитивно понятны) | Низкое (интерфейс перегружен множеством кнопок, назначение которых рядовому работнику не всегда будут интуитивно понятны) | Высокое (наиболее современный интерфейс, дружелюбный к пользователю) |
| Защищенность | - | Есть | Есть |
| Надёжность | Есть | Есть | Есть |

Таким образом, можно сделать вывод:

Учёт книг является наименее предпочтительным выбором, так как имеет относительно небольшую гибкость и эффективность.

1С: Библиотека является наиболее оптимальным выбором системы учёта книг, но только для крупных библиотек, так как имеют не самую низкую стоимость обслуживания и поддержки.

АБИС «Марк» будет наиболее предпочтительной системой для более мелких библиотек, так как она имеет небольшую стоимость обслуживания, отличается высокой гибкостью в настройке, а также имеет неплохие показатели эффективности и надёжности.

1. **Лабораторная работа 3.**

Назначение и цель создания программы

Назначением создания системы управления библиотекой является информатизация и автоматизация некоторых бизнес-процессов библиотеки, а именно: составление, хранение и представление списка книг, содержащихся в библиотеке; ведение библиотечных карт читателей. Цель – оптимизация бизнес-процессов, обслуживание большего количества посетителей, облегчение процесса выдачи/приёма книг.

Ниже приведена модель «Чёрный ящик» для рассматриваемого бизнес-процесса (Рис. 1).

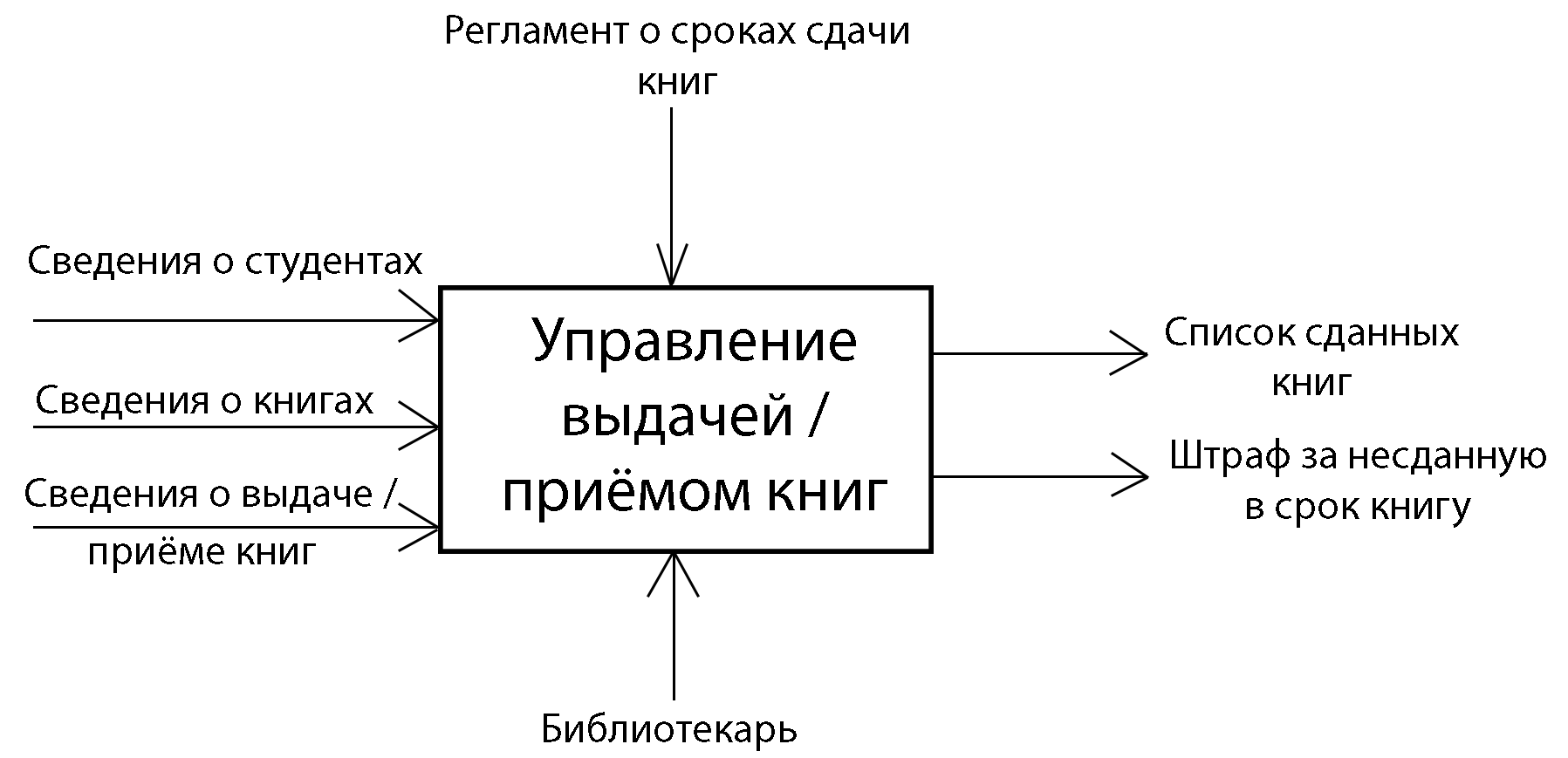


Рисунок 1 – Модель «Чёрный ящик»

Структура программы и состав функциональных задач

Разрабатываемая систему управления библиотекой должна быть веб-приложением, соответственно разделяться на серверную (бэкенд) и клиентскую (фронтенд) составляющие.

Структура программы:

* Главная страница системы;
* Страница, на которой будет происходить добавление данных о новом читателе в базу данных;
* Страница, на которой будет происходить добавление данных о недавно поступивших книгах в базу данных;
* Страница, отображающая книги, которые выданы в данный момент.

Задачи, которые должны быть реализованы:

* Работа со списком книг в базе данных;
* Работа со списком посетителей в базе данных;
* Выдача книги читателю;
* Принятие книги обратно в библиотеку.

Функциональные требования к программе

* Требования к задаче “Работа со списком книг в базе данных”:

Как только в библиотеку поступает новая книга, которая ещё не значится в базе данных системы, в Таблицу “Книги” должна заноситься информация о неё. На вход должна подаваться информация о книге (Название, автор, год издания, статус). Пункт “статус” должен принимать только два значения: “в наличии” и “выдано”, что, соответственно, будет отображать, есть ли данная книга в наличии или же её уже забрали. На выходе должны получить таблицу, которая будет показывать все книги, значащиеся в базе данных библиотеки.

На случай, если книга была уничтожена или каким-то образом изъята из библиотеки, должна быть реализована функция по удалению записи о книги.

* Требования к задаче “ Работа со списком посетителей в базе данных”:

Как только в библиотеку приходит новый посетитель, который ещё не значится в базе данных системы, в Таблицу “Посетители” должна заноситься информация о нём. На вход должна подаваться информация о новом посетителе библиотеки (Фамилия, имя, отчество, паспортные данные, телефонный номер). На выходе должны получить таблицу, которая будет показывать всех посетителей данной библиотеки.

* Требования к задаче “Выдача книги читателю”:

Посетитель просит выдать ему книгу. Если в таблице “Книги” эта книга в поле статус не имеет значение “выдано”, то система должна изменить значение этого поля на значение “выдано”, а также добавить новые данные в таблицу “Выданные книги”, на вход которой подаются следующие данные: идентификатор книги, идентификатор посетителя, дата выдачи книги. Идентификаторы студента и книги должны добавляться в таблицы “Книги” и “Посетители” автоматически при добавлении новых данных.

* Требования к задаче “Принятие книги обратно в библиотеку”:

Посетитель приносит книгу обратно в библиотеку. Система должна удалить соответствующую запись из таблицы “Выданные книги”, сменить значение поля “статус” в таблице “Книги” на “в наличии”, а также выписать штраф, если дата возврата книги была просрочена.

Таблица 1. Поля формы по добавлению новой книги в базу данных

|  |  |  |
| --- | --- | --- |
| № | Наименование поля | Формат данных |
| 1 | Название книги | Строчный |
| 2 | ФИО автора | Строчный |
| 3 | Год издания | Численный |
| 4 | Статус | Строчный |

Таблица 2. Поля формы по добавлению новой посетителей библиотеки в базу данных

|  |  |  |
| --- | --- | --- |
| № | Наименование поля | Формат данных |
| 1 | Фамилия | Строчный |
| 2 | Имя | Строчный |
| 3 | Отчество | Строчный |
| 4 | Серия, номер паспорта | Численный |
| 5 | Телефонный номер | Численный |

Модель требований в нотации UML

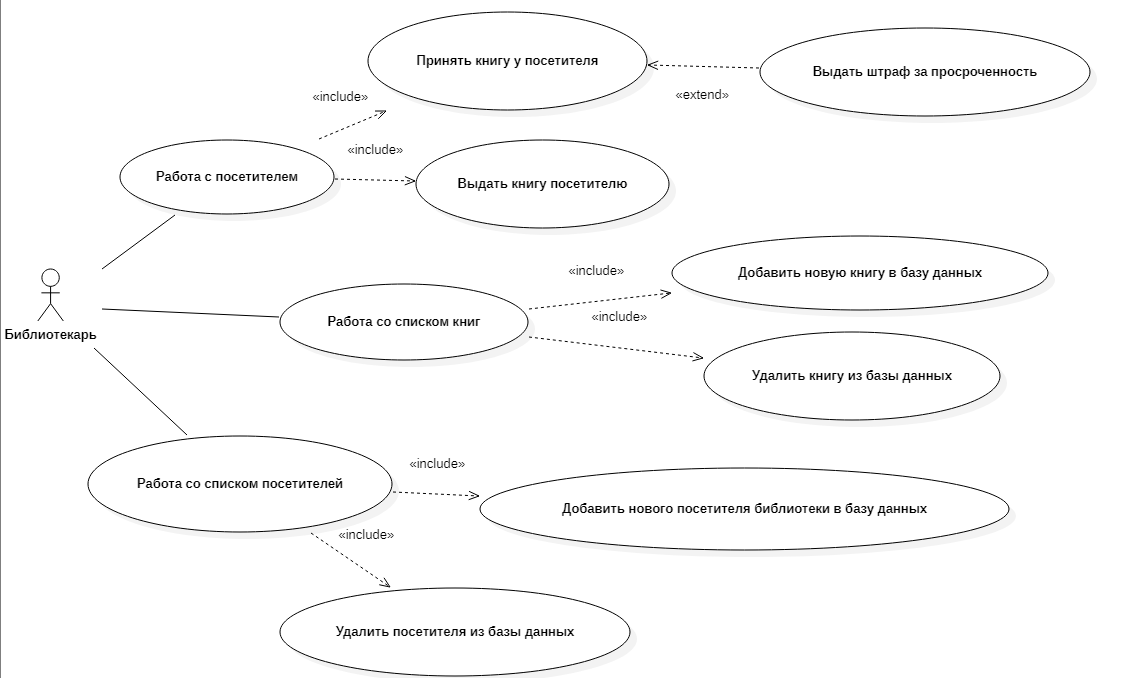


Рисунок 2 – Модель требований в нотации UML

Требования к инструментальному программному обеспечению

* Данные должны быть хорошо организованы, входные и выходные данные должны соответствовать своим таблицам и параметрам;
* данные должны быть достоверными;
* необходима защита от несанкционированного доступа к данным, их порчи и уничтожения.

Требования к инструментальному программному обеспечению

* Операционная система должна быть современной (ОС Windows 7/8/8.1/10);
* Программные пакеты не должны обладать высокой стоимостью;
* Программные пакеты должны быть легки в освоении;
* Для разработки серверной части веб-приложения следует использовать язык программирования Python 3 на фреймворке Django версии 2.3 или выше;
* Код должен быть написать согласно методологии ООП;
* Клиентская часть веб-приложения должна быть написана на стеке HTML/CSS/JS.

Нефункциональные требования

* Программа не должна позволять себе порчу входных и выходных данных;
* Программа должна иметь технологию безопасного хранения данных (таких, как паспортные данные посетителей);
* Программа должна иметь простой и понятный для пользователя интерфейс;
* Программа должна быть легка в освоении рабочим персоналом;
* Программа должна быть легко модифицируемой;

1. **Лабораторная работа 4.**

Построим функциональную модель системы управления библиотекой в виде контекстной диаграммы в нотации IDEF0 на рис. 1.

Затем построим диаграмму декомпозиции A0 на рис. 2.

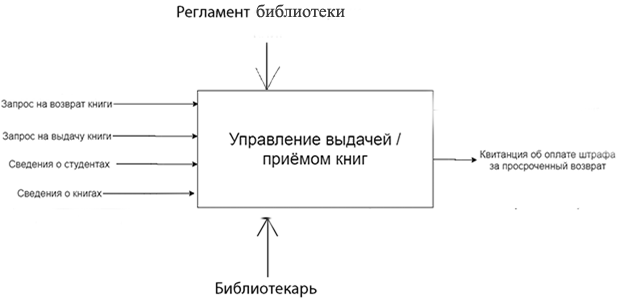


Рис. 1. - Диаграмма «Чёрный ящик»

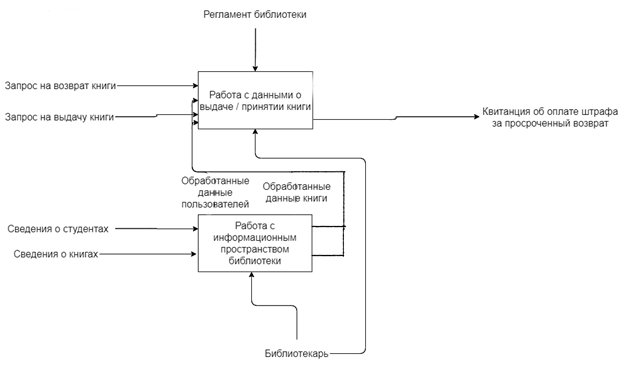


Рис. 2. Диаграмма декомпозиции A0

Рассмотрим отдельно декомпозиции блоков “Работа с данными о выдаче / принятии книг”, “Работа с информационным пространством библиотеки”.

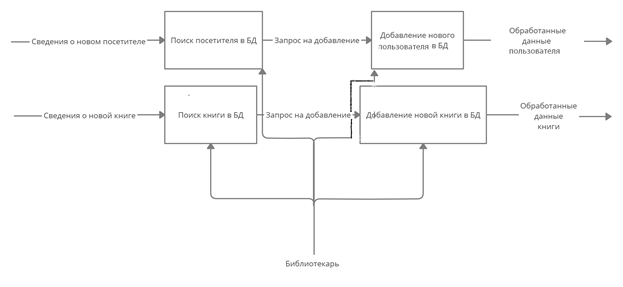


Рис. 3. Диаграмма декомпозиции блока “Работа с информационным пространством библиотеки”

Таблица 1. Activity Dictionary блока “Работа с информационным пространством библиотеки”

|  |  |
| --- | --- |
| Наименование действия | Определение действия |
| Поиск посетителя в БД | Ищет данные посетителя в БД (ФИО, серия, номер паспорта, телефонный номер). |
| Поиск книги в БД | Ищет данные книги в БД (Название, ФИО автора, год издания, статус). |
| Добавление нового пользователя в БД | Добавляет данные о новом посетителе в БД. |
| Добавление новой книги в БД | Добавляет данные о новой книге в БД. |

Таблица 2. Arrow Dictionary блока “Работа с информационным пространством библиотеки”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование стрелки | Источник стрелки | Тип стрелки источника | Приемник стрелки | Тип стрелки приемника |
| Сведения о новом посетителе | Внешняя граница | Output | Поиск посетителя в БД | Input |
| Сведения о новой книге | Внешняя граница | Output | Поиск книги в БД | Input |
| Запрос на добавление | Поиск посетителя в БД | Output | Добавление нового посетителя в БД | Input |
| Запрос на добавление | Поиск книги в БД | Output | Добавление новой книги в БД | Input |
| Обработанные данные пользователя библиотеки | Добавление нового посетителя в БД | Output | Внешняя граница | Input |
| Обработанные данные новой книги | Добавление новой книги в БД | Output | Внешняя граница | Input |
| Библиотекарь | Внешняя граница | Mechanism | Поиск книги в БД | Mechanism |
| Библиотекарь | Внешняя граница | Mechanism | Поиск нового посетителя в БД | Mechanism |
| Библиотекарь | Внешняя граница | Mechanism | Добавление нового посетителя в БД | Mechanism |
| Библиотекарь | Внешняя граница | Mechanism | Добавление новой книги в БД | Mechanism |

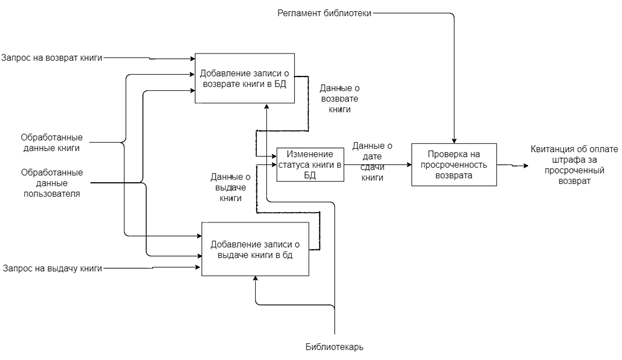


Рис. 4. Диаграмма декомпозиции блока “ Работа с данными о выдаче / принятии книг”

Таблица 3. Activity Dictionary блока “Работа с данными о выдаче / принятии книг”

|  |  |
| --- | --- |
| Наименование действия | Определение действия |
| Добавление записи о выдаче книги в БД | Добавляет идентификатор книги, идентификатор посетителя и дату выдачи книги в таблицу “Выданные книги” в БД. |
| Добавление записи о возврате книги в БД | Удаляет соответствующую запись в таблице “Выданные книги” в БД. |
| Изменение статуса книги в БД | Меняет статус книги в таблице “Книги”. |
| Проверка на просроченность вовзврата. | Проверяет, является ли возврат книги просроченным согласно регламенту библиотеки. |

Таблица 4. Arrow Dictionary блока “Работа с данными о выдаче / принятии книг”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование стрелки | Источник стрелки | Тип стрелки источника | Приемник стрелки | Тип стрелки приемника |
| Обработанные данные книги | Внешняя граница | Output | Добавление записи о возврате книги в БД | Input |
| Обработанные данные пользователя | Внешняя граница | Output | Добавление записи о возврате книги в БД | Input |
| Обработанные данные книги | Внешняя граница | Output | Добавление записи о выдаче книги в БД | Input |
| Обработанные данные пользователя | Внешняя граница | Output | Добавление записи о выдаче книги в БД | Input |
| Данные о выдаче книги | Добавление записи о выдаче книги в БД | Output | Изменение статуса книги в БД | Input |
| Данные о возврате книги | Добавление записи о возврате книги в БД | Output | Изменение статуса книги в БД | Input |
| Данные о дате сдачи книги | Изменение статуса книги в БД | Output | Проверка на просроченность возврата | Input |
| Запрос на возврат книги | Внешняя граница | Output | Добавление записи о возврате книги в БД | Input |
| Запрос на выдачу книги | Внешняя граница | Output | Добавление записи о выдаче книги в БД | Input |

Продолжение таблицы 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Список выданных книг | Изменение статуса книги в БД | Input | Внешняя граница | Output |
| Квитанция о выплате штрафа за просроченность | Проверка на просроченность возврата | Input | Внешняя граница | Output |
| Регламент библиотеки | Внешняя граница | Control | Проверка на просроченность возврата | Control |
| Библиотекарь | Внешняя граница | Mechanism | Добавление записи о выдаче книги в БД | Mechanism |
| Библиотекарь | Внешняя граница | Mechanism | Добавление записи о возврате книги в БД | Mechanism |

1. **Лабораторная работа 4.1**

При анализе описания функционирования программной системы были выявлены следующие абстракции: библиотекарь, данные о книге, данные о посетителе библиотеки, данные о выдаче книги, данные о возврате книги, регламент библиотеки, квитанция за просроченный возврат книги.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракция | Тип | Описание |
| 1 | Библиотекарь | Абстракция сущности | Человек, работающий с программой. Осуществляет работу со списком книг, посетителей в БД. Отвечает за выдачу/приём книги. |
| 2 | Данные о книге | Абстракция сущности | Данные книг: автор, название, год издательства. |
| 3 | Данные о посетителе библиотеки | Абстракция сущности | Данные человека, посещающего библиотеку: ФИО, серия, номер паспорта, телефонный номер. |
| 4 | Данные о выдаче книги | Абстракция сущности | Данные о выдаче книги посетителю: id посетителя, id книги, дата выдачи книги. |
| 5 | Данные о возврате книги | Абстракция сущности | Данные о выдаче книги посетителю: id посетителя, id книги, дата возврата книги. |
| 6 | Регламент библиотеки | Абстракция сущности | Нормативный правовой акт, устанавливающий порядок предоставления муниципальной услуги и стандарт предоставления муниципальной услуги. |
| 7 | Выход за сроки возврата книги | Абстракция поведения | Событие, возникающее, если посетитель просрочил возврат книги в соответствии с регламентом библиотеки. |

Таблица 1 – Абстракции подсистемы

Проведем классификацию обнаруженных абстракций по классическому подходу. Результат отображен в Таблице 2.

|  |  |  |
| --- | --- | --- |
| № | Класс | Список абстракций |
| 1 | Люди | Библиотекарь |
| 2 | Места | - |
| 3 | Предметы | Данные о книге, данные о посетителе библиотеки, данные о выдаче/возврате книги |
| 4 | Организации | - |
| 5 | Концепции | Регламент библиотеки |
| 6 | События | Выход за сроки возврата книги |

Проанализируем поведение выделенных абстракций. Выделим возможное поведение каждой абстракции в пределах рассматриваемой функции и составим Таблицу 3.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракция | Поведение | Описание поведения |
| 1 | Библиотекарь | Локальное | Добавляет новые записи в базу данных |
| 2 | Регламент библиотеки | Элементарное | Устанавливает правила предоставления услуги библиотекой посетителю по выдаче / принятию книг |
| 3 | Выход за сроки возврата книги | Элементарное | Выписывает квитанцию в случае, если дата возврата книги является просроченной |

Таблица 3 -Абстракции подсистемы и их поведение.

Построим диаграмму классов UML. Результат приведен на Рисунке 1.

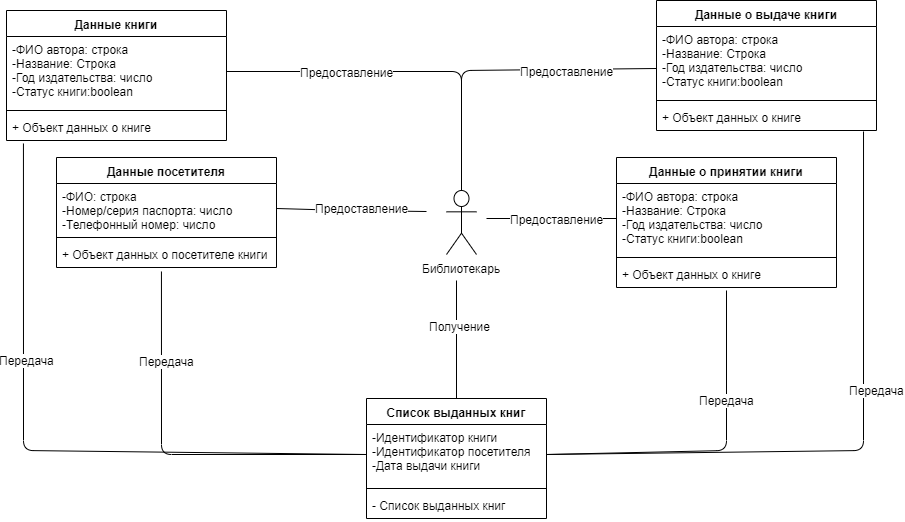


Рисунок 1 – Расширенная диаграмма классов UML.

Выберем класс «Список выданных книг», и построим для него диаграмму состояния. Результат представлен на Рисунках 2 и 3.

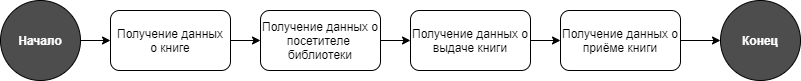


Рисунок 2 – Диаграмма состояний для класса «Список выданных книг»

На основе анализа функциональных моделей, разработанных при выполнении лабораторной работы №4, для каждого из базовых вариантов использования построим диаграмму деятельности.

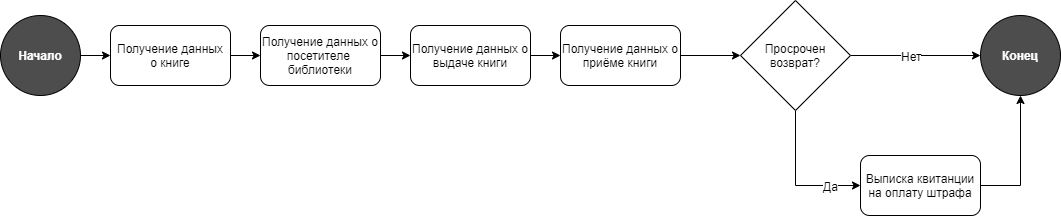


Рисунок 3 – Расширенная диаграмма деятельности для класса «Список выданных книг»

Для каждого варианта использования выделим список объектов, участвующих во взаимодействии в этом прецеденте, заполним таблицу 4.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Прецендент | Объект | Описание объекта |
| 1.1 | Выписка квитанции на оплату штрафа | Регламент | Устанавливает нормы |
| 1.2 | Выписка квитанции на оплату штрафа | ПС | Выписывает квитанцию на оплату штрафа, если срок возврата книги превысил срок, указанный в регламенте |
| 1.3 | Выписка квитанции на оплату штрафа | Данные о возврате книги | Дата возврата книги |

Таблица 4 – Список объектов для каждого потока событий

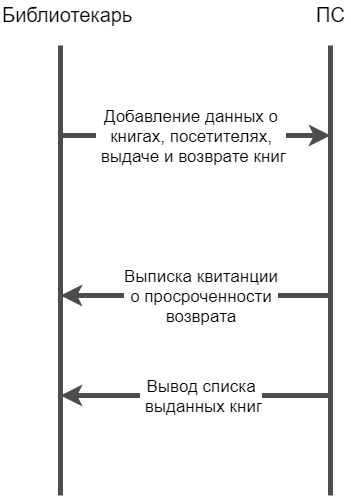


Рисунок 4 – Диаграмма последовательности для прецедента «Список выданных книг»

1. **Лабораторная работа 5**

На основе анализа предметной области, проведенного в Лабораторной работе №3 и №1 произведем идентификацию сущностей информационной базы ИС и связей между ними.

Нормативно-справочная информация:

1. Регламент библиотеки – нормативный правовой акт, устанавливающий порядок предоставления муниципальной услуги и стандарт предоставления муниципальной услуги. Используется на этапе проектирования программы с целью установить нормы сдачи книг в срок.

Входная (текущая) информация:

1. Сведения о посетителе библиотеки – содержит сведения о посетителях библиотеки (ФИО, серия/паспорт, телефонный номер).
2. Сведения о книге – содержит сведения о книге (название, количество, автор, издательство).
3. Сведения об авторе книги – содержит сведения об авторе книги (ФИО).
4. Сведения об издательстве – содержит сведения об издательстве (название, год).
5. Сведения о дате выдаче книги посетителю библиотеки – содержит сведения о дате выдачи книги посетителю.
6. Сведения о дате возврата книги библиотеке – содержит сведения о дате возврата книги библиотеке.

Проведем нормализацию сущностей логической модели данных и разработаем ER-диаграмму физической модели данных. Имена, атрибуты и назначение сущностей физической модели данных приведем в таблице 1.

Таблица 1. Сущности физической модели данных

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сущность | Атрибуты | Описание |
| Справочные | | | |
| 1 | Посетители | ID посетителя, фамилия, имя, отчество, серия паспорта, номер паспорта, телефонный номер. | Информация о посетителях библиотеки |
| 2 | Книги | ID книги, название, фамилия автора, имя автора, отчество автора, год издательства, статус выдачи | Информация о книгах, имеющихся в расположении библиотеки |
| 3 | Авторы | ID автора, фамилия, имя, отчество | Информация об авторах, написавших книги |
| 4 | Издательства | ID издательства, имя издательства | Информация об издательствах, выпускающих книги |
| Оперативные | | | |
| 5 | Выданные книги | ID транзакции, ID книги, ID посетителя библиотеки, дата выдачи книги, дата возврата книги | Информация о всех выдачах/возвратах книг в библиотеке. |

Средствами ERWin на основе физической модели данных выполним генерацию SQL-кода для создания реляционной базы данных ПС и представим его на рисунке 1.

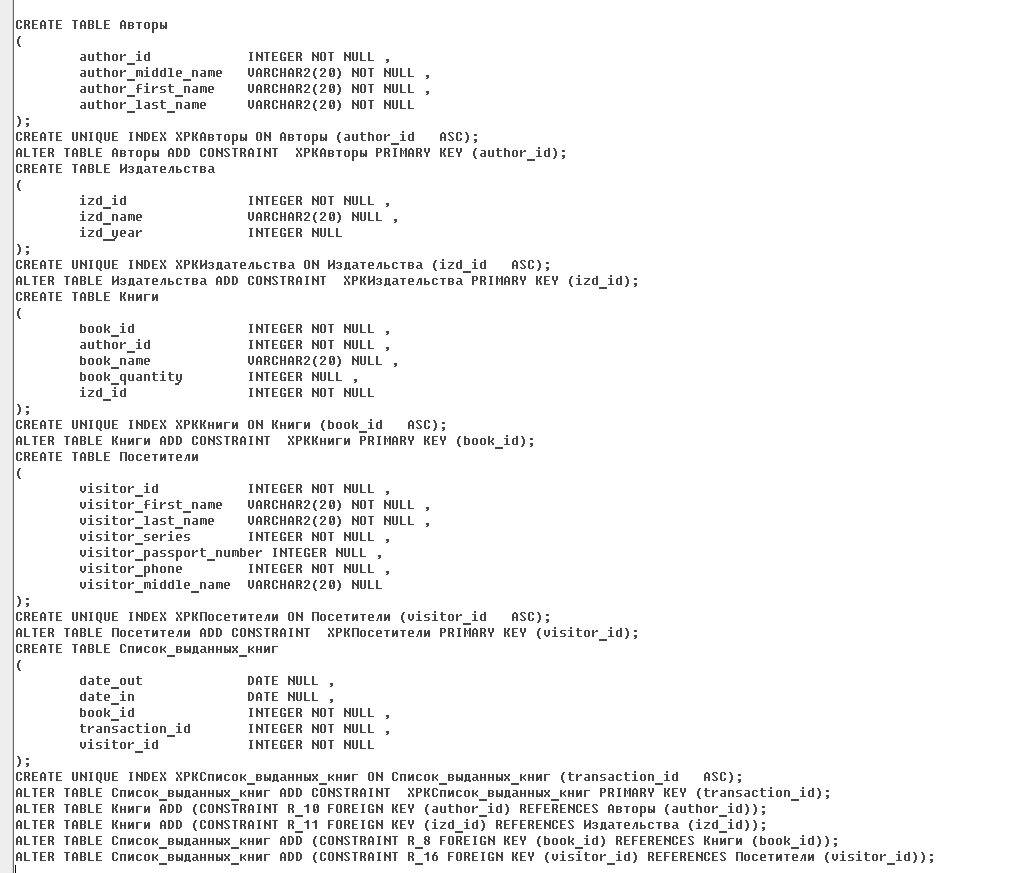


Рисунок 1 – SQL-код для создания реляционной базы данных ПС

В выбранной СУБД –SQLite3 развернём БД, доработаем её структуру с учетом возможной нормализации отношений, а также доработаем структуру таблиц с учетом ограничений на значения полей и сформируем ER-диаграмму БД на рисунке 2.

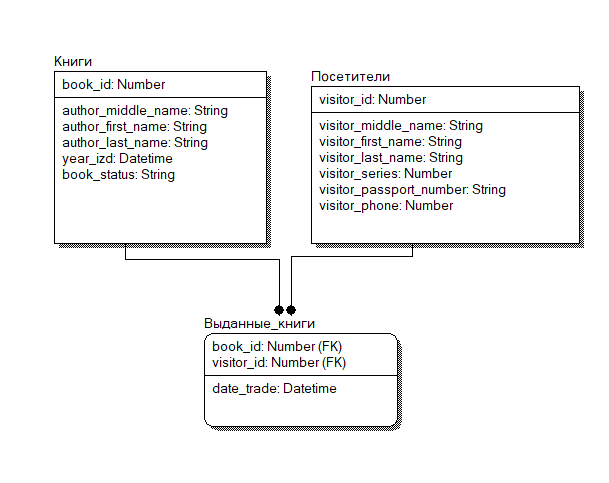


Рисунок 2 - ER-диаграмма физической модели данных

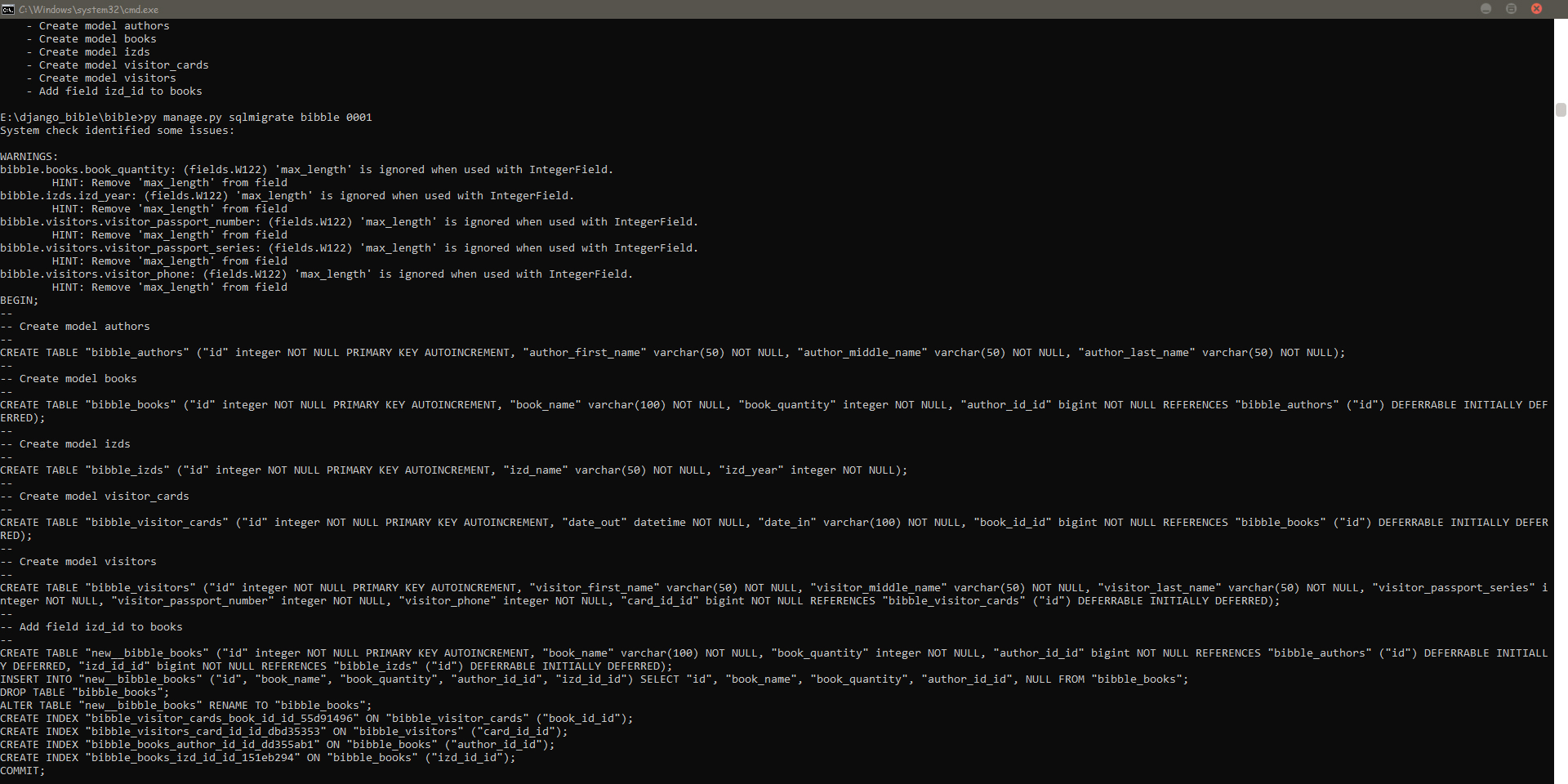


Рисунок 3 – развёртывание БД при помощи Django

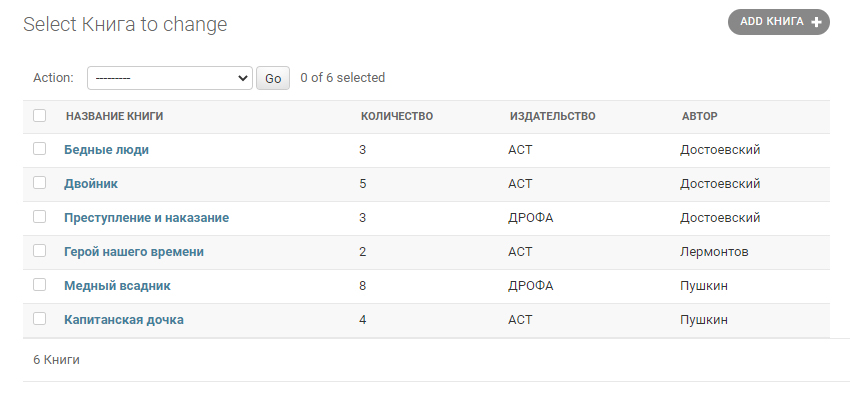


Рисунок 4 – пример заполнения таблицы Книги

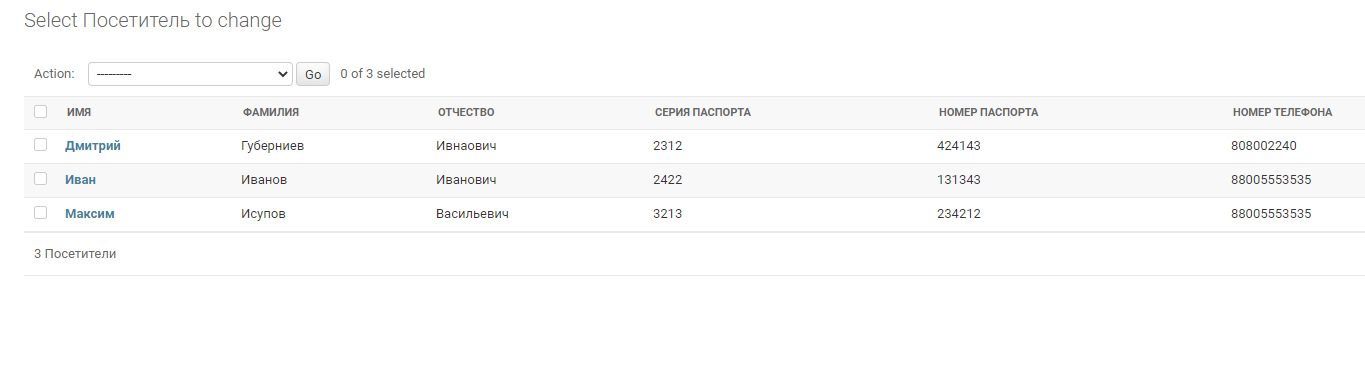


Рисунок 5 – пример заполнения таблицы Посетители

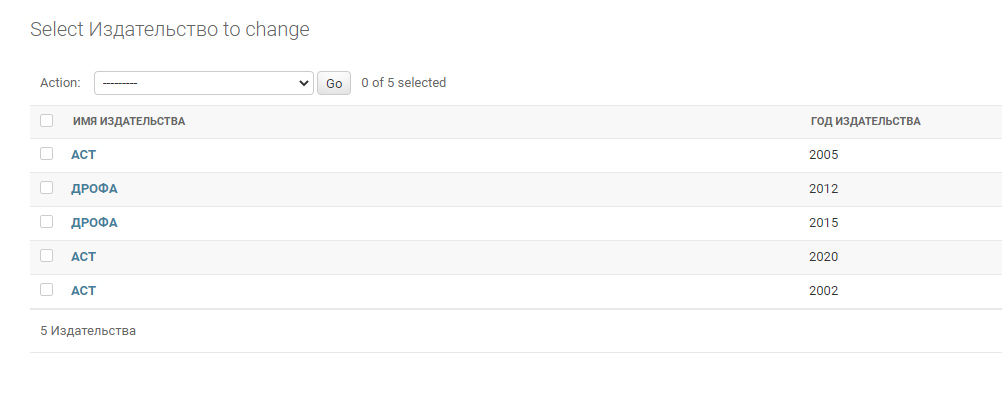


Рисунок 6 – пример заполнения таблицы Издательства

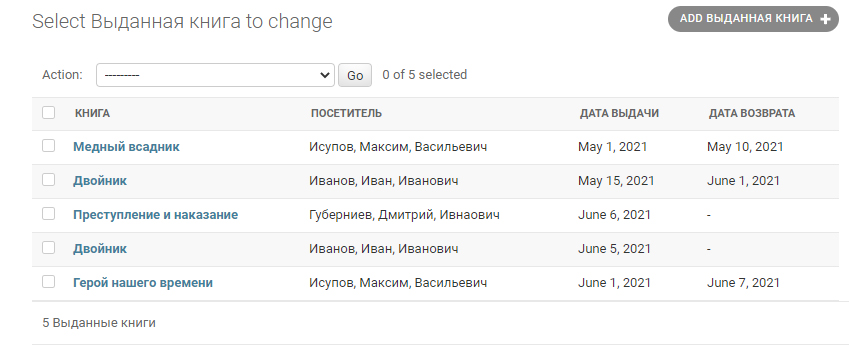


Рисунок 7 – пример заполнения таблицы Выданные книги

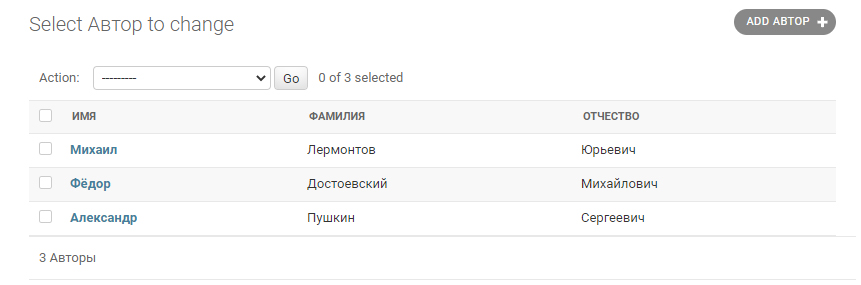


Рисунок 8 – пример заполнения таблицы Авторы

Приведём описание таблиц БД в таблицах 2 – 4.

Таблица 2. Структура таблицы Книги

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Значение по умолчанию | Примечание |
| book\_id | integer | 4Б |  | - | Обязательное поле |
| book\_name | varchar(20) | 40Б |  | - | Обязательное поле |
| book\_quantity | integer | 4Б |  | - | Обязательное поле |
| author\_id | integer | 4Б |  | - | Обязательное поле |
| Izd\_id | integer | 3Б |  | - | Обязательное поле |

Таблица 3. Структура таблицы Посетители

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Значение по умолчанию | Примечание |
| visitor\_id | integer | 4Б |  | - | Обязательное поле |
| visitor\_middle\_name | varchar(20) | 40Б |  | - | Обязательное поле |
| visitor\_first\_name | varchar(20) | 40Б |  | - | Обязательное поле |
| visitor\_last\_name | varchar(20) | 40Б |  | - | Обязательное поле |
| visitor\_series | integer | 4Б |  | - | Обязательное поле |
| visitor\_passport\_number | integer | 4Б |  | - | Обязательное поле |
| visitor\_phone | integer | 4Б |  | - |  |

Таблица 4. Структура таблицы Выданные книги

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Значение по умолчанию | Примечание |
| transaction\_id | integer | 4Б |  | - | Обязательное поле |
| visitor\_id | integer | 4Б |  | - | Обязательное поле |
| book\_id | integer | 4Б |  | - | Обязательное поле |
| date\_out | date | 3Б |  | - | Обязательное поле |
| date\_out | date | 3Б |  | - | - |

Таблица 5. Структура таблицы Авторы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Значение по умолчанию | Примечание |
| author\_id | integer | 4Б |  | - | Обязательное поле |
| author\_middle\_name | integer | 4Б |  | - | Обязательное поле |
| Author\_first\_name | integer | 4Б |  | - | Обязательное поле |
| Author\_last\_name | date | 3Б |  | - | Обязательное поле |

Таблица 6. Структура таблицы Издательства

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Значение по умолчанию | Примечание |
| izd\_id | integer | 4Б |  | - | Обязательное поле |
| izd\_name | Varchar(20) | 40Б |  | - | Обязательное поле |
| izd\_year | integer | 4Б |  | - | Обязательное поле |

Опишем связи между таблицами БД в таблице 5.

Таблица 5. Связи между таблицами БД

|  |  |  |
| --- | --- | --- |
| Родительская таблица | Дочерняя таблица | Тип связи |
| Книги | Выданные книги | Один ко многим |
| Посетители | Выданные книги | Один ко многим |
| Авторы | Книги | Один ко многим |
| Издательства | Книги | Один ко многим |

1. **Лабораторная работа 6**

В соответствии с требованиями технического задания, разработанного при выполнении лабораторной работы №3, проведём обоснованный выбор средства разработки специального ПО. Язык программирования Python был выбран по причине того, что он имеет модуль Django а также встроенную базу данных Sqlite3. Разработаем схему общесистемного ПО.



Рисунок 1 – Структурная схема общесистемного программного обеспечения

Приложение было создано на базе модуля для Python – Django. В качестве базы данных использовалась встроенная база данных sqlite3, задаётся она в модуле models.

 Рисунок 2 – Часть кода для создания БД

Выполним описание разработанных компонентов приложения в таблице 1.

Таблица 12. Перечень разработанных компонентов приложения

|  |  |  |
| --- | --- | --- |
| № | Имя компонента | Описание |
| 1 | Главная страница | Содержит ссылки на все остальные компоненты приложения |
| 2 | Книги | Содержит данные обо всех книгах, присутствующих в БД библиотеки. Здесь осуществляется переход на форму добавления книг. |
| 3 | Авторы | Содержит данные обо всех авторах, присутствующих в БД библиотеки. Здесь осуществляется переход на форму добавления авторов. |
| 4 | Издательства | Содержит данные обо всех издательствах, присутствующих в БД библиотеки. Здесь осуществляется переход на форму добавления издательств. |
| 5 | Посетители | Содержит данные обо всех посетителях, зарегистрированных в библиотеке. Здесь осуществляется переход на форму регистрации новых посетителей. |
| 6 | Список выданных книг | Содержит записи о всех процессах выдачи/возврата книг в библиотеке. |
| 7 | Форма добавления/изменения книг | Представляет собой форму, которая содержит поля для заполнения таблицы Книги |
| 8 | Форма добавления/изменения авторов | Представляет собой форму, которая содержит поля для заполнения таблицы Авторы |
| 9 | Форма добавления/изменения издательств | Представляет собой форму, которая содержит поля для заполнения таблицы Издательства |
| 10 | Форма регистрации/ изменения посетителей | Представляет собой форму, которая содержит поля для заполнения таблицы Посетители |
| 11 | Форма добавления/изменения списка выданных книг | Представляет собой форму, которая содержит поля для заполнения таблицы Выданные книги |

Представим экранные формы компонентов приложения в виде рисунков



Рисунок 3 – Представление компонента Книги

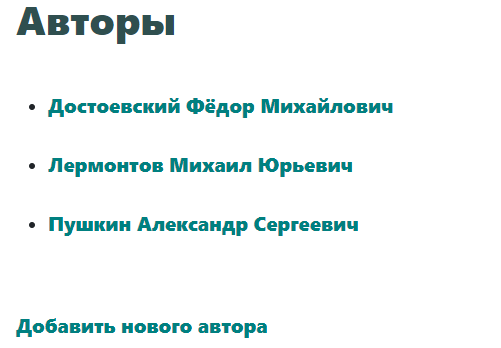


Рисунок 4 – Представление компонента Авторы

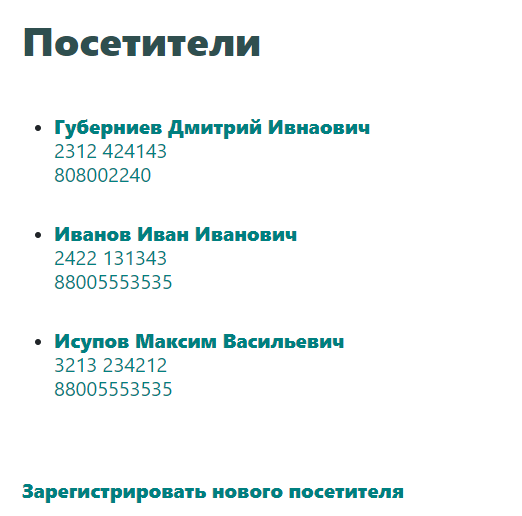


Рисунок 5 – Представление компонента Авторы

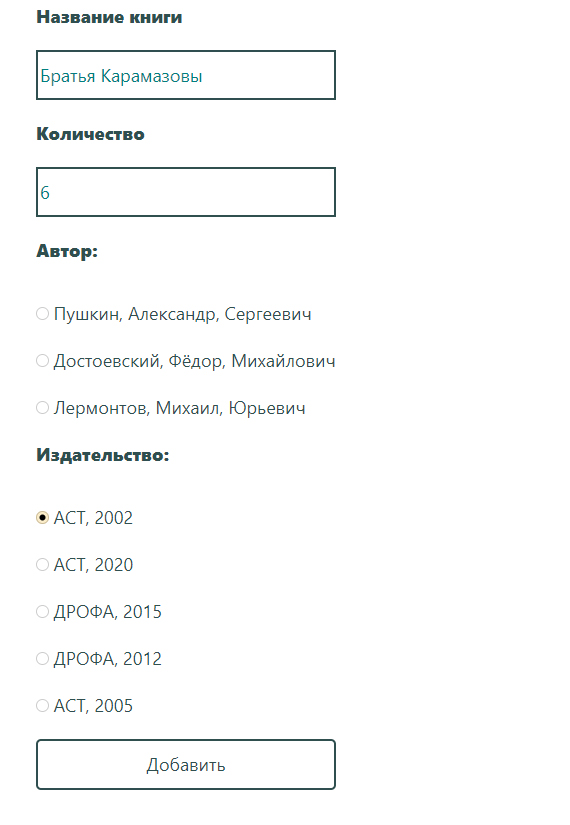


Рисунок 6 – Фрагмент формы по добавлению новой книги в БД библиотеки

Выполним описание разработанных отчетов в таблице 2.

Таблица 2. Перечень разработанных отчетов в приложении

|  |  |  |
| --- | --- | --- |
| №  пп | Имя отчета | Описание |
| 1 | Вывод списка выданных книг | Вывод данных таблицы в БД “Список выданных книг” на экран |

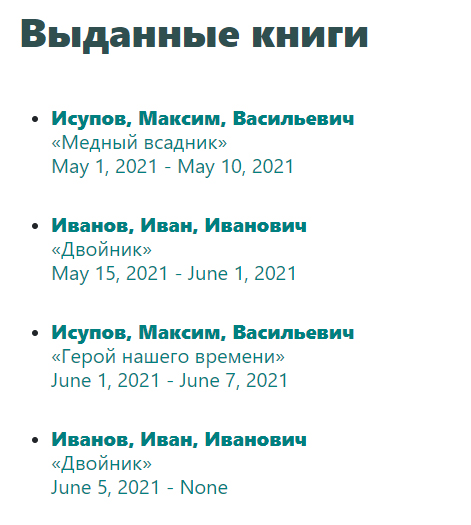


Рисунок 7 – Вывод списка выданных книг на экран

Разработаем структурную схему приложения, выражающую взаимодействие его компонентов в процессе функционирования приложения. Построим декомпозицию ПС управления библиотекой на рисунке 8.

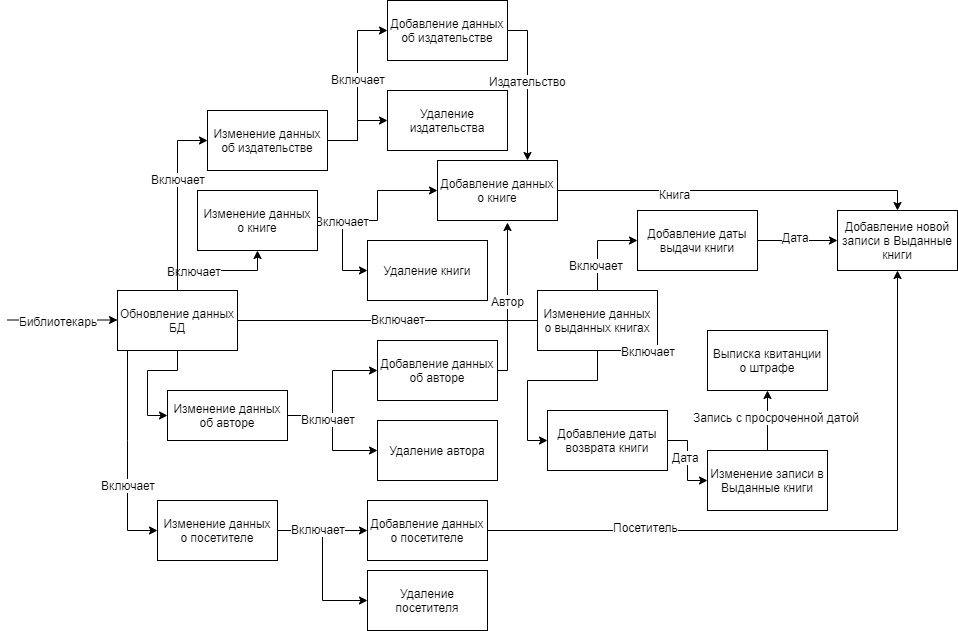


Рисунок 8 – Декомпозиция ПС управления библиотекой

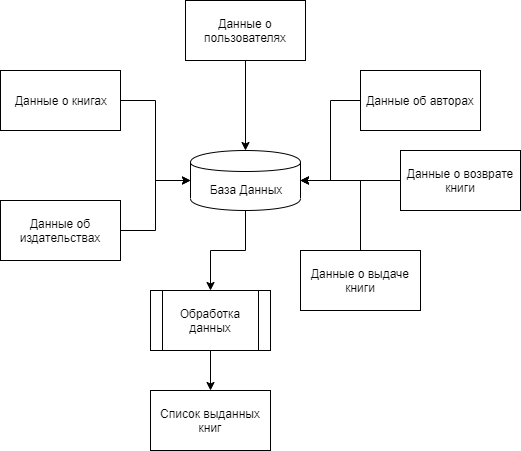


Рисунок 9 – Схема взаимодействия компонентов специального ПО

После компиляции и запуска приложения синтаксических ошибок не обнаружено, результаты соответствуют заявленным и не имеют логических ошибок.

1. **Лабораторная работа 7.**

Выполним системное пользовательское тестирование работоспособности ПС посредством воздействия на её интерфейсную часть. При проведении тестов были обнаружены следующие ошибки:

* Отображение пустого значения в таблице Выданные книги как none. Для решения данной проблемы была написана функция, которая проверяет данную строку на пустоту.

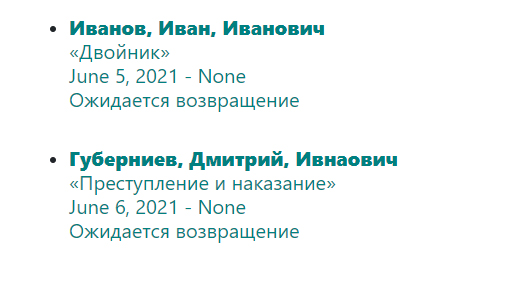


Рисунок 1 – Ошибка с отображением пустого значения в таблице Выданные книги

Выполним нагрузочное тестирование программы и оценим эффективность разработанных автоматизированных функций ПС, запросов к БД. В качестве критерия эффективности будем использовать время выполнения функции. Предварительно подготовим 5 массивов исходных данных на 10, 50, 100, 500 и 1000 записей.



Рисунок 2 – График зависимости времени работы программы от количества записей

Разработаем модульный тест для проверки работоспособности кода с помощью встроенной библиотеки в Django TestCase, реализующего автоматизированные функции ПС.

Результаты автоматизированного модульного теста приведены на рисунках 3-4.



Рисунок 3 – Фрагмент кода автоматизированного тестирования

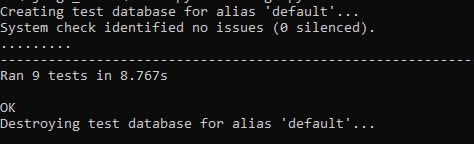


Рисунок 4 – Результат выполнения автоматизированного тестирования

1. **Руководство пользователя**

Данное ПО является:

* свободным,
* сделанным на заказ,
* платформозависимым.

Технические требования для запуска программы:

Данная программа работает с помощью IDLE или другой среды разработки на Python 3. Требуется установить python фреймворк Django.

Меры безопасности:

Программа не защищена от копирования и является свободно распространяемым ПО с открытым исходным кодом.

Руководство:

Для того, чтобы развернуть сайт, требуется в командной строке перейти в каталог Django приложения и ввести следующую команду: python manage.py runserver.

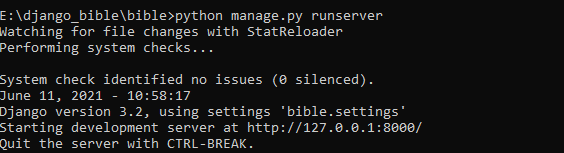


Рисунок 1 – Запуск сайта с помощью командной строки

Для создания аккаунта администратора следует воспользоваться командой python manage.py createsuperuser.

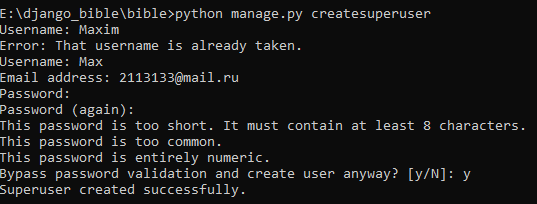


Рисунок 1 – Создание аккаунта администратора

1. **Выводы**

**Лабораторная работа 1.** Проведен системный анализ бизнес-процесса управления выдачей / приёмом книг и выполнена декомпозиция на подпроцессы. Построены модель «Чёрный ящик» и модель прецедентов. Выполнено описание процессов, операций управления выдачей / приёмом книг.

**Лабораторная работа 2.** Были рассмотрены три популярные системы учёта книг в библиотеках, произведен их анализ, выделены достоинства и недостатки каждой системы, а также показана их применимость для библиотек.

**Лабораторная работа 3.** Были установлены общая цель и назначение программы, определена структура программы и её функциональный состав, определены функциональные и нефункциональные требования к программе, разработаны требования к инструментальному ПО.

**Лабораторная работа 4.** Была спроектирована функциональная структура системы управления библиотеки. Составлены диаграммы «Чёрных ящик», её декомпозиция и диаграммы дальнейших декомпозиций на операции.

**Лабораторная работа 4.1.** Были изучены методики объектно-ориентированного подхода программной инженерии для системы управления библиотекой.

**Лабораторная работа 5.** Были изучены программные средства для разработки моделей информационной базы ПС, проработан метод нормализации отношений в базе данных и приобретены навыки применения CASE-средста ERwin для моделирования базы данных ПС.

**Лабораторная работа 6.** Приобретены навыки разработки интерфейсной части программной системы при помощи современных средств разработки.

**Лабораторная работа 7.** Были освоены методики тестирования ПС в ручном режиме, выполнено системное пользовательское тестирование работоспособности ПС посредством воздействия на её интерфейсную часть, нагрузочное тестирование ПС, а также был разработанный автоматизированный модульный тест для тестирования ПС.

**Лабораторная работа 8.** Были изучены основы документации программ, составлено техническое описание и характеристики программы.