# Логическая схема базы данных

# Изображение выглядит как текст, диаграмма, документ, Шрифт Автоматически созданное описаниеДиаграмма классов

# Тестирование приложения

Тестирование проводилось методом «Чек-лист». Тип тестирования – функциональное тестирование.

## Чек-лист тестирования:

A) Вход под аккаунтом студента;

B) Отображение отправленных заявок;

C) Отправка новой заявки;

D) Отображение отправленной заявке в списке отправленных;

E) Отображение списка заявок студента в личном кабинете администратора;

F) Рассмотрение выбранной заявки администратором;

G) Обновления списка заявок после рассмотрения;

## Тестирование входа в аккаунт студента

### «Позитивное тестирование»

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описаниеИзображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

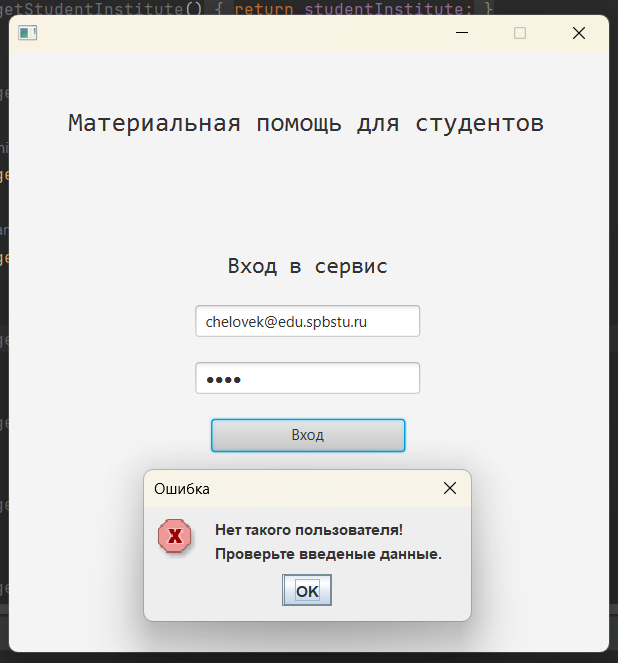
Вход в аккаунт ученика (Петров Петр П. (пароль - 1111)).

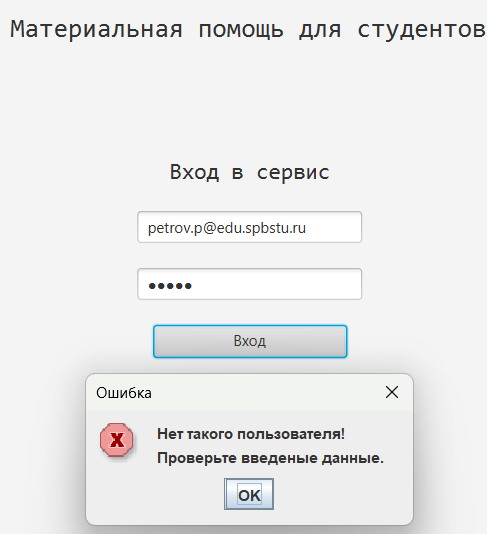
По нажатию кнопки «Вход», успешно открывается следующая сцена приложения («Личный кабинет»).

Изображение выглядит как текст, снимок экрана, дисплей, число

Автоматически созданное описание**Позитивное тестирование для входа ученика успешно пройдено.**

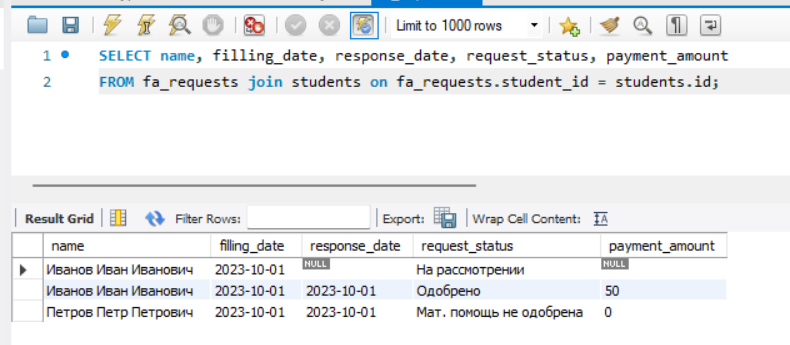
### «Негативное тестирование»

На ввод несуществующих данных аккаунта должен сработать отказ входа и всплыть окошко с предупреждением.

****На ввод неправильного пароля должно произойти то же самое. (На следующем скриншоте был введен пароль 12345)

**Негативное тестирование для входа студента успешно пройдено.**

## Отображение отправленных заявок

Имеющиеся заявки в базе данных:

Изображение выглядит как текст, снимок экрана, число, Шрифт

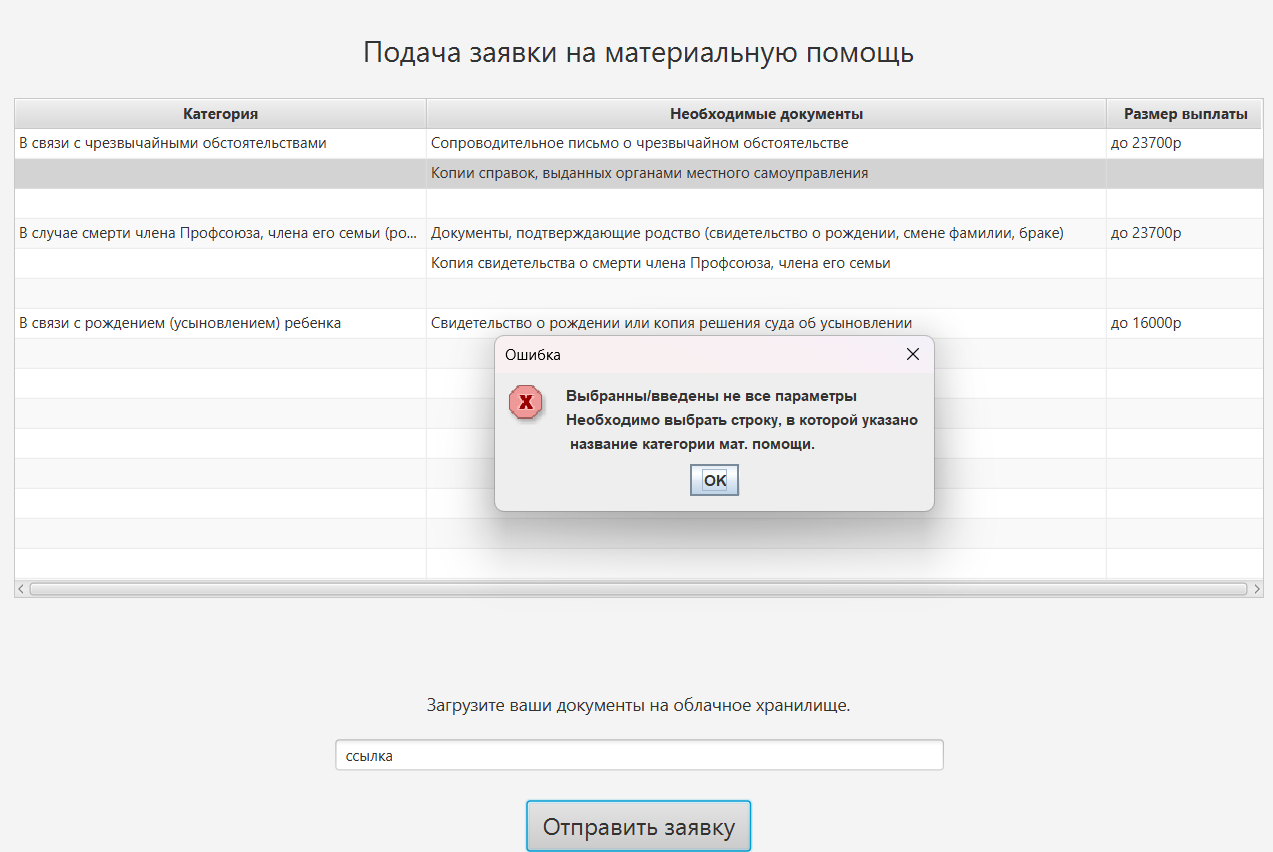
Автоматически созданное описаниеЛичный кабинет Петра:

Отправленные заявки отображаются корректно.

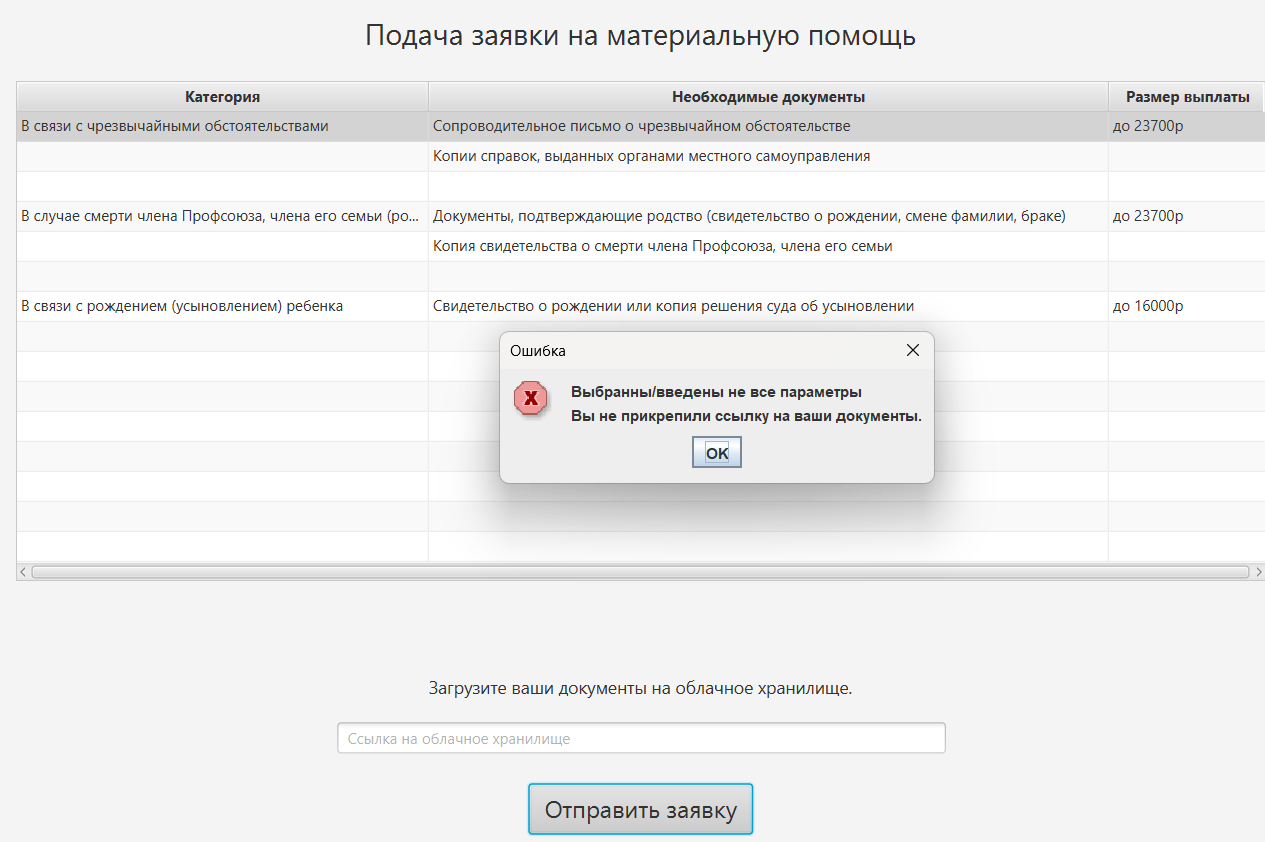
## Отправка новой заявки

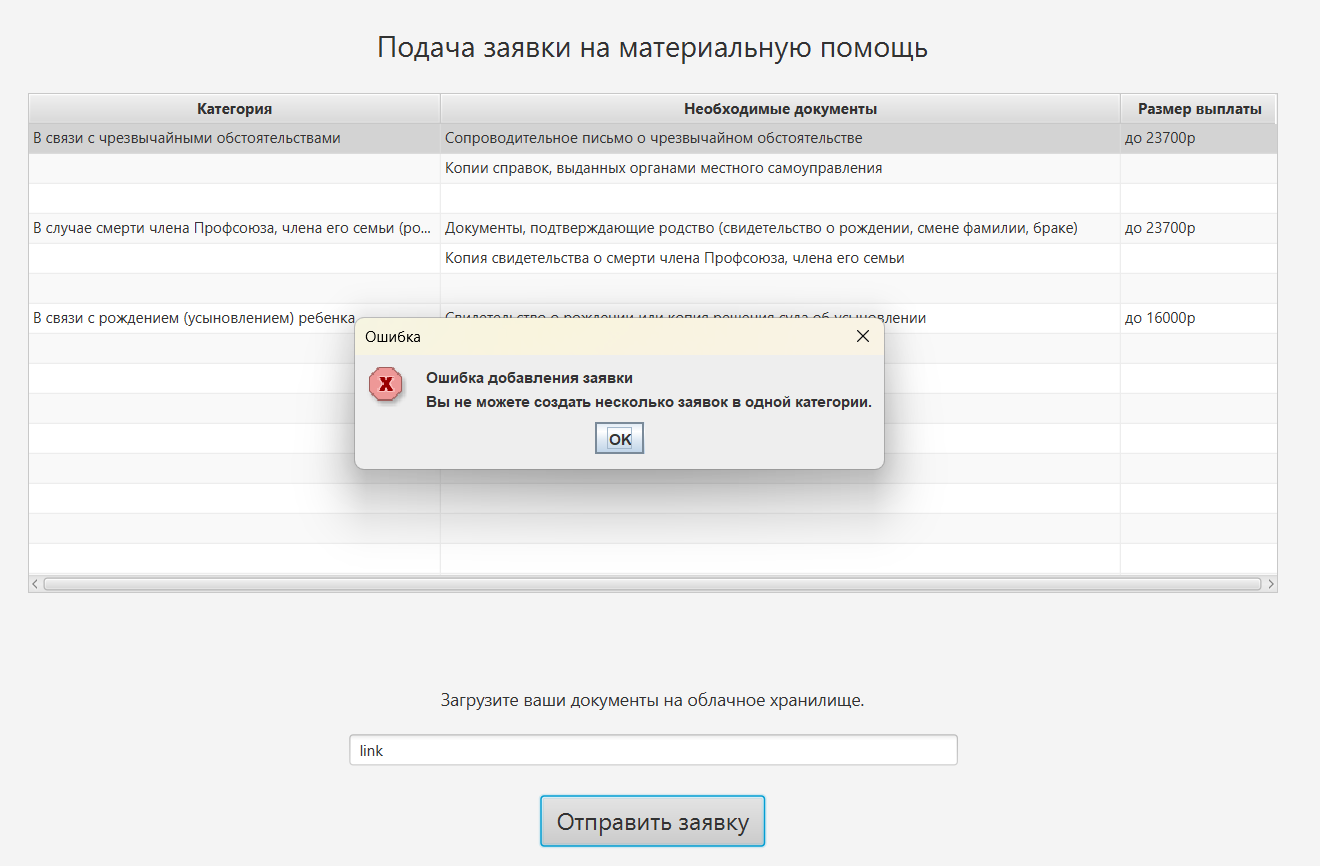
Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеМеню отправки заявки:

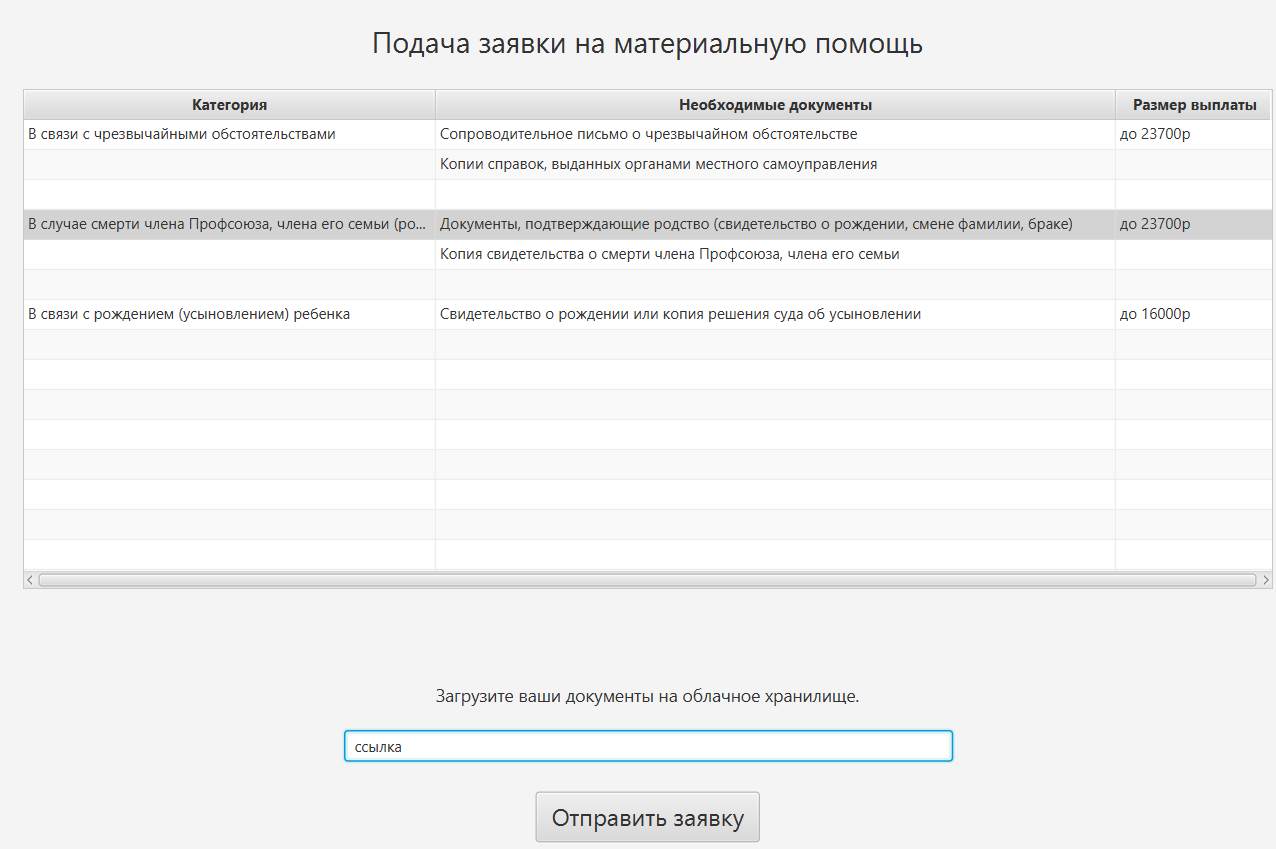
При выборе неполной строки должно появиться сообщение об ошибке выбора.

Сообщение появилось, как и ожидалось.

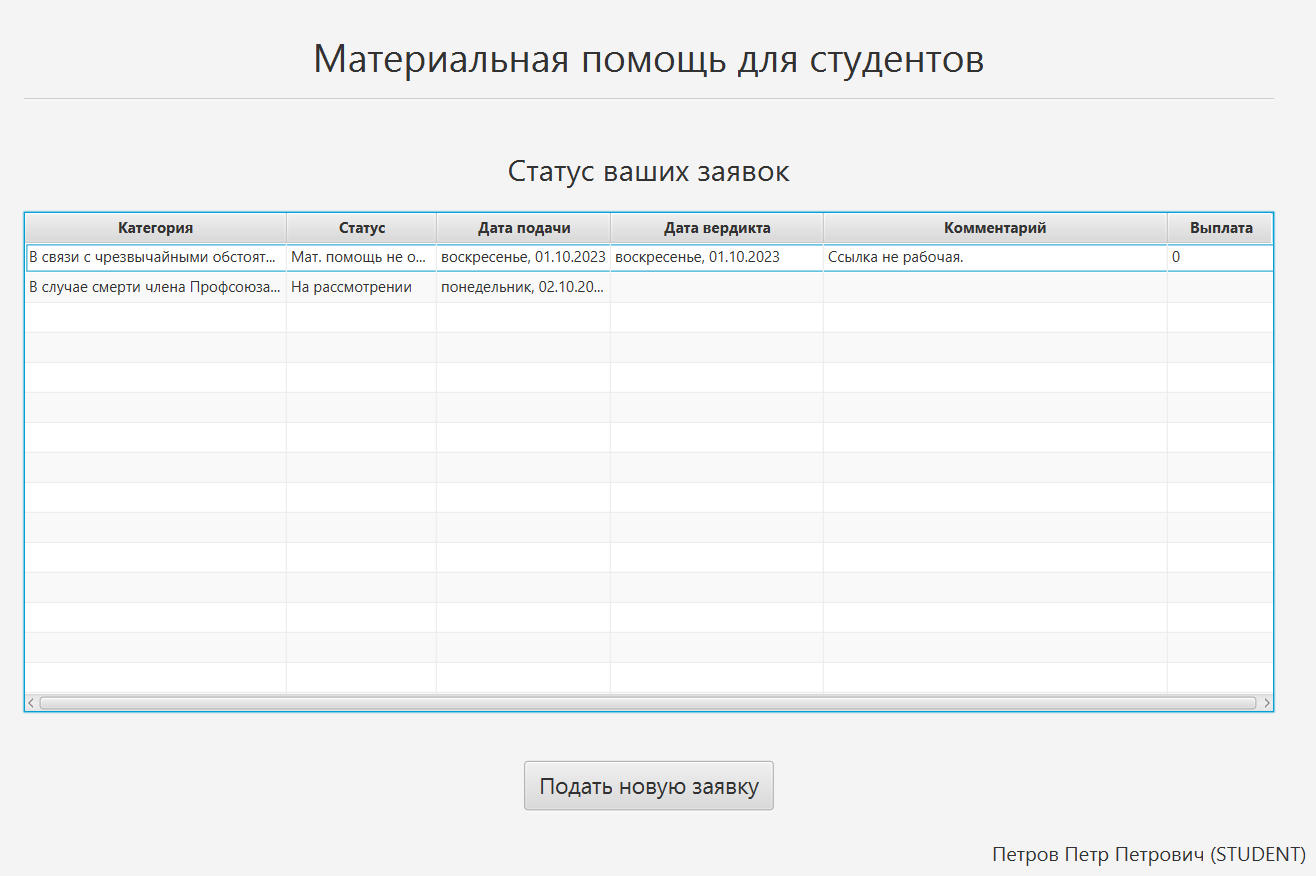
При попытке отправить заявку с пустым полем для ссылки также всплывет предупреждение:

При попытке отправить еще одну заявку в категории, в которой ранее уже была отправлена заявка, должно появиться предупреждение:

Сообщения выводятся по плану.

Правильная отправка заявки:

## Отображение отправленной заявке в списке отправленных

В списке отправленных заявок появилась новая заявка:

Тест успешно пройден.

## Изображение выглядит как текст, снимок экрана, Шрифт, число Автоматически созданное описаниеОтображение списка заявок студента в личном кабинете администратора

Заявка корректно отображается.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

## Рассмотрение выбранной заявки администратором

По

После отправки ответа заявка должна пропасть из списка заявок на рассмотрение у администратора и ее статус должен обновиться в личном кабинете студента.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описаниеРассмотрение заявки отработало, как и ожидалось.

## Вывод

Приложение протестировано по составленному чек-листу. В ходе тестирования имеющийся функционал работал корректно. Тестирование завершено успешно.

# Доказательство удовлетворения НФБК

Для доказательства, что данная схема базы данных соответствует Нормальной Форме Бойса-Кодда (BCNF), нам нужно проверить, что все функциональные зависимости в базе данных удовлетворяют правилам BCNF:

BCNF требует, чтобы каждая нетривиальная функциональная зависимость (то есть зависимость, в которой атрибуты на правой стороне не являются частью ключа) имела ключ как свой левый атрибут. Другими словами, если X -> Y - функциональная зависимость, то X должно быть суперключом (ключом или частью ключа). Также это должно выполняться для каждой нетривиальной зависимости в каждой таблице.

* Таблица `institutes` содержит поля `id` и `name`. Поле `id` является первичным ключом. Это удовлетворяет требованиям первой нормальной формы (1НФ), так как все столбцы являются атомарными и уникальными.
* Таблица `specialties` содержит поля `id`, `name`, и `institute\_id`. Поле `id` является первичным ключом. Поле `institute\_id` ссылается на таблицу `institutes`, что создает связь между этими двумя таблицами. Это также удовлетворяет требованиям 1НФ и 2НФ, так как оно имеет первичный ключ и все столбцы зависят только от первичного ключа.
* Таблица `admins` содержит поля `id`, `name`, `mail`, и `pswd`. Поле `id` является первичным ключом. Поле `mail` имеет уникальное ограничение, что удовлетворяет требованиям 1НФ и 2НФ. Таблица не содержит множественных значений, и все столбцы зависят от первичного ключа.
* Таблица `students` содержит поля `id`, `name`, `mail`, `pswd`, и `specialty\_id`. Поле `id` является первичным ключом. Поле `mail` также имеет уникальное ограничение, что удовлетворяет требованиям 1НФ и 2НФ. Таблица имеет внешний ключ `specialty\_id`, который ссылается на таблицу `specialties`, обеспечивая связь между студентами и их специальностями.
* Таблица `required\_documents` содержит поля `id` и `description`. Поле `id` является первичным ключом. Эта таблица также удовлетворяет 1НФ и 2НФ, так как все столбцы атомарны и уникальны.
* Таблица `fa\_categories` содержит поля `id`, `title`, и `payment\_amount`. Поле `id` является первичным ключом. Эта таблица также соответствует требованиям 1НФ и 2НФ, так как она имеет первичный ключ и все столбцы зависят от него.
* Таблица `required\_documents\_list` связывает таблицы `required\_documents` и `fa\_categories`. Она содержит поля `id`, `required\_document\_id`, и `fa\_category\_id`, которые являются внешними ключами, ссылаясь на соответствующие таблицы. Это обеспечивает связь между документами и категориями, и удовлетворяет требованиям 1НФ и 2НФ.
* Таблица `fa\_requests` содержит поля `id`, `filling\_date`, `response\_date`, `conf\_doc\_link`, `request\_status`, `admin\_coment`, `payment\_amount`, `admin\_id`, `student\_id`, и `fa\_category\_id`. Она имеет первичный ключ `id` и несколько внешних ключей, которые связаны с другими таблицами. Это обеспечивает связи между заявками, администраторами, студентами и категориями. Эта таблица также удовлетворяет 1НФ и 2НФ.

Таким образом, данная база данных удовлетворяет требованиям НФБК, так как она находится в 1НФ и 2НФ, и все ее столбцы атомарны и зависят только от первичных ключей или внешних ключей.