**User stories**

Как пользователь я могу добавлять, редактировать и удалять выбранную книгу в таблице, просматривать поля, а также иметь возможность поиска книги в базе по любому параметру

**Use case**

**Название:** «Может быть инициирован»

**Цель:** Отображение пользовательского интерфейса

**Начальное состояние:** Выполнение открытия программы и инициализация

**Основной сценарий:**

1. Пользователь открывает программу
2. Отображается таблица с данными

**Альтернативный сценарий:**

* 1. Если сервер не отвечает, по тем или иным причинам, выводится сообщение об ошибке
  2. Пользователь закрывает программу

**Сценарий обработки ошибок**

Предусловие: на шаге 1 основного сценария происходит ошибка отображения книг из базы (сбой сети и т. п.)

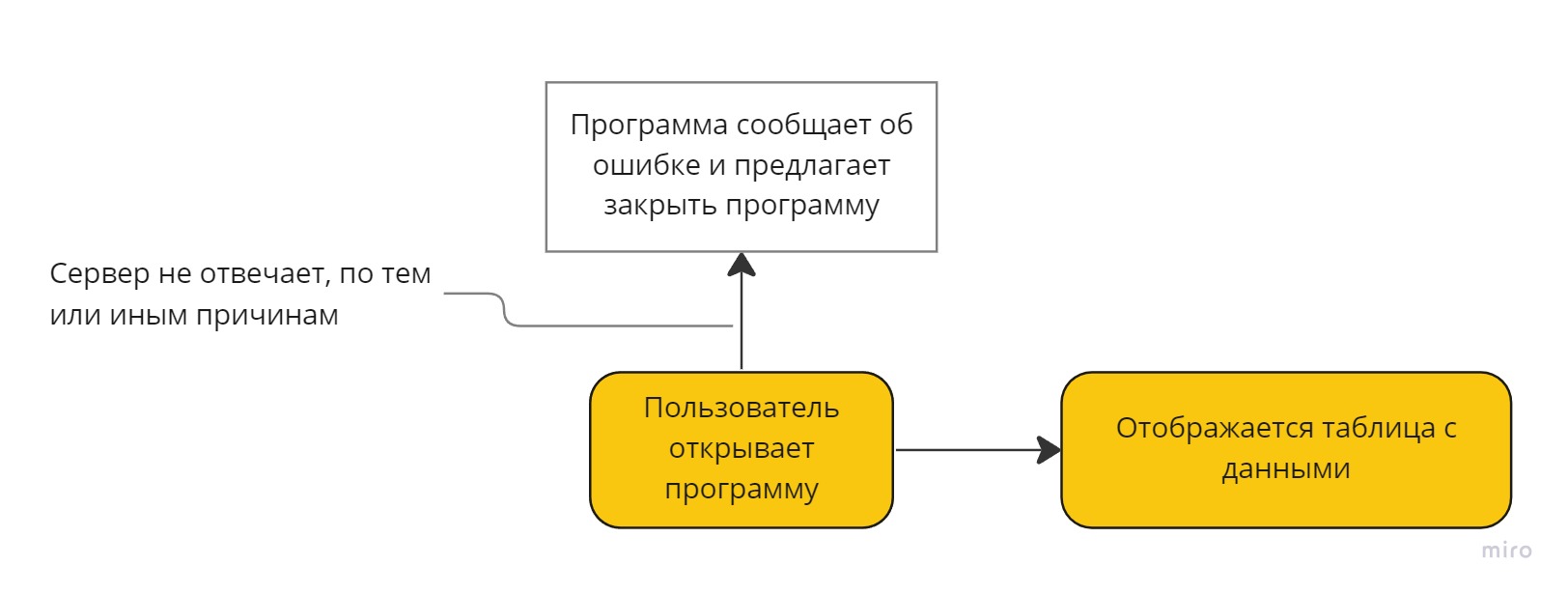


Рис. 1 Графическое изображение альтернативного сценария

**Название:** «Можно добавить новую книгу»

**Цель:** Добавить книгу в базу

**Начальное состояние:** Программа инициализирована

**Основной сценарий:**

1. Происходит нажатие на кнопку «добавить книгу»
2. Пользователь заполняет поля
3. Нажимает на кнопку «сохранить»
4. Проверка корректности ввода
5. Сохранение книги в базе

**Альтернативный сценарий:**

* 1. На шаге 3 пользователь имеет возможность закрыть диалоговое окно с редактированием новой книги, и она не добавится
  2. Отображение ошибки при неправильном вводе полей
  3. Если сервер не отвечает, по тем или иным причинам, выводится сообщение об ошибке

**Сценарий обработки ошибок**

Предусловие: на шаге 5 основного сценария происходит ошибка сохранения книги в базе (сбой сети и т. п.)

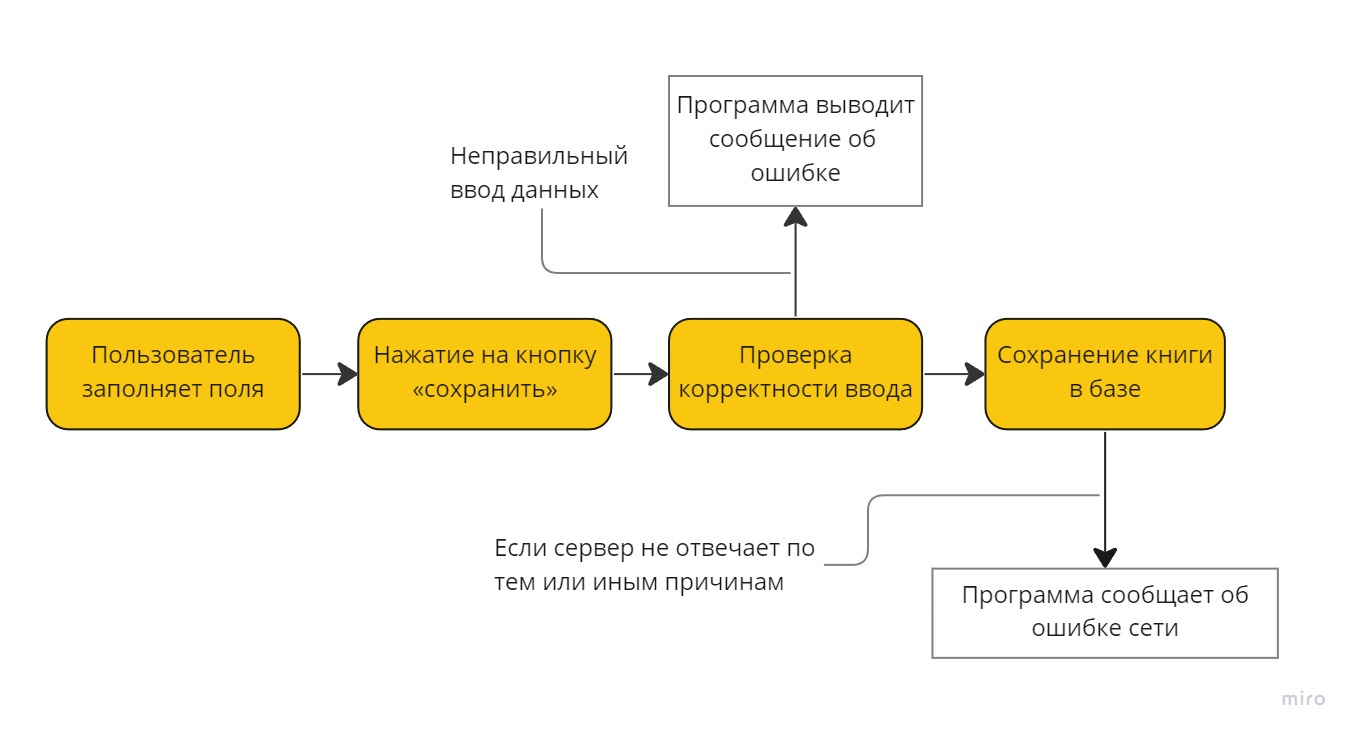


Рис. 2 Графическое изображение альтернативного сценария

**Название:** «Можно найти книгу»

**Цель:** Получить результат поискового запроса

**Начальное состояние:** Программа инициализирована

**Основной сценарий:**

1. Юзер набирает текст в строку поиска
2. Программа в режиме реального времени выводит книги, подходящие под запрос

**Название:** «Можно редактировать книгу»

**Цель:** Изменить данные книги

**Начальное состояние:** Программа инициализирована

**Основной сценарий:**

1. Юзер выбирает в таблице запись
2. Нажимает кнопку редактировать
3. Появляется окно редактирования книги
4. Пользователь вносит изменения в поля
5. Нажимает кнопку «сохранить»
6. Изменения отображаются в таблице

**Альтернативный сценарий:**

* 1. Книга не выбрана – всплытие окна, уведомляющем об ошибке
  2. На шаге 4 пользователь может не изменять книгу
  3. Сервер может не отвечать, по тем или иным причинам, в таком случае выводится сообщение об ошибке

**Сценарий обработки ошибок**

Предусловие: на шаге 5 основного сценария происходит ошибка изменения книги в базе (сбой сети и т. п.)

1. Программа сообщает об ошибке

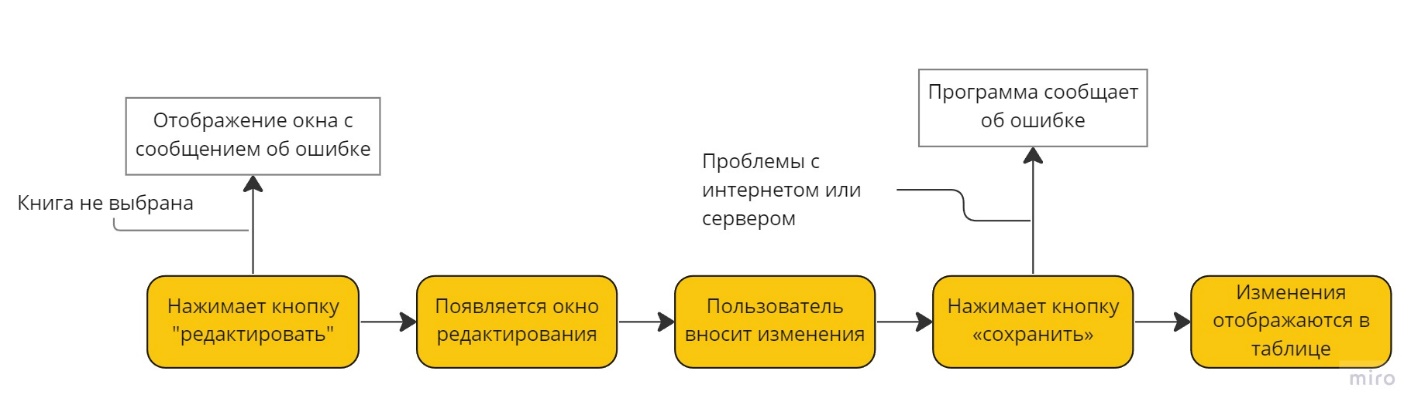


Рис. 3 Графическое изображение альтернативного сценария

**Название:** «Можно удалить книгу»

**Цель:** Удаление данных книги

**Начальное состояние:** Программа инициализирована, пользователь выбрал книгу

**Основной сценарий:**

1. Пользователь выбирает в таблице запись
2. Нажимает кнопку удалить
3. В таблице исчезнет выбранная запись

**Альтернативный сценарий:**

* 1. Книга не выбрана – всплытие окна, уведомляющем об ошибке
  2. Если сервер не отвечает, по тем или иным причинам, выводится сообщение об ошибке и пользователю предлагается сохранить данные в файл

**Сценарий обработки ошибок**

Предусловие: на шаге 2 основного сценария происходит ошибка удаления книги (сбой сети и т. п.)

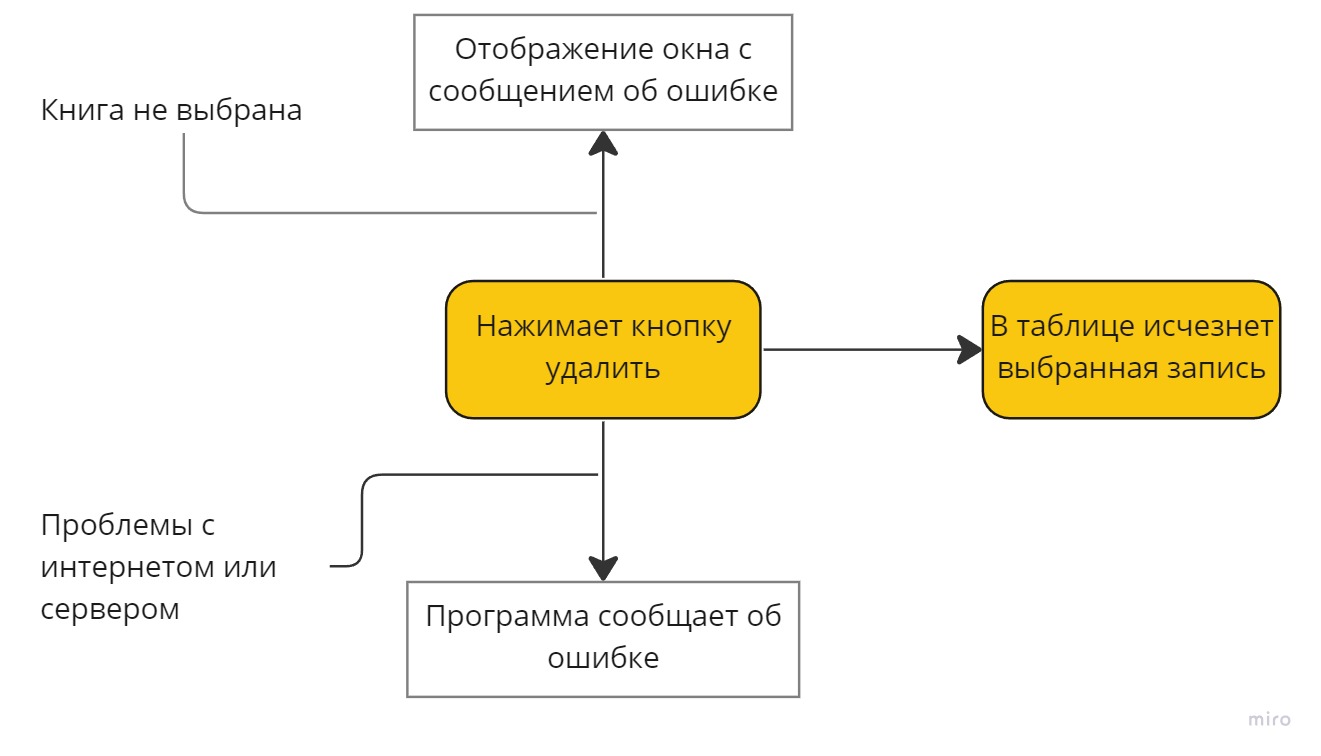
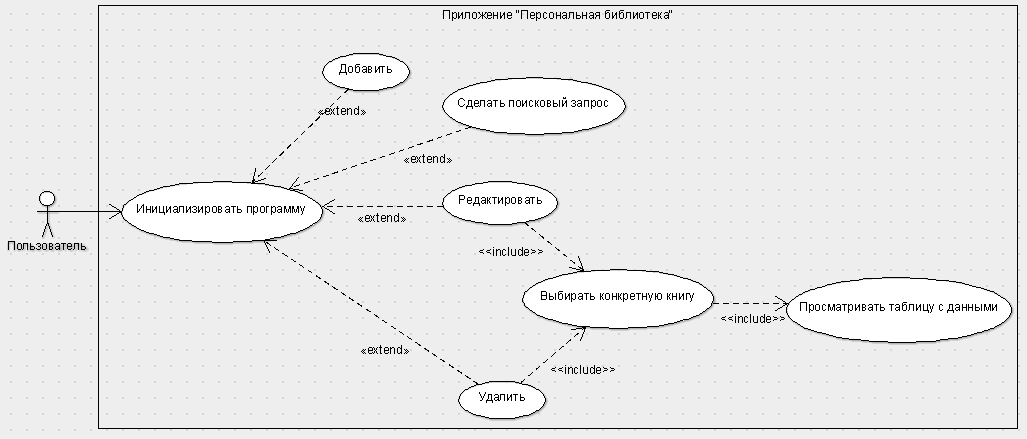


Рис. 4 Графическое изображение альтернативного сценария

**Use case diagram**



**Specification**

1. **Общие сведения**
2. **Полное наименование системы и ее условное обозначение**

Программа для администрирования библиотеки через сеть «Личная библиотека»

1. **Наименование разработчика системы и реквизиты заказчика.**

Заказчик -

Разработчик - студент группы ИС-3 Молчанов П.Д.

1. **Основания для разработки АС.**

Работа по созданию клиент-серверной программы, которая позволяет администрировать библиотеку

1. **Порядок оформления и предъявления заказчику результатов работ по созданию системы:**

К результатам труда разработчика относится:

* оригинальное программное обеспечение
* типовые проектные решения и особенности построения клиент-серверной архитектуры
* проектная и рабочая документация

Заказчику передаются:

* программное обеспечение
* документация

Заказчик приобретает у третьих лиц:

* пассивное сетевое оборудование
* серверное оборудование
* аппаратное оборудование

1. **Назначение и цели создания системы**
2. **Назначение системы**

Программа позволяет вести и администрировать личную электронную библиотеку, имея возможность добавления, редактирования, удаления и поиска книг в этой системе

1. **Цели создания системы**

Целью создания системы является:

* Учится разрабатывать клиент-серверное приложение
* Получение новых знаний в сфере архитектуры
* Изучение потоков данных в программе
* Изучение технологий java fx и spring framework
* Упростить взаимодействие с программой UI
* Ускорить процессы внутри программы

1. **Характеристика объекта автоматизации**
2. **Краткие сведения об объекте автоматизации**

Объектом автоматизации является личная библиотека

1. **Требования к системе**
2. **Требование к структуре и функционированию системы**

ИС администрирования библиотеки должна представлять собой систему, реализованную по концепции MVC, включающую в себя подсистемы:

п/с функционал клиента:

* добавление книги
* удаление выбранной книги
* редактирование выбранной книги
* поиска по любым полям

п/с функционал сервера:

* получение списка книг
* добавление книги
* редактирование книги
* удаления книги по идентификатору
* коннект с базой данных
* выводы ошибок, при их наличии
* отправка данных через api
* получение данных через api

1. **Требования к средствам и способам связи для информационного обмена между компонентами системы**

Для информационного обмена между компонентами системы должна быть организована локальная сеть. ИС функционирует как на клиенте, так и на сервере, к которому имеют доступ пользователи этой программой по средствам локальной сети

1. **Требования по эргономике и технической эстетике.**

Программа, работающая на клиенте, должна иметь минималистичный интерфейс:

* Основной цвет: светло-синий
* Другой цвет должен гармонировать с основным, к примеру белый
* Шрифт должен хорошо читаться
* Размер шрифта в диапазоне от 12 до 32 пунктиров
* Все кнопки должны находится в ожидаемых местах

1. **Требование к структуре и функционированию системы**