  
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«Дальневосточный федеральный университет»ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ  
Департамент программной инженерии и искусственного интеллекта**

**ОТЧЕТ**  
по лабораторной работе №2.3 на тему  
«Проектирование и разработка базы данных (MySQL)   
в выбранной предметной области»  
по дисциплине «Проектирование и разработка баз данных»  
Специальность 09.03.04 «Программная инженерия»

Выполнили:  
студенты гр. Б9119-09.03.04прогин  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Агапова Д.Р.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Сорокина П.Д.

г. Владивосток  
2022

**Содержание**

[Задачи 3](#_Toc106463552)

[1. Описание предметной области «Создание выкроек» 4](#_Toc106463553)

[2 Построение концептуальной модели 6](#_Toc106463554)

[3 Проектирование и разработка БД 8](#_Toc106463555)

[3.1. Процесс прямого проектирования БД на основе EER модели 8](#_Toc106463556)

[3.2. Перечень реализованных функций 8](#_Toc106463557)

[3.3. Выбор среды разработки 9](#_Toc106463558)

[4 Руководство пользователя. 10](#_Toc106463559)

[4.1. Основные элементы интерфейса 10](#_Toc106463560)

[4.1.1. LoginForm 10](#_Toc106463561)

[4.1.2. RegisterForm 12](#_Toc106463562)

[4.1.3. MainForm 14](#_Toc106463563)

[4.1.4. CategoryForm 15](#_Toc106463564)

[4.1.5. PatternListForm 17](#_Toc106463565)

[4.1.6. PrivateForm 19](#_Toc106463566)

[4.1.7. FavoritesListForm 21](#_Toc106463567)

[4.1.8. DataChangeForm 23](#_Toc106463568)

[4.2. Пример работы приложения 26](#_Toc106463569)

[Приложение 1. Прямое проектирование 41](#_Toc106463570)

[Приложение 2. Реализованные функции 44](#_Toc106463571)

# Задачи

1. Провести анализ предметной области (по выбранному варианту).
2. Построить концептуальную модель (ER-модель).
3. Преобразовать ER-модель в реляционную модель в СУБД MySQL.
4. Создать скрипт, выполняющий первоначальную загрузку данных в БД (наполнение справочных таблиц, с этими данными система будет передана заказчику).
5. Создать скрипт, наполняющий БД модельными данными (для отладки системы, демонстрации работы заказчику).
6. Реализовать все функции, которыми должна обладать ваша БД (запросы, процедуры, функции).
7. Создать диалоговый интерфейс взаимодействия БД с пользователями (в выбранной среде разработки).
8. Написать отчет по лабораторной работе, согласно требованиям.

1. Описание предметной области «Создание выкроек»

Требуется разработать программную систему для автоматизации работы создания выкроек по собственным параметрам тела и построению 3D-модели. Это идеальный конечный результат.

Поэтому данная работа представлена первым этапом, а именно: разработка информационной системы для автоматизации работы с каталогом готовых выкроек и личным кабинетом пользователя.

Необходимо учесть следующие зависимости между объектами предметной области и их характеристиками:

* Регистрация невозможна с логином, который уже зарегистрирован.
* У логина пользователя может быть только один пароль, один набор личных параметров, один список избранного.
* Список избранного не может содержать одинаковые выкройки.
* В списке избранного выкройка может содержать только одну заметку.
* Каталог выкроек не может содержать одинаковые категории выкроек.
* Каталог выкроек не может содержать одинаковых выкроек.
* Категория выкроек может содержать несколько выкроек, но не может содержать одинаковых выкроек.
* У каждой выкройки может быть только один номер, одна категория, одно название один набор основных мерок и один набор дополнительных мерок.

Система будет использоваться пользователем и администратором.

При работе с системой **Пользователь** должен иметь возможность решать следующие задачи:

1. Авторизация в системе.

2. Регистрация в системе.

3. Просмотр своей личной учетной записи.

4. Редактирование логина.

5. Редактирование пароля.

6. Добавление и редактирование личных параметров.

7. Просмотр списка избранного.

8. Добавление выкроек в список избранного.

9. Удаление выкроек из списка избранного.

10. Добавление и редактирование заметок у выкроек в списке избранного.

11. Просмотр всего каталога выкроек.

12. Удаление аккаунта.

При работе с системой **Администратор** должен иметь возможность решать следующие задачи:

1. Просмотр всех учетных записей пользователей.

2. Изменение данных любой учетной записи пользователя.

3. Удаление любой учетной записи пользователя.

4. Просмотр всего каталога выкроек.

5. Добавление выкройки в каталог выкроек.

6. Редактирование любой выкройки в каталоге выкроек.

7. Удаление любой выкройки из каталога выкроек.

8. Добавление и редактирование любой категории в каталоге выкроек.

9. Удаление любой категории из каталога выкроек.

# 2 Построение концептуальной модели

Необходимо построить такую базу данных, в которой хранится информация о категориях выкроек, самих выкройках, их 3D-моделях и математических формулах, а также о пользователях и их избранных выкройках.

Таким образом, можно выделить следующие сущности с их атрибутами:

1) Выкройки (для каталога):

* номер выкройки [INT];
* название [VARCHAR(60)];
* категория [VARCHAR(45)];
* мерки [MEDIUMTEXT];

2) 3D-модели

* номер 3D-модели [INT];
* номер выкройки [INT] (FK) (1:1);
* 3d-модель по стандартным меркам [JSON];

3) Пользователи:

* логин [VARCHAR(20)];
* пароль [VARCHAR(30)];
* личные параметры [VARCHAR(100)] (1:N);

4) Избранное пользователей:

* + логин [VARCHAR(20)] (FK) (N:N);
  + номер выкройки [INT] (FK) (N:N) (допустимо Null-значение);

5) Математические формулы:

* + номер математической формулы [INT];
  + формула [MEDIUMTEXT];

6) Формулы выкроек:

* + номер выкройки [INT] (FK) (N:N);
  + номер математической формулы [INT] (FK) (N:N).

Результатом построения концептуальной модели является ER-диаграмма, приведенная на рис. 1.

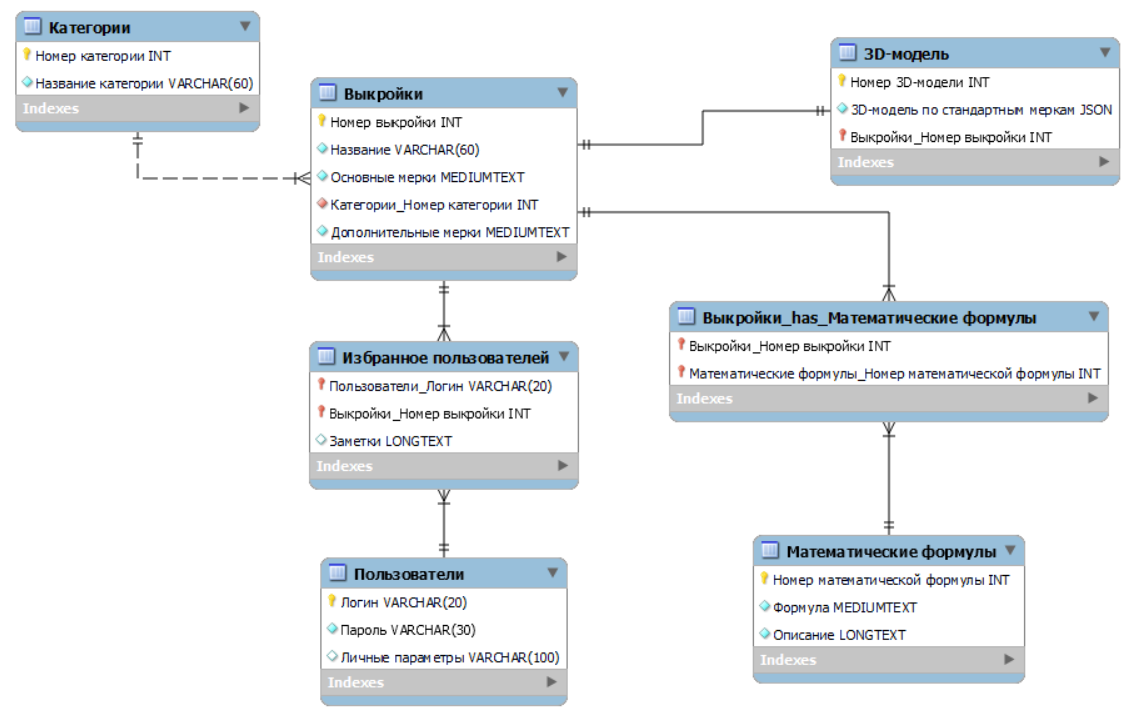


Рис. 1. ER-диаграмма

# 3 Проектирование и разработка БД

## Процесс прямого проектирования БД на основе EER модели

Для проектирования базы данных был использован MySQL Workbench и инструмент Forward Engineer, предоставляющий возможность сгенерировать скрипт создания базы данных из ER-диаграммы. Полученный скрипт, сгенерированный на основе созданной схемы (рис. 1), можно найти в приложении 1.

## Перечень реализованных функций

Все описанные далее запросы можно найти в приложении 2.

Запрос 1. Заполнение справочника категорий выкроек первичными данными.

Запрос 2. Заполнение справочника выкроек первичными данными.

Запрос 3. Заполнение справочника математических формул выкроек первичными данными.

Запрос 4. Заполнение справочника 3D-моделей выкроек первичными данными.

Запрос 5. Заполнение таблицы математических формул каждой выкройки первичными данными.

Запрос 6. Заполнение таблицы пользователей первичными данными.

Запрос 7. Заполнение таблицы избранное пользователей первичными данными.

Запрос 8. Поиск пользователя по логину в системе.

Запрос 9. Вывод общего числа всех зарегистрированных пользователей и общего числа выкроек в каталоге.

Запрос 10. Добавление в таблицу пользователей нового (зарегистрировавшегося) пользователя.

Запрос 11. Вывод всех выкроек из справочника выкроек по определенной категории.

Запрос 12. Добавление выкройки в список избранного пользователя.

Запрос 13. Изменение личных данных пользователя в таблице пользователей.

Запрос 14. Вывод списка избранного конкретного пользователя.

Запрос 15. Удаляет выбранную выкройку из списка избранного конкретного пользователя.

Запрос 16. Обновляет заметки к выбранной выкройки в списке избранного конкретного пользователя.

Запрос 17. Удаляет конкретного пользователя из системы.

Так как база MySql будет подключена через локальный хост, то администратор будет иметь доступ ко всем своим функциям через сервер.

## Выбор среды разработки

Информационная система реализована в интегрированной среде разработки (IDE) – Microsoft Visual Studio, на платформе .NET.

Данный продукт позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, с помощью которой был разработан интерфейс для работы с базой данных.

Такой выбор обосновывается следующими факторами:

• содержит более единую модель программирования;

• создает более однородную структуру в программной модели и устраняет многие ошибки и противоречия от использования Windows API;

• поддерживает использования любого языка программирования;

• позволяет разрабатывать интеллектуальные клиенты;

*Прим. Интеллектуальный клиент — это приложение с полнофункциональным графическим интерфейсом, простое в развертывании и обновлении, способное работать при наличии или отсутствии подключения к Интернету и использующее более безопасный доступ к ресурсам на локальном компьютере.*

• предоставляет гибкий элемент управления с именем DataGridView для отображения таких табличных данных в традиционном формате строк и столбцов так, что каждый фрагмент данных занимает свою собственную ячейку.

Для выполнения системой запросов к базе данных использовано бесплатное программное обеспечение MAMP, которое устанавливает локальную серверную среду для тестирования и разработки. На нее выгружается вся предварительно написанная база данных, а затем уже с помощью включенной библиотеки «MySql.Data.MySqlClient» в среде разработки появляется возможность отправлять SQL-запросы прямо с программы.

# 4 Руководство пользователя.

Система сконструирована в виде нескольких окон, переход между которыми пользователь осуществляет с помощью выполнения определенных действий.

## Основные элементы интерфейса

* + 1. LoginForm

Окно авторизации (рис. 2). Первое окно, которое открывается при загрузке системы.

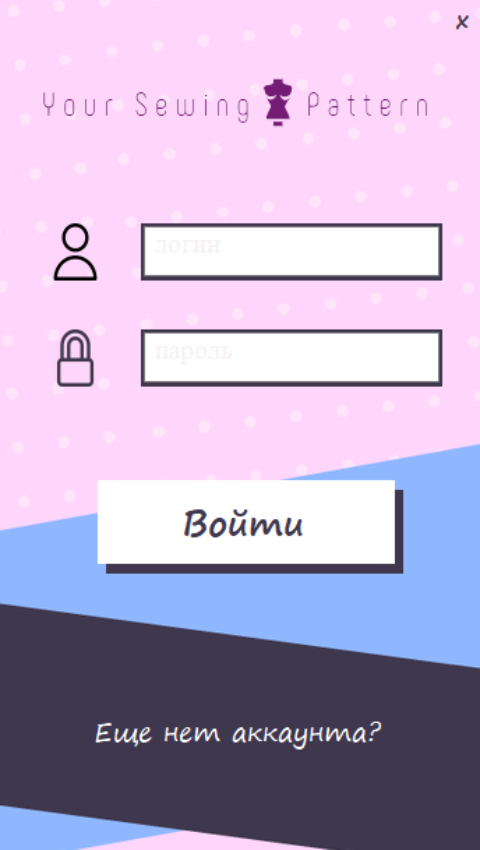


Рис. 2. LoginForm

Основные элементы управления представлены на рис. 3.

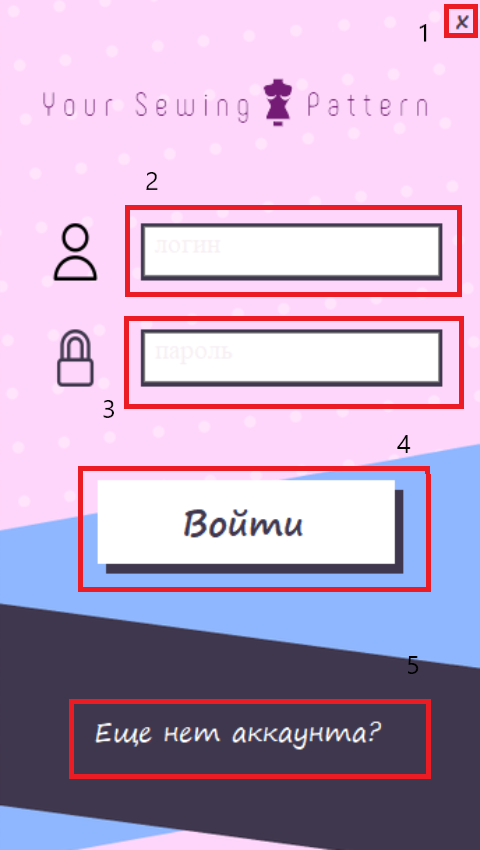
****

Рис. 3. Основные элементы управления в окне LoginForm

Поля для ввода:

* логина (2);
* пароля (3).

Кнопки:

* кнопка закрытия приложения (1);
* авторизация и вход в приложение под указанным логином (2) и паролем (3);
* переход на окно регистрации (5).
  + 1. RegisterForm

Окно регистрации (рис. 4). Чтобы воспользоваться функционалом системы, пользователю необходимо зарегистрироваться, т.е. ввести данные, которые затем будут указываться при авторизации.

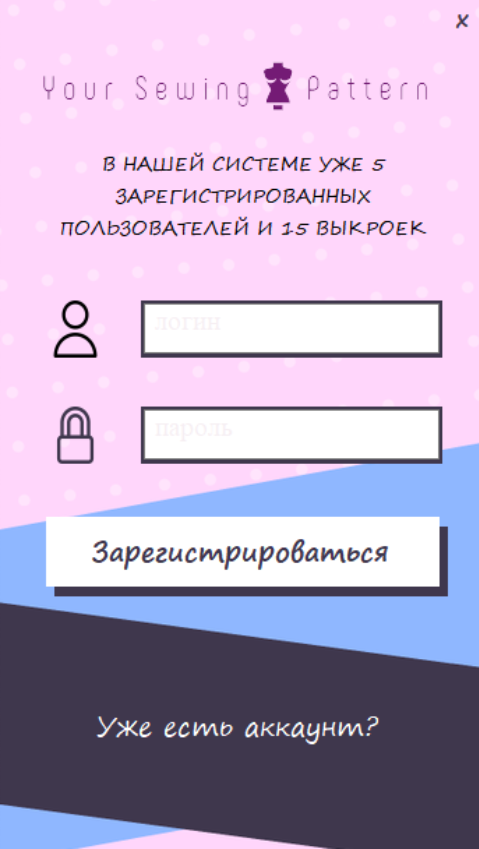


Рис. 4. RegisterForm

Основные элементы управления представлены на рис. 5.

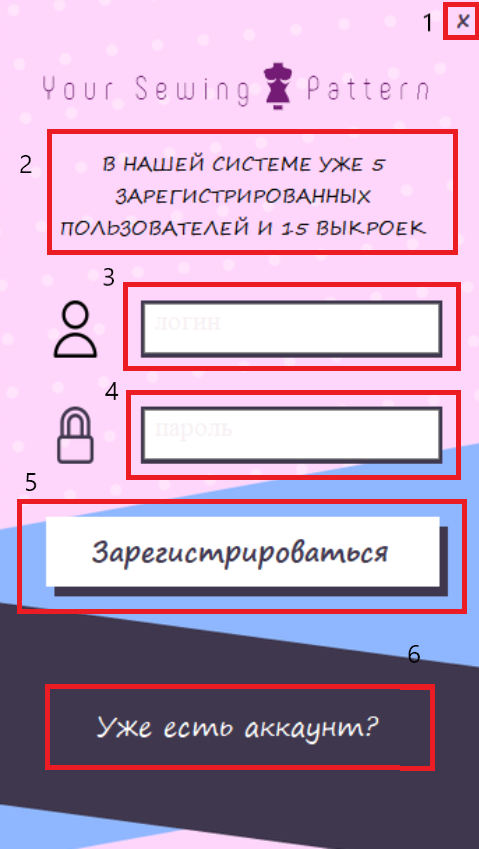
****

Рис. 5. Основные элементы управления в окне RegisterForm

Поля для ввода:

* логина (3);
* пароля (4).

Поле для вывода:

* информации об общем количестве пользователей системы и выкроек в каталоге (2).

Кнопки:

* закрытие приложения (1);
* регистрация (сохранение данных о пользователе в системе) и переход на окно авторизации (5);
* переход на окно авторизации (6).
  + 1. MainForm

Окно главной формы (рис. 6). Здесь для пользователя открывается выбор категории одежды, выкройки которой необходимо найти.

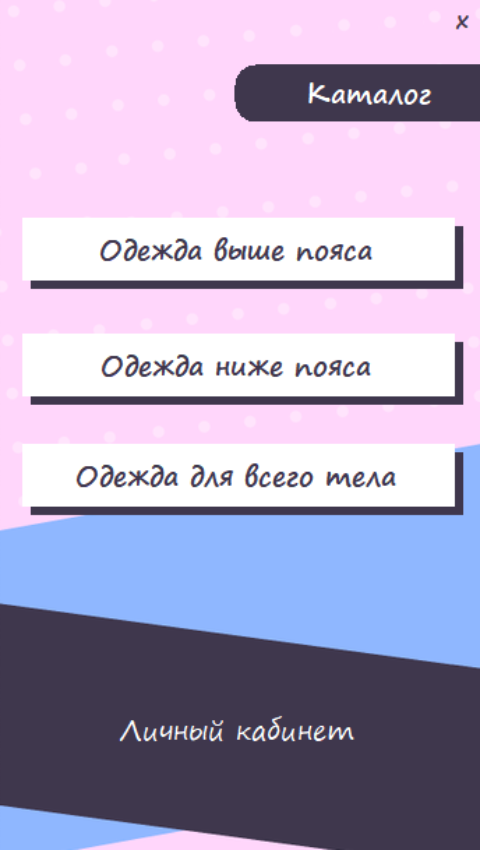


Рис. 6. MainForm

Основные элементы управления представлены на рис. 7.

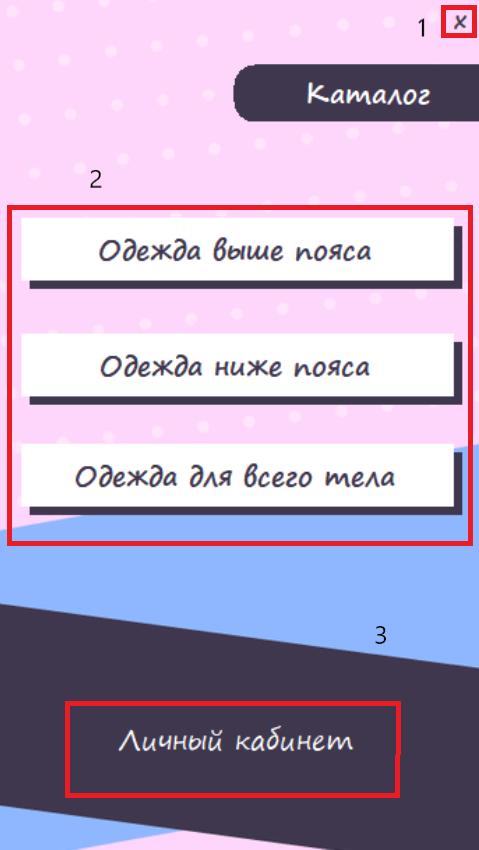
****

Рис. 7. Основные элементы управления в окне MainForm

Кнопки:

* закрытие приложения (1);
* выбор категории одежды (2);
* переход в личный кабинет (3).
  + 1. CategoryForm

Окно выбора категории выкройки (рис. 8). В данном окне пользователя предоставляется выбор категории выкроек.



Рис. 8. CategoryForm

Основные элементы управления представлены на рис. 9.

****

Рис. 9. Основные элементы управления в окне CategoryForm

Кнопки:

* закрытие приложения (1);
* возвращение на предыдущий экран – главную форму (2).
* выбор категории выкроек (3);
* переход в личный кабинет (4).
  + 1. PatternListForm

Окно всех выкроек по выбранной категории (рис. 10). Здесь пользователь может ознакомиться по выбранной категории с полным списком выкроек с прилагающейся информацией о них.

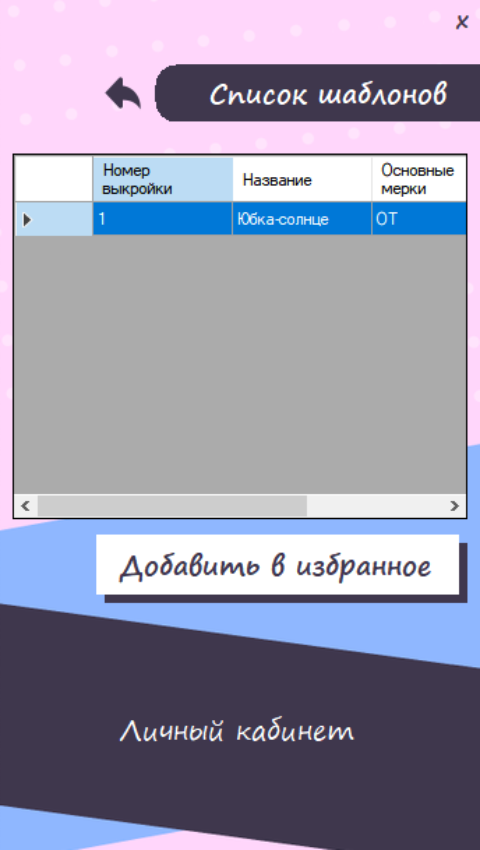


Рис. 10. PatternListForm

Основные элементы управления представлены на рис. 11.

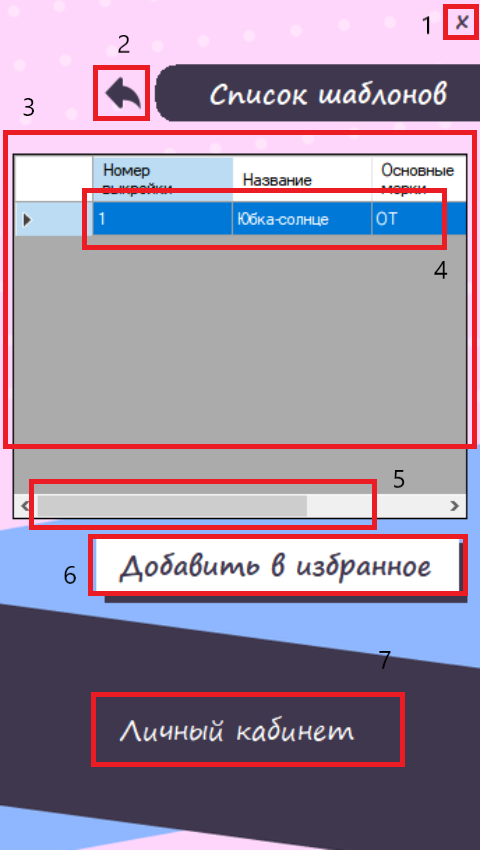
****

Рис. 11. Основные элементы управления в окне PatternListForm

Поле вывода:

* таблица, содержащая список всех выкроек (3).

Элемент выбора:

* строка в таблице (3) при нажатии становится активной (4).

*Прим. При начальном открытии формы первая строка является по умолчанию активной.*

Кнопки:

* закрытие приложения (1);
* возвращение на предыдущий экран – форму категорий выкроек (2);
* добавление выбранной выкройки в список избранного (6);
* переход в личный кабинет (7).

Полоса прокрутки (5), позволяющая посмотреть все содержимое таблицы (3).

После нажатия кнопки добавление в избранное (6), высвечивается специальное окно ввода, для добавления заметки к выбранной выкройке. Внешний вид этого окна представлен на рис 12.

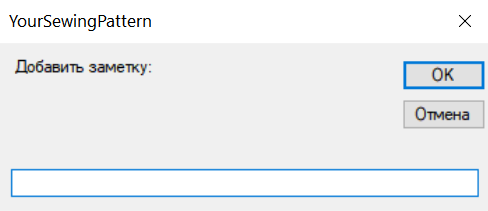


Рис. 12. Окно добавление заметки

* + 1. PrivateForm

Окно личного кабинета пользователя (рис. 13). Окно предназначено для взаимодействия пользователя с данными своего аккаунта.

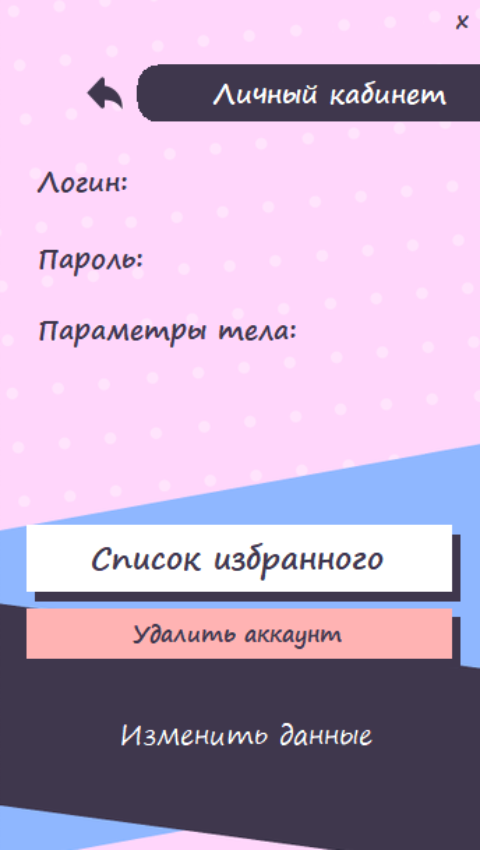


Рис. 13. PrivateForm

Основные элементы управления представлены на рис. 14.

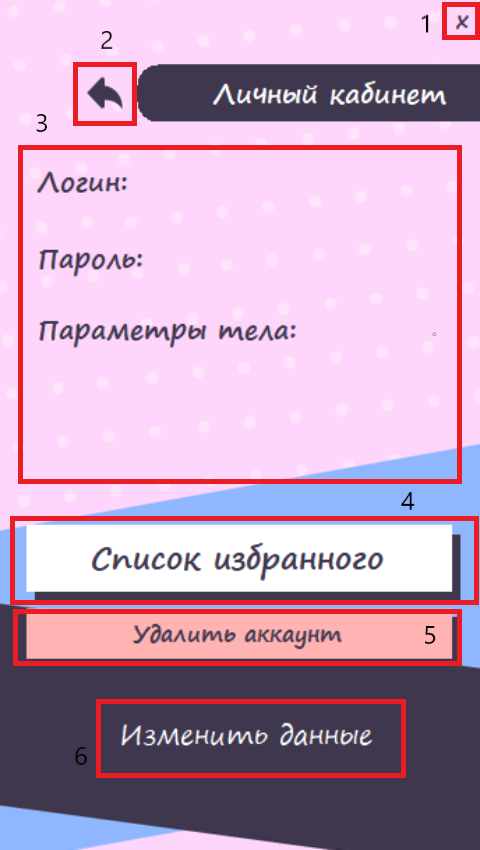
****

Рис. 14. Основные элементы управления в окне PrivateForm

Поле вывода:

* личной информации о пользователе (3).

Кнопки:

* закрытие приложения (1);
* возвращение на предыдущий экран – любая форма, с которой пользователь перешел (2);
* переход в список избранного (4);
* удаление аккаунта (5);
* переход на окно изменения данных (6).
  + 1. FavoritesListForm

Окно избранных выкроек пользователя (рис. 15). Предназначено для взаимодействия пользователя с данными своего списка избранного.

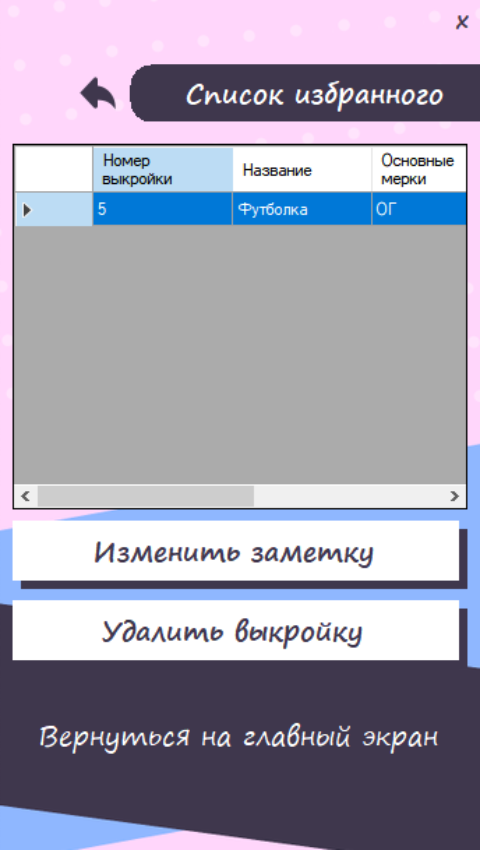


Рис. 15. FavoritesListForm

Основные элементы управления представлены на рис. 16.

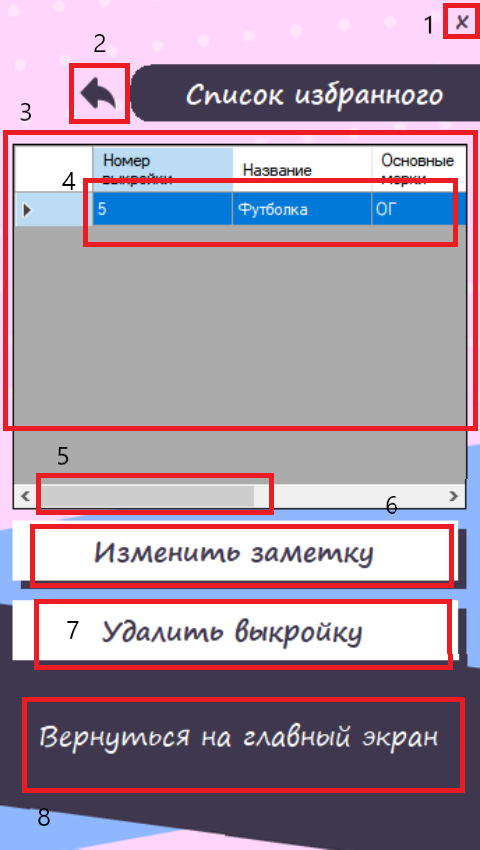
****

Рис. 16. Основные элементы управления в окне FavoritesListForm

Поле вывода:

* таблица, содержащая список всех избранных выкроек пользователя (3).

Элемент выбора:

* строка в таблице (3), при нажатии становится активной (4).

*Прим. При начальном открытии формы первая строка является по умолчанию активной.*

Кнопки:

* закрытие приложения (1);
* возвращение на предыдущий экран – форму личного кабинета пользователя (2);
* изменение заметки у выбранной выкройки (6);
* удаление выбранной выкройки из списка избранного (7);
* переход на главную форму – MainForm (8).

Полоса прокрутки (5), позволяющая посмотреть все содержимое таблицы (3).

После нажатия кнопки изменения заметки (6) высвечивается специальное окно ввода для изменения заметки к выбранной выкройке. Внешний вид этого окна представлен на рис 17.

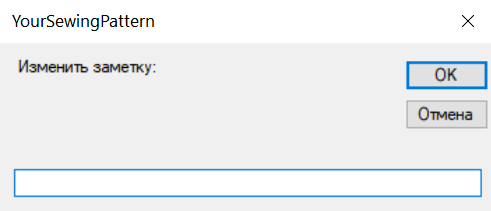


Рис. 17. Окно изменения заметки

* + 1. DataChangeForm

Окно изменения данных в личном кабинете (рис. 18). Выполняет функцию редактирование логина, пароля и личных параметров.

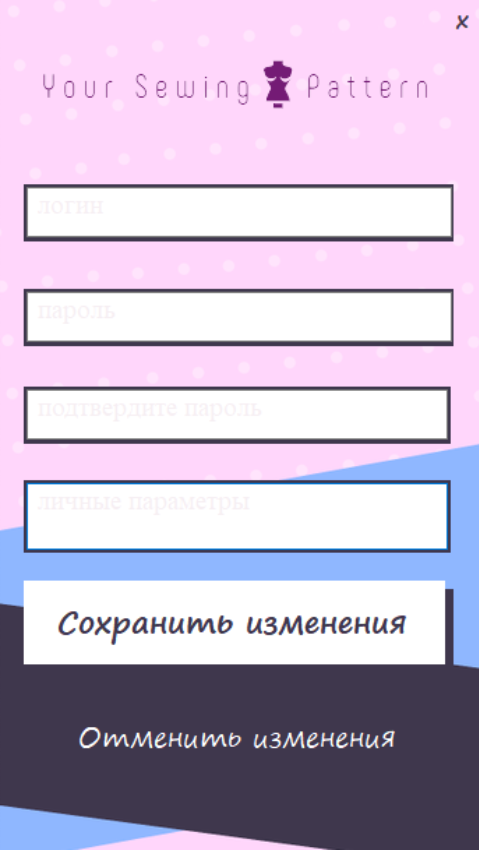


Рис. 18. DataChangeForm

Основные элементы управления представлены на рис. 19.

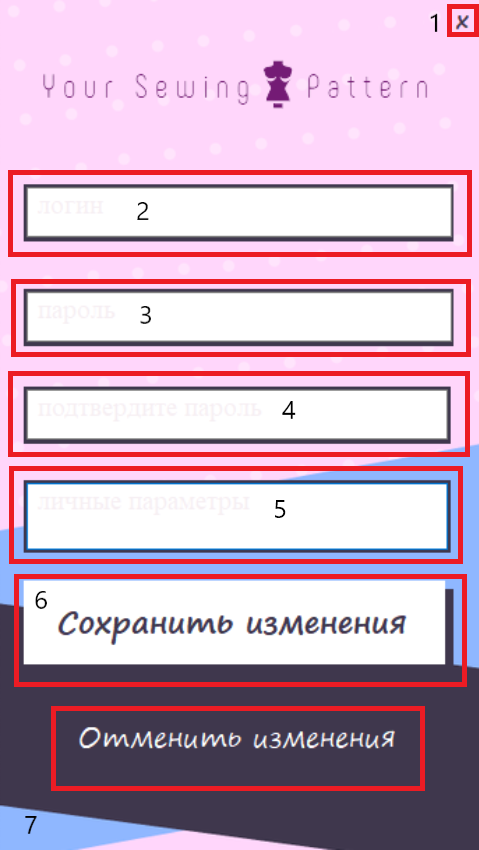
****

Рис. 19. Основные элементы управления в окне DataChangeForm

Поля вывода:

* логина (2);
* пароля (3);
* значений личных параметров (5).

*Прим. При открытии окна в полях вывода отображается информация о пользователе актуальная на данный момент.*

Поля ввода:

* логина (2);
* пароля (3);
* подтверждения пароля (4);
* значений личных параметров (5).

Кнопки:

* закрытие окна изменения данных (1);
* сохранение введенных данных (6);
* возвращение к первоначальным данным (7).

При наведении на поле личных параметров высвечивается информационное сообщение о том, в каком порядке вводить значения. Вид информационного сообщения приведено на рис. 20.

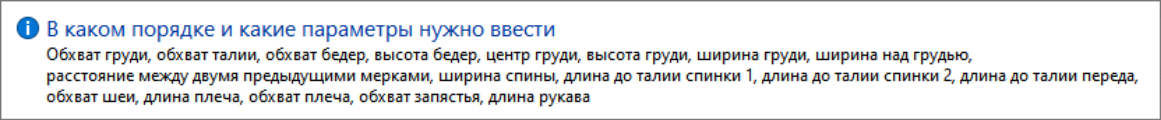


Рис. 20. Информационное сообщение о порядке ввода личных параметров

На рис. 21 показана общая схема переходов между окнами. Обычной стрелкой показана связь, при которой пользователь, перемещаясь к конечному окну, не имеет возможности вернуться обратно к начальному. Пунктирной стрелкой обозначена связь, при которой можно переключаться от одного окна к другому в любом порядке.

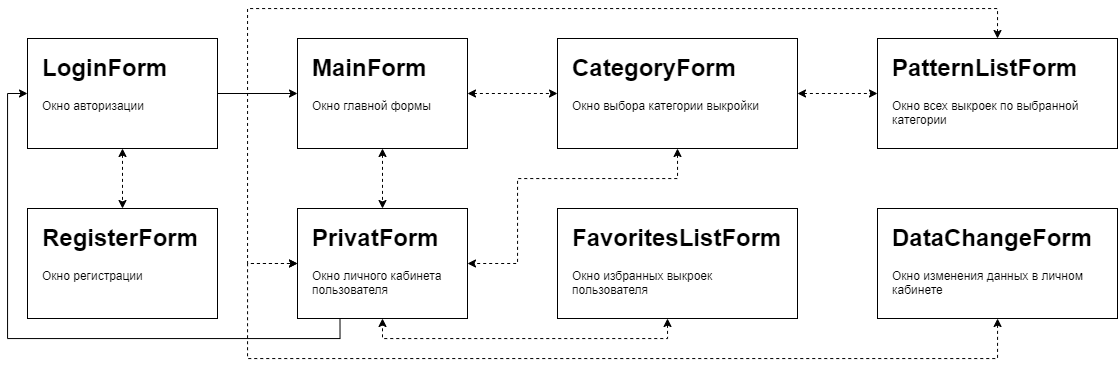


Рис. 21. Диаграмма связи между окнами

## Пример работы приложения

В данной главе рассмотрим последовательно работу всех функций системы для конкретного пользователя.

Для начала зарегистрируем нашего пользователя.

Так как аккаунта еще нет, то с открывшегося окна авторизации переключимся на окно регистрации, далее введем данные нашего пользователя и нажмем на кнопку «Зарегистрироваться», рис. 22.

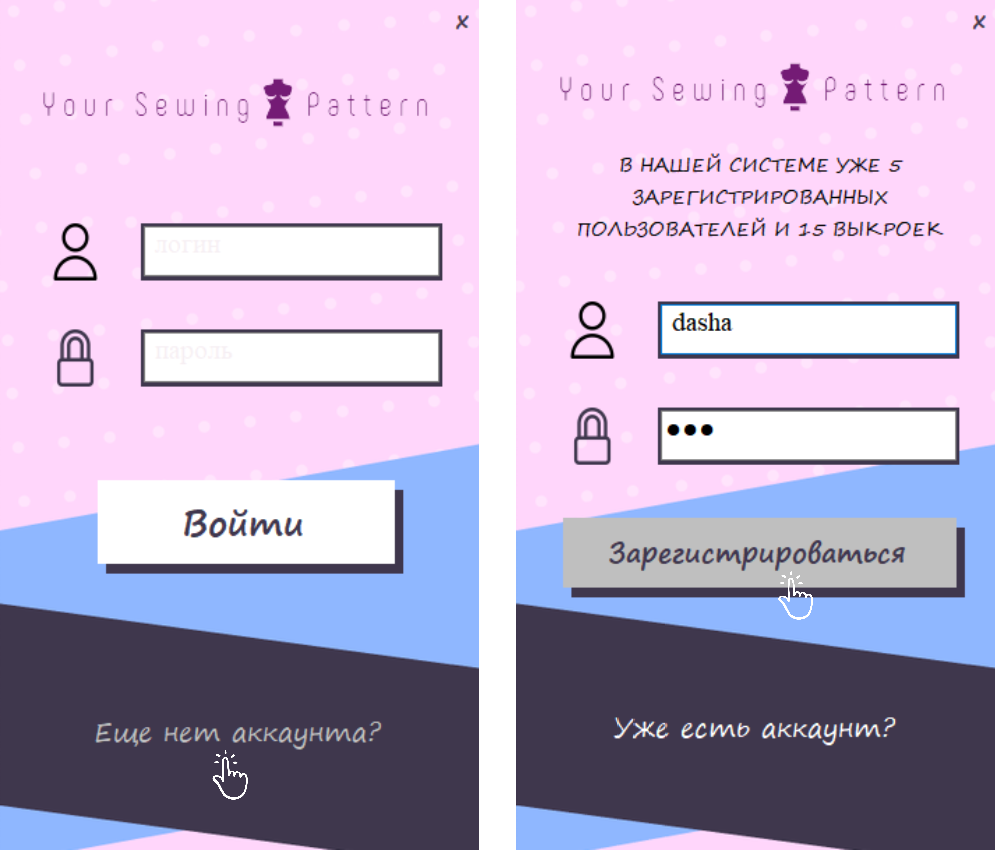


Рис. 22. Регистрация пользователя

После того как мы внесли данные пользователя в базу данных, откроется снова окно авторизации. Введем данные нашего только что зарегистрировавшегося пользователя и нажмем «Войти».

Появится главное окно с категориями одежды и доступом к личному кабинету. Посмотрим, что у нас есть в каталоге. Нажмем на нижнюю кнопку «Одежда для всего тела».

Высветится сообщение о том, что пока мы не можем посмотреть данную категорию, рис 23.



Рис. 23. Авторизация и главное окно

Посмотрим тогда другой тип одежды, а именно «Одежда ниже пояса». Откроется окно с последующим выбором категории выкроек. Перейдем сразу на вкладку «Шорты», рис. 24.



Рис. 24. Переключение по категориям

Пользователь увидит окно со списком всех доступных выкроек с возможностью добавить понравившуюся вещь в избранное.

Как только мы определились с выбором, высветится окно с возможностью ввода заметки к выкройке. Введем и посмотрим, что произойдет, рис. 25.

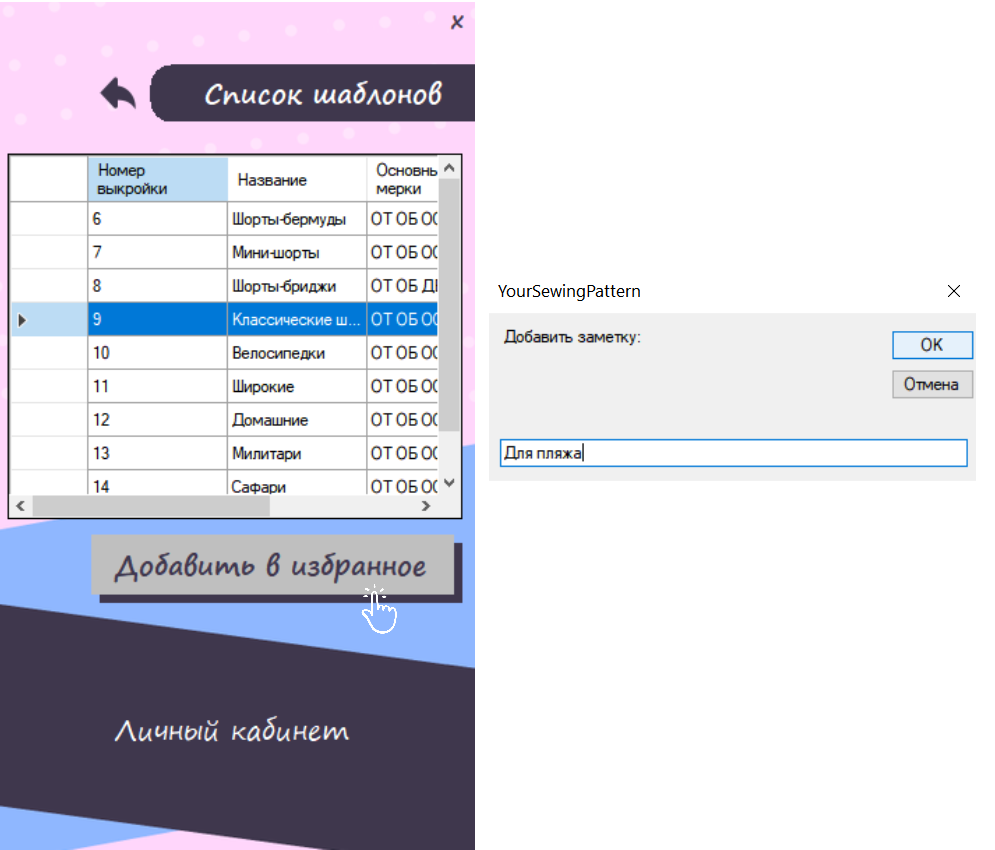


Рис. 25. Добавление выкройки в избранное

Но перед этим проверим кнопку переключения на предыдущее окно и уже с открывшегося окна перейдем в личный кабинет, рис 26.

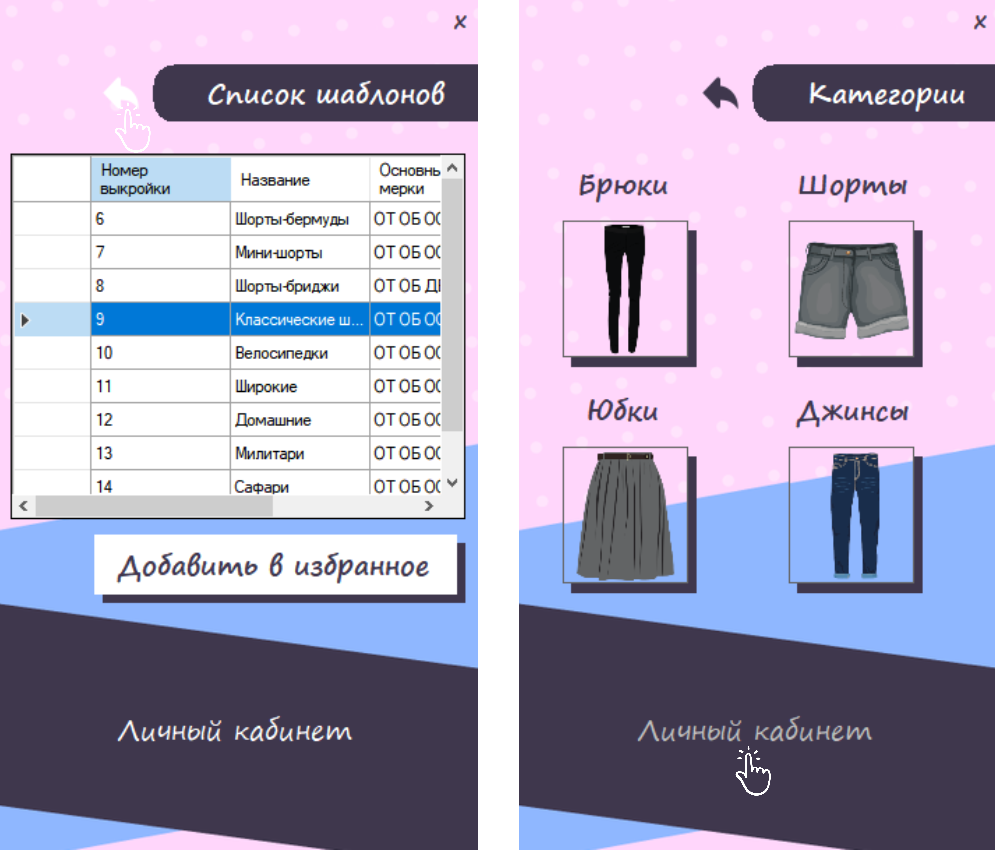


Рис. 26. Переключение между окнами

Первое, что мы увидим – это данные аккаунта пользователя (пока без параметров тела). Посмотрим список избранных, рис. 27.

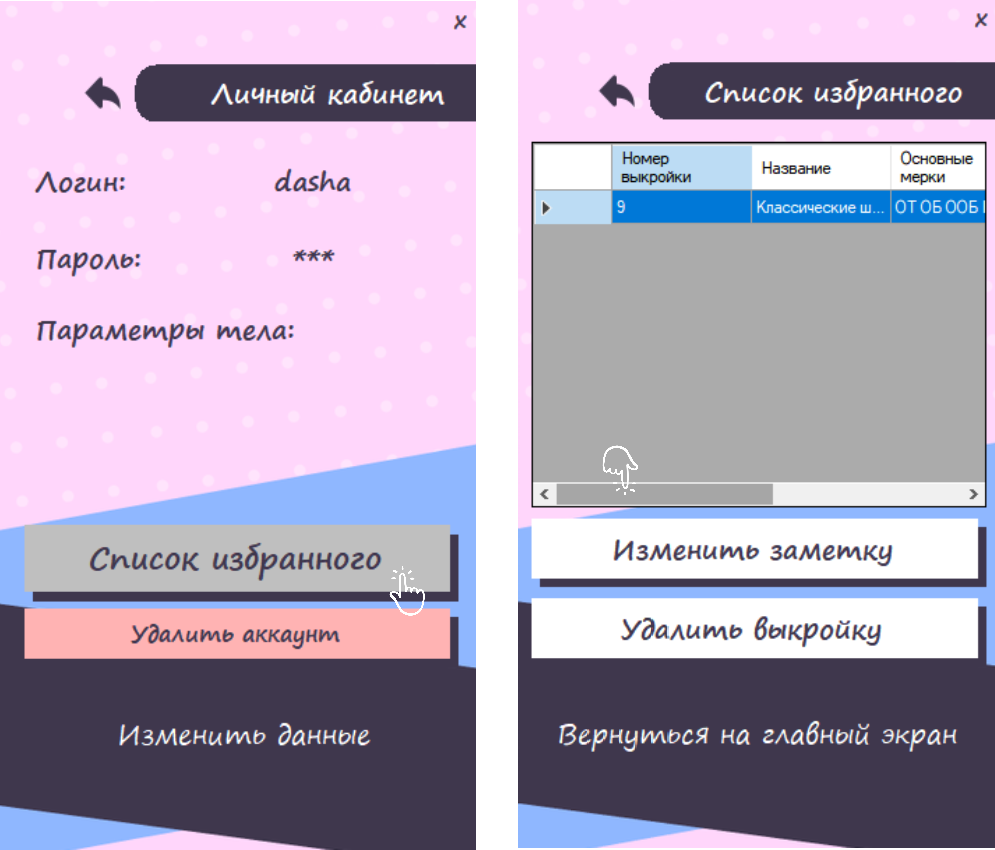


Рис. 27. Личный кабинет и список избранного

Перемещая полосу прокрутки, убеждаемся, что данные добавлены точно. Теперь можем подкорректировать заметки, рис. 28.

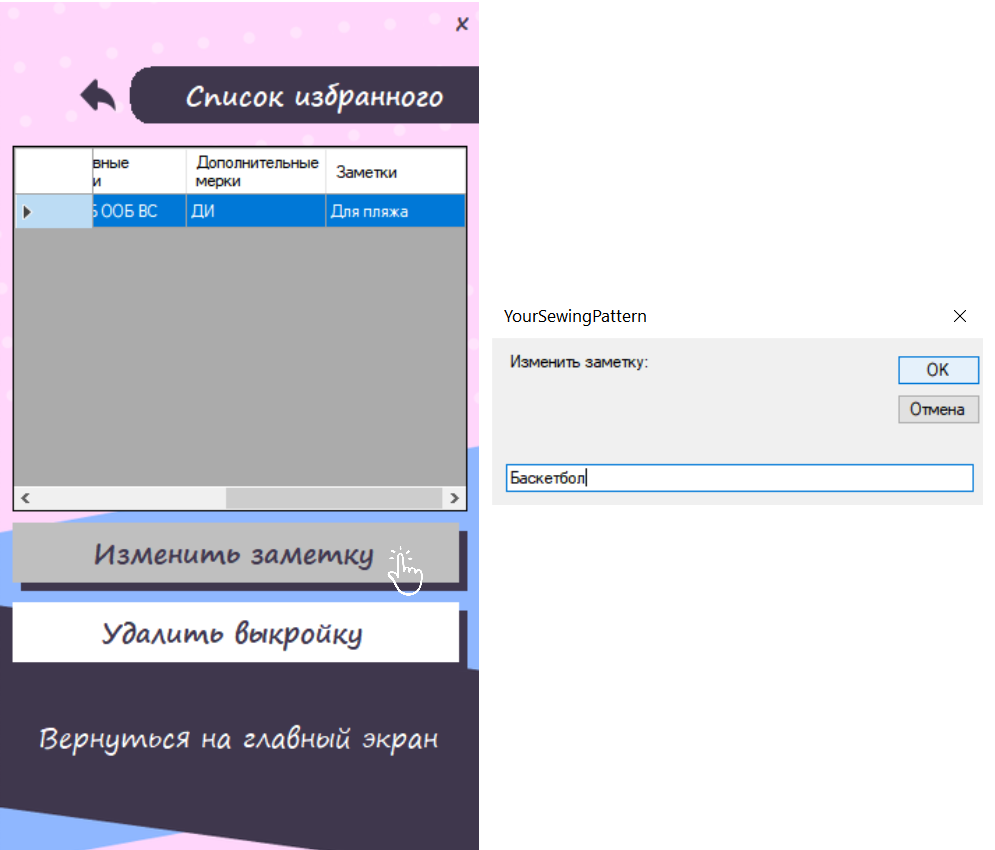


Рис. 28 Редактирование заметки

Все верно. А теперь удалим выкройку, рис. 29.

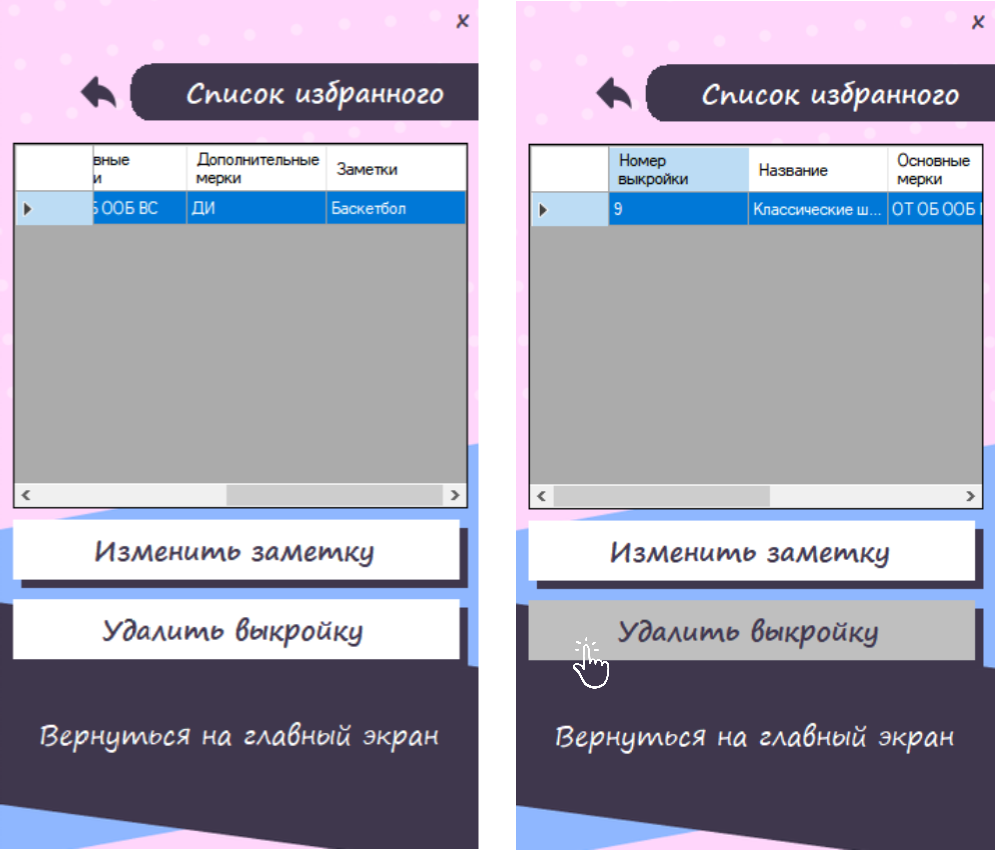


Рис. 29. Удаление выкройки

Список очистился. Вернемся на главную страницу, а затем опять в личный кабинет, рис. 30.

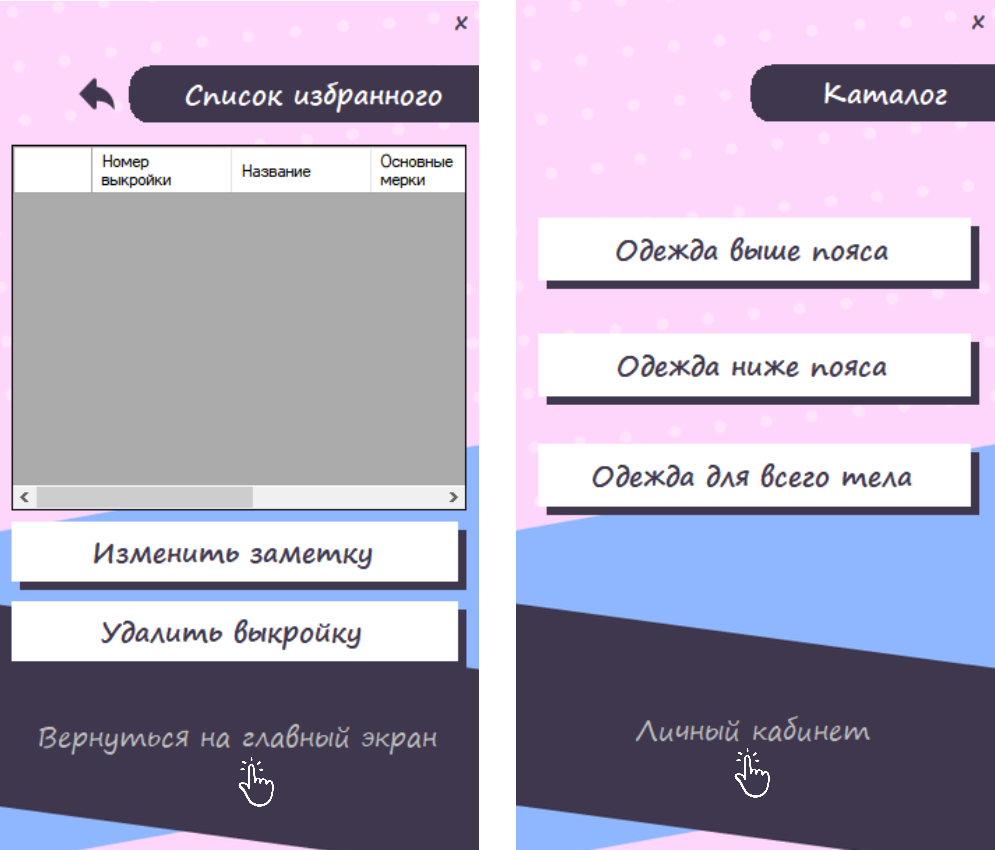


Рис. 30. Переход в личный кабинет с главной страницы.

У пользователя не хватает личных параметров. Добавим их, рис. 31.

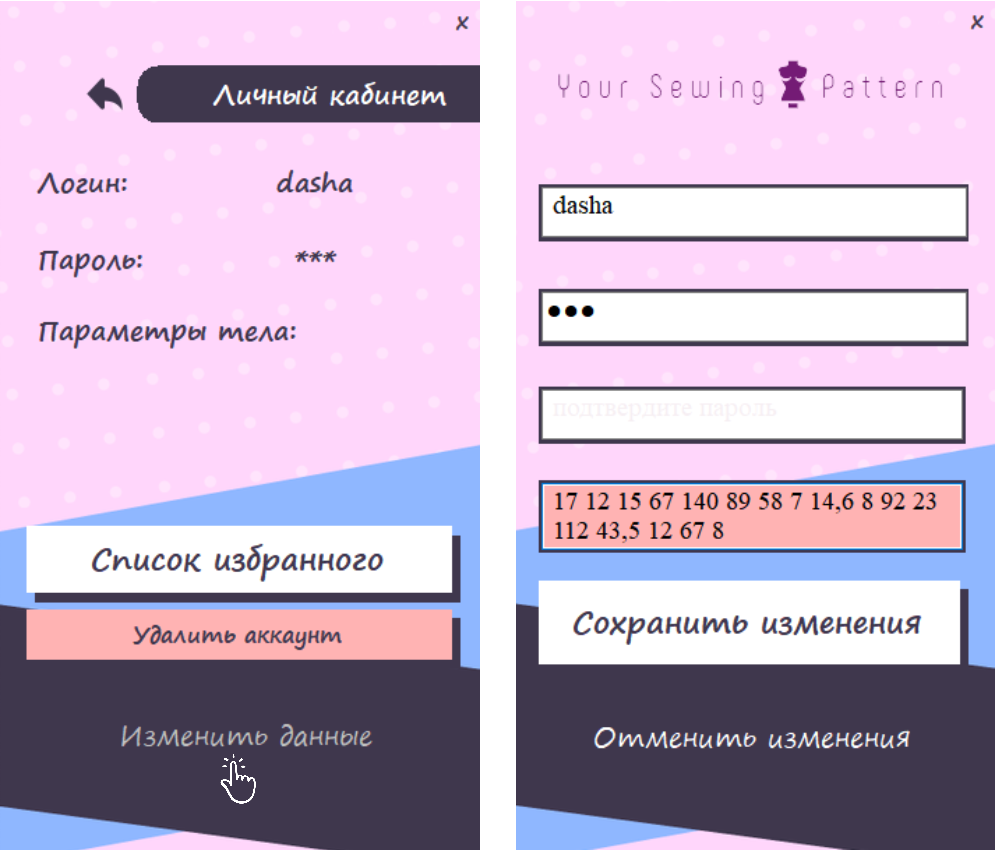


Рис. 31. Добавление параметров и проверка на их правильность

Неверные данные – не проблема, при наведении на данное поле будет высвечиваться информация о вводимых параметрах. Пропишем их в том порядке и количестве, в котором это необходимо. А затем попробуем изменить пароль.

Снова неверно заполнено. Но если ввести идентичный пароль в поле «Подтвердить пароль», то данные можно будет изменить, рис. 32.

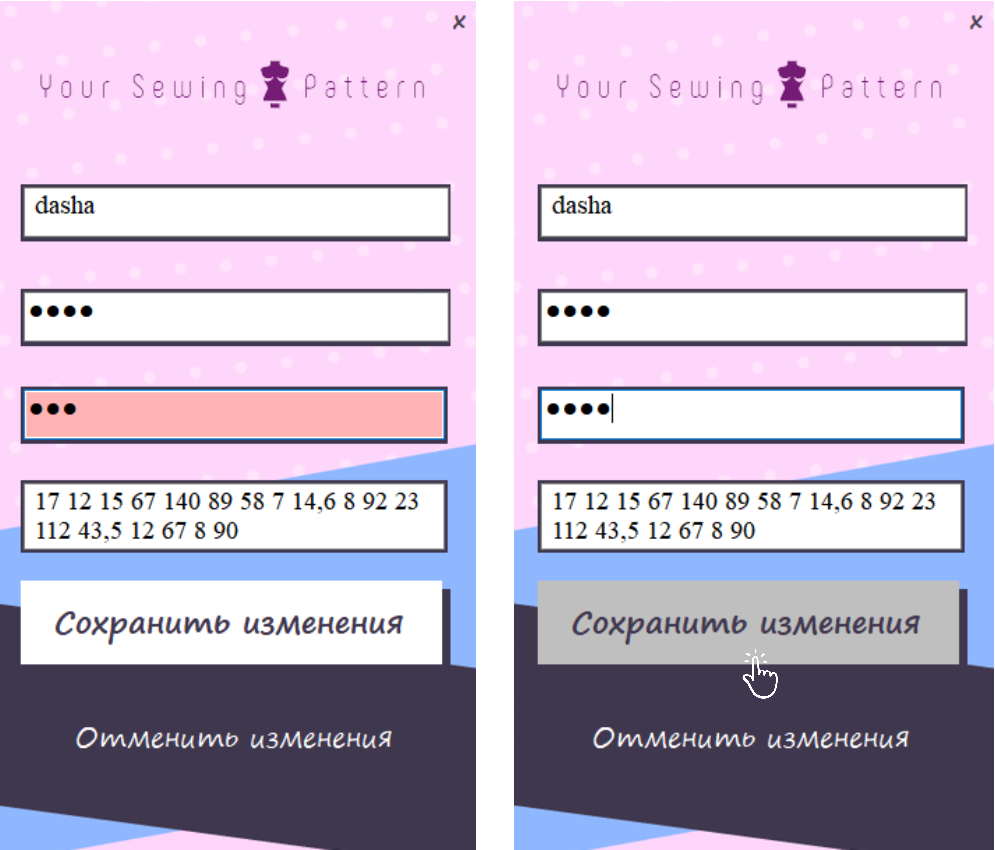


Рис. 32. Демонстрация изменения пароля

Дополнительно показана работоспособность кнопки «Отменить изменения». Она необходима, если пользователь ввел слишком много неправильной информации, рис. 33.

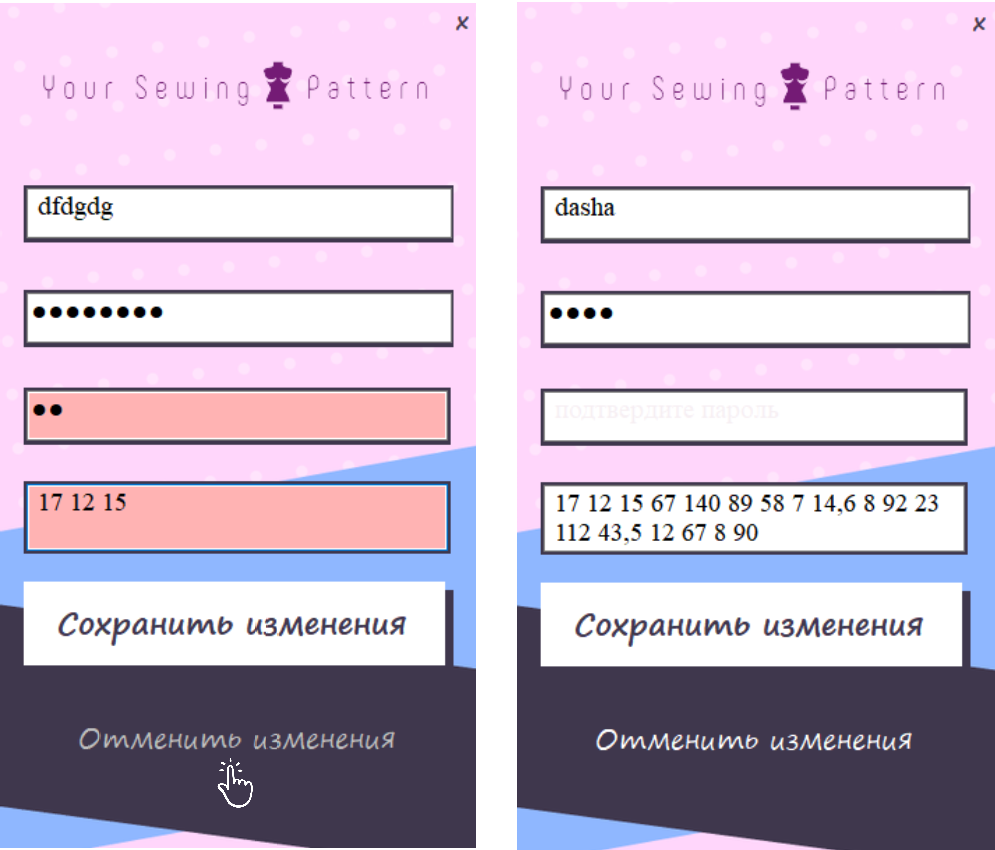


Рис. 33. Отменить изменения

Данные успешно обновлены. Информация тут же отобразилась в личном кабинете. Проверим, как теперь будет работать авторизация, рис. 34.

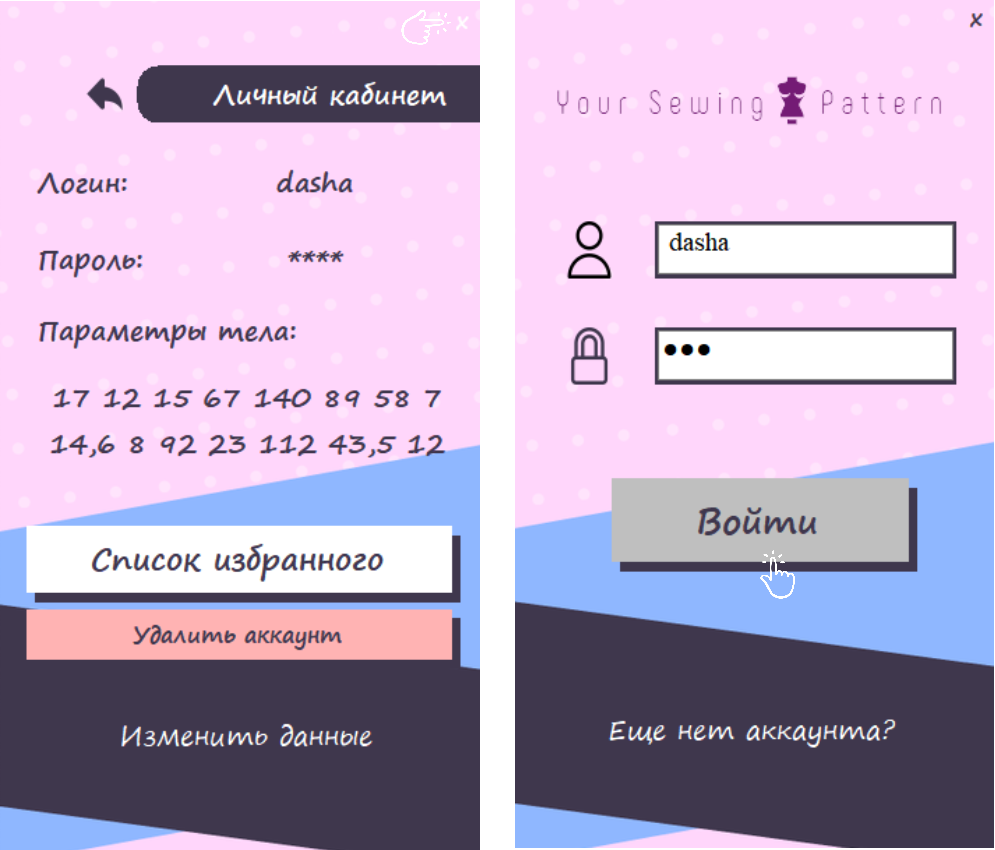


Рис. 34 Проверка авторизации с новым паролем

Система, конечно же, сообщит пользователю, что пароль неверный и не даст зайти. Но как только мы введем правильный обновленный пароль, система откроет главное окно, что будет значить, что пользователь авторизовался, рис. 35.

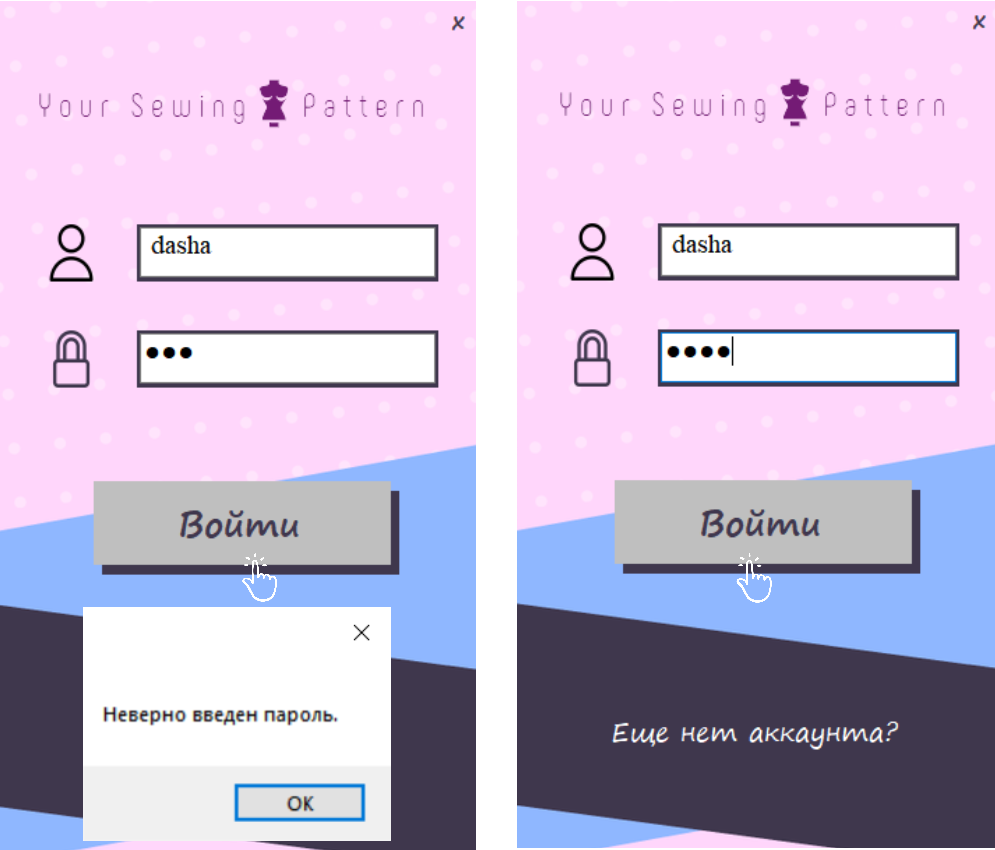


Рис. 35. Авторизация с новым паролем

В завершении примера удалим аккаунт нашего пользователя и попробуем вновь пройти авторизацию, рис. 36.

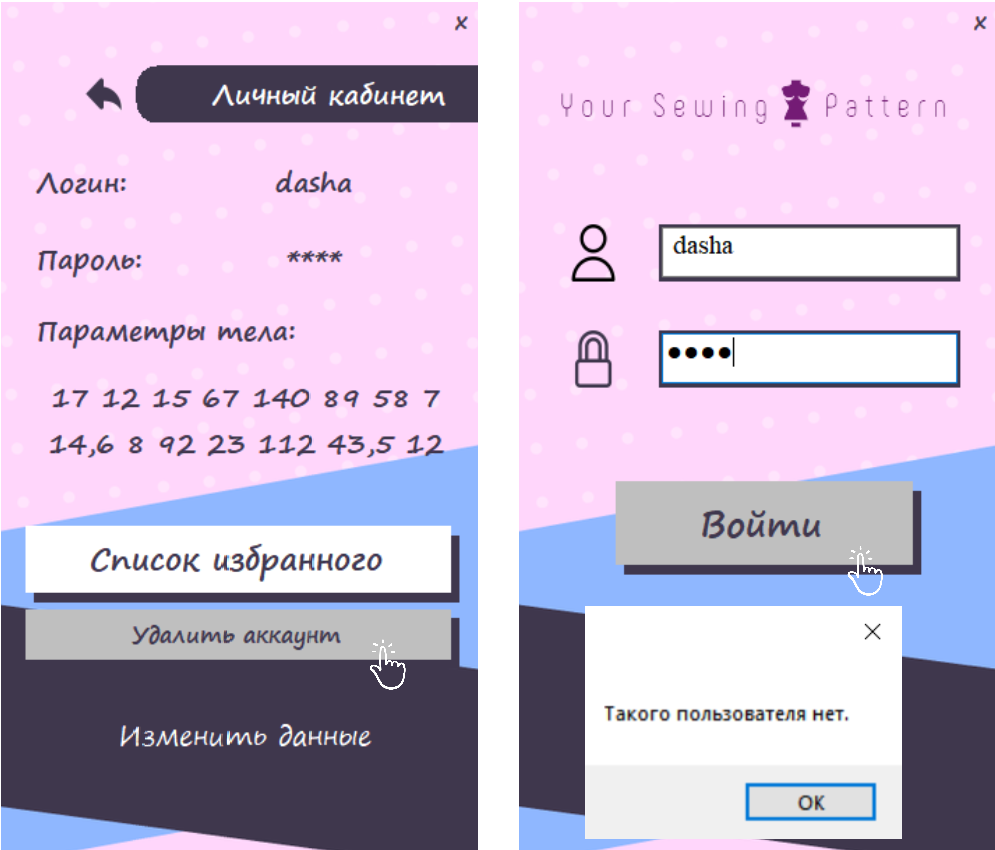


Рис. 36. Удаление аккаунта

В этом, собственно, и заключается основной функционал системы. В будущем она будет не только работать с данными справочника и пользователей, но и выполнять более сложные операции такие как, например, автоматическое изменение рисунка по введенным данным пользователя или печать в файл.

Это будет полностью работоспособное приложение, а пока все задачи данной работы выполнены.

# Приложение 1. Прямое проектирование

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema YSP

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema YSP

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `YSP` DEFAULT CHARACTER SET utf8 ;

USE `YSP` ;

-- -----------------------------------------------------

-- Table `YSP`.`Категории`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `YSP`.`Категории` (

`Номер категории` INT NOT NULL,

`Название категории` VARCHAR(60) NOT NULL,

PRIMARY KEY (`Номер категории`),

UNIQUE INDEX `Название категории\_UNIQUE` (`Название категории` ASC) VISIBLE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `YSP`.`Выкройки`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `YSP`.`Выкройки` (

`Номер выкройки` INT NOT NULL,

`Название` VARCHAR(60) NOT NULL,

`Основные мерки` MEDIUMTEXT NOT NULL,

`Категории\_Номер категории` INT NOT NULL,

`Дополнительные мерки` MEDIUMTEXT NOT NULL,

PRIMARY KEY (`Номер выкройки`),

UNIQUE INDEX `Название\_UNIQUE` (`Название` ASC) VISIBLE,

INDEX `fk\_Выкройки\_Категории1\_idx` (`Категории\_Номер категории` ASC) VISIBLE,

CONSTRAINT `fk\_Выкройки\_Категории1`

FOREIGN KEY (`Категории\_Номер категории`)

REFERENCES `YSP`.`Категории` (`Номер категории`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `YSP`.`3D-модель`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `YSP`.`3D-модель` (

`Номер 3D-модели` INT NOT NULL,

`3D-модель по стандартным меркам` LONGTEXT NOT NULL,

`Выкройки\_Номер выкройки` INT NOT NULL,

PRIMARY KEY (`Номер 3D-модели`, `Выкройки\_Номер выкройки`),

INDEX `fk\_3D-модель\_Выкройки\_idx` (`Выкройки\_Номер выкройки` ASC) VISIBLE,

CONSTRAINT `fk\_3D-модель\_Выкройки`

FOREIGN KEY (`Выкройки\_Номер выкройки`)

REFERENCES `YSP`.`Выкройки` (`Номер выкройки`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `YSP`.`Пользователи`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `YSP`.`Пользователи` (

`Логин` VARCHAR(20) NOT NULL,

`Пароль` VARCHAR(30) NOT NULL,

`Личные параметры` VARCHAR(100) NULL,

PRIMARY KEY (`Логин`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `YSP`.`Избранное пользователей`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `YSP`.`Избранное пользователей` (

`Пользователи\_Логин` VARCHAR(20) NOT NULL,

`Выкройки\_Номер выкройки` INT NOT NULL,

`Заметки` LONGTEXT NULL,

PRIMARY KEY (`Пользователи\_Логин`, `Выкройки\_Номер выкройки`),

INDEX `fk\_Пользователи\_has\_Выкройки\_Выкро\_idx` (`Выкройки\_Номер выкройки` ASC) VISIBLE,

INDEX `fk\_Пользователи\_has\_Выкройки\_Польз\_idx` (`Пользователи\_Логин` ASC) VISIBLE,

CONSTRAINT `fk\_Пользователи\_has\_Выкройки\_Пользо1`

FOREIGN KEY (`Пользователи\_Логин`)

REFERENCES `YSP`.`Пользователи` (`Логин`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Пользователи\_has\_Выкройки\_Выкрой1`

FOREIGN KEY (`Выкройки\_Номер выкройки`)

REFERENCES `YSP`.`Выкройки` (`Номер выкройки`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `YSP`.`Математические формулы`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `YSP`.`Математические формулы` (

`Номер математической формулы` INT NOT NULL,

`Формула` MEDIUMTEXT NOT NULL,

`Описание` LONGTEXT NOT NULL,

PRIMARY KEY (`Номер математической формулы`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `YSP`.`Выкройки\_has\_Математические формулы`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `YSP`.`Выкройки\_has\_Математические формулы` (

`Выкройки\_Номер выкройки` INT NOT NULL,

`Математические формулы\_Номер математической формулы` INT NOT NULL,

PRIMARY KEY (`Выкройки\_Номер выкройки`, `Математические формулы\_Номер математической формулы`),

INDEX `fk\_Выкройки\_has\_Математические фор\_idx` (`Математические формулы\_Номер математической формулы` ASC) VISIBLE,

INDEX `fk\_Выкройки\_has\_Математические фор\_idx1` (`Выкройки\_Номер выкройки` ASC) VISIBLE,

CONSTRAINT `fk\_Выкройки\_has\_Математические форм1`

FOREIGN KEY (`Выкройки\_Номер выкройки`)

REFERENCES `YSP`.`Выкройки` (`Номер выкройки`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Выкройки\_has\_Математические форм2`

FOREIGN KEY (`Математические формулы\_Номер математической формулы`)

REFERENCES `YSP`.`Математические формулы` (`Номер математической формулы`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

# Приложение 2. Реализованные функции

Запрос 1. Справочник категорий

insert into `YSP`.`категории` values

(1,'Юбки'),

(2,'Брюки'),

(3,'Рубашки'),

(4,'Блузки'),

(5,'Платья'),

('6', 'Шорты'),

('7', 'Джинсы');

Запрос 2. Справочник математических формул

insert into `YSP`.`математические формулы` values

(1,'(x+7\*y)^2-4\*x\*z+9\*y\*z','Короткий рукав'),

(2,'2\*y-(z+x-3)^3-6\*y\*x+5','Низ юбки солнце'),

(3,'3\*(z-x-y)^3-(z+6\*y)^2-10\*x','Воротник рубашки'),

(4,'4\*x-(y-z)^3-8\*x-6\*z','Брюки карго'),

(5,'5-12\*x\*y+(x-y-z+5)^3-y','Платье трапеция');

Запрос 3. Справочник пользователей

insert into `YSP`.`пользователи` values

('NiShiGara','123456789qwerty','104 84 110 21 22 31 44 34 11 38 44 46,5 49 40 12 32 17 62'),

('srf\_adlr','qwertasdfzxc','80 60 86 18 19 25 32 28 8 32 41 43 43 34 11 26 14 60'),

('burakov.a.a','12348765bur','112 96 122 22 23 33 48 36 12 40 45 48 51 42 13 34 18 63'),

('Aitan','qwertzxcFdW35','108 90 116 22 22,5 32 46 35 11,5 39 44,5 47 50 41 13 33 17,5 63'),

('Nathaniel','Po3jv83fvF','88 68 94 18 20 27 36 30 9 34 42 44 45 36 11 28 15 61');

Запрос 4. Справочник выкроек

insert into `YSP`.`выкройки` values

(1,'Юбка-солнце','ОТ',1,'ДИ'),

(2,'Брюки карго','ОТ ОБ',2,'ДИ'),

(3,'Платье трапеция','ОГ ОТ ОБ',5,'ДИ'),

(4,'Классическая рубашка','ДС',3,'ДВ ДР ШИ'),

(5,'Футболка','ОГ',4,' ДИ ДР'),

('6', 'Шорты-бермуды', 'ОТ ОБ ООБ ВС', '6', 'ДИ'),

('7', 'Мини-шорты', 'ОТ ОБ ООБ ВС', '6', 'ДИ'),

('8', 'Шорты-бриджи', 'ОТ ОБ ДБС ВС ДШ ВК ШПП ШЗП', '6', 'ДИ'),

('9', 'Классические шорты', 'ОТ ОБ ООБ ВС', '6', 'ДИ'),

('10', 'Велосипедки', 'ОТ ОБ ООБ ВС', '6', 'ДИ'),

('11', 'Широкие', 'ОТ ОБ ООБ ВС', '6', 'ДИ'),

('12', 'Домашние', 'ОТ ОБ ООБ ВС', '6', 'ДИ'),

('13', 'Милитари', 'ОТ ОБ ООБ ВС', '6', 'ДИ'),

('14', 'Сафари', 'ОТ ОБ ООБ ВС', '6', 'ДИ'),

('15', 'Чинос', 'ОТ ОБ ООБ ВС', '6', 'ДИ');

Запрос 5. Справочник Выкройки\_has\_Математические формулы

insert into `YSP`.`выкройки\_has\_математические формулы` values

(1,2),

(2,4),

(3,5),

(4,1),

(4,3),

(5,1);

Запрос 6. Справочник 3D-модель

insert into `YSP`.`3d-модель` values

(1,'модель юбки-солнце',1),

(2,'модель брюк карго',2),

(3,'модель платья трапеции',3),

(4,'модель классической рубашки',4),

(5,'модель футболки',5);

Запрос 7. Справочник Избранное пользователей

insert into `YSP`.`избранное пользователей` values

('NiShiGara',1,null),

('NiShiGara',2,null),

('NiShiGara',3,null),

('srf\_adlr',1,null),

('srf\_adlr',3,null),

('srf\_adlr',4,'Для похода в театр'),

('srf\_adlr',5,null),

('burakov.a.a',4,null),

('burakov.a.a',5,null),

('Aitan',5,'Для занятий спортом'),

('Nathaniel',4,null),

('Nathaniel',1,null),

('Nathaniel',3,'Для Дня рождения подруги'),

('srf\_adlr', '12', null),

('srf\_adlr', '7', null),

('srf\_adlr', '8', 'Для прогулок в жаркие дни'),

('srf\_adlr', '9', null),

('srf\_adlr', '11', null),

('srf\_adlr', '14', null);

Запрос 8.

SELECT \* FROM `пользователи` WHERE `Логин` = @uL;

Где @uL – логин пользователя.

Запрос 9.

SELECT count(\*) AS Пользователи, (SELECT count(\*) FROM `выкройки`) AS Выкройки FROM `пользователи`;

Запрос 10.

INSERT INTO `пользователи` values(@uL, @uP, @uM);

Где @uL – логин пользователя, @uP – пароль пользователя, @uM - мерки пользователя (по умолчанию Null).

Запрос 11.

SELECT `Номер выкройки`,`Название`,`Основные мерки`,`Дополнительные мерки` FROM `выкройки` e WHERE e.`Категории\_Номер категории` = @nC;

Где @nC – номер категории.

Запрос 12.

INSERT INTO `избранное пользователей` values(@uL, @vN, @nts);

Где @uL – номер категории, @vN – номер выкройки, @nts – заметки пользователя (могут быть Null).

Запрос 13.

UPDATE `пользователи` set `Логин` = @uLN, `Пароль` = @uPN, `Личные параметры` = @uMN where `Логин` = @uL;

Где @uLN – новый логин пользователя, @uPN – новый пароль пользователя, @uMN - новые параметры пользователя, @uL – изначальный логин пользователя.

Запрос 14.

SELECT v.`Номер выкройки`, v.`Название`, v.`Основные мерки`, v.`Дополнительные мерки`, izb.`Заметки` FROM `избранное пользователей` izb, `выкройки` v WHERE izb.`Пользователи\_Логин` = @uL AND izb.`Выкройки\_Номер выкройки` = v.`Номер выкройки`;

Где @uL – логин пользователя.

Запрос 15.

DELETE FROM `избранное пользователей` WHERE `избранное пользователей`.`Пользователи\_Логин` = @uL AND `избранное пользователей`.`Выкройки\_Номер выкройки` = @vN;

Где @uL – логин пользователя, @vN – номер выкройки.

Запрос 16.

UPDATE `избранное пользователей` SET `Заметки` = @nts WHERE `избранное пользователей`.`Пользователи\_Логин` = @uL AND `избранное пользователей`.`Выкройки\_Номер выкройки` = @vN;

Где @nts – заметка, @uL – логин пользователя, @vN – номер выкройки.

Запрос 17.

DELETE FROM `пользователи` WHERE `пользователи`.`Логин` = @uL;

Где @uL – логин пользователя.