**ОТЧЁТ ПО ПРАКТИЧЕСКИМ РАБОТАМ**

**Выполнил:**

**Студент группы ИСП-41-19**

**\_\_\_\_\_\_\_\_\_\_\_\_/Шепелёв С.А./**

г.Химки, 2022

**СОДЕРЖАНИЕ**

Введение

1 Основание для разработки.

2 Назначение разработки.

3 Требования к программе или программному изделию.

4 Требования к программной документации.

5 Технико-экономическое обоснование.

6 Стадии и этапы разработки.

7 Порядок контроля и приемки.

8 Приложения.

# ВВЕДЕНИЕ

Целью разработки программного продукта – создания программного решения, реализующего приложение автоматизированной рабочей системы «Электронный каталог CD-дисков». Данное приложение является хорошим решением для составления списка дисков, регулировать состав каталога, оно позволяет формировать в удобной и простой форме формировать полный список дисков, например, в личной коллекции.

**1 Основание для разработки**

Основанием для разработки программного продукта послужило выданное задание в филиале «РКТ» МАИ.

**2 Назначение разработки**

Программный продукт в первую очередь предназначен для упрощения процесса составления и упорядочения, обновления дискового ассортимента в магазинах, специализированных на продаже дискового носителя информации.

**3 Требования к программе или программному изделию**

Требования к функциональным характеристикам;

1. Пользователь должен иметь возможность вносить данные в БД;
2. Должна быть предусмотрена возможность редактировать, удалять и сохранять записи в базу данных;
3. У программы должен быть предусмотрен графический интерфейс взаимодействия с данными каталога.

Требования к надёжности и безопасности:

1. В коде программы необходимо предусмотреть обработку исключительных ситуаций;
2. Во избежание «падение» программы при неверном вводе данных пользователем должны быть введены всплывающие окна, предупреждающие пользователя о совершённых ошибках, и предотвращающих выход программы из строя.

Условие эксплуатации:

Программу можно эксплуатировать по прямому её назначению – для составления полного каталога магазина дисков, либо же использовать в личных целях (например, составить каталог дисков личной коллекции).

Требования к составу и параметрам технических средств:

Для использования программы потребуется компьютер со следующими аппаратными характеристиками:

1. 2-х ядерный процессор архитектуры x86, amd64 и выше;
2. 512 мб ОЗУ и больше;
3. Наличие видеочипа с 128 мб видеопамяти и выше;
4. 150 мб свободного дискового пространства.

Требования к информационной и программной совместимости:

Операционная система: Windows XP и выше.

Требования к маркировке и упаковке:

Требования к хранению и транспортированию:

Файлы проекта должны хранится на резервном жёстком диске, флэш накопителе либо на ресурсе github.com.

Специальные требования:

В коде должны присутствовать поясняющие комментарии, разъясняющие работу определённых функций, классов внутри программного кода.

**4 Требования к программной документации**

Документация должна иметь описание программного продукта, а также руководство по эксплуатации программного продукта для пользователя.

**5 Технико-экономическое обоснование**

Разрабатываемое программное решение позволит оптимизировать процесс составления каталога дисков, тем самым облегчив ему задачу составления каталога и при этом сэкономить время и силы.

# 6 Стадии и этапы разработки.

## **6.1. Проектирование программного обеспечения**

Прежде чем приступать к непосредственной разработке программного продукта, необходимо определить структуру программы, примерный алгоритм её работы, а также каким образом пользователь будет взаимодействовать с программным продуктом. Для этих целей, я составил ряд диаграмм и схем, описывающих работу программы и поясняющие принцип взаимодействия пользователя с её интерфейсом.

Диаграмма переходов состояния

Данная диаграмма демонстрирует поведение разрабатываемой программной системы при получении управляющих воздействий. Главное её предназначение – описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла.

Для создания данного вида диаграммы я воспользовался инструментами веб-сервиса Lucidchart, представленного на рисунке 1. Ключевой особенность данного ресурса является возможность строить любые нужные диаграммы и схемы без необходимости оформлять платную подписку. Интерфейс сайта прост и понятен. Слева находится панель с объектами, а справа поле для рисования.

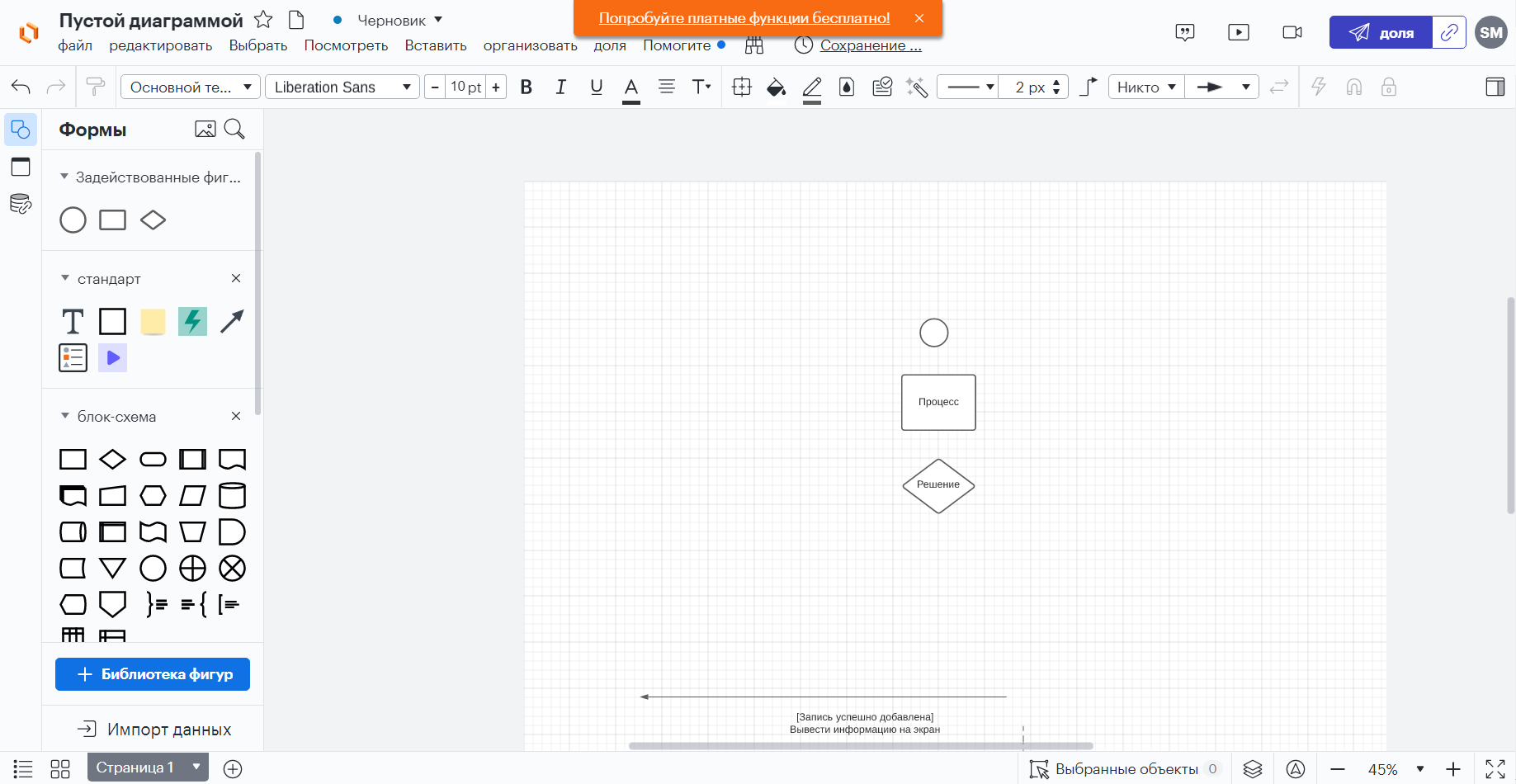


Рисунок 1 – Интерфейс Lucidchart.

Путём перетаскивания на поле можно отразить все ключевые схемы и диаграммы на итоговом рисунке. Помимо этого есть возможность соединить объекты между собой путём перетаскивания стрелочек на промежуток между двумя объектами, либо посредством последовательного нажатия левого клика с одного объекта на другой. В результате проделанных манипуляций была сделана диаграмма, представленная на рисунке 2.

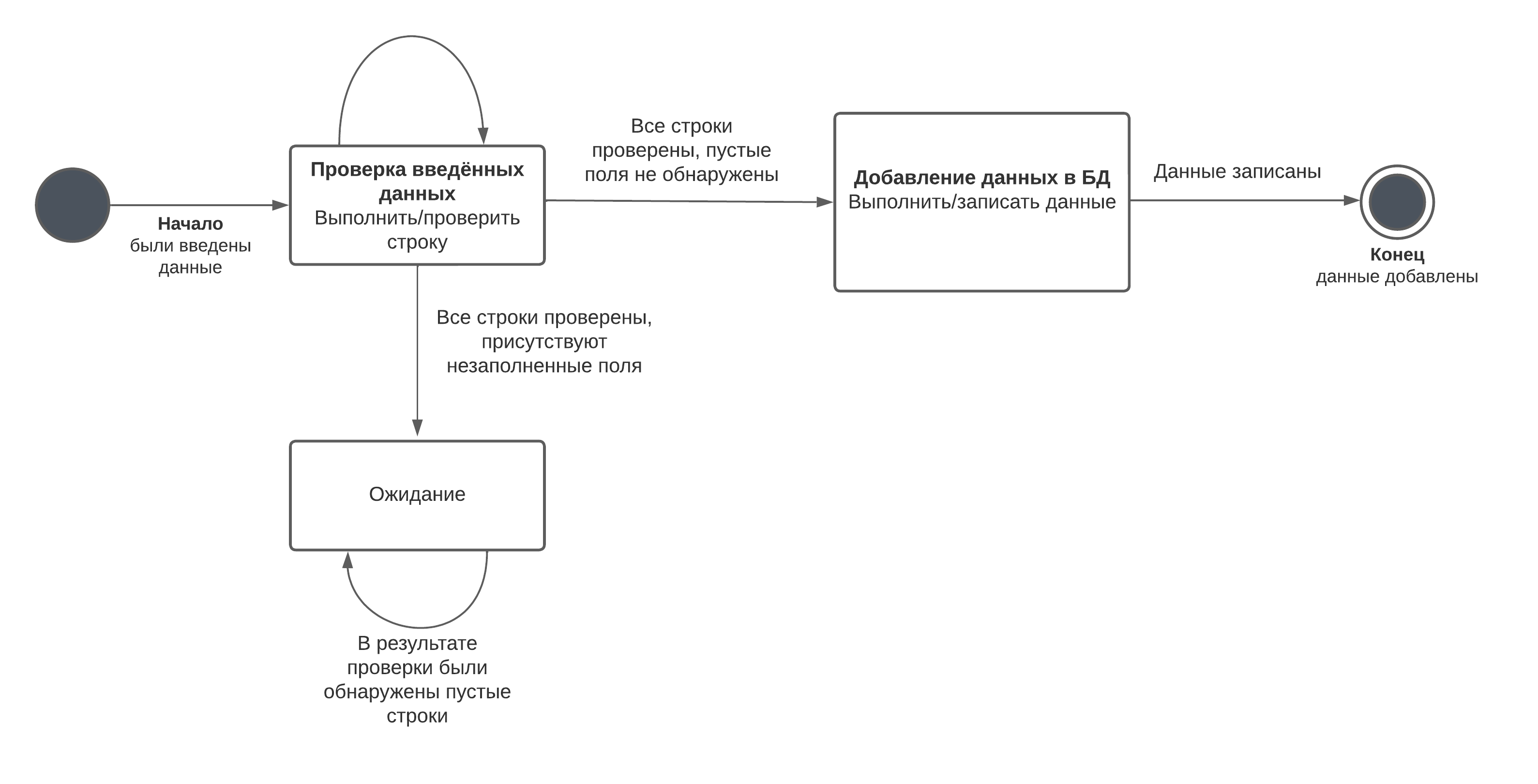


Рисунок 2 – Диаграмма переходов.

Функциональная диаграмма:

Данные диаграммы отражают взаимные связи функций разрабатываемого программного обеспечения. Они создаются на ранних стадиях проектирования систем, для того чтобы помочь проектировщику выявить основные функции и составные части проектируемой  
программной системы и, по возможности, обнаружить и устранить существенные ошибки.

Путём перетаскивания на поле можно отразить все ключевые схемы и диаграммы на итоговом рисунке. Помимо этого есть возможность соединить объекты между собой путём перетаскивания стрелочек на промежуток между двумя объектами, либо посредством последовательного нажатия левого клика с одного объекта на другой. Таким образом, была нарисована диаграмма, представленная на рисунке 3.

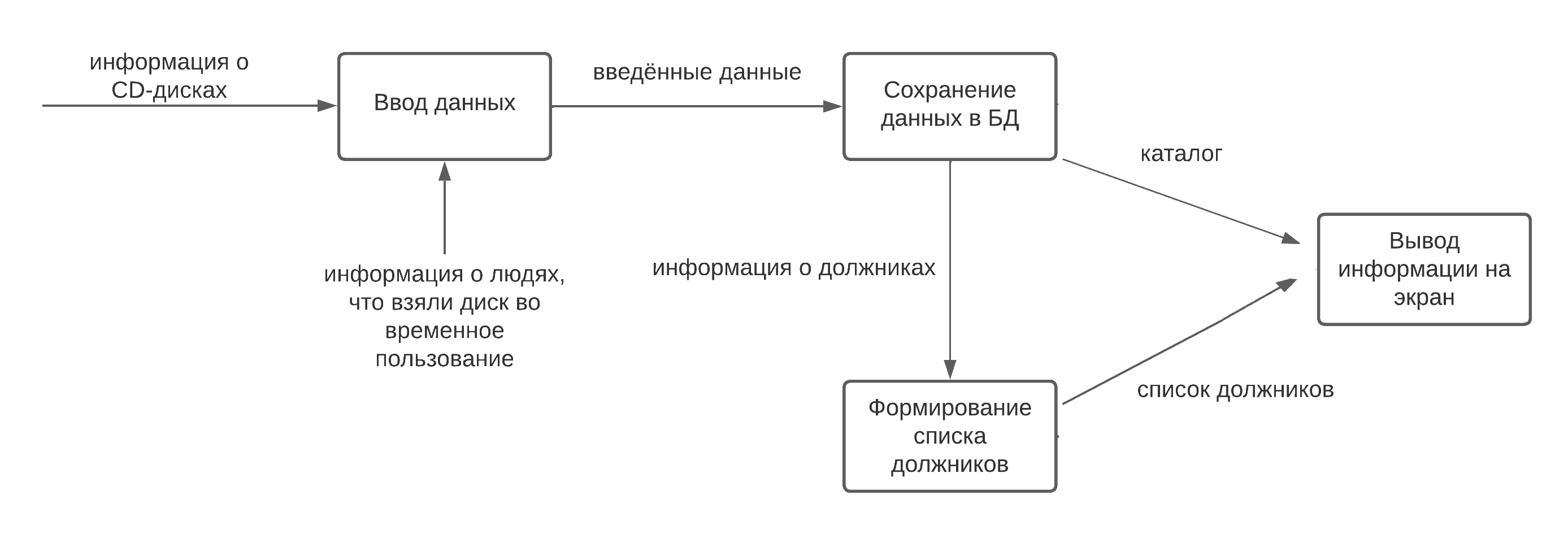


Рисунок 3 – Функциональная диаграмма.

Диаграмма потоков данных:

Диаграмма потоков данных — основное средство моделирования функциональных требований к системе, проектируемой или реально существующей. В основе модели лежат понятия внешней сущности, процесса, хранилища (накопителя) данных потока данных. Источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам; те, в свою очередь, преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных или внешним сущностям — потребителям информации.

Средствами всё того же ресурса была построена диаграмма, представленная на рисунке 4.

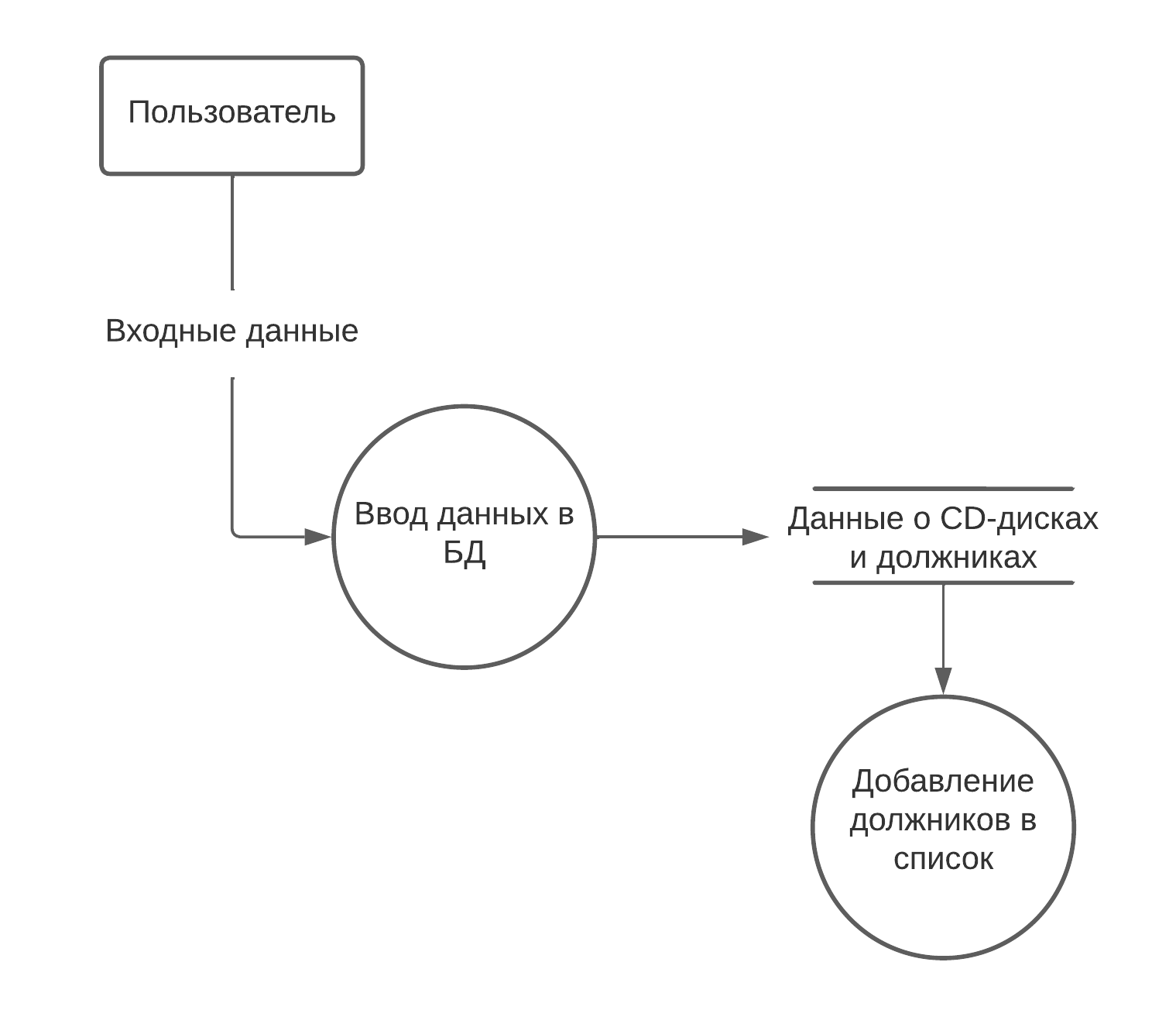


Рисунок 4 – Диаграмма потоков данных.

Диаграмма «сущность-связь»:

Данная диаграмма (ER-модель данных) обеспечивает стандартный способ определения данных и отношений между ними. Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. Диаграммы «сущность— связь» в отличие от функциональных диаграмм определяют спецификации структур данных программного обеспечения.

Средствами Lucidchart была создана диаграмму «сущность-связь» для своей программы. Итоговый вид диаграммы представлен на рисунке 5.

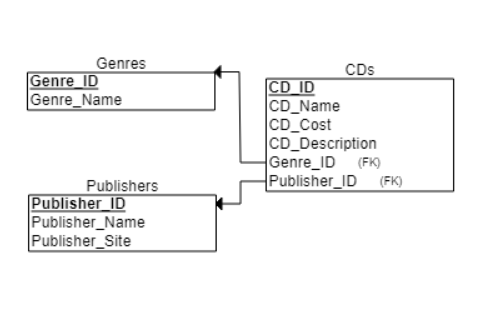


Рисунок 5 – Диаграмма «сущность-связь»

Структурная схема:

Процесс проектирования программного обеспечения включает в себя определение структурных компонентов программной системы и связей между ними. Результат уточнения структуры может быть представлен в виде структурной схемы, которая даёт достаточно полное представление о проектируемом программного обеспечении.

С помощью веб-ресурса Lucidchart, была составлена структурную схему для разрабатываемого мной программного продукта. Итоговый результат проделанной работы представлен на рисунке 6.

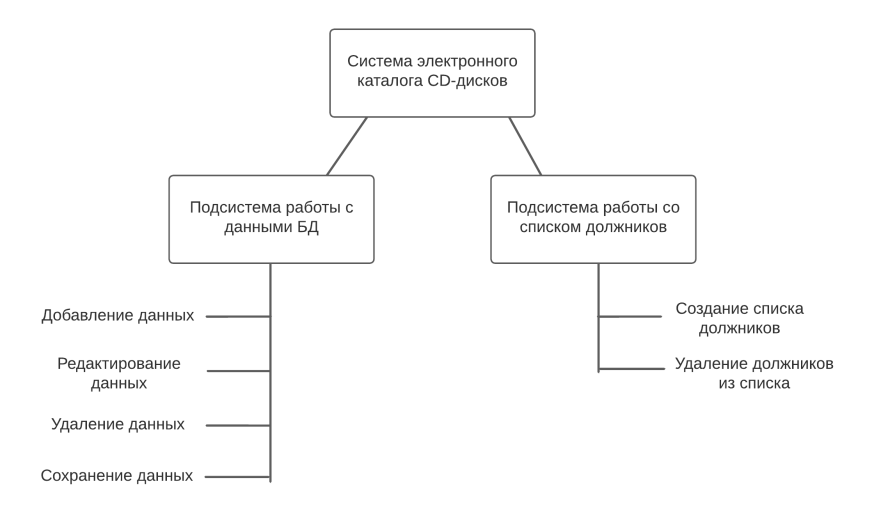


Рисунок 6 – Структурная схема

Функциональная схема:

Функциональная схема — это схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств. Схемы данных отображают путь данных при решении задач и определяют этапы обработки, а также различные применяемые носители данных.

Итоговая функциональная схема, созданная с помощью инструмента веб-ресурса, представлена на рисунке 7.

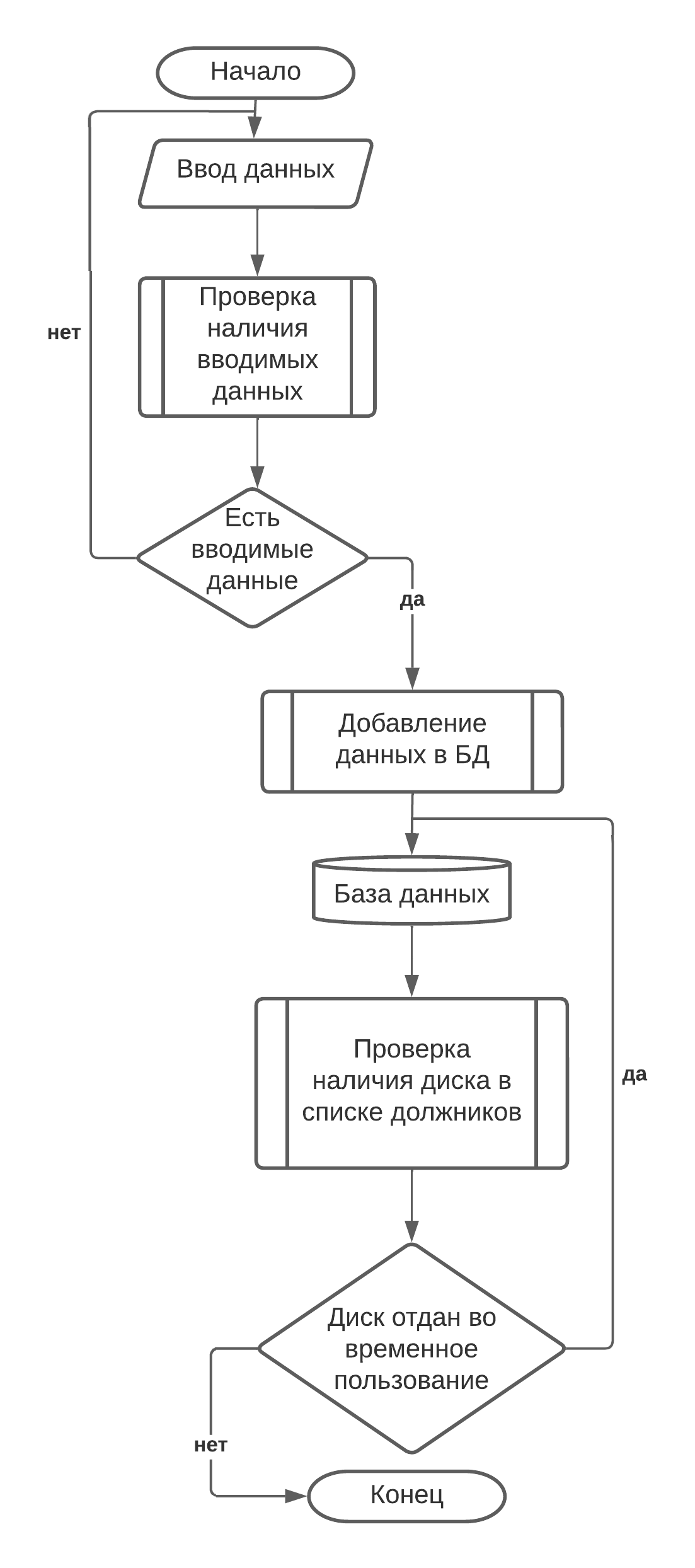


Рисунок 7 – Функциональная схема.

UML-диаграмма:

Язык UML предоставляет в распоряжение пользователей легко воспринимаемый и выразительный язык визуального моделирования, предназначенный для разработки и документирования сложных моделей систем различного целевого назначения.

С помощью средств Lucidchart можно создать и UML-диаграмму для разрабатываемого программного продукта.

Перетаскивая объекты на холст, и соединяя их между собой, была составлена диаграмма, представленная на рисунке 8.

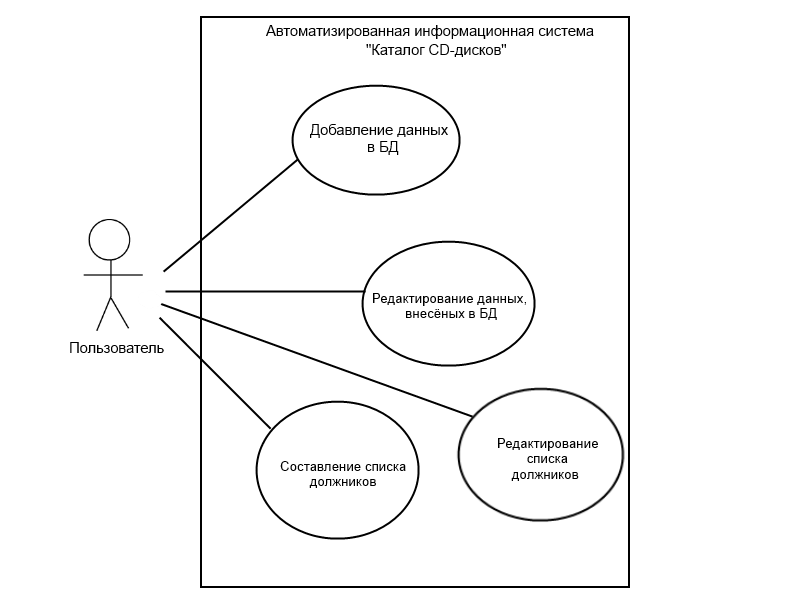


Рисунок 8 – UML-диаграмма.

Диаграмма последовательностей:

Диаграмма последовательности системы — графическая модель, которая для определенного сценария варианта использования, показывает динамику взаимодействия объектов во времени. На диаграмме последовательности изображаются исключительно те объекты, которые непосредственно участвуют во взаимодействии и не показываются возможные статические ассоциации с другими объектами. Для диаграммы последовательности ключевым моментом является именно динамика взаимодействия объектов во времени.

В результате проделанной работы, с помощью веб-инструмента создание схем и диаграмм мною была сделана диаграмма, представленная на рисунке 9.

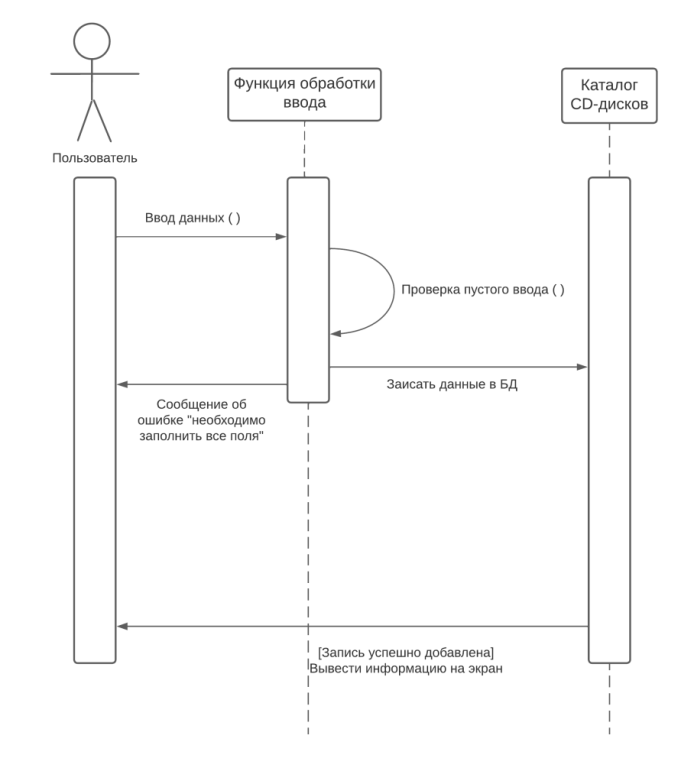


Рисунок 9 – Диаграмма последовательности.

Диаграмма деятельности:

Под деятельностью понимают задачу (операцию), которую необходимо выполнить вручную или с помощью средств автоматизации. Каждому варианту использования соответствует своя последовательность задач. В теоретическом плане диаграмма деятельности — это обобщенное представление алгоритма, реализующего анализируемый вариант использования. Диаграммы деятельности позволяют описывать альтернативные и параллельные процессы.

В результате проделанной работы была составлена диаграмма, представленная на рисунке 10.

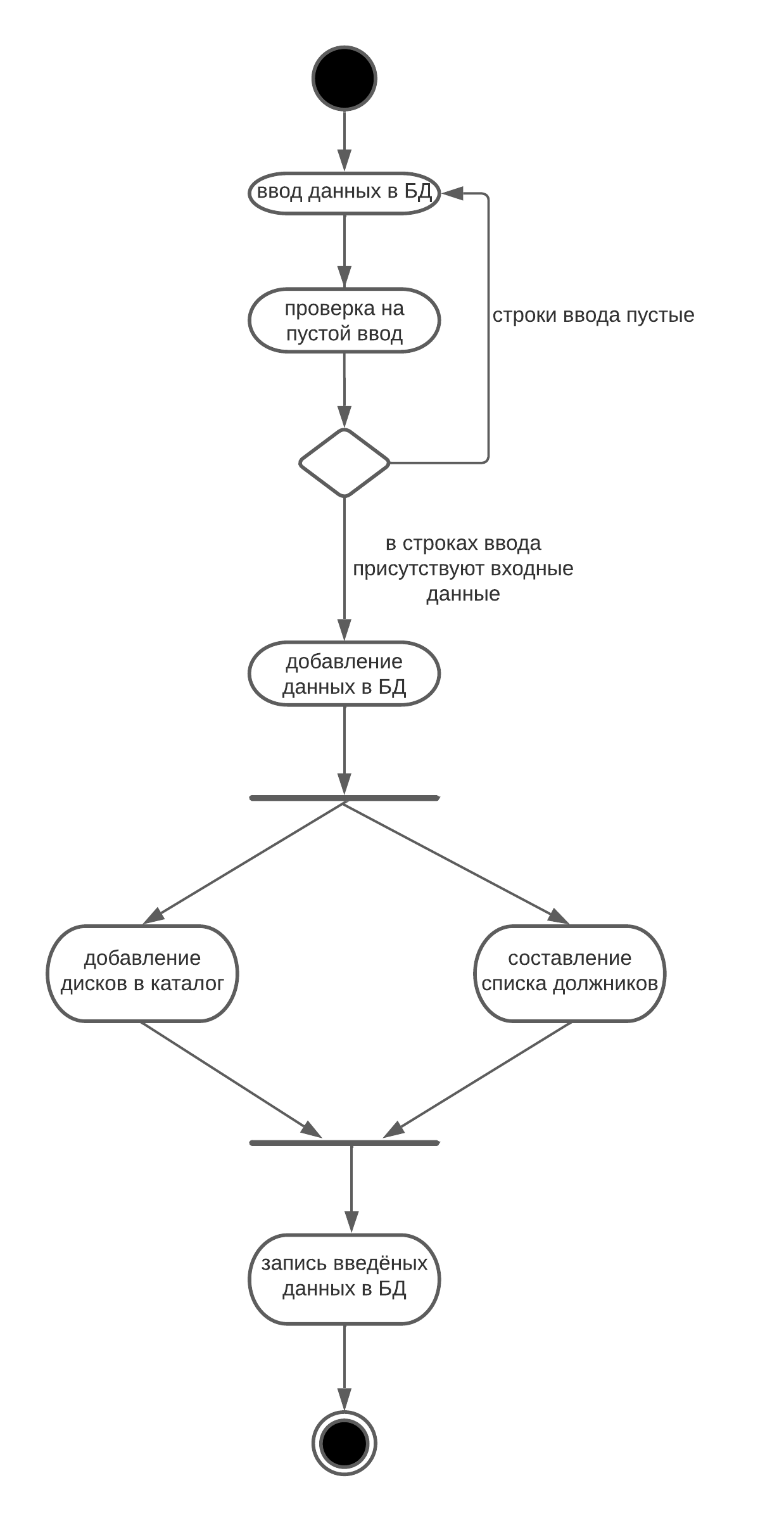


Рисунок 10 – Диаграмма деятельности.

# 6.2 Создание интерфейса

Для обеспечения наиболее удобного взаимодействия пользователя с функциями программы необходимо разработать пользовательский интерфейс, удовлетворяющий этим условиям. Для этого необходимо

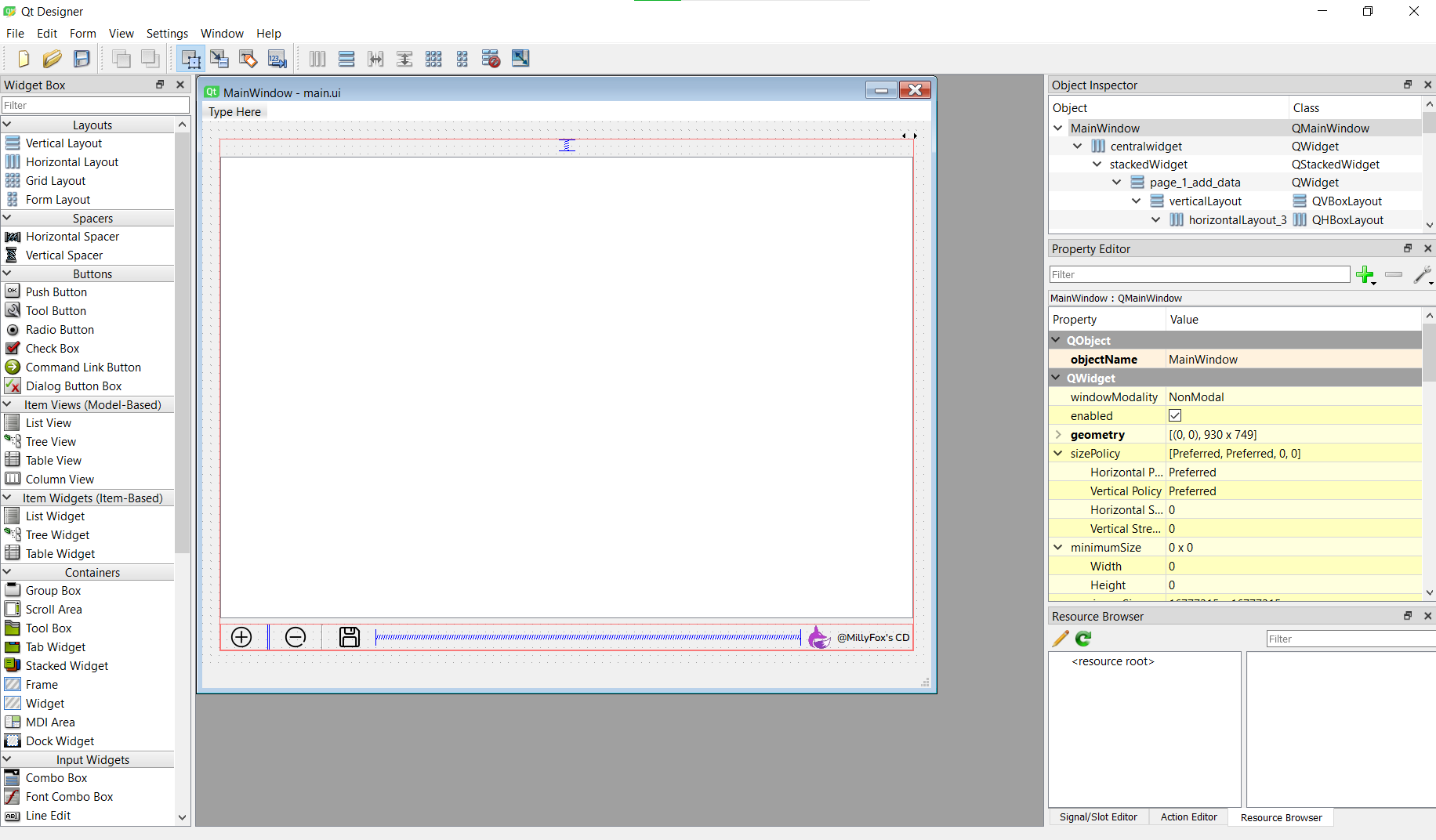


Рисунок 11 – Интерфейс Qt Designer.

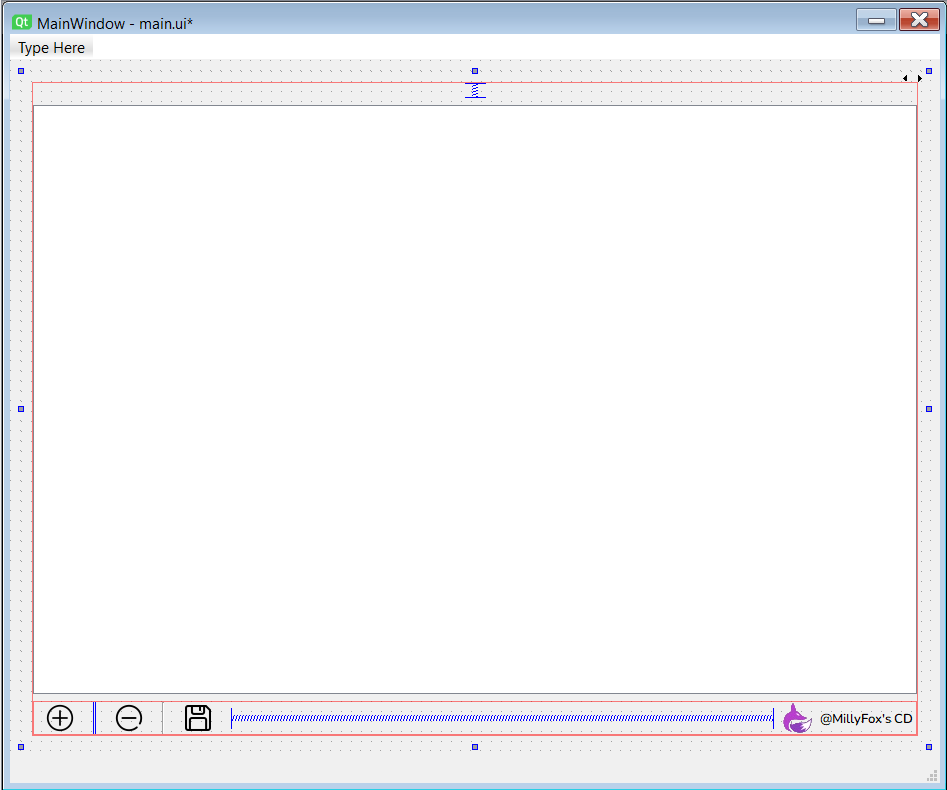


Рисунок 12 – Окно программы.