Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Национальный исследовательский университет**

**«Высшая школа экономики»**

Факультет компьютерных наук

ООП «Прикладная математика и информатика»

**Отчёт по проекту**

**Студент:** Коновалов Евгений (165 гр.)

**Место выполнения проекта:** НИУ ВШЭ

Москва, 2017

Оглавление

[Техническое задание 3](#_Toc532735965)

[Концептуальное проектирование 7](#_Toc532735966)

[Краткое описание предметной области 7](#_Toc532735967)

[Процесс построения инфологической модели. 8](#_Toc532735968)

[ER-Диаграмма с комментариями. 9](#_Toc532735969)

[Проектирование реляционной модели: 10](#_Toc532735970)

[Процесс перехода к реляционной модели. 10](#_Toc532735971)

[Диаграмма реляционной модели (схемы БД) 10](#_Toc532735972)

[Дополнительные механизмы обеспечения целостности данных. 10](#_Toc532735973)

[Развёртывание БД в выбранной СУБД 10](#_Toc532735974)

[DDL-скрипт создания схемы БД. 10](#_Toc532735975)

[Примеры DML-операторов вставки текстовых данных. 10](#_Toc532735976)

[Контроль работы ограничений целостности с примерами. 10](#_Toc532735977)

[Разработка клиентского приложения 10](#_Toc532735978)

[Бизнес-логика тестового приложения. 10](#_Toc532735979)

[Техническое описание тестового приложения. 10](#_Toc532735980)

[Примеры ввода и редактирования данных в БД 10](#_Toc532735981)

[Запросы и отчёты. 10](#_Toc532735982)

[Результаты функционального тестирования. 11](#_Toc532735983)

[Заключение. 11](#_Toc532735984)

База данных «Сеть библиотек»

# Техническое задание

**1. Введение**

**1.1 Наименование программы**

Наименование программы: «База данных сети библиотек».

**1.2 Краткая характеристика области применения**

База данных предназначена для организации деятельности сети библиотек города Москвы.

**1.3 Условные обозначения и сокращения**

БД - База данных;

ТЗ - Техническое задание

СУБД-Система управления базой данных

**2. Основания для разработки**

Основанием для разработки БД является задание на проектное проектирование по междисциплинарному курсу «Теория баз данных», выданное 01 октября 2018 года. преподавателем Незнановым А.А.

**2.1 Наименование и условное обозначение темы разработки**

Наименование темы разработки - Разработка БД «Сеть библиотек»

**3. Назначение разработки**

**3.1 Функциональное назначение**

Функциональное назначением БД является информационное обеспечение сотрудников

сети о клиентах, книгах, заказах, сроках их исполнения, и наличии книг в соответствующих библиотеках.

**4. Требования к программе или программному изделию**

**4.1 Требования к функциональным характеристикам**

Требования к составу выполненных функций

Программа должна выполнять следующие функции:

1. Добавление, редактирование, просмотр персональной информации о клиентах;
2. Добавление, редактирование, просмотр информации по заказам клиентов;
3. Добавление, редактирование, просмотр, удаление информации по книгам в наличии;

4. Поиск информации:

- О клиенте по его ФИО;

- О клиенте по его домашнему адресу;

- О клиентах, чей штраф превышает заданный;

- О клиентах, зарегистрировавшихся в заданный день;

- О (активных) заказах клиента (в конкретной библиотеке);

- О просроченных заказах клиентов (в конкретной библиотеке);

- О (активных) заказах книги (в конкретной библиотеке);

- О книге по её названию;

- О книгах по жанру

4.1.1 Требования к организации входных данных.

Большая часть данных была взята из открытого источника kaggle-соревнования

<https://www.kaggle.com/applecrazy/cmu-book-summary-dataset>,

Недостающие данные в виде пользователей и библиотек были внесены вручную символами латиницы.

**4.1.3 Требования к организации выходных данных**

Выходные данные соответствуют входным данным. Просмотр выходных данных осуществляется на мониторе и должен отображаться в виде таблицы.

|  |  |  |  |
| --- | --- | --- | --- |
| **Client Table** |  |  |  |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Client ID | Int | 4 | Primary Key |
| Name | Text | Dynamic |  |
| Address | Text | Dynamic |  |
| Check-In Date | Date | 4 |  |
| Fine | Int | 4 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Order Table** |  |  |  |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Order ID | Int | 4 | Primary Key |
| Date From | Date | 4 |  |
| Date To | Date | 4 |  |
| Active | Bool | 1 |  |
| Client ID | Int | 4 | Link to the Client’s Table |
| Book ID | Int | 4 | Link to the Book’s Table |
| Library ID | Int | 4 | Link to the Library’s Table |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Library Table** |  |  |  |  |
| **Field Name** | **Field Type** | **Size** | **Description** |  |
| Library ID | Int | 4 | Primary Key |  |
| Name | String | Dynamic |  |  |
| Address | String | Dynamic |  |  |
| Working Hours Begin | Time | 4 |  |  |
| Working Hours End | Time | 4 |  |  |
| **Book Table** |  |  |  |  |
| **Field Name** | **Field Type** | **Size** | **Description** | |
| Book ID | Int | 4 | Primary Key | |
| Name | String | Dynamic |  | |
| ISNB | String | Dynamic |  | |
| Quantity | Int | 4 |  | |
| Library ID | Int | 4 | Link to the Library’s Table | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Author Table** |  |  |  |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Author ID | String | 30 | Primary Key |
| Name | String | 30 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Genre Table** |  |  |  |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Genre ID | Int | 4 | Primary Key |
| Name | String | Dynamic |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Connector Author-Book** |  |  |  |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Author ID | Int | 4 | Paired Primary Key |
| Book ID | Int | 4 | Paired Primary Key |

|  |  |  |  |
| --- | --- | --- | --- |
| **Connector Genre-Book** |  |  |  |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Genre ID | Int | 4 | Paired Primary Key |
| Book ID | Int | 4 | Paired Primary Key |

**4.1.4 Требование к временным характеристикам**

Требования к временным характеристикам БД не предъявляются.

**4.2 Требования к надежности**

**4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы**

Надежное функционирование БД должно быть обеспечено сотрудниками библиотек и сотрудником компьютерного отдела:

1) Своевременным ведение БД;

2) Организацией бесперебойного питания серверного и коммуникационного оборудования;

3) Использованием лицензионного программного обеспечения.

**4.2.2 Время восстановление после отказа**

Время восстановления после отказа, вызванного сбоем электропитания технических средств ОС или файловой системы, не должно превышать 15 минут при соблюдении условий эксплуатации технических и программных средств и правильной настройки операционной системы.

**4.2.3 Отказы из-за некорректных действий оператора**

Возможными считаются отказы БД вследствие некорректных действий персонала, обслуживающего СУБД, операционную систему, под управлением которой работает База данных. Меры безопасности по недопущению некорректных действий персонала должны определяться руководством оператора.

**4.3 Условия эксплуатации**

**4.3.1 Климатические условия эксплуатации**

Требования не предъявляются

**4.3.2 Требования к видам обслуживания**

Обслуживание БД включает в себя:

1) информационное обслуживание - ввод и редактирование информации БД;

2) системное администрирование БД «Сеть библиотек»

**4.3.3 Требование к численности персонала**

Минимальное количество персонала, требуемое для нормального функционирования, должно составлять не менее 1 человека.

**4.4 Требования к параметру и составу технических средств**

Минимальные аппаратные требования:

· Процессор Intel совместимый, тактовая частота не ниже 500 MHz;

· Объем свободной оперативной памяти - не менее 512 Мб;

· Не менее 1 ГБ свободного дискового пространства;

· Клавиатура;

· Мышь;

· Монитор с минимальным разрешением - 800х600 пикселей;

· Принтер.

**4.5 Требования к информационной и программной совместимости**

**4.5.1 Требования к информационным структурам и методам решения**

Проектирование структуры БД должно быть выполнено в рамках разработки технического проекта.

**4.5.2 Требования к программным средствам, используемым программой**

Система должна работать под управлением Unix-подобных ОС.

**4.6 Требования к защите информации и программ**

Доступ к информации БД на чтение или редактирование предоставляется только авторизованным пользователям по логину и паролю с соответствующими правами доступа. Для авторизации необходимо правильно ввести логин и пароль. Права на соответствующий доступ определяются системным администратором в соответствии с должностными инструкциями.

**4.7 Требования к упаковке и маркировке**

Особые требования не предъявляются.

**4.8 Специальные требования**

**4.8.1 Требования к пользовательскому интерфейсу**

Интерфейс должен быть понятным для пользователя.

**4.8.2 Требования к архивированию и резервному копированию данных**

Требования к архивированию и резервному копированию данных не предоставляются.

# Концептуальное проектирование

## Краткое описание предметной области

Библиотека представляет собой хранилище книг. Основные ее функции - выдавать читателям книги и принимать их. Остальные функции относятся к обслуживанию процесса выдачи и приема. Экземпляр каждой книги, хранящейся в библиотеке, должен иметь индивидуальный номер. Каждый читатель должен иметь свой абонемент, который хранит как историю заказов, так и информацию о самом читателе. Кроме этого, имеет значение учет должников.

Книги выдаются читателям на руки, на определенный срок. Индивидуальный номер книги, дата выдачи и дата возврата записываются в читательский билет. В пределах указанного срока читатель должен вернуть книгу в библиотеку. При необходимости срок использования книги может быть продлен читателем. Если читатель не вернул книгу в срок и не сообщил о продлении, то у него возникает задолженность. Наличие задолженностей влияет на дальнейшие отношения библиотеки с читателем.

Каждый читатель имеет индивидуальный номер и читательский билет. По индивидуальному номеру работник библиотеки может определить ФИО читателя, дату рождения и его контактные данные.

Придя в библиотеку, читатель, делает заказ на книгу. Сотрудник библиотеки заполняет бланк заказа с указанием всей необходимой информации о читателе, и о книгах, которые он собирается взять. Далее начинается процесс приёмки заказа – сотрудник библиотеки проверяет наличие ассортимента и по возможности выдает необходимые книги. Информация о выданной книге заносится библиотекарем в читательский билет, где указывается дата выдачи книги и предполагаемая дата возврата. Это позволяет регулировать возврат книг и определять должников. В случае невозврата книги в указанные сроки, сотрудник библиотеки может, воспользовавшись абонементом читателя, определить его личные данные и обратится к нему с помощью почтового уведомления. В случае утери читатель обязуется выплатить полную стоимость экземпляра или же возместить ущерб покупкой такой же самой книги для библиотеки.

## Процесс построения инфологической модели.

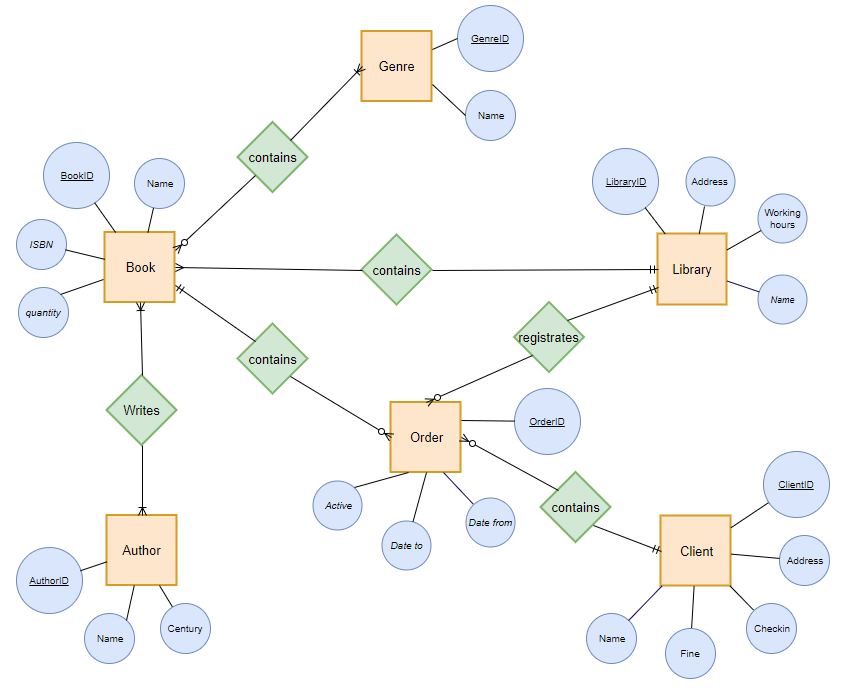
Описывая процесс построения стержневых сущностей стоит упомянуть:

1. Библиотека (Код\_библиотеки, Название, Адрес, Часы\_работы) - эта сущность отводится для хранения информации о конкретном филиале сети библиотек. Утверждается, что название и адрес библиотеки являются уникальными, обеспечивая тем самым безошибочный и однозначный поиск по названию и адресу библиотеки. Для простоты кодирования передаваемой информации о библиотеки, используется целочисленный атрибут «Код\_библиотеки».
2. Книга (Код\_книги, Название, ISBN\_книги, Количество) – эта сущность отводится для хранения всех книг, содержащихся в филиалах библиотек. Естественно название поле с названием книги в этой таблице не может быть уникальным, именно поэтому книге необходимо иметь свой собственный внутренний библиотечный номер. Для этого было предложено ввести некий целочисленный атрибут «Код\_книги», автоматически инкрементируемый при внесении новой книги в базу данных объединенной сети библиотек.   
   По аналогии предлагается создавать также все подобные атрибуты других сущностей, такие как: «Код\_библиотеки», «Код\_жанра», «Код\_автора», «Код\_заказа», «Код\_клиента».
3. Жанр (Код\_жанра, Жанр) – выделение этой сущности позволяет реализовать одну из важнейших функций библиотеки - подбор книги для клиентов исходя из их личностных предпочтений в виде конкретных жанров.
4. Автор (Код\_автора, Автор) – выделение этой сущности позволяет реализовать одну из важнейших функций библиотеки - подбор книги для клиентов исходя из их личностных предпочтений в виде конкретного автора.
5. Заказ (Код\_заказа, Актуальность, Дата\_выдачи, Дата\_завершения) – эта сущность являются ключевой в нашей системе. Она является логическим соединение клиента с получаемой книгой, которая в свою очередь имеет свой жанр, автора, и находится в конкретной библиотеке. Из-за этого сущность имеет огромное количество зависимостей первого уровня, и ещё большее – второго.
6. Клиент (Код\_клиента, Имя, Дата\_регистрации, Домашний\_адрес, Штраф) – сущность отводится для хранения информации о клиентах, зарегистрировавшихся в базе библиотеки, для дальнейшей возможности пользования её книжными ресурсами. Уникальность имени гарантирует нам безошибочное изменение персональных данных клиентов при условии поиска клиента по его имени.

Все вышеописанные сущности подвязаны между собой следующими связями:

1. Клиент заказал - [Клиент 1, Заказ N] (Код\_клиента, Код\_заказа)
2. Место исполнения (заказа) – [Заказ 1, Библиотека 1] (Код\_заказа, Код\_библиотеки)
3. Книга заказана – [Заказ 1, Книг N] (Код\_заказа, Код\_книги)
4. Находится в библиотеке (книга) – [Книг N, Библиотека M] (Код\_книги, Код\_библиотеки)
5. Авторство [Автор N, Книга M] (Код\_автора, Код\_книги)
6. Жанр – [Жанр N, Книга M] (Код\_жанра, Код\_книги)

## ER-Диаграмма с комментариями.



Исходя из ключевых особенностей и обоснования связей стоить отметить:

1. каждая книга должна иметь от нуля до многих жанров; каждый жанр должен содержать от нуля до многих книг
2. каждая книга должна иметь от одного до многих авторов; каждый автор должен содержать от одной до многих книг
3. каждая книга должна принадлежать ровно одной библиотеке; каждая библиотека должна содержать многие книги
4. каждый заказ должен содержать ровно одну книгу; каждая книга может содержаться от нуля до нескольких заказах
5. каждый заказ должен содержать ровно одну библиотеку; каждая библиотека может содержаться в одном или нескольких заказах
6. каждый заказ должен содержать ровно одного клиента; каждый клиент может содержаться в одном или нескольких заказах

# Проектирование реляционной модели:

## Процесс перехода к реляционной модели.

Инфологическая модель позволяет понять суть разрабатываемой базы данных, но она не подходит для непосредственной реализации структуры БД. Необходимо преобразование инфологической модели в даталогическую, с учетом особенностей выбранной даталогической модели.

Разработанная структура реляционной модели БД позволяет обойтись без сложных механизмов наследования таблиц. Это позволяет освободить нашу БД от лишней повторяющейся информации, потому как все данные хранятся в отдельных смысловых таблицах, и не нуждаются в дублировании информации создавая новую таблицу на основе уже имеющейся. Достаточно связать две разные сущности, получив тем самым то же преимущество, но за меньшие затраты по памяти.

Среди всех равных следует выделить сложные виды связей «от многих ко многим»:

1. Так связаны, например, сущности Автор и Книга. Это вполне естественно, ведь, один автор за свою жизнь пишет много книг, и наоборот, одна книга может иметь несколько соавторов.
2. Аналогичную связь имеют сущности Жанр и Книга. Действительно, ведь одну книгу бывает сложно классифицировать в какой-то один жанр по её сюжету. И, конечно же, книг с одним жанром за всю историю литературы было написано несчитанное количество.

## Диаграмма реляционной модели (схемы БД)

## Дополнительные механизмы обеспечения целостности данных.

Дополнительных механизмов обеспечения целостности данных (помимо использования уникальных и обязательных полей) не используется.

# Развёртывание БД в выбранной СУБД

В качестве архитектурных решений выбор был остановлен на свободной объектно-реляционной СУБД PostgreSQL. Для управления базой данных при помощи языка Python используется пакет psycopg2, написанный на языке С.

## DDL-скрипт создания схемы БД.

Предполагается что на тестируемой машине уже установлен [PostgreSQL 11.1](https://www.postgresql.org/) и [Python 3.7.0](https://www.python.org/downloads/release/python-370/), с заранее предустановленным пакетом [psycopg2](http://www.psycopg.org/psycopg/download/).

В качестве SQL-клиента и, по совместительству, системы администрирования базой данных используется [DBeaver – Universal Database Client](https://dbeaver.com/download/).

Алгоритм запуска создания БД:

1. Инициализировать базу данных при помощи программного обеспечения PostgreSQL.
2. Подключиться к базе при помощи SQL-клиента DBeaver.
3. Развернуть скелет базы данных, запустив [script.sql](https://www.dropbox.com/s/r3f2y6rtc2u8rwi/Script.sql?dl=0) внутри инициализированной базы.
4. Скачать [открытые данные](https://www.kaggle.com/applecrazy/cmu-book-summary-dataset) и поместить их в одну директорию с файлом [parser.py](https://www.dropbox.com/s/rc4xq8fsfu29zbx/parser.py?dl=0).
5. Запустить файл parser.py, начав процесс наполнения базы данными. ( ̴̴1.5 минуты)
6. После завершения работы скрипта, в базу данных будет добавлена информация по книгам и библиотекам, в которых они располагаются. На данном этапе БД готова к дальнейшему наполнению, и в то же время имеет необходимый минимум для начала своего функционирования.

## Примеры DML-операторов вставки текстовых данных.

Манипулировать базой данных на этом этапе можно при помощи интерфейса CLI. Для этого необходимо загрузить файл [db.py](https://www.dropbox.com/s/wiyer5w43lxqldt/db.py?dl=0), запуская который можно при помощи, написанных на языке Python, функций-декораторов над DML-операторами, записывать и редактировать данные в БД.

Ниже представлены несколько примеров использований таких функций по шаблону:



1. Добавление нового клиента:  
   
2. Добавление нового заказа:  
   
3. Добавление новой книги:  
   

## Контроль работы ограничений целостности с примерами.

Идеологически таблица должна хранить всю историю заказов, а значит, удалять из базы ничего нельзя. Однако, в функциях добавления, описанных выше – ограничения всё же присутствуют: при добавлении нового пользователя с именем, уже имеющемся в базе – внесения нового не произойдёт. Аналогично с новыми книгами, их жанрами, и авторами. Стоит также заметить, что автор и жанр могут существовать только в связке с книгой, и не могут быть добавлены отдельно от неё.

# Разработка клиентского приложения

## Бизнес-логика тестового приложения.

Библиотеки, для которых разрабатывалось служебное ПО, являются государственными учреждениями, а значит не нуждаются ни в каком дополнительном финансировании извне.

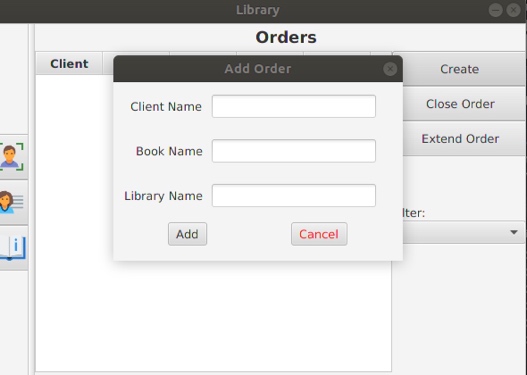
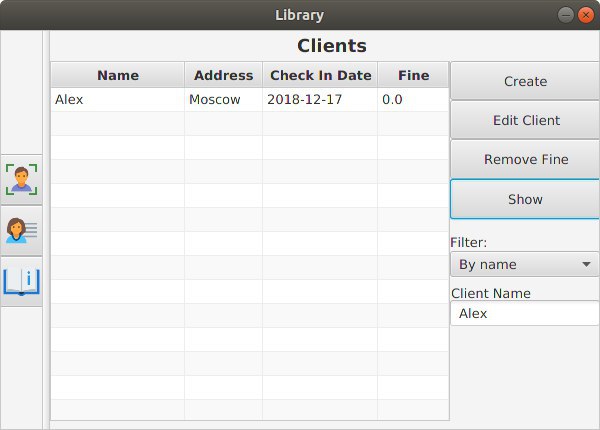
Тем не менее система управления базой данных позволяет ускорить время обработки запросов клиентов, а это находится в зоне интересов заказчика, потому как увеличивается проходимость, в свою очередь уменьшая время работы, тратившееся на одного клиента.

Клиентское приложение позволяет сотрудникам библиотеки избавиться от некоторого количества бумажной работы, таких как: ведение читательского билета во время процедур оформления и получения заказов, внесения данных в читательский билет во время процедуры регистрации нового клиента, и контроля должников путём автоматического пересчёта штрафов.

## Техническое описание тестового приложения.

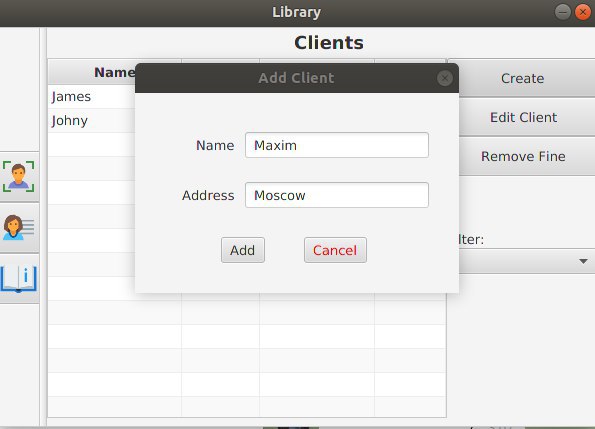
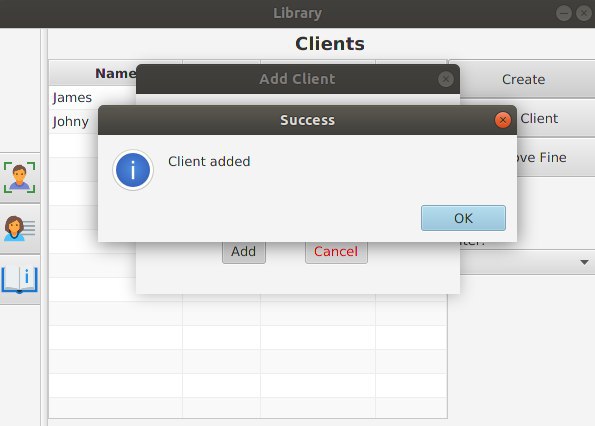
Клиентское приложение «Сеть библиотек» представляет собой простое в применение приложение, работающее на основе Unix-подобных систем, которое имеет возможность ведения отчётной деятельности и контроля над всеми процессами среды библиотеки.

Программное обеспечение имеет удобный интерфейс управления, включая выбор категории и множества кнопок для каждой из категорий, в зависимости от текущей задачи сотрудника, и вопроса клиента.

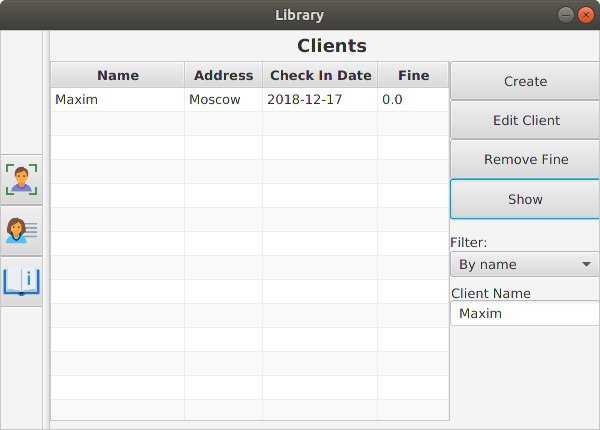
 

Управление всем функционалом происходит непосредственно в окнах приложения, при помощи кнопок бокового правого меню, после выбора соответствующей категории в боковом левом меню.

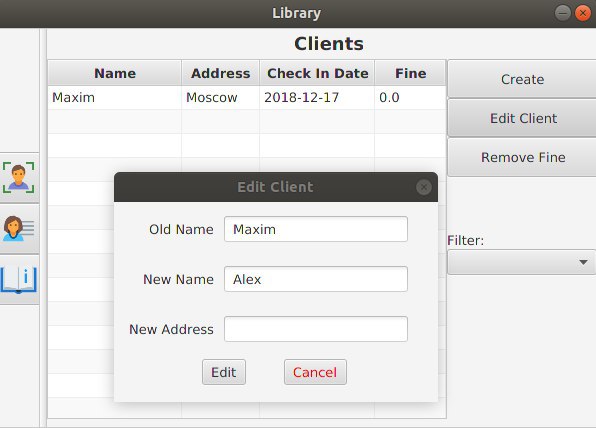
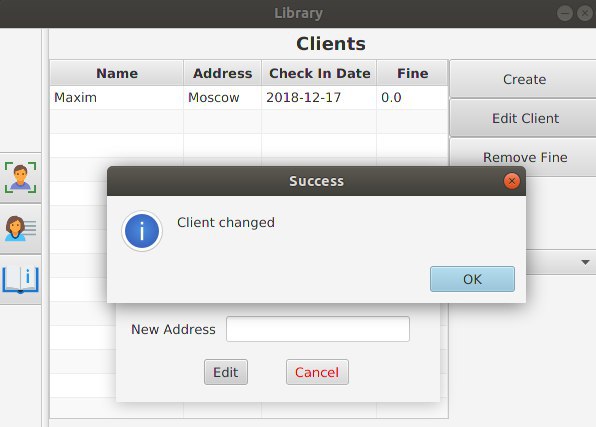
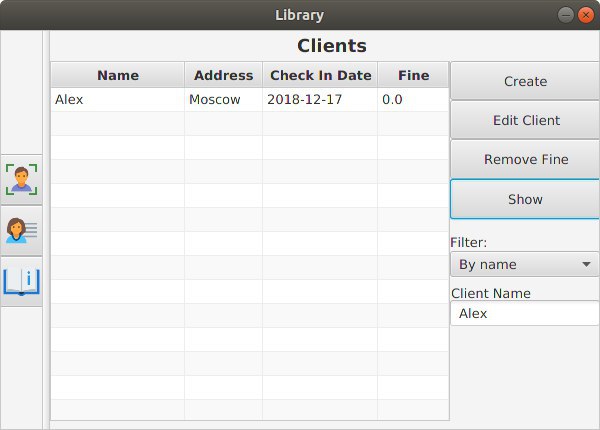
Например, добавление клиента осуществляется при выборе соответствующего раздела работы с клиентом и нажатии кнопки Create следующим образом:

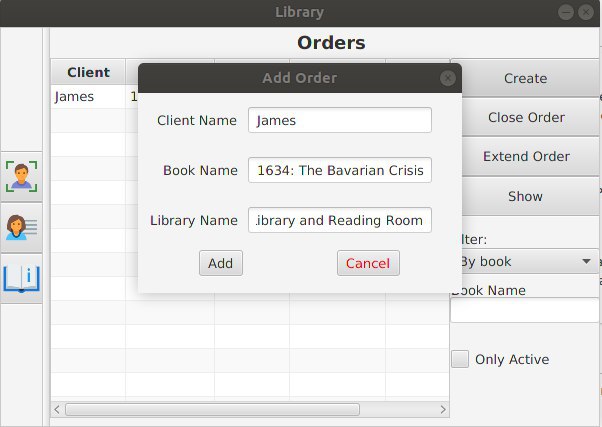
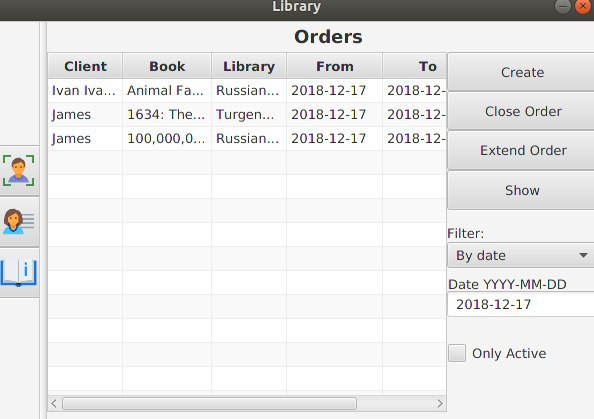
При этом происходит инициализация и добавление соответствующей записи в БД, с которой можно ознакомиться, выбрав в правом меню кнопку Show.



Редактирование уже существующих клиентов осуществляется при помощи нажатия кнопки Edit client в правом меню, и заполнения предложенных полей. При этом, поиск клиента осуществляется по его имени.

Также была реализована система работы с заказами, а именно при размещении заказа на имя James, по книге «1634: The Bavarian Crisis» происходят следующие операции в приложении:

Далее заказ размещается в системе, книга резервируется в библиотеке, и клиент ожидает её получения.

## Результаты функционального тестирования.

Тестирование ПО не выявило серьёзных недостатков. Клиентское приложение работает корректно и способно к решению поставленных задач, описанных в п.2 «Техническое задание». Функциональное тестирование состояло из следующих частей:

1. Функциональная пригодность приложения.
2. Точность работы.
3. Надёжность и достоверность хранимых данных.
4. Соответствие стандартам и правилам.

# Заключение.

В результате выполнения командного проекта, нами была разработана программная система управления базой данных "Сеть Библиотека". Эта программа упрощает работу сотрудникам, переводя все их обязанности из бумажных в электронные. Пропадает необходимость также в лишнем обременении читателя иметь при себе читательский билет, и лишней траты времени для сотрудника на его заведение и правильное ведение. В разработанной нами программе доступны функции добавления и редактирования информации как по посетителям, так и по обороту книг среди всей сети библиотек.

Преимущества автоматизированной системы в том, что она будет экономить время ведения документации, при должном уровне настройки может вестись из каждого филиала библиотек, упростит бумажную волокиту, и увеличит надежность и отказоустойчивость всей системы работы библиотеки. Электронная система также обеспечивает глубокую достоверность внесенных в неё данных (исключая человеческий фактор), и предусматривает возможность хранения истории заказов всего времени членства клиента в библиотеке. Также была предусмотрена безошибочная система ведения штрафов за просрочки и возможность продления активного заказа. Текущий функционал может быть легко усовершенствован под новые нужды.