**Министерство образования и науки Российской Федерации**

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

**ДАЛЬНЕВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

**(ДВФУ)**

|  |
| --- |
| **ШКОЛА ЕСТЕСТВЕННЫХ НАУК**  **Кафедра информатики, математического и компьютерного моделирования** |

|  |
| --- |
| **ПРОЕКТНАЯ ДОКУМЕНТАЦИЯ**  По дисциплине «Языки и методы программирования»  Специальность 01.03.02 «Прикладная математика и информатика»  Курсовой проект «**Система управления библиографическими базами данных**» |

|  |  |
| --- | --- |
|  | Выполнили студенты группы Б8203Б:  Горячкин Данила Михайлович\_\_\_\_\_\_\_\_\_  Шулепкин Виктор Александрович\_\_\_\_\_\_  Василенко Иван Максимович\_\_\_\_\_\_\_\_\_\_  Проверил преподаватель:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Сущенко  (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (зачтено/не зачтено)  «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |

г. Владивосток

2018

Оглавление

[Оглавление 2](#_Toc534311350)

[Глава 1. Анализ предметной области 4](#_Toc534311351)

[1.1 Введение 4](#_Toc534311352)

[1.2 Сравнительная характеристика 7](#_Toc534311353)

[1.3 Заключение 9](#_Toc534311354)

[Глава 2. Анализ требований к системе 10](#_Toc534311355)

[2.1 Введение 10](#_Toc534311356)

[2.2 Глоссарий 10](#_Toc534311357)

[2.3 Функциональные требования к системе для управления библиографическими базами данных 11](#_Toc534311358)

[2.4 Требования к реализации 12](#_Toc534311359)

[2.5 Аппаратные требования 12](#_Toc534311360)

[2.6 Заключение 12](#_Toc534311361)

[Глава 3. Проектирование программного средства 13](#_Toc534311362)

[3.1 Введение. 13](#_Toc534311363)

[3.2 Диаграмма прецедентов. 14](#_Toc534311364)

[3.3 UserFlow диаграмма. 17](#_Toc534311365)

[3.4 Проект пользовательского интерфейса. 17](#_Toc534311366)

[3.5 Диаграмма потоковых данных. 24](#_Toc534311367)

[Глава 4. Тестирование. 25](#_Toc534311368)

[5.1 Введение 25](#_Toc534311369)

[5.2. Описание пользовательских функций 25](#_Toc534311370)

[5.3 Функциональное тестирование 27](#_Toc534311371)

[Глава 5. Архитектурный проект. 31](#_Toc534311372)

[5.1 Введение 31](#_Toc534311373)

[5.2 Описание классов. 31](#_Toc534311374)

[Ссылка на диаграмму классов: 33](#_Toc534311375)

[5.3 Вывод. 35](#_Toc534311376)

[Глава 6. Проектирование БД 36](#_Toc534311377)

[6.1 Введение 36](#_Toc534311378)

[6.2 ER-диаграмма, демонстрирующая структуру основной БД 36](#_Toc534311379)

[6.3 Таблицы БД. 36](#_Toc534311380)

[6.4 Выбор СУБД 59](#_Toc534311381)

[6.5 Заключение 59](#_Toc534311382)

[Приложения. 60](#_Toc534311383)

[Приложение 1. UserFlow диаграмма 60](#_Toc534311384)

[Приложение 2. ER-диаграмма, демонстрирующая структуру основной БД 61](#_Toc534311385)

# Глава 1. Анализ предметной области

## 1.1 Введение

Система управления библиографической информацией — электронная система, в которой размещают, хранят, обменивают данные и информацию, а также позволяют исследователям, ученым и писателям создавать и повторно использовать библиографические ссылки. Библиографические базы данных заменили традиционные печатные каталоги и являются общими или специальными, которые пытаются собрать данные по всем статьям по выбранной дисциплине или группе дисциплин.

Библиографическая информация — это информация об источниках информации, что позволяет их эффективно обнаруживать и однозначно идентифицировать, а также служит их заменителем в информационных процессах и выполняет роль посредника между потребителем и информационными ресурсами. Библиографическая информация существует как совокупность библиографических записей, каждый из которых является отдельным информационным ресурсом. Библиографическая запись, в свою очередь, состоит из элементов, каждый из которых раскрывает один из признаков информационного ресурса.

Специалисты, которые отвечают в таких областях как стратегическое планирование, учёт и архивы систем, бизнес-анализ, библиотеки и хранилища данных, выразили стратегию управления информацией. Как правило, они пытаются сформулировать подходы к решению организации и распространения информации с акцентом на качество, определение потребностей пользователей информации или для анализа окружающей среды, которая ценит информацию, знания и ресурсы и строит их в корпоративное планирование. Эффективным инструментом управления развитием библиографических ресурсов была и остается информация. Персональные средства управления библиографической информацией позволяют пользователю вести свою базу данных и использовать её для ведения ссылочного аппарата печатной или учебной работы.

В системах управления библиографической информацией создаются значительно меньшие базы публикаций, чем в библиографических базах данных, и используются одним человеком или группой людей. Данные системы легко устанавливаются на отдельном персональном компьютере или мобильном устройстве.

Программное обеспечение обычно включает базу данных и систему генерации отобранных ссылок в форматах, необходимых научными журналами. Современные системы управления библиографической информацией могут быть интегрированы с текстовыми процессорами таким образом, что список ссылок (в выбранном формате) автоматически создается и добавляется к документу. Это избавляет от неприятности забыть включить ссылку на источник, процитированный в тексте.

Также системы имеют возможность импортировать (экспортировать) детали библиографических ссылок из библиографических баз данных. С развитием Web 2.0 появились онлайновые системы управления библиографической информацией. Возможность доступа с любого компьютера, коллективная работа над одной темой — особенности таких систем.

Разработка систем управления библиографической информацией объясняется стремительным ростом большого количества научной, учебной и профессиональной литературы. Известно, что «основным фактором, определяющим слаженную коллективную работу ученых, является организация передачи информации».

Составление библиографии и вставка ссылок вручную — очень кропотливая работа. Именно для облегчения библиографической работы и вставки ссылок вручную давно существует множество специальных программ — менеджеров библиографической информации.

Библиографической менеджер — это программное обеспечение, которое поможет вам собирать, сохранять и обрабатывать литературные ссылки. Программное обеспечение дает возможность просматривать и сортировать собраны ссылки на различные записи (автор, дата публикации, ключевые слова и тому подобное), а также создавать библиографию в различных стилях и добавлять те или отдельные названия на них.

Таким образом, библиографические менеджеры предоставляют следующие возможности:

* ведение локальной базы библиографических данных;
* поиск библиографической информации в электронных каталогах непосредственно из программы;
* импорт найденных библиографических записей в свою картотеку для дальнейшего использования;
* возможность работать с полнотекстовыми ресурсами, в том числе с ресурсами Интернета;
* возможность генерировать систему библиографических ссылок в разных форматах;
* опровержение шаблонов оформления.

## 1.2 Сравнительная характеристика

На сегодняшний день признанным лидером в области менеджеров библиографии является EndNote, что является стандартом. Основными возможностями данной программной системы является:

* создание своей собственной поисковой базы данных ссылок на статьи, книги, диссертации и тому подобное;
* организация PDF-статьи и легкое их получение с функцией поиска в EndNote;
* легко вставлять цитаты в статью в процессе написания;
* возможность форматирования цитат и библиографии в различных стилях;
* вставка файлов или чисел в ссылке;
* импорт фильтров;
* экспорт ссылок в базе данных непосредственно в EndNote;
* поиск ссылки.

Традиционно EndNote считается наиболее эффективным при поиске в удаленных библиографических базах. Ссылки сохраняются в файлах с названиями библиотек. Каждая ссылка в библиотеке имеет уникальный идентификационный номер. Каждая ссылка может содержать до 45 файлов в поле Вложения файла, но только один графический файл или приложение на рисунке поле.

По EndNote, то при запуске макроса во вставленных ссылках генерируются библиографические ссылки в соответствии с выбранным стилем и автоматически формируется список литературы в конце текста. Список может быть выведен или по мере упоминания в тексте, либо по алфавиту. Bibus может импортировать ссылки из восьми разных форматов (Включая EndNote и BibTeX) и экспортировать в пяти форматах. Существует возможность подключения фильтров для других форматов, что позволяет легко переносить существующие базы из других программ.

Bibus и EndNote — хороший выбор для тех, кому приходится подгонять форматирование библиографии под ГОСТ. Прекрасный редактор стилей позволяет учесть все «фантазии» отечественных чиновников и при необходимости быстро подстроиться под внезапно изменившиеся требования. В то же время Bibus недостаточно «социальный»: не умеет синхронизировать базы литературы из сети и не может импортировать ссылки. Недостатком Bibus является то, что его стиле несовместимы со стилями других библиографических программ с открытым стандартом стилей библиографии.

Mendeley и Zotero имеют очень схожий вид и отличаются только деталями реализации, то есть то, что реализовано хорошо в одной системе, выглядит несколько хуже в другой и наоборот. Обе системы могут свободно обмениваться информацией, экспортируя её в универсальные форматы RIS или BibTeX, а также используя автоматический экспорт из локальной базы Zotero. Поскольку функциональность у обеих систем практически идентичны, то проще проанализировать их недостатки.

Сравнивая два менеджера библиографической информации, определено, что претензии к Zotero носят в основном "технологический" характер. Единственным существенным недостатком в функциональности откровенно странный диалог вставки цитат в документы. По Mendeley хорошая техническая установка противоречит именно функциональным погрешностям: отсутствие вложенных коллекций. В Zotero больше внимания уделяется именно вставке цитат в документы и локальном клиенту, а социальная сеть существует как удобное дополнение. Mendeley Desktop незаменим для «разбора завалов» PDF-документов, тогда как Zotero не умеет импортировать и проверять их автоматически. Mendeley не умеет импортировать цитаты из RTF-файлов. Общим недостатком является отсутствие визуального редактора стилей. Относительно социальных сетей Zotero и Mendeley, то они практически идентичны и отличаются только дизайном и географией своей аудитории.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Возможности** | **Bibus** | **EndNote** | **Zotero** | **Mendeley** |
| Наличие хранилищ для файлов | — | — | + | — |
| Экспорт ссылок в базе данных | + | + | + | + |
| Деление пользователей на группы | + | + | + | + |
| Жесткая привязка к браузеру | — | — | + | — |
| Сложная инсталляция | — | — | + | — |
| Создание собственной поисковой базы данных ссылок на статьи | + | + | + | + |

Таблица . Сравнение возможностей популярных систем управления библиографической информацией.

## 1.3 Заключение

Современные системы управления библиографическими базами данных позволяют пользователям не только получать и добавлять информацию о библиографических записях, но и работать со ссылками на различные записи.

Изучив сравнительную характеристику наиболее популярных систем, можно сформировать пользовательские требования к будущему программному продукту. Мы будем разрабатывать наше приложение исходя из них и нашего технического задания.

# Глава 2. Анализ требований к системе

## 2.1 Введение

Основываясь на анализе предметной области и задачах, необходимо разработать систему для управления библиографическими базами данных, поддерживающих клиент-серверную архитектуру. Внедрение такого программного средства (ПС) позволит систематизировать данные о публикациях и предоставлять информацию о них пользователям.

## 2.2 Глоссарий

1) **ПС** – программное средство.

2) **Читатель** – пользователь, который может получать библиографическую информацию в режиме чтения.

3) **Автор** – пользователь, который может получать библиографическую информацию в режиме чтения и редактировать информацию о привязанных к нему публикациям.

4) **Издательство** – пользователь, который может получать библиографическую информацию в режиме чтения и редактировать информацию о привязанных к нему публикациям.

5) **Библиотекарь** - пользователь, который может добавлять, удалять, обновлять библиографические данные, перемещать пользователей из группы «читатели» в группы «авторы» или «издательства».

6) **Администратор** - пользователь, который может добавлять, удалять, обновлять библиографические данные, добавлять/удалять пользователей, редактировать их анкетные данные, перемещать пользователей между группами.

## 2.3 Функциональные требования к системе для управления библиографическими базами данных

1. **Авторизация пользователей**

Приложение должно позволять авторизовываться уже зарегистрированным пользователям.

1. **Доступ к библиографической информации**

Приложение должно позволять пользователям получать библиографическую информацию о публикациях.

1. **Редактирование библиографической информации**

Приложение должно позволять отдельным группам пользователей («автор», «издательство») редактирование библиографической информации привязанных к ним публикаций.

1. Изменение группы пользователя

Приложение должно позволять отдельным группам пользователей («библиотекарь», «администратор») изменять группы других пользователей.

1. **Верификация**

Приложение должно позволять пользователям посылать запрос на верификацию их профилей для изменения группы.

1. **Интерфейс**

Приложение должно иметь удобный для пользователя интерфейс, обеспечивающий удобный просмотр информации о публикациях и ее редактирование.

1. **Данные (База данных) на сервере**

Данные на сервере должны быть структурированы. Имеется несколько таблиц, реализующиx следующие сущности:

* Публикации
* Журналы
* Авторы
* Организации

## 2.4 Требования к реализации

Необходимо разработать клиентское приложение на Java с интерфейсом, созданным при помощи JavaFX, и серверное приложение, реализованное с использованием Hibernate. Должна поддерживаться база данных, хранимая на сервере. База данных создается при помощи СУБД PostgreSQL.

## 2.5 Аппаратные требования

Минимальные системные требования:

Операционная система: Windows 7 / 8 / 8.1 /10;

Процессор: AMD Athlon 64 X2 2,4 ГГц или Intel Core 2 Duo 1,8 ГГц; Оперативная память: 2 Гб;

Место на жестком диске: 1 Гбайт.

## 2.6 Заключение

В ходе выявления требований к будущей систему для управления библиографическими базами данных был сформирован ряд четких условий, необходимых для ее реализации. На основании проведенного анализа можно приступать к проектированию программного средства.

# Глава 3. Проектирование программного средства

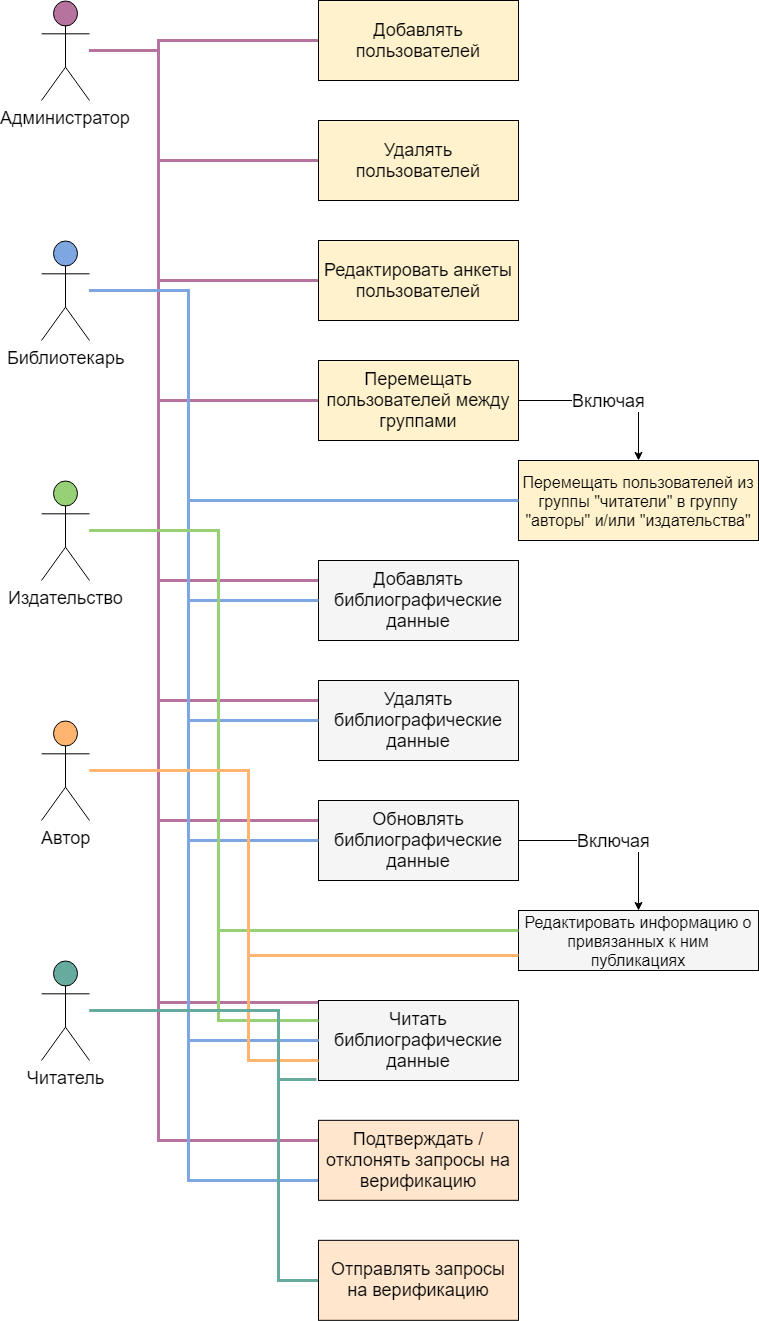
## 3.1 ***Введение***.

Проектирование программного средства – это последний этап разработки ПО, который контролируется заказчиком. В нем утверждаются все подсистемы системы, основные функции и подфункции пользователей, их тестирование. Также прогнозируется результата ответа системы в любой момент работы. Описывается связь объектов системы. Процесс проектирования сопровождается построением разных диаграмм, отражающих пользовательское восприятие системы, и поясняющих разработчику поставленные перед ним задачи.

На основании анализа предметной области и маркетингового исследования необходимо построить:

* Диаграмму прецедентов
* UserFlow диаграмму
* Проект пользовательского интерфейса
* Диаграмму потоков данных

## 3.2 Диаграмма прецедентов.



Модель вариантов использования.

* Администраторы:
  + могут добавлять/удалять и обновлять библиографические данные
  + могут добавлять/удалять пользователей, редактировать их анкетные данные
  + могут перемещать пользователей между группами
  + могут подтверждать/отклонять запросы на верификацию.
* Библиотекари:
  + могут добавлять/удалять и обновлять библиографические данные
  + могут перемещать пользователей из группы “читатели” в группы “авторы” и/или “издательства”
  + могут подтверждать/отклонять запросы на верификацию.
* Издательства:
  + могут получать доступ к любой библиографической информации в режиме чтения;
  + могут редактировать информацию о привязанных к ним публикациях
* Авторы:
  + могут получать доступ к любой библиографической информации в режиме чтения;
  + могут редактировать информацию о привязанных к ним публикациях
* Читатели:
  + могут получать доступ к библиографической информации в режиме чтения
  + отправлять запросы на верификацию.

## 3.3 UserFlow диаграмма.

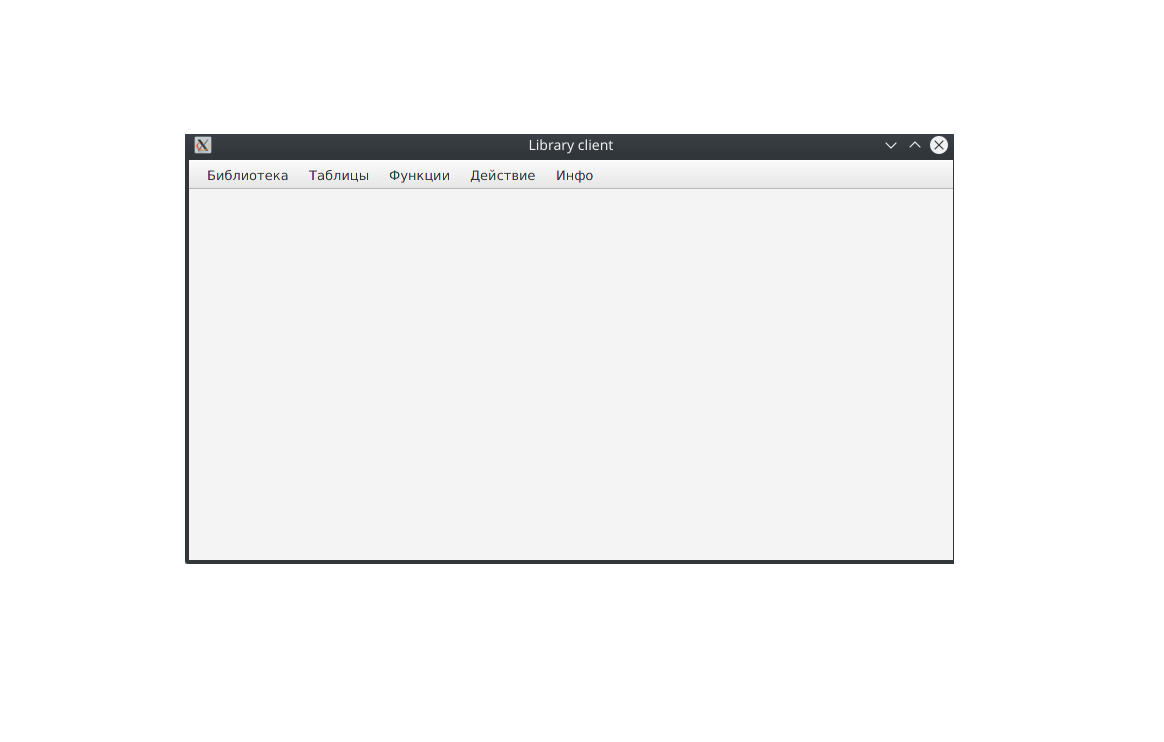
User Flow диаграмма представляет собой путеводитель по графической части программы. По диаграмме видно, что происходит после непосредственного запуска программы и какие возможности открываются перед пользователем в зависимости от его группы. На схеме показано, какие действия доступны пользователю каждой группы. Так, например, пользователи группы «Администраторы» имеют полномочия редактировать все записи, а пользователи группы «Читатели» имею доступ к записям в режиме чтения. Стоит обратить внимание, что все группы ведут к одним и тем же таблицам, то есть все группы имеют возможность как минимум, просматривать их.

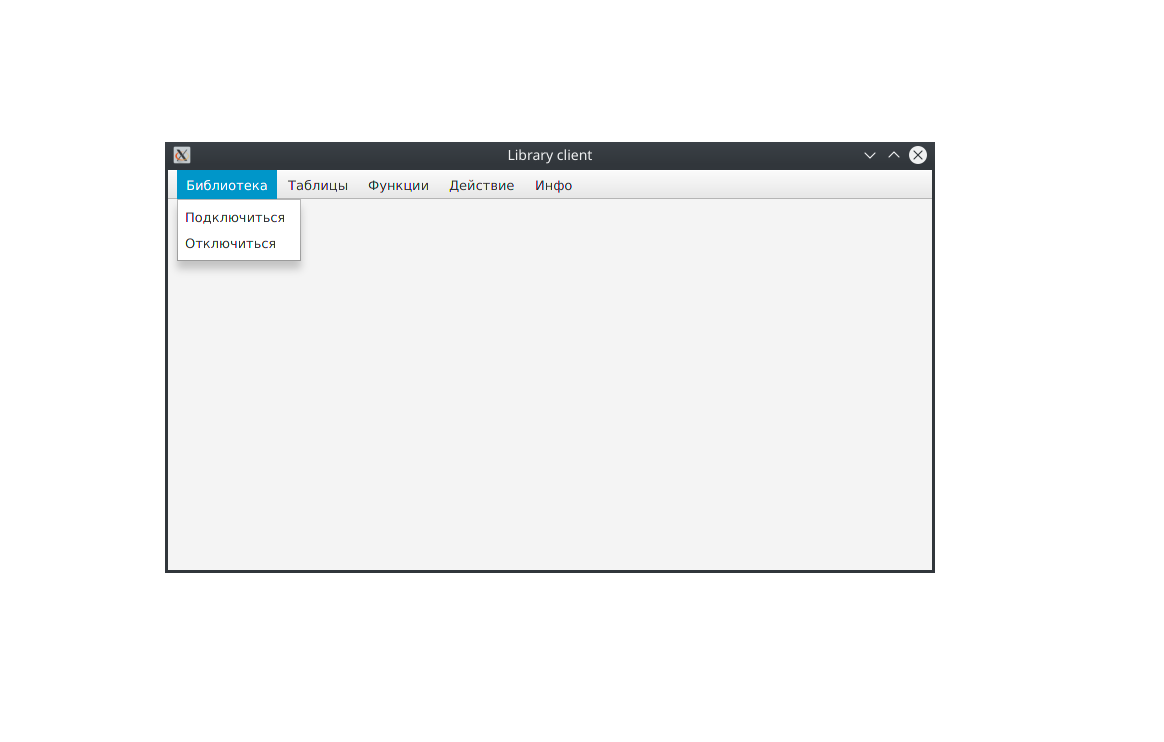
Ознакомится с User Flow диаграммой можно в [приложение 1](#_Приложение_1._UserFlow) или по ссылке.

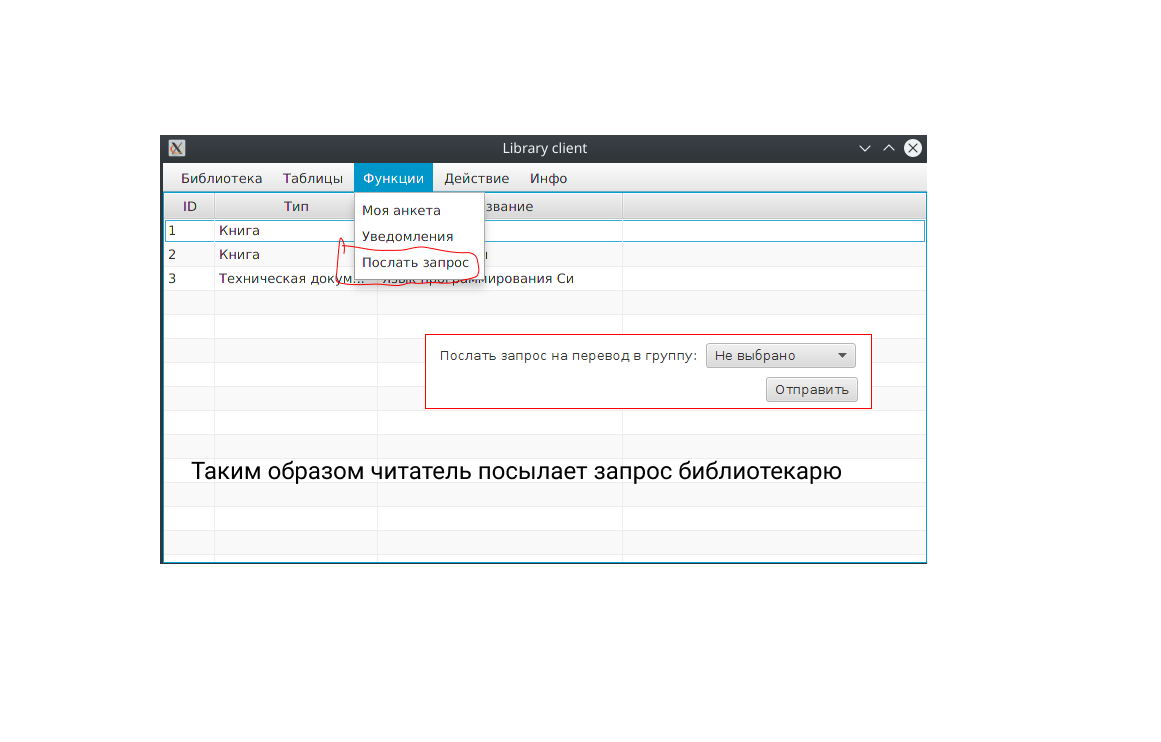
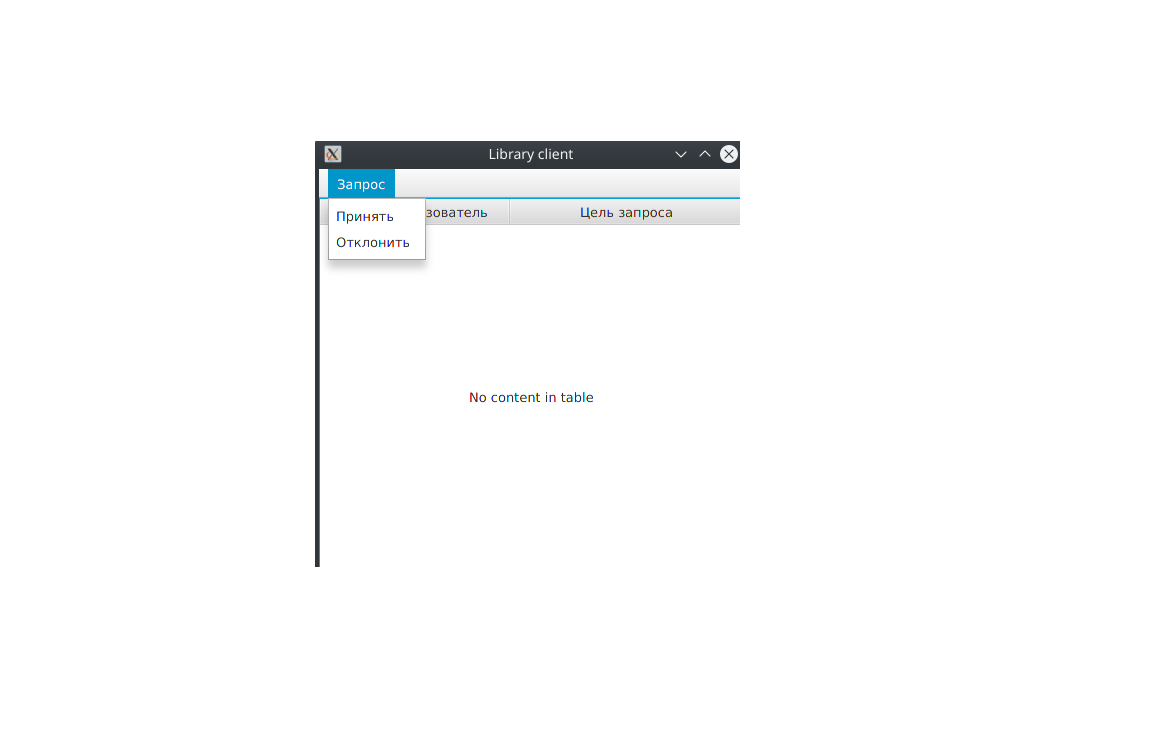
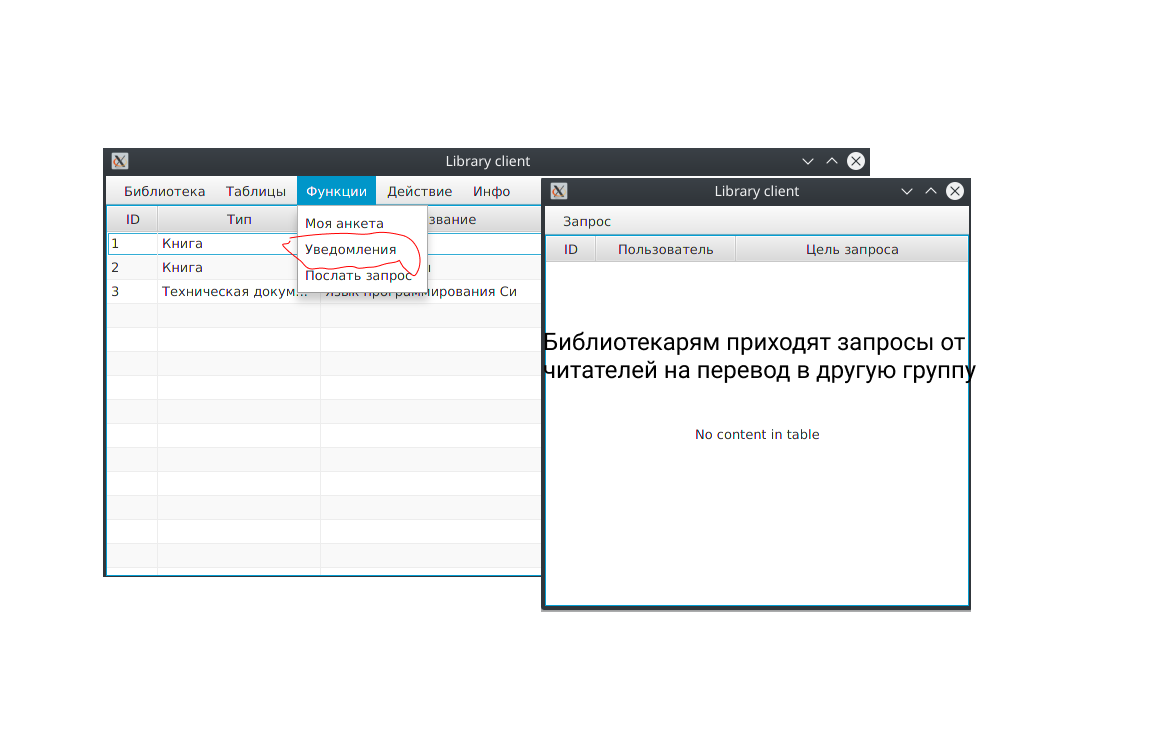
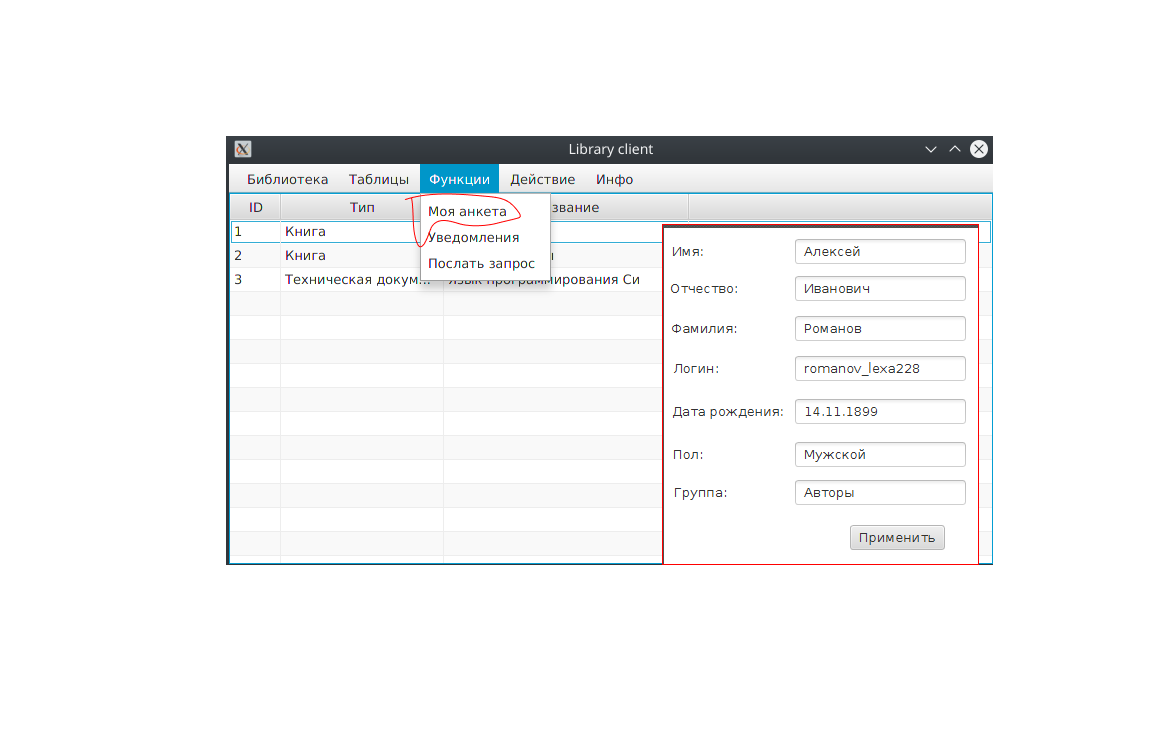
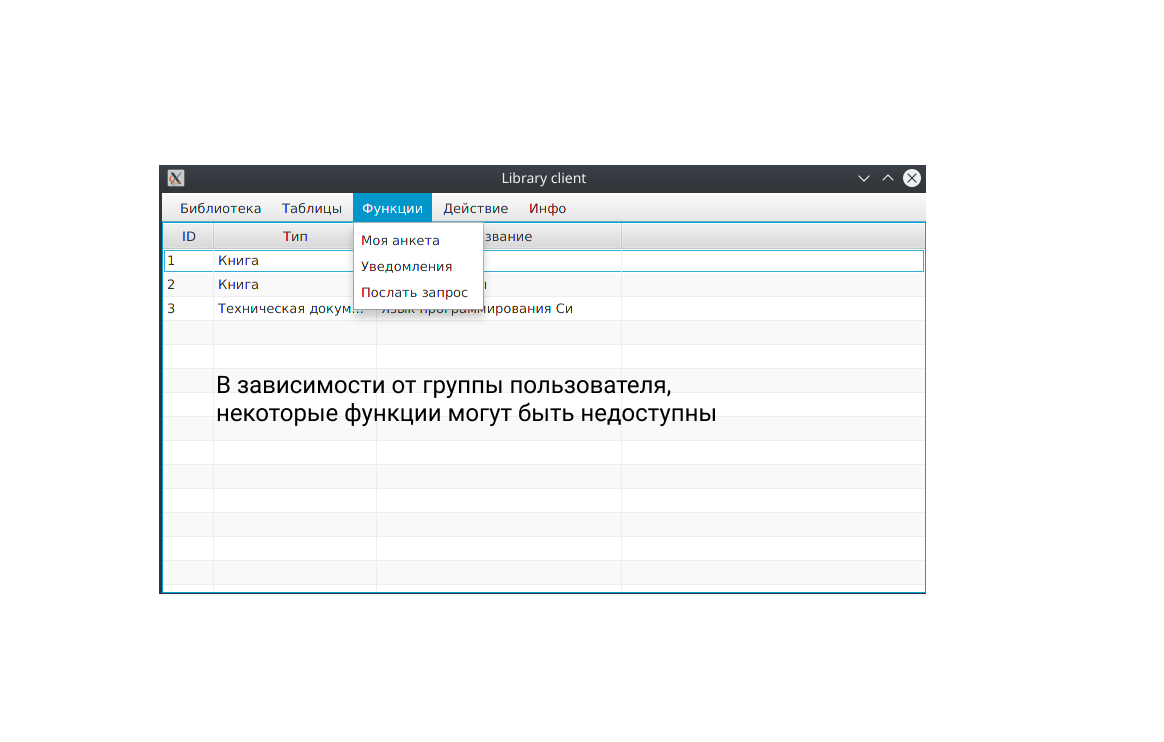
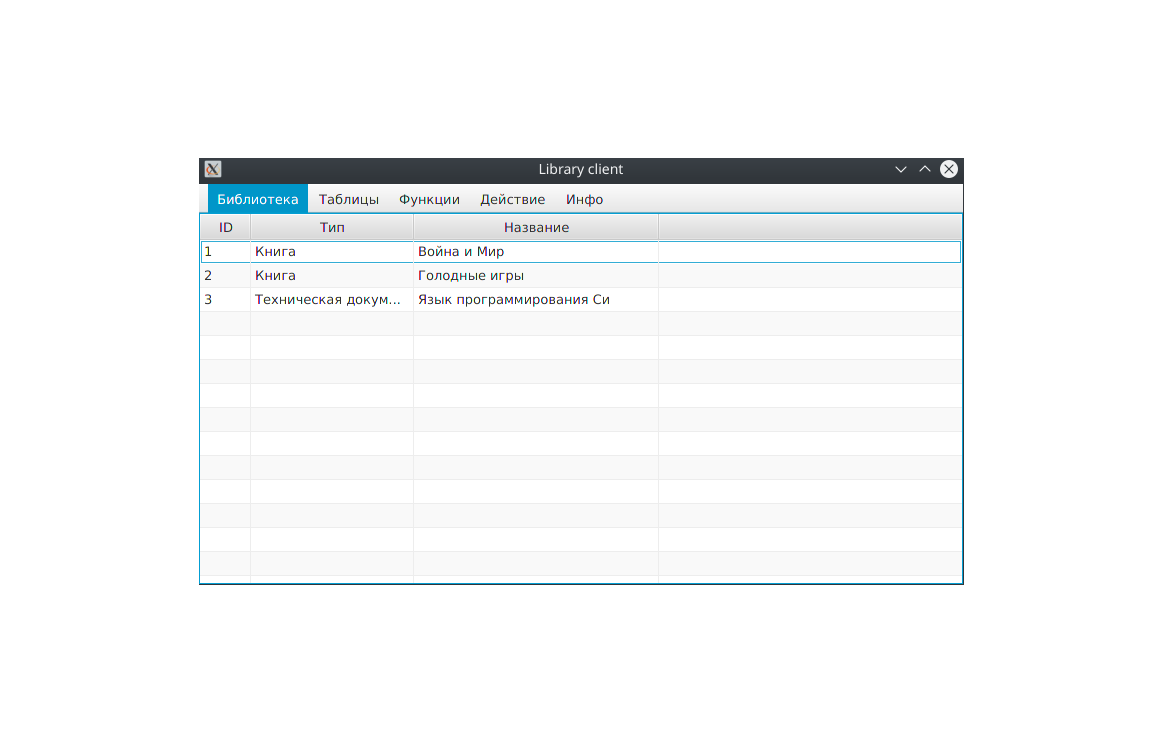
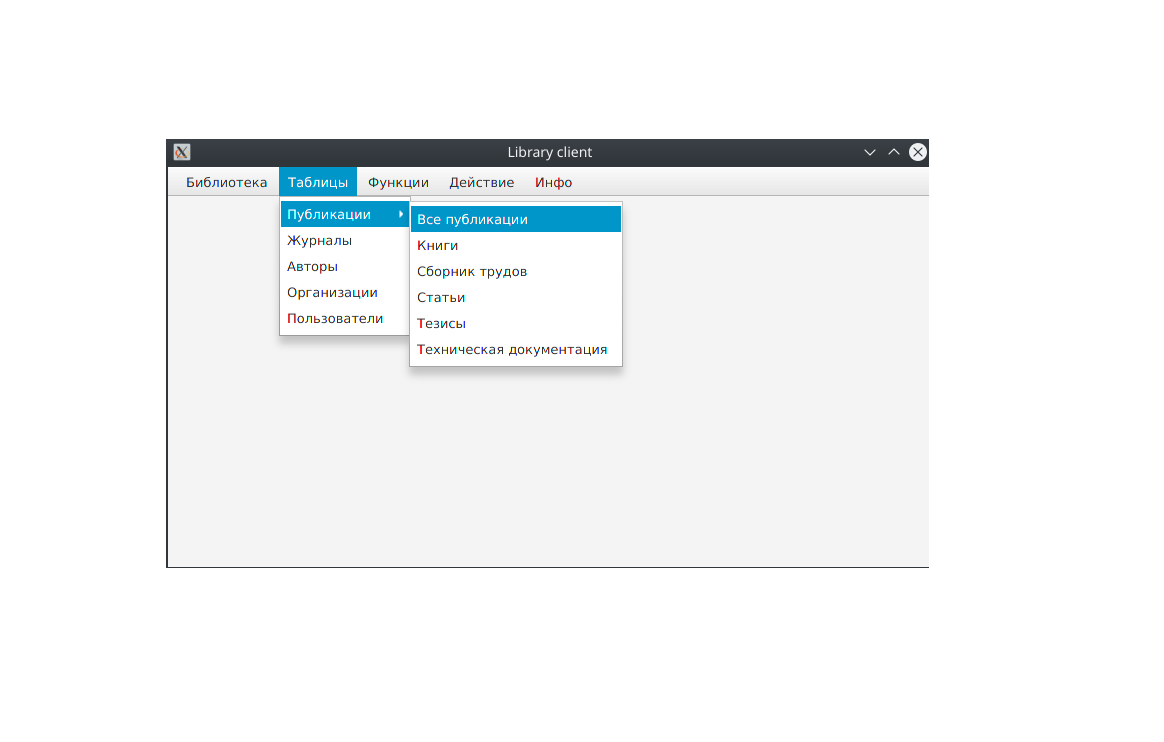
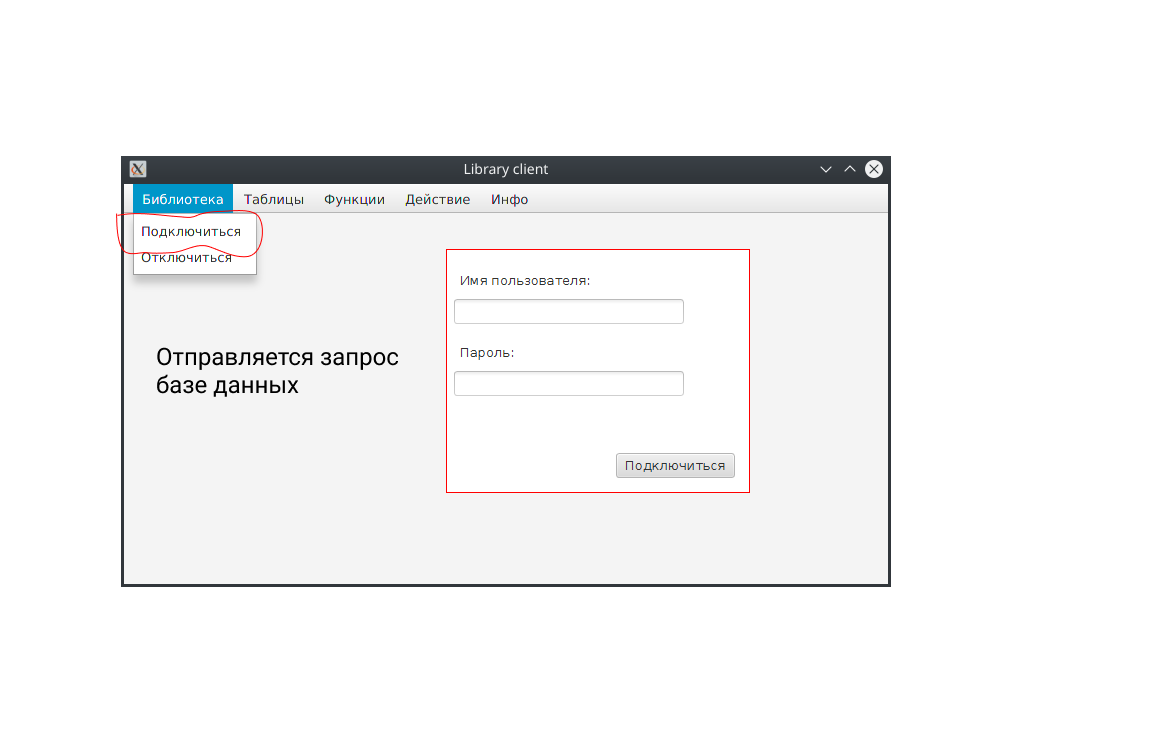


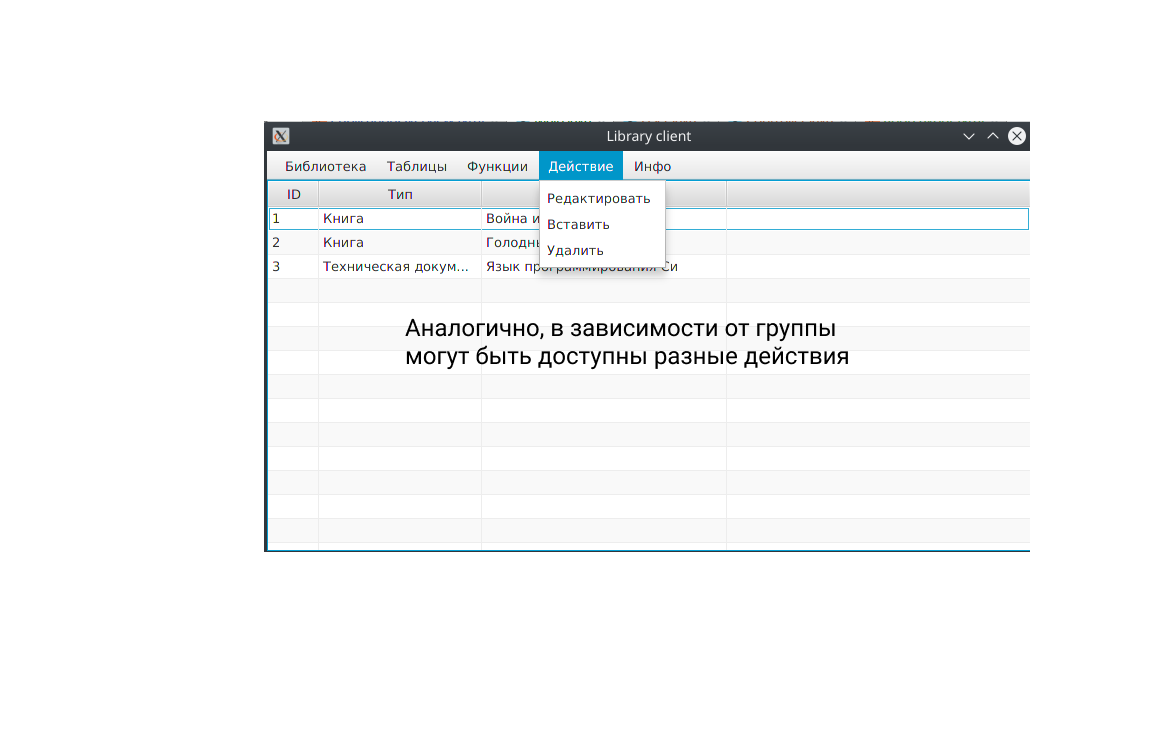
## 3.4 Проект пользовательского интерфейса.

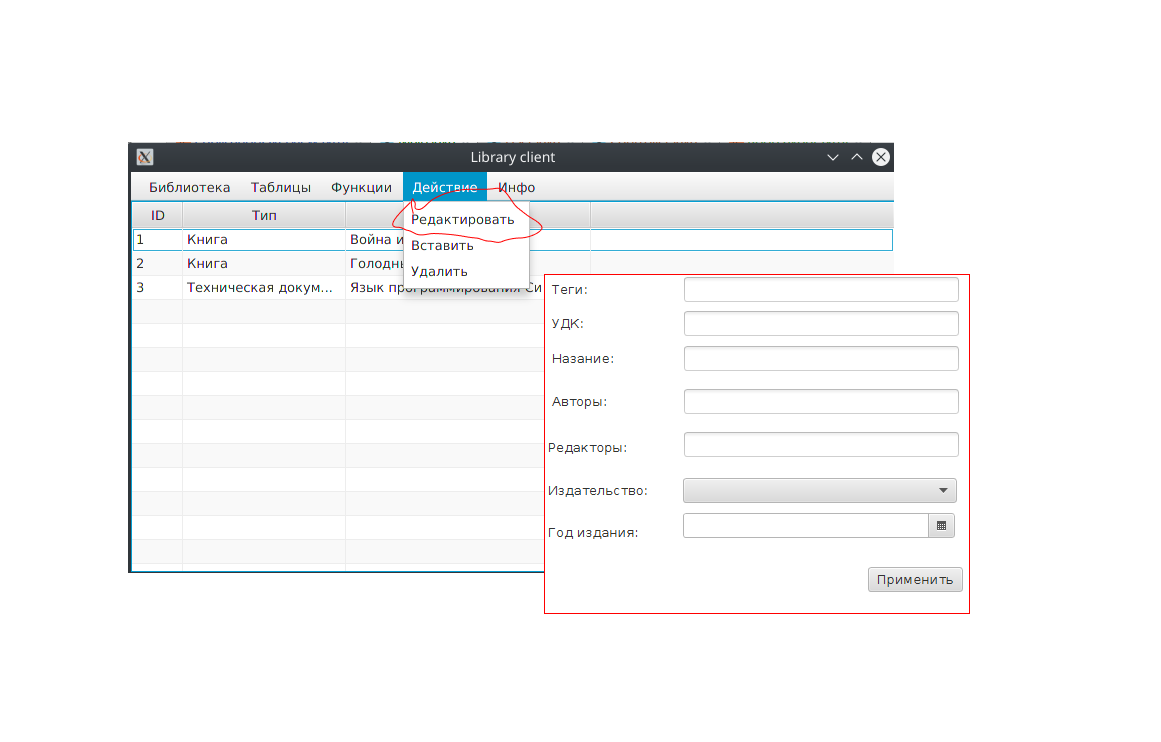
Стартовое окно программы:



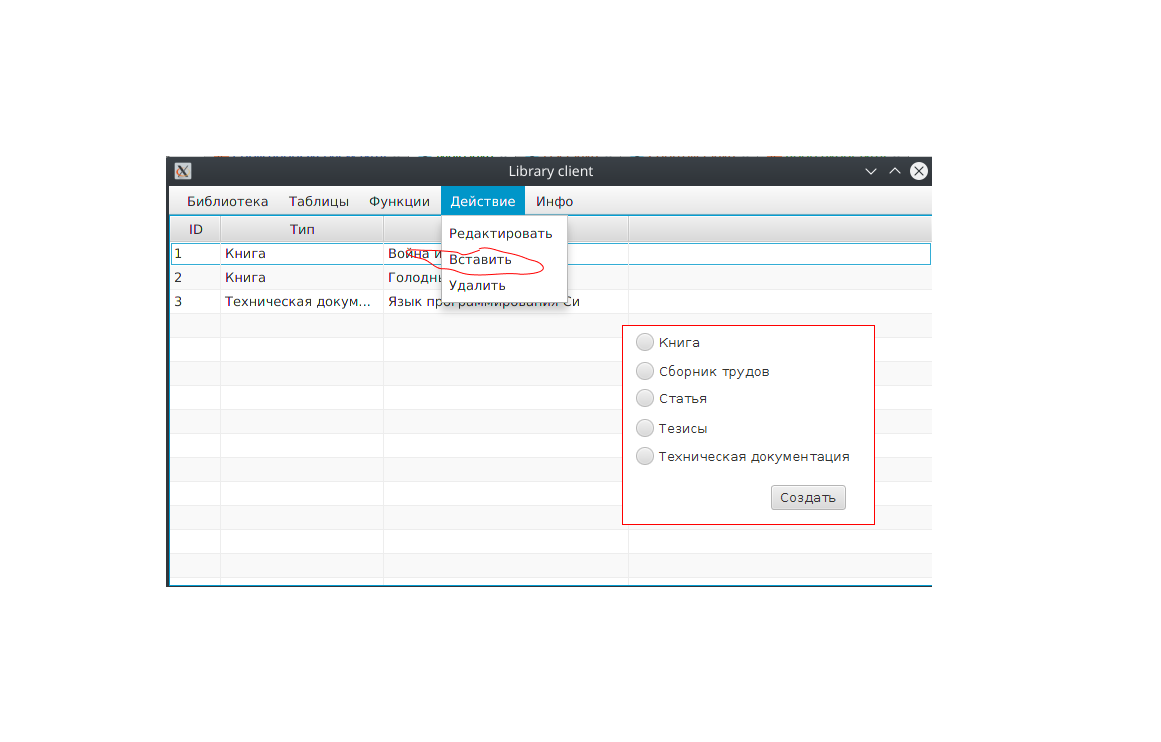


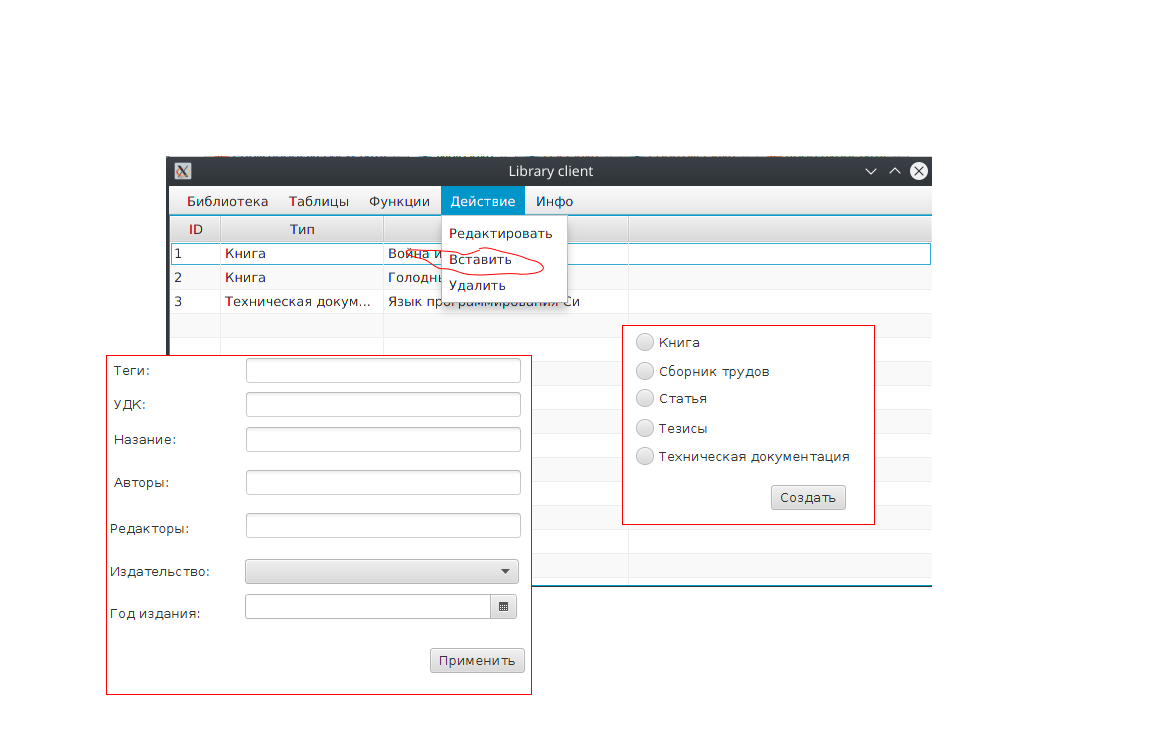


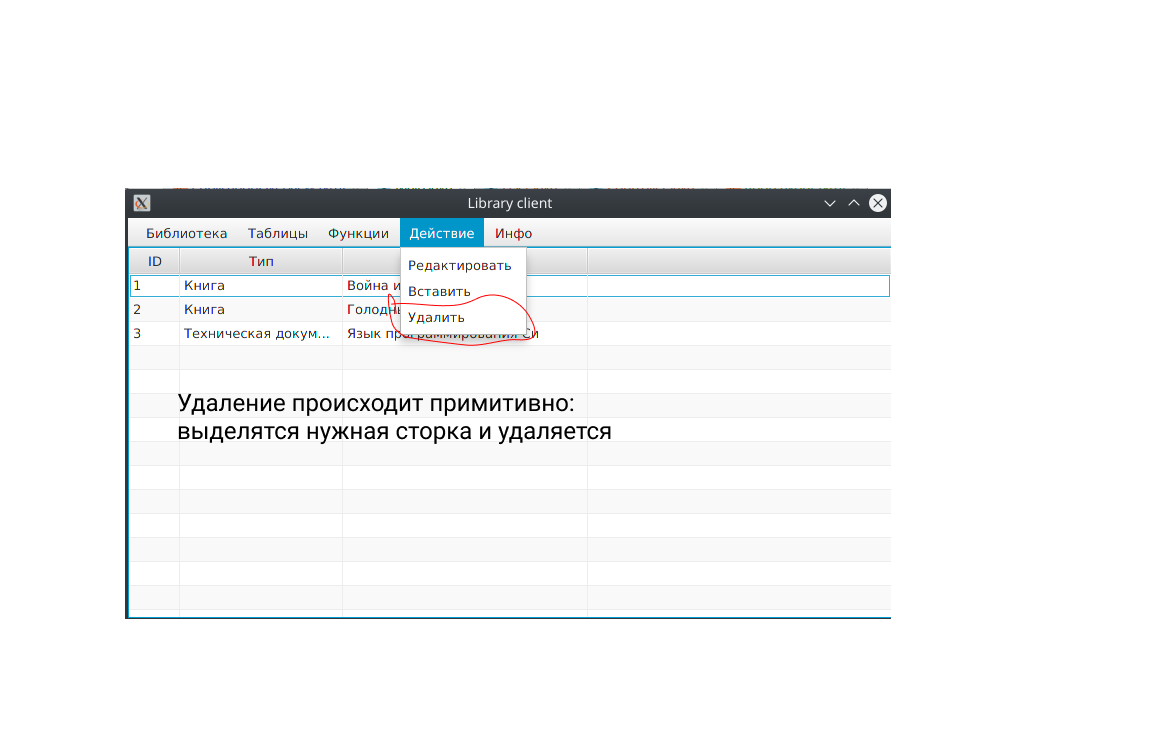


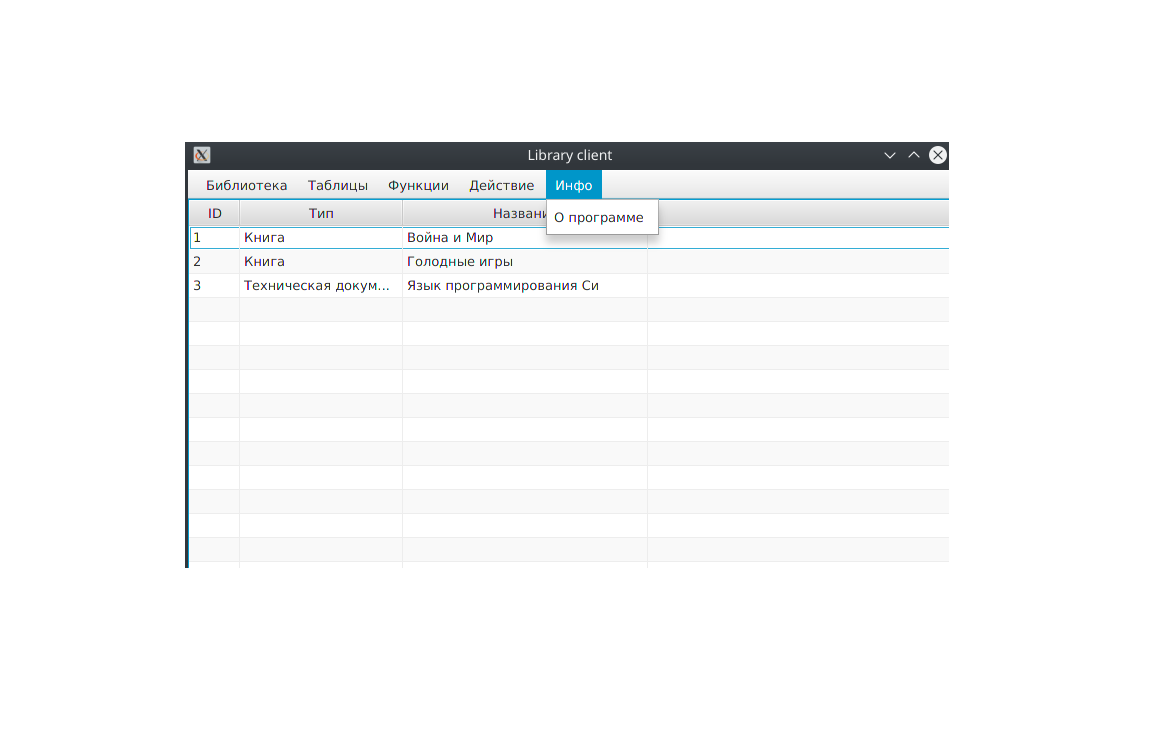


Выбираем нужный тип публикации (В зависимости от этого будут появляться разные окна для создания)









## 3.5 Диаграмма потоковых данных.

На основании результатов выявленных при диаграммы прецедентов можно спроектировать диаграмму потоков данных.

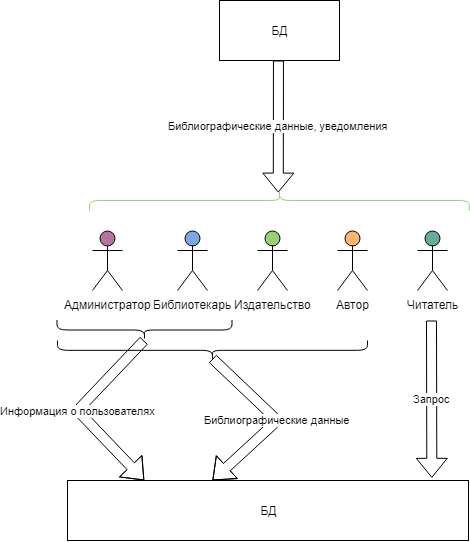


Диаграмма потоковых данных.

# Глава 4. Тестирование.

## 5.1 Введение

Тестирование — это процесс проверки соответствия реального и ожидаемого поведения программы на некотором наборе тестов, выбранных специальным образом.

Функциональное тестирование — один из методов тестирования программного обеспечения, нацеленное на проверку реализуемости функциональных требований и способности решать необходимые пользователю задачи.

Функциональное тестирование состоит из шести шагов:

1) Определить набор функций, которые программное обеспечение должно уметь выполнять.

2) Создание входных данных, основанных на спецификации программного обеспечения.

3) Создание выходных данных, основанных на спецификации программного обеспечения.

4) Выполнение тестов.

5) Сравнение действительных и ожидаемых выходных данных.

6) Проверить, удовлетворяет ли программное обеспечение всем требованиям.

## 5.2. Описание пользовательских функций

Основные пользовательские функции:

* Редактирование анкеты
* Редактирование записей
* Удаление
* Вставка
* Просмотр уведомлений
* Отправка запроса на верификацию

***1) “Редактирование анкеты”*** – функция, позволяющая пользователю редактировать данные в анкете, а именно: ФИО, дата рождения, пол. Данная функция доступна только группе «Администраторы».

Для каждого из входных данных существуют ограничения:

ФИО – это строковая константа, длина ФИО не должна превышать 50 символов, ФИО состоит минимум из одного слова, символов латинского алфавита, кириллицы, символов пробела и {‘-’, ‘.’,}.

При попытке ввести данные, не удовлетворяющие формату, будет выведено соответствующее сообщение об ошибке.

Дата рождения — поле, состоящее из трех чисел, между которыми находятся разделители.

***2) “Редактирование записей”*** – функция, позволяющая пользователям, имеющим соответствующие полномочия, редактировать записи таблиц. Разные группы пользователей имеют разный доступ к редактированию записей. Пользователи из групп «Администраторы» и «Библиотекари» имеют возможность редактировать все записи. Пользователи из групп «Издательство» и «Авторы» имею право редактировать только свои записи. «Читатели» не могут редактировать никакие записи.

***3) “Удаление”*** – функция, позволяющая пользователю удалять записи из таблиц. Разные группы пользователей имеют разный доступ к данной функции. Пользователи из групп «Администраторы» и «Библиотекари» имеют возможность удалять любые записи. Пользователи из групп «Издательство», «Авторы» и «Читатели» не имеют доступа к данной функции.

При попытке удалить запись, поля которых могут быть связаны с другими таблицами, будет получено соответствующее сообщение об ошибке.

***4) “Вставка”*** – функция, позволяющая пользователю добавлять записи в таблицы. Разные группы пользователей имеют разный доступ к данной функции. Пользователи из групп «Администраторы» и «Библиотекари» имеют возможность вставлять записи в любые таблицы. Пользователи из групп «Издательство», «Авторы» и «Читатели» не имеют доступа к данной функции.

***5) “Просмотр уведомлений”*** – функция, позволяющая пользователю из группы «Библиотекари» просматривать запросы на верификацию пользователей и принимать решение о переводе их в другую группу. Пользователь из группы «Читатели» может быть переведен в группу «Авторы» или «Издательство» при условии, что у читателя в анкете присутствуют контактные данные: e-mail и/или номер телефона.

***6) “Отправка запроса на верификацию”*** – функция, позволяющая пользователям из группы «Читатели» отправить запрос на перевод их в группу «Издательство» или «Авторы». Обязательным условием является наличие у пользователя контактных данных: e-mail и/или номера телефона.

## 5.3 Функциональное тестирование

Функция №1: **“Редактирование анкеты”**

1.) Длина ФИО должна не превышать 50 символов:

a.) Правильный класс эквивалентности: |ФИО|<50

**Вход**: “Иван”

**Выход**: “Иван”

b.)Неправильный класс эквивалентности 1: |имя|=0

**Вход**: ‘’

**Выход**: Неверные данные, имя не может быть пустой строкой.

c.) Неправильный класс эквивалентности 2: |имя|>50

**Вход**: “AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA”

**Выход**: Неверные данные, имя не должно превышать длину в 50 символов

d.) Правильный класс эквивалентности: |фамилия| = 0

**Вход**: “”

**Выход**: “”

Функция №2: **“Редактирование записей”**

1.) Ключевые слова (теги) имею ограничение на длину в 1000 символов.

2.) Ключевые слова — обязательное поле.

a.) Ошибка: ключевые слова — обязательное поле.

**Вход**: “”

**Выход**: “”

3.) УДК код — обязательное поле.

a.) Ошибка: УДК код — обязательное поле..

**Вход**: “”

**Выход**: “”

Аналогичное ограничение накладывается на функцию «Вставка».

Функция №3: **“Удаление”**

1. «Читатели», «Авторы» и «Издательство» не имеют доступ к данной функции. Элементы для вызова функции не будут активны, если группа пользователей входит в одну из вышеназванных.
2. «Администраторы» и «Библиотекари» имеют полномочия удалять записи.

Функция №4: **“Редактирование записей”**

1. «Читатели», «Авторы» и «Издательство» не имеют доступ к данной функции. Элементы для вызова функции не будут активны, если группа пользователей входит в одну из вышеназванных.
2. «Администраторы» и «Библиотекари» имеют полномочия вставлять записи в любые таблицы

Функция №5: **“Просмотр уведомлений”**

1. Функция активна только для «Библиотекарей».
2. «Библиотекарь» не может перевести пользователя в группу «Библиотекари» или «Администраторы».

Функция №6: **“Отправка запроса на верификацию”**

1. Данная функция доступна только у группы «Читатели».
2. Если пользователь не имеет в анкете номера телефона или электронной почты, то при отправке запроса он получит соответствующее сообщение.
3. Если пользователь повторно попробовал отправить запрос, то он получит сообщение о том, что его профиль уже отправлен на рассмотрение.

# Глава 5. Архитектурный проект.

## 5.1 Введение

На основе анализа предметной области был построен проект системы для управления библиографическими базами данных, понятный как заказчику так и исполнителю. В проекте были сформированы четкие требования, выявлены пользовательские функции, описаны процессы передачи данных, структуры данных. Все эти достижения способствуют созданию архитектурного проекта, который будет понятен только исполнителю. Данный переход необходим для создания документа, в который будет включено реальное представление программного средства на языке Java. Так как Java – объектно-ориентированный язык, то яркой моделью перехода является диаграмма классов.

Каждый класс имеет свои атрибуты и методы, описанные на конкретном языке реализации. На атрибуты накладываются ограничения, для методов описываются типы возвращаемых значений и передаваемые в них данные.

Между классами существуют следующие отношения:

класс может быть вызван другим классом(🡪).

Также, в данном разделе представлены все введённые перечисляемые типы данных, используемые для повышения читабельности кода.

## 5.2 Описание классов.

Используемые в программе перечисляемые типы данных:

* Тип прав пользователя (UserType):

администратор, библиотекарь, издательство, автор, читатель.

* Тип публикации (PublicationType):

книга, сборник трудов, статья, тезис, техническая документация.

* Тип ошибки (ErrorType):

отсутствие ошибки, ошибка подключения к БД, несоответствие прав доступа, прочие ошибки.

Используемые в программе классы:

* Пользователь (User)
* Запрос на верификацию (Verification)
* Уведомление (Notification)
* Класс взаимодействия с БД (DataBase)

Класс пользователя включает в себя:

* Поля:
  + ID пользователя (id) – целое неотрицательное число
  + Тип прав (type) – перечисляемый тип данных UserType
  + Имя пользователя (name) – строка, длиной не более 50 символов
  + Массив уведомлений (notifications)
* Конструктор класса:
  + Принимает все указанные выше поля

Класс запросов на верификацию включает в себя:

* Поля:
  + ID запроса (id) - целое неотрицательное число
  + ID пользователя (userID) - целое неотрицательное число
  + Текущий тип пользователя (userType) - перечисляемый тип данных UserType
  + Желаемый тип пользователя (toType) - перечисляемый тип данных UserType
* Конструктор класс:
  + Принимает все указанные выше поля
* Методы класса:
  + Генерация заголовка уведомления для верификации (getTitle) – генерируются автоматически, т. к. имеют общий для всех шаблон, изначально заложенный в программу. Возвращает строку не превышающую 100 символов.
  + Генерация текста уведомления для верификации (getMessage) – генерируются автоматически, т. к. имеют общий для всех шаблон, изначально заложенный в программу. Возвращает строку не превышающую 1000 символов.

Класс уведомлений:

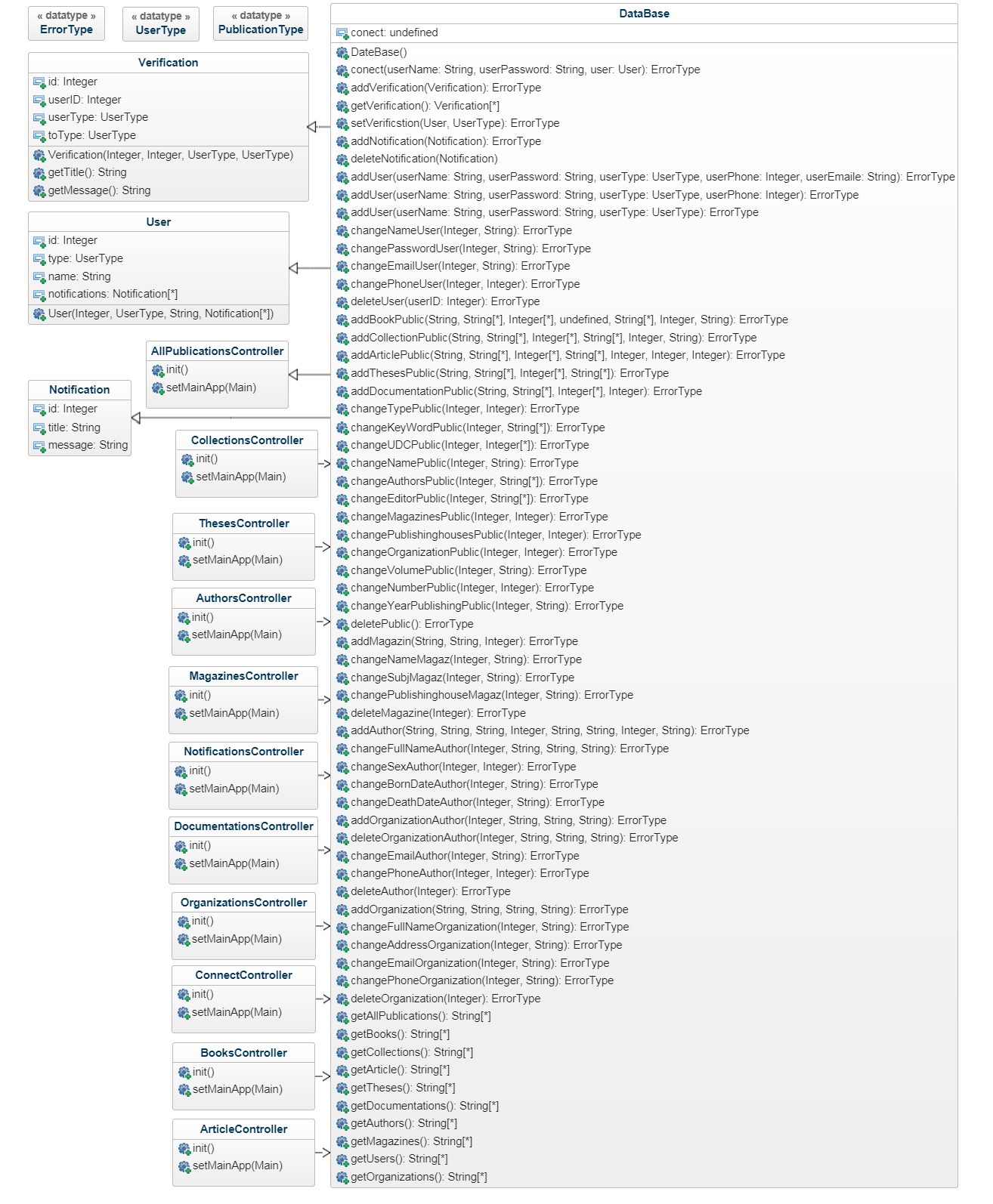
* Поля:
  + ID уведомления (id) - целое неотрицательное число
  + Заголовок уведомления (title) - строка, длиной не более 100 символов
  + Содержание уведомления (message) - строка, длиной не более 1000 символов

Методы и атрибуты остальных классов, а также их взаимосвязи можно увидеть в диаграмме классов.

### Ссылка на диаграмму классов:



Диаграмма классов:



(на диаграмме опущена часть классов и типов связанных с визуализацией)

## 5.3 Вывод.

На основе данных, выявленных на ранних этапах в данной главе был сформирован проект бедующей программы написанной на Java. Он включает в себя перечисляемые типы данных, схему классов, взаимосвязь между ними, позволяет утверждать однозначность их представления. Результаты, полученные в данной главе, позволяют перейти от абстрактной модели системы к реальному представлению программного средства на объектно-ориентированном языке.

# Глава 6. Проектирование БД

## 6.1 Введение

Cистема управления библиографическими базами данных – это клиент-серверное приложение, работающее с большим количеством информации о публикациях. Для успешной работы системы требуется разработать клиентское приложение, с помощью которого пользователь сможет получать доступ к информации, и серверное приложение, которое будет обрабатывать запросы пользователя с помощью базы данных.

## 6.2 ER-диаграмма, демонстрирующая структуру основной БД

[См. Приложение 2.](#_Приложение_2._ER-диаграмма,)

## 6.3 Таблицы БД.

База данных состоит из следующих информационных объектов:

1. Пользователи (users):

1. Идентификатор пользователя (users\_id):

1. Данный реквизит служит для удобства учета пользователей. Данный реквизит является уникальным;
2. Данный реквизит - это целое число, занимающее 4 байта;
3. Данный реквизит генерируется автоматически СУБД при создании нового пользователя системным администратором;
4. Степень важности – высокая;
5. Идентификатор является одним из реквизитов информационных объектов “ Пользователи”, “Другие источники”, “Уведомления”, “Запросы на верификацию”, “Привязка пользователя к издательству”, “Привязка пользователя к автору”.

2. Имя пользователя (name):

1. Данный реквизит служит для удобства учета пользователей;
2. Данный реквизит - это имя из множества имен на планете Земля (максимальная длина – 50 символов, допускаемых СУБД);
3. Данный реквизит заносится системным администратором

при создании нового пользователя;

1. Степень важности – средняя.

3. Фамилия пользователя (surname):

1. Данный реквизит служит для хранения дополнительной информации о пользователе;
2. Данный реквизит - это фамилия из множества фамилий на планете Земля (максимальная длина – 50 символов, допускаемых СУБД);
3. Данный реквизит заносится системным администратором

при создании нового пользователя;

1. Степень важности – низкая.

4. Отчество пользователя (patronymic):

1. Данный реквизит служит для хранения дополнительной информации о пользователе;
2. Данный реквизит - это отчество из множества отчеств на планете Земля (максимальная длина – 50 символов, допускаемых СУБД);
3. Данный реквизит заносится системным администратором

при создании нового пользователя;

1. Степень важности – низкая.

5. Пол (sex):

1. Данный реквизит служит для удобства учета пользователей;
2. Данный реквизит - это элемент множества {0, 1, 2}, где 0 ставится, если пользователь – женщина, 1 – если человек, 2 – во всех остальных случаях;
3. Данный реквизит заносится системным администратором

при создании нового пользователя;

1. Степень важности – средняя.

6. Логин (login):

1. Данный реквизит служит для идентификации пользователей при прохождении авторизации;
2. Данный реквизит – это уникальная строковая константа максимальной длины 15;
3. Данный реквизит заносится системным администратором

при создании нового пользователя;

1. Степень важности – высокая.

7. Пароль (password):

1. Данный реквизит служит для идентификации пользователей при прохождении авторизации;
2. Данный реквизит – это строковая константа длины от 8 до 50 символов;
3. Данный реквизит заносится системным администратором

при создании нового пользователя;

1. Степень важности – высокая.

8. Дата рождения (birth\_date):

1. Данный реквизит служит для хранения дополнительной информации о пользователе;
2. Данный реквизит - это дата;
3. Данный реквизит заносится системным администратором

при создании нового пользователя;

1. Степень важности – низкая.

9. Тип учетной записи (type):

1. Данный реквизит служит для определения группы пользователя;
2. Данный реквизит - это элемент множества {0, 1, 2, 3, 4}, где 0 ставится, если пользователь – администратор, 1 – если библиотекарь, 2 – если издательство, 3 – если автор, 4 – если читатель;
3. Данный реквизит по умолчанию равен 4 после создания нового пользователя;
4. Данный реквизит может быть изменен библиотекарем или администратором;
5. Степень важности – высокая.

10. Номер телефона (phone\_number):

1. Данный реквизит служит для хранения дополнительной информации о пользователе;
2. Данный реквизит - это номер телефона, хранящийся в виде целого числа;
3. Данный реквизит заносится системным администратором

при создании нового пользователя;

1. Степень важности – низкая.

11. Почта (email):

1. Данный реквизит служит для хранения дополнительной информации о пользователе;
2. Данный реквизит - это строковая константа максимальной длины 100;
3. Данный реквизит заносится системным администратором

при создании нового пользователя;

1. Степень важности – низкая.
2. Авторы (authors):
3. Идентификатор автора (authors \_id):
4. Данный реквизит служит для удобства учета авторов. Данный реквизит является уникальным;
5. Данный реквизит - это целое число, занимающее 4 байта;
6. Данный реквизит генерируется автоматически СУБД при создании нового автора системным администратором;
7. Степень важности – высокая;
8. Идентификатор является одним из реквизитов информационных объектов “Авторы”, “ Привязка автора к организации”, “Привязка авторов к публикации”, “Привязка пользователя к автору”.
9. Имя автора (name):
10. Данный реквизит служит для удобства учета авторов;
11. Данный реквизит - это имя из множества имен на планете Земля (максимальная длина – 50 символов, допускаемых СУБД);
12. Данный реквизит заносится системным администратором

при создании нового автора;

1. Степень важности – средняя.
2. Фамилия автора (surname):
3. Данный реквизит служит для хранения дополнительной информации об авторе;
4. Данный реквизит - это фамилия из множества фамилий на планете Земля (максимальная длина – 50 символов, допускаемых СУБД);
5. Данный реквизит заносится системным администратором

при создании нового автора;

1. Степень важности – низкая.
2. Отчество автора (patronymic):
3. Данный реквизит служит для хранения дополнительной информации об авторе;
4. Данный реквизит - это отчество из множества отчеств на планете Земля (максимальная длина – 50 символов, допускаемых СУБД);
5. Данный реквизит заносится системным администратором

при создании нового автора;

1. Степень важности – низкая.
2. Пол (sex):
3. Данный реквизит служит для удобства учета авторов;
4. Данный реквизит - это элемент множества {0, 1, 2}, где 0 ставится, если автор – женщина, 1 – если человек, 2 – во всех остальных случаях;
5. Данный реквизит заносится системным администратором

при создании нового автора;

1. Степень важности – средняя.
2. Дата рождения (birth\_date):
3. Данный реквизит служит для хранения дополнительной информации об авторе;
4. Данный реквизит - это дата;
5. Данный реквизит заносится системным администратором

при создании нового автора;

1. Степень важности – низкая.
2. Дата смерти (death \_date):
3. Данный реквизит служит для хранения дополнительной информации об авторе;
4. Данный реквизит - это дата;
5. Данный реквизит заносится системным администратором

при создании нового автора;

1. Степень важности – низкая.
2. Номер телефона (phone\_number):
3. Данный реквизит служит для хранения дополнительной информации об авторе;
4. Данный реквизит - это номер телефона, хранящийся в виде целого числа;
5. Данный реквизит заносится системным администратором

при создании нового автора;

1. Степень важности – низкая.
2. Почта (email):
3. Данный реквизит служит для хранения дополнительной информации об авторе;
4. Данный реквизит - это строковая константа максимальной длины 100;
5. Данный реквизит заносится системным администратором

при создании нового автора;

1. Степень важности – низкая.
2. Организации (organizations):
3. Идентификатор организации (organizations \_id):
4. Данный реквизит служит для удобства учета организаций. Данный реквизит является уникальным;
5. Данный реквизит - это целое число, занимающее 4 байта;
6. Данный реквизит генерируется автоматически СУБД при создании новой организации системным администратором;
7. Степень важности – высокая;
8. Идентификатор является одним из реквизитов информационных объектов “ Организации”, “ Привязка автора к организации”, “Привязка публикации к организации”, “ Сборники”, “ Журналы”.
9. Полное наименование (title):
10. Данный реквизит служит для определения наименования организации. Данный реквизит является уникальным;
11. Данный реквизит - это строковая константа максимальной длины 1000;
12. Данный реквизит заносится системным администратором

при создании новой организации;

1. Степень важности – высокая.
2. Юридический адрес (legal\_adress):
3. Данный реквизит служит для определения

юридического адреса организации;

1. Данный реквизит - это строковая константа максимальной длины 1000;
2. Данный реквизит заносится системным администратором при создании новой организации;
3. Степень важности – высокая.
4. Номер телефона (phone\_number):
5. Данный реквизит служит для хранения дополнительной информации об организации;
6. Данный реквизит - это номер телефона, хранящийся в виде целого числа;
7. Данный реквизит заносится системным администратором

при создании новой организации;

1. Степень важности – низкая.
2. Почта (email):
3. Данный реквизит служит для хранения дополнительной информации об организации;
4. Данный реквизит - это строковая константа максимальной длины 100;
5. Данный реквизит заносится системным администратором

при создании новой организации;

1. Степень важности – низкая.
2. Тематики (subjects):
3. Идентификатор тематики (subjects \_id):
4. Данный реквизит служит для удобства учета тематик. Данный реквизит является уникальным;
5. Данный реквизит - это целое число, занимающее 4 байта;
6. Данный реквизит генерируется автоматически СУБД при создании новой тематики системным администратором;
7. Степень важности – высокая;
8. Идентификатор является одним из реквизитов информационных объектов “ Тематики”, “ Сборники”, “Журналы”.
9. Название (title):
10. Данный реквизит служит для определения тематики. Данный реквизит является уникальным;
11. Данный реквизит - это строковая константа максимальной длины 100;
12. Данный реквизит заносится системным администратором

при создании новой тематики;

1. Степень важности – высокая.
2. Издательства (publishing\_houses):
3. Идентификатор издательства (subjects \_id):
4. Данный реквизит служит для удобства учета издательств. Данный реквизит является уникальным;
5. Данный реквизит - это целое число, занимающее 4 байта;
6. Данный реквизит генерируется автоматически СУБД при создании нового издательства системным администратором;
7. Степень важности – высокая;
8. Идентификатор является одним из реквизитов информационных объектов “Издательства”, “ Привязка пользователя к издательству”, “Привязка публикации к издательству”.
9. Полное наименование (title):
10. Данный реквизит служит для определения наименования издательства. Данный реквизит является уникальным;
11. Данный реквизит - это строковая константа максимальной длины 1000;
12. Данный реквизит заносится системным администратором

при создании нового издательства;

1. Степень важности – высокая.
2. Юридический адрес (legal\_adress):
3. Данный реквизит служит для определения

юридического адреса издательства;

1. Данный реквизит - это строковая константа максимальной длины 1000;
2. Данный реквизит заносится системным администратором при создании нового издательства;
3. Степень важности – высокая.
4. Номер телефона (phone\_number):
5. Данный реквизит служит для хранения дополнительной информации об издательстве;
6. Данный реквизит - это номер телефона, хранящийся в виде целого числа;
7. Данный реквизит заносится системным администратором

при создании нового издательства;

1. Степень важности – низкая.
2. Почта (email):
3. Данный реквизит служит для хранения дополнительной информации об издательстве;
4. Данный реквизит - это строковая константа максимальной длины 100;
5. Данный реквизит заносится системным администратором

при создании нового издательства;

1. Степень важности – низкая.
2. Журналы (magazines):
3. Идентификатор журнала (magazines \_id):
4. Данный реквизит служит для удобства учета журналов. Данный реквизит является уникальным;
5. Данный реквизит - это целое число, занимающее 4 байта;
6. Данный реквизит генерируется автоматически СУБД при создании нового журнала системным администратором;
7. Степень важности – высокая;
8. Идентификатор является одним из реквизитов информационных объектов “Журналы”, “ Привязка публикации к журналу”.
9. Тематика журнала (subjects\_id):

Реквизит описан в объекте (Тематики).

1. Организация журнала (organizations\_id):

Реквизит описан в объекте (Организации).

1. Публикации (publications):
2. Идентификатор публикации (publications \_id):
3. Данный реквизит служит для удобства учета публикаций. Данный реквизит является уникальным;
4. Данный реквизит - это целое число, занимающее 4 байта;
5. Данный реквизит генерируется автоматически СУБД при создании новой публикации системным администратором;
6. Степень важности – высокая;
7. Идентификатор является одним из реквизитов информационных объектов “Публикации”, “ Привязка публикации к журналу”, “Привязка публикации к издательству”, “Привязка авторов к публикации”, “Привязка публикации к организации”, “Привязка редактора к публикации”, “Привязка публикации к сборнику”, “Информация о журнальной статье”, “Годы издания книг и сборников”, “Привязка удк кода к публикации”, “Привязка ключевого слова к публикации”.
8. Тип публикации (type):
9. Данный реквизит служит для определения типа публикации;
10. Данный реквизит - это элемент множества {0, 1, 2, 3, 4}, где 0 ставится, если публикация – книга, 1 – если сборник трудов, 2 – если статья, 3 – если тезисы, 4 – если техническая документация;
11. Данный реквизит заносится системным администратором

при создании новой публикации;

1. Степень важности – высокая.
2. Название (title):
3. Данный реквизит служит для определения названия публикации;
4. Данный реквизит - это строковая константа максимальной длины 1000;
5. Данный реквизит заносится системным администратором

при создании новой публикации;

1. Степень важности – высокая.
2. Сборники (digests):
3. Идентификатор сборника (digests \_id):
4. Данный реквизит служит для удобства учета сборников. Данный реквизит является уникальным;
5. Данный реквизит - это целое число, занимающее 4 байта;
6. Данный реквизит генерируется автоматически СУБД при создании нового сборника системным администратором;
7. Степень важности – высокая;
8. Идентификатор является одним из реквизитов информационных объектов “ Сборники”, “Привязка публикации к сборнику”.
9. Тематика сборника (subjects\_id):

Реквизит описан в объекте (Тематики).

1. Организация сборника (organizations\_id):

Реквизит описан в объекте (Организации).

1. Коды удк (udc\_codes):
2. Идентификатор кода удк (udc\_codes \_id):
3. Данный реквизит служит для удобства учета кодов удк. Данный реквизит является уникальным;
4. Данный реквизит - это целое число, занимающее 4 байта;
5. Данный реквизит генерируется автоматически СУБД при создании нового кода удк системным администратором;
6. Степень важности – высокая;
7. Идентификатор является одним из реквизитов информационных объектов “ Коды удк”, “Привязка удк кода к публикации”.
8. Код (udc\_code):
9. Данный реквизит служит для хранения кода удк. Данный реквизит является уникальным;
10. Данный реквизит - это строковая константа максимальной длины 50;
11. Данный реквизит заносится системным администратором

при создании нового кода;

1. Степень важности – высокая.
2. Ключевые слова (keywords):
3. Идентификатор ключевого слова (keywords \_id):
4. Данный реквизит служит для удобства учета ключевых слов. Данный реквизит является уникальным;
5. Данный реквизит - это целое число, занимающее 4 байта;
6. Данный реквизит генерируется автоматически СУБД при создании нового ключевого слова системным администратором;
7. Степень важности – высокая;
8. Идентификатор является одним из реквизитов информационных объектов “Ключевые слова”, “Привязка ключевого слова к публикации”.
9. Ключевое слово (keyword):
10. Данный реквизит служит для хранения ключевого слова. Данный реквизит является уникальным;
11. Данный реквизит - это строковая константа максимальной длины 50;
12. Данный реквизит заносится системным администратором

при создании нового ключевого слова;

1. Степень важности – высокая.
2. Редакторы (editors):
3. Идентификатор редактора (editors \_id):
4. Данный реквизит служит для удобства учета редакторов. Данный реквизит является уникальным;
5. Данный реквизит - это целое число, занимающее 4 байта;
6. Данный реквизит генерируется автоматически СУБД при создании нового редактора системным администратором;
7. Степень важности – высокая;
8. Идентификатор является одним из реквизитов информационных объектов “Редакторы”, “Привязка редактора к публикации”.
9. Имя редактора (name):
10. Данный реквизит служит для удобства учета редакторов;
11. Данный реквизит - это имя из множества имен на планете Земля (максимальная длина – 50 символов, допускаемых СУБД);
12. Данный реквизит заносится системным администратором при создании нового редактора;
13. Степень важности – средняя.
14. Фамилия редактора (surname):
15. Данный реквизит служит для хранения дополнительной информации о редакторе;
16. Данный реквизит - это фамилия из множества фамилий на планете Земля (максимальная длина – 50 символов, допускаемых СУБД);
17. Данный реквизит заносится системным администратором при создании нового редактора;
18. Степень важности – низкая.
19. Отчество редактора (patronymic):
20. Данный реквизит служит для хранения дополнительной информации о редакторе;
21. Данный реквизит - это отчество из множества отчеств на планете Земля (максимальная длина – 50 символов, допускаемых СУБД);
22. Данный реквизит заносится системным администратором при создании нового редактора;
23. Степень важности – низкая.
24. Пол (sex):
25. Данный реквизит служит для удобства учета редакторов;
26. Данный реквизит - это элемент множества {0, 1, 2}, где 0 ставится, если редактор – женщина, 1 – если человек, 2 – во всех остальных случаях;
27. Данный реквизит заносится системным администратором при создании нового редактора;
28. Степень важности – средняя.
29. Дата рождения (birth\_date):
30. Данный реквизит служит для хранения дополнительной информации о редакторе;
31. Данный реквизит - это дата;
32. Данный реквизит заносится системным администратором при создании нового редактора;
33. Степень важности – низкая.
34. Дата смерти (death \_date):
35. Данный реквизит служит для хранения дополнительной информации о редакторе;
36. Данный реквизит - это дата;
37. Данный реквизит заносится системным администратором при создании нового редактора;
38. Степень важности – низкая.
39. Номер телефона (phone\_number):
40. Данный реквизит служит для хранения дополнительной информации о редакторе;
41. Данный реквизит - это номер телефона, хранящийся в виде целого числа;
42. Данный реквизит заносится системным администратором при создании нового редактора;
43. Степень важности – низкая.
44. Почта (email):
45. Данный реквизит служит для хранения дополнительной информации о редакторе;
46. Данный реквизит - это строковая константа максимальной длины 100;
47. Данный реквизит заносится системным администратором

при создании нового редактора;

1. Степень важности – низкая.
2. Привязка автора к организации (authors\_organizations):
3. Автор (authors\_id):

Реквизит описан в объекте (Авторы);

1. Организация, в которой работал автор (organizations\_id):

Реквизит описан в объекте (Организации);

1. Начало работы (start):
2. Данный реквизит служит для хранения даты приема на работу;
3. Данный реквизит - это дата;
4. Данный реквизит заносится системным администратором при создании новой привязки;
5. Степень важности – средняя.
6. Конец работы (finish):
7. Данный реквизит служит для хранения даты ухода с работы;
8. Данный реквизит - это дата;
9. Данный реквизит заносится системным администратором при создании новой привязки;
10. Степень важности – средняя.
11. Привязка пользователя к автору (users\_authors):
12. Пользователь (users\_id):

Реквизит описан в объекте (Пользователи);

1. Автор (authors\_id):

Реквизит описан в объекте (Авторы);

1. Привязка авторов к публикации (authors\_publications):
2. Публикации (publications\_id):

Реквизит описан в объекте (Публикации);

1. Автор публикации (authors\_id):

Реквизит описан в объекте (Авторы);

1. Запросы на верификацию (verifications):
2. Идентификатор запроса (verifications \_id):
3. Данный реквизит служит для удобства учета запроса. Данный реквизит является уникальным;
4. Данный реквизит - это целое число, занимающее 4 байта;
5. Данный реквизит генерируется автоматически СУБД при создании нового редактора системным администратором;
6. Степень важности – высокая;
7. Пользователь, который запрашивает (users\_id):

Реквизит описан в объекте (Пользователи);

1. Запрашиваемая группа (to\_type):
2. Данный реквизит служит для определения группы, в которую хочет вступить пользователь;
3. Данный реквизит - это элемент множества {0, 1, 2, 3, 4}, где 0 ставится, если пользователь – администратор, 1 – если библиотекарь, 2 – если издательство, 3 – если автор, 4 – если читатель;
4. Степень важности – высокая.
5. Уведомления (notifications):
6. Идентификатор уведомления (notifications \_id):
7. Данный реквизит служит для удобства учета уведомлений. Данный реквизит является уникальным;
8. Данный реквизит - это целое число, занимающее 4 байта;
9. Данный реквизит генерируется автоматически СУБД при создании нового уведомления;
10. Степень важности – высокая;
11. Заголовок (title):
12. Данный реквизит служит для хранения заголовка сообщения;
13. Данный реквизит - это строковая константа максимальной длины 100;
14. Данный реквизит заносится при создании нового уведомления;
15. Степень важности – низкая.
16. Сообщение (message):
17. Данный реквизит служит для хранения сообщения;
18. Данный реквизит - это строковая константа максимальной длины 1000;
19. Данный реквизит заносится при создании нового уведомления;
20. Степень важности – низкая.
21. Пользователь, которому отправлено уведомление (users\_id):

Реквизит описан в объекте (Пользователи);

1. Привязка пользователя к издательству (users\_publishing\_houses):
2. Пользователь (users\_id):

Реквизит описан в объекте (Пользователи);

1. Издательство (publishing\_houses \_id):

Реквизит описан в объекте (Издательства);

1. Привязка публикации к организации (organizations\_publications):
2. Публикация (publications\_id):

Реквизит описан в объекте (Публикации);

1. Организация (organizations\_id):

Реквизит описан в объекте (Организации);

1. Привязка публикации к издательству (publishing\_houses\_publications):
2. Публикация (publications\_id):

Реквизит описан в объекте (Публикации);

1. Издательство (publishing\_houses \_id):

Реквизит описан в объекте (Издательства).

1. Привязка публикации к журналу (magazines\_publications):
2. Публикация (publications\_id):

Реквизит описан в объекте (Публикации);

1. Журнал (magazines \_id):

Реквизит описан в объекте (Журналы).

1. Информация о журнальной статье (magazine\_article\_info):
2. Журнальная статья (publications\_id):

Реквизит описан в объекте (Публикации);

1. Том (volume):
2. Данный реквизит служит для хранения номера тома;
3. Данный реквизит – это целое неотрицательное число;
4. Степень важности – средняя.
5. Номер выпуска (release\_number):
6. Данный реквизит служит для хранения номера выпуска;
7. Данный реквизит – это целое неотрицательное число;
8. Степень важности – средняя.
9. Годы издания и сборников (publishing\_years):
10. Издание или сборник (publications\_id):

Реквизит описан в объекте (Публикации);

1. Год издания (year):
2. Данный реквизит служит для хранения года издания;
3. Данный реквизит – это целое неотрицательное число;
4. Степень важности – средняя.
5. Привязка публикации к сборнику (digests\_publications):
6. Публикация (publications\_id):

Реквизит описан в объекте (Публикации);

1. Сборник (digests \_id):

Реквизит описан в объекте (Сборники);

1. Привязка удк кода к публикации (publications\_udc\_codes):
2. Публикация (publications\_id):

Реквизит описан в объекте (Публикации);

1. Код удк (udc\_codes \_id):

Реквизит описан в объекте (Коды удк);

1. Привязка ключевого слова к публикации (publications\_ keywords):
2. Публикация (publications\_id):

Реквизит описан в объекте (Публикации);

1. Ключевое слово (keywords\_id):

Реквизит описан в объекте (Ключевые слова);

1. Привязка редактора к публикации (editors \_publications):
2. Публикация (publications\_id):

Реквизит описан в объекте (Публикации);

1. Редактор (editors\_id):

Реквизит описан в объекте (Редакторы);

## 6.4 Выбор СУБД

При решении данной задачи СУБД должна удовлетворять требованиям:

• Поддержка многопользовательского использования.

• Возможность управления распределенными БД.

• Возможность реализации хранимых процедур.

• Возможность добавления новых объектов (таблиц).

• Возможность добавления триггеров

• Кроссплатформенность

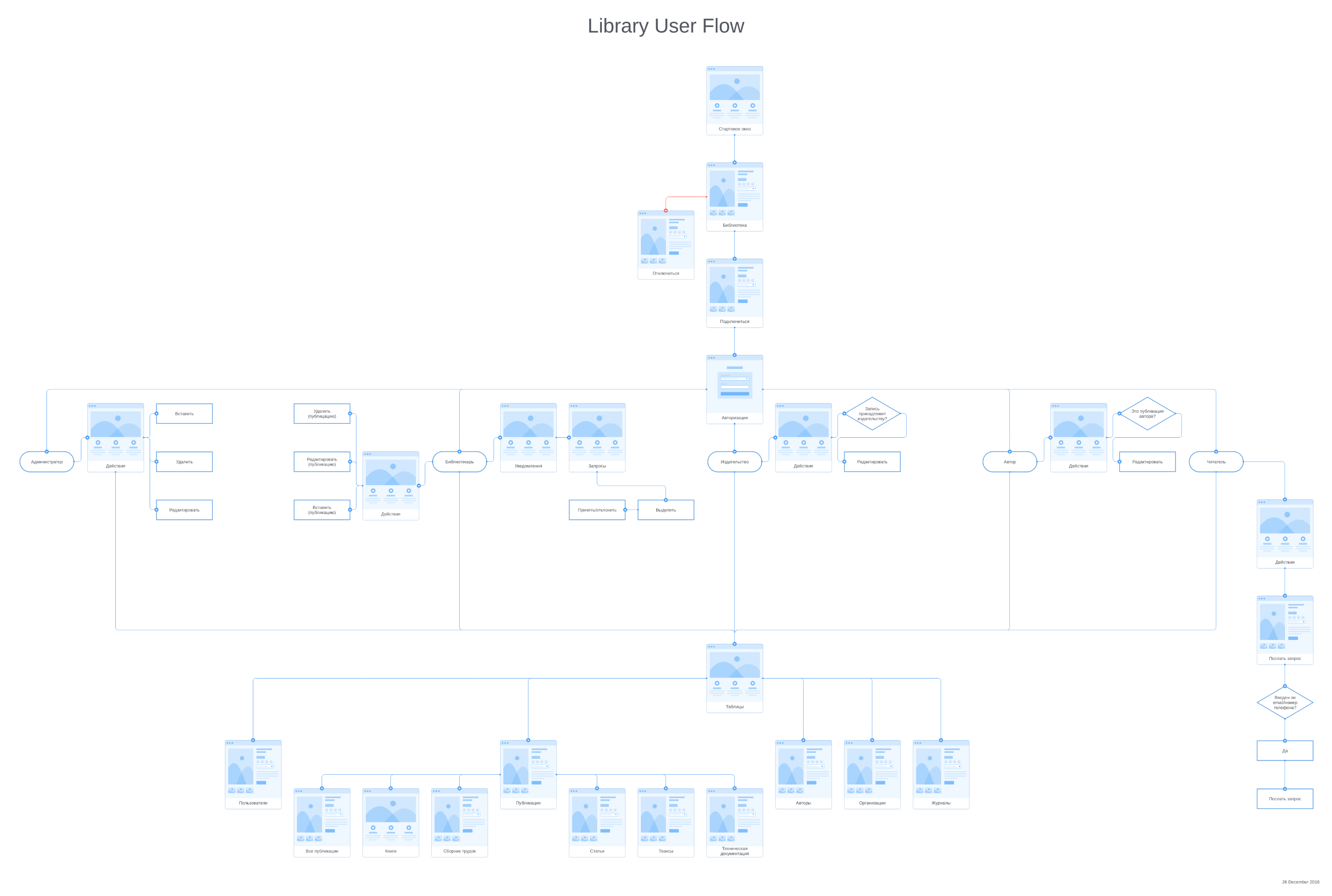
• Возможность добавлять транзакции

## 6.5 Заключение

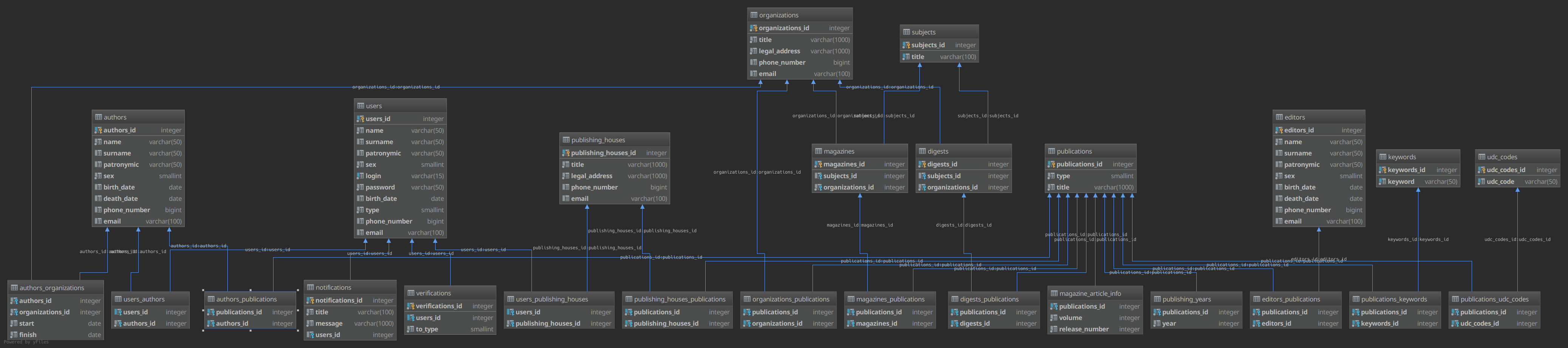
На основе данных, полученных при анализе предметной области, было сформировано четкое представление БД, необходимой для создания системы управления библиографическими базами данных. Было составлено описание объектов предметной области. Были установлены взаимосвязи между объектами. В качестве СУБД была выбрана бесплатная СУБД PostgreSQL. Таким образом, был создан проект БД.

# Приложения.

## Приложение 1. UserFlow диаграмма



## Приложение 2. ER-диаграмма, демонстрирующая структуру основной БД



Стрелки обозначают связь «многие к одному». Исходящий конец обозначает «многих», входящий - «одного».