**Министерство цифрового развития, связи и массовых коммуникаций РФ**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Сибирский государственный университет телекоммуникаций и информатики»**

**(СибГУТИ)**

**КУРСОВАЯ РАБОТА**

Вариант 30

«Учет занятий спортивной школы»

Выполнил студент группы ИП-113 Шпилев Д. И.

Проверил доцент кафедры ПМиК к.т.н Мерзлякова Е.Ю.

Новосибирск \_\_\_\_\_\_г.

Новосибирск 2023

Содержание

[Введение 3](#_Toc149239098)

[1. Анализ задач и пользователей 4](#_Toc149239099)

[2. Выбор репрезентативных задач 5](#_Toc149239100)

[3. Заимствование 6](#_Toc149239101)

[4. Прототип интерфейса 9](#_Toc149239102)

[5. Реализация 13](#_Toc149239103)

[Скриншоты работы программы 14](#_Toc149239104)

[CWT-анализ 23](#_Toc149239105)

[GOMS-анализ 25](#_Toc149239106)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 28](#_Toc149239107)

[ПРИЛОЖЕНИЕ 29](#_Toc149239108)

# Введение

В современном мире все больше людей уделяют внимание спорту и здоровому образу жизни. Это приводит к росту числа спортивных школ, секций и клубов, которые предлагают разнообразные занятия для детей и взрослых. Однако, для эффективного управления таким учреждением необходимо вести учет занятий, контролировать посещаемость, анализировать результаты учащихся, планировать расписание и многое другое. Именно для решения этих задач создаются специализированные приложения для учета занятий в спортивной школе. Я выбрал данную тему курсовой работы потому, что увлекаюсь спортом, а также в моем родном городе недавно появилась секция по плаванию. Поэтому я и решил разработать приложение, где можно производить учет для данного вида спорта.

На просторах интернета я столкнулся с приложением «Учет клиентов» и с ее конфигурацией «Спортивная школа». Именно оттуда я позаимствовал интерфейс приложения.

# 1. Анализ задач и пользователей

Для варианта задания «Учет занятий спортивной школы» при рассмотрении заинтересованных лиц были выбраны люди, которые имеют опыт с работой за компьютером.

Из данной категории были выбраны одногруппник и мама.

1 Одногруппник: возраст – 20 лет, образование – среднее общее, студент,

навыки владения компьютером – высокие.

2 Мама: возраст – 43 года, образование – высшее педагогическое, мед. Сестра в поликлинике, навыки владения компьютером – средние.

Пользователи программного продукта для учета спортивной школы являются лицами женского пола, имеют незаконченное или полное высшее образование. Навык владения компьютером обычно

на уровне базового.

Основные термины пользователя:

* Учет занятий - процесс регистрации и анализа данных о проведенных занятиях, посещаемости, результатах учащихся и т.д.
* Результаты учащихся (разряды) - показатели, отражающие уровень достижений учащихся в различных видах спорта, а также их прогресс в развитии физических качеств.

Возможности существующих программ по схожей тематике: добавление, удаление и редактирование данных о учениках, тренерах, соревнованиях, инвентаре. Просмотр разрядов по плаванию.

# 2. Выбор репрезентативных задач

1. Добавление, редактирование и удаление данных об учениках.
2. Добавление, редактирование и удаление данных об соревнования.
3. Добавление, редактирование и удаление результатов прошедших или идущих соревнований для отдельного ученика.
4. Добавление, редактирование и удаление данных об тренерах.
5. Добавление, редактирование и удаление данных об инвентаре в спортивной школе и его месте.
6. Просмотр результатов для получения разряда по плаванию для определенной дистанции и бассейна.
7. Импорт, экспорт и создание новой базы данных учета.
8. Предоставление помощи пользователю по интерфейсу.

# 3. Заимствование

**Учет клиентов**

Приложение для учета клиентов. У нее есть отдельная конфигурация «Спортивная школа», от куда и будет позаимствован основой функционал и дизайн приложения. Например, при заходе в приложение будет отображаться примерно следующее (рис.1):

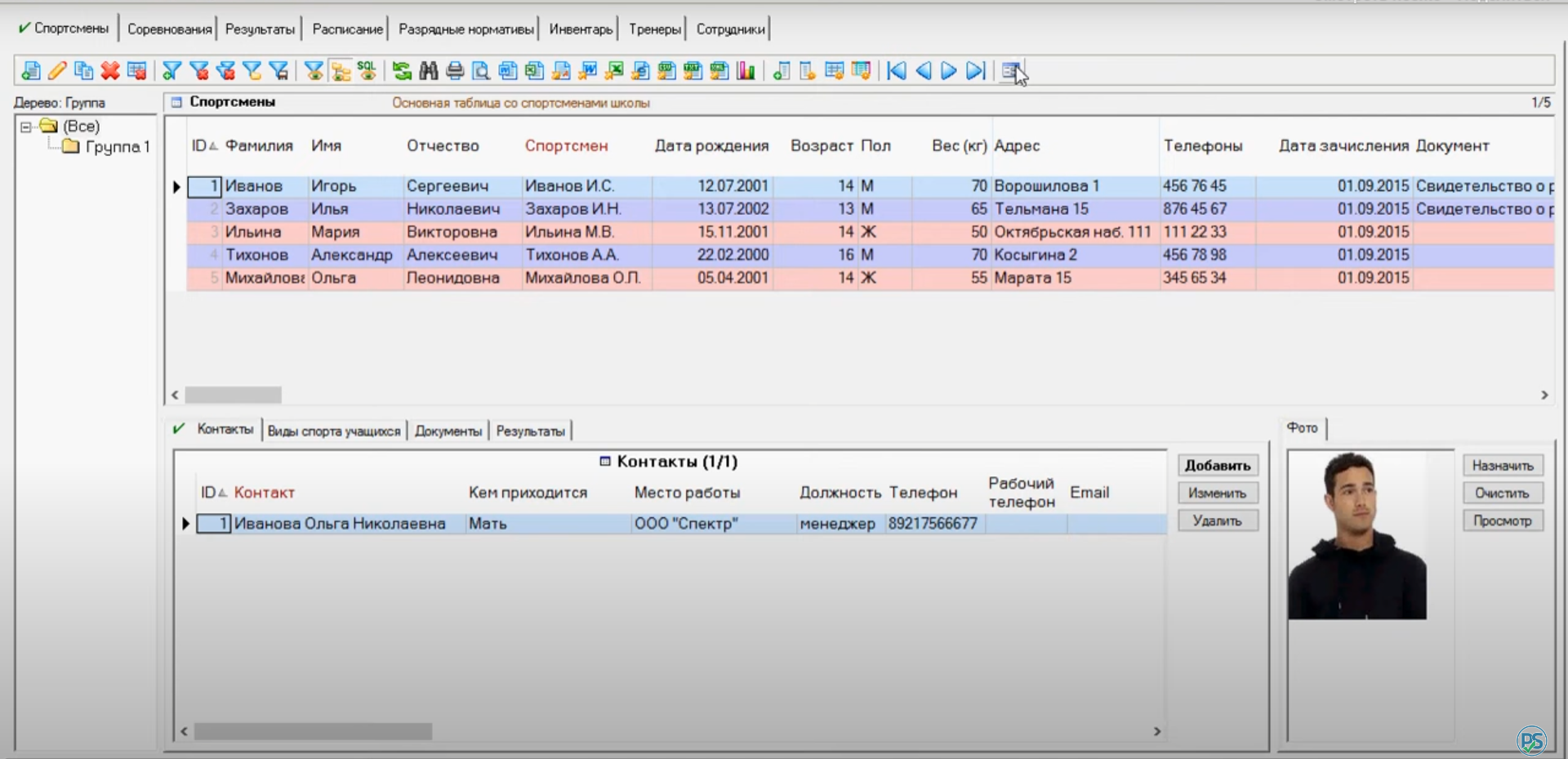


Рисунок 1. Первое окно приложения «Учет клиентов»

Добавление и редактирование спортсменов будет выглядеть примерно вот так (рис.2):

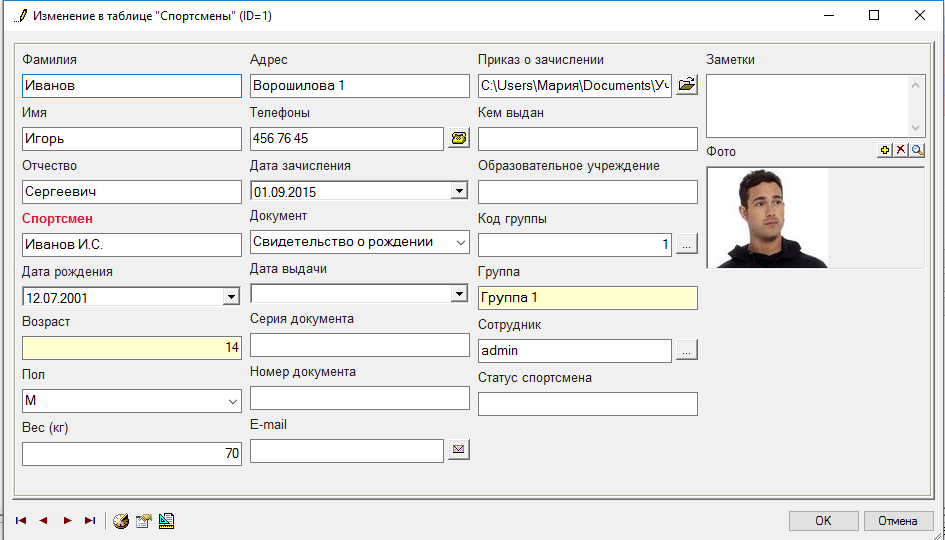


Рисунок 2. Добавление/Редактирование спортсмена в «Учет клиентов»

Учет соревнований будет выглядеть следующем образом (рис.3):

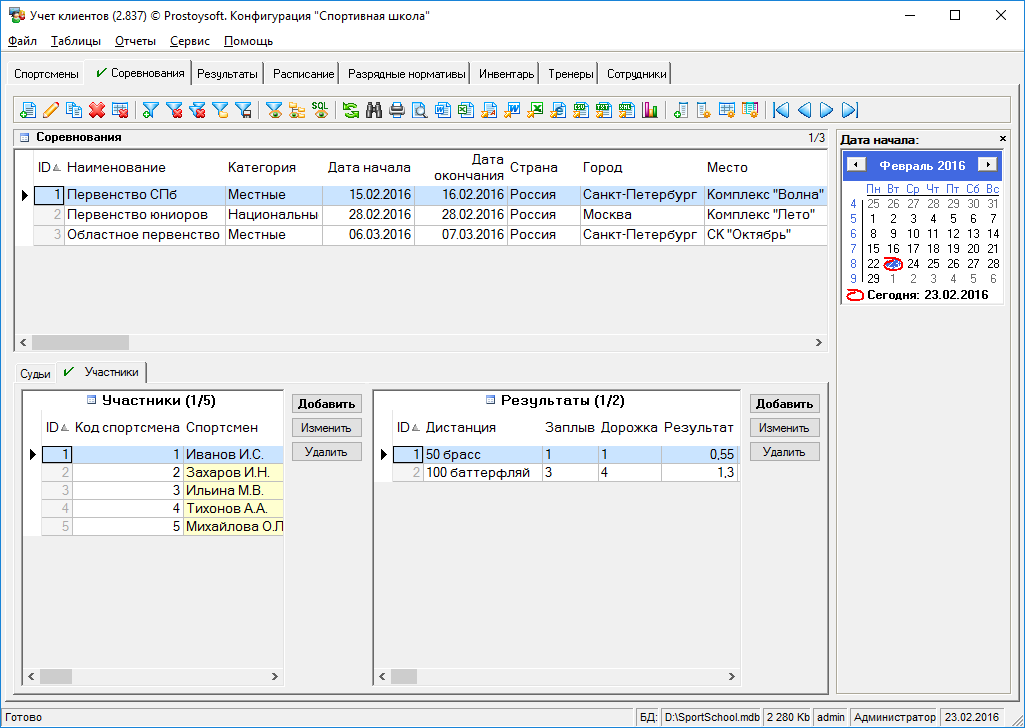


Рисунок 3. Учет соревнований в «Учет клиентов»

Добавление результатов по отдельному спортсмену в соревновании (рис.4):

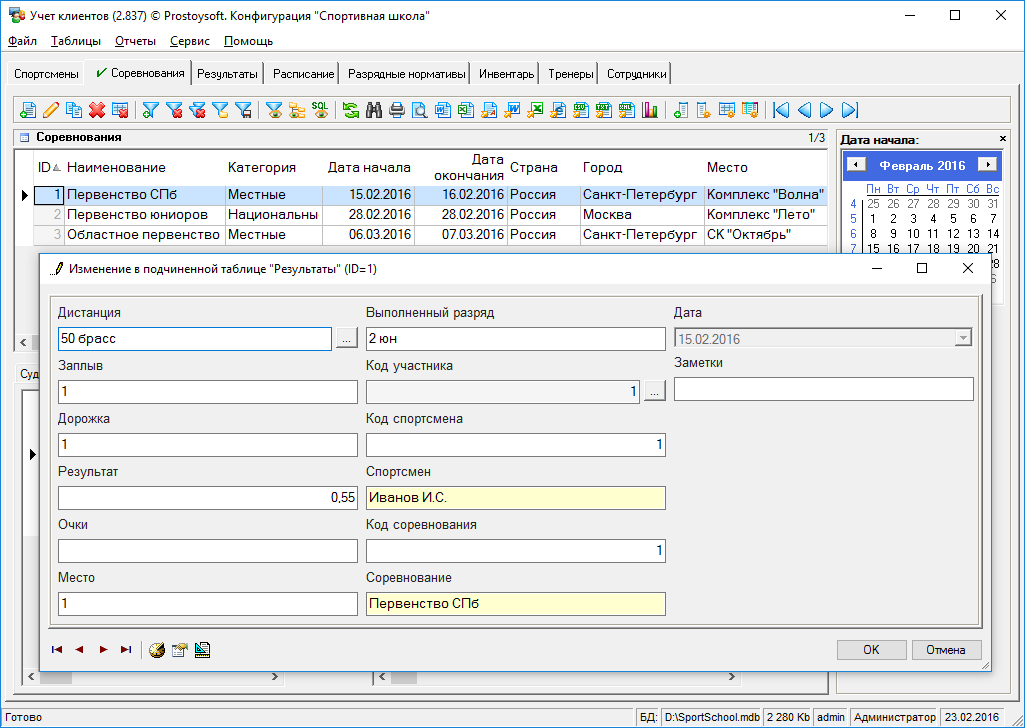


Рисунок 4. Добавление результатов

Просмотр разрядов (рис.5):

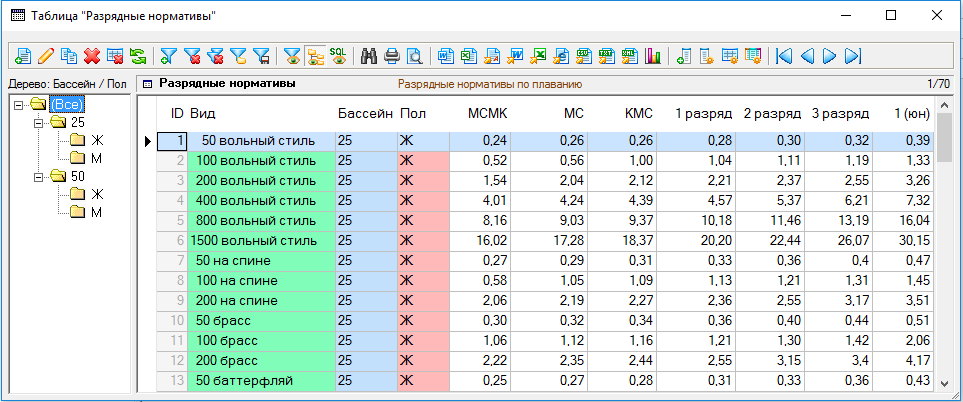


Рисунок 5. Разряды

Просмотр расписания, а также добавление, редактирование и удаление (рис.6):

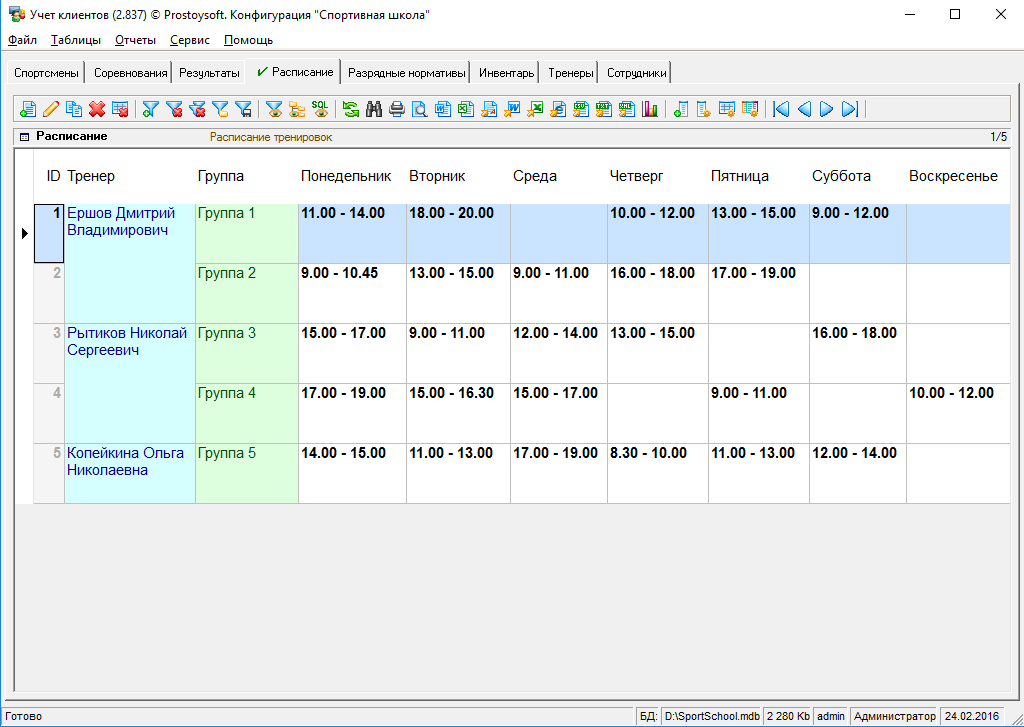


Рисунок 6. Расписание

# 4. Прототип интерфейса

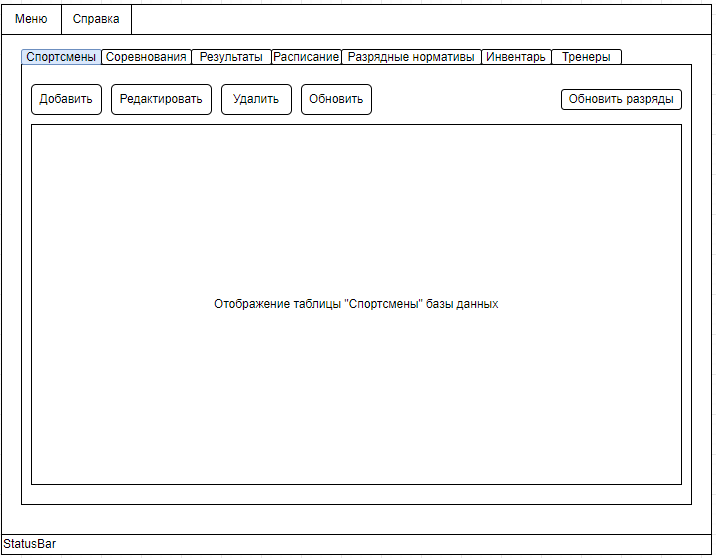


Рисунок 7. Черновое описание спортсменов

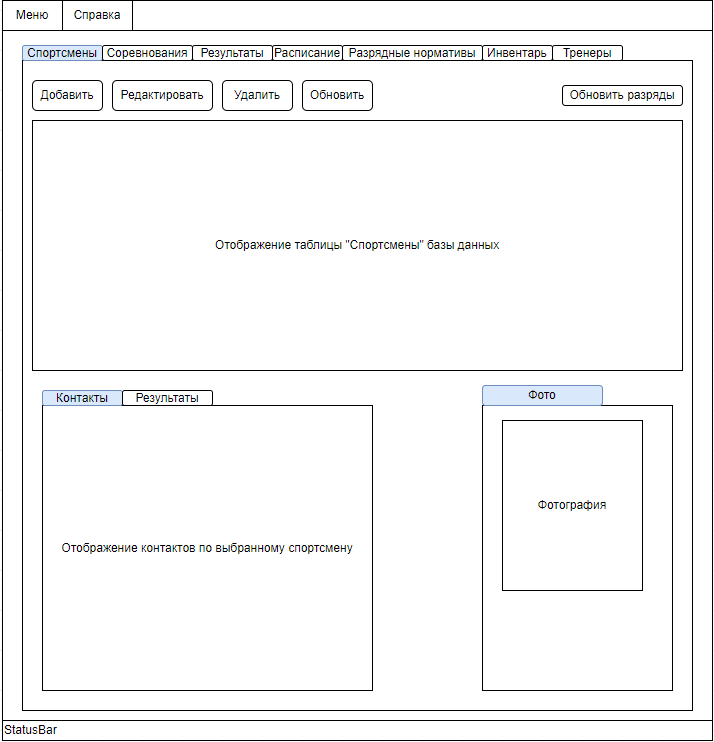
На данной вкладке будет производиться отображение базы данных. При нажатии на кнопку «Изменить» должно открыться окно изменения записи. При нажатии на кнопку «Удалить» запись должна удаляться из базы. При нажатии на кнопку «Добавить» должно открываться окно создания новой записи. При нажатии на кнопку «Обновить разряды» происходит пересчет разрядов для всех спортсменов на основе соревнований.

Меню: содержит создание новой, экспорт и импорт базы данных.

Справка: содержит инструкцию пользования программой.

StatusBar: показывает статус базы данных, открыта она или нет.

Все вкладки отличаются лишь тем, какая в них показывается таблица базы данных и в наличии кнопок.



Удалить

Редактировать

Добавить

Рисунок 8. Отображение информации об спортсмене

При выбранном спортсмене показывается информация о его контактах и результатах, а также показывается фотография. Так же имеется возможность добавить, изменить или удалить их.

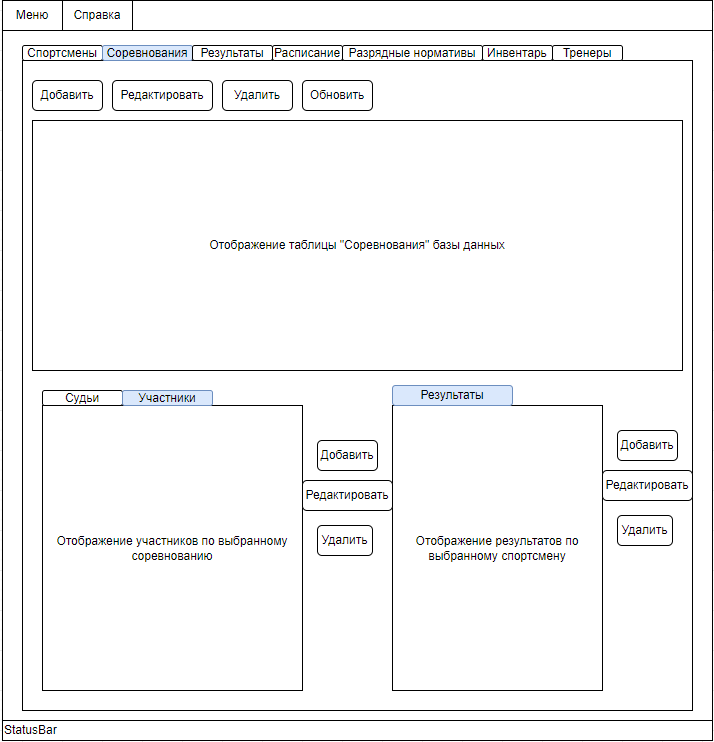


Рисунок 9. Отображение информации об соревновании

По выбранному соревнованию будут отображаться их участники и судьи. А по выбранному участнику отображаются результаты.

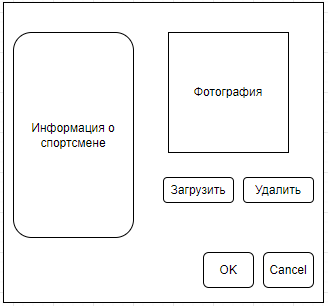


Рисунок 10. Диалоговое окно Добавления/Удаления спортсмена

Кнопка «Загрузить» открывает диалоговое окно выбора изображения. Если нужно удалить текущее изображение необходимо нажать на кнопку «Удалить». Если не заполнить все нужные поля информации, программа не даст сохранить запись и попросит заполнить все необходимые поля.

# 5. Реализация

Функционал:

* Импорт и экспорт базы данных, а также создание новой.
* Просмотр, создание, редактирование и удаление таблиц базы данных спортивной школы, таких как: спортсмены, соревнования, результаты, разряды, тренеры, инвентарь.
* Автоматическое определение текущего разряда для каждого из спортсменов.
* Навигатор помощи с описанием возможности приложения.

Приложение состоит из окон:

* Главное окно с вкладками.
* Окно по добавлению/редактированию спортсмена.
* Окно по добавлению/редактированию контактов спортсмена.
* Окно по добавлению/редактированию соревнования.
* Окно по добавлению/редактированию результатов спортсмена.
* Окно по добавлению/редактированию судей соревнований.
* Окно по добавлению/редактированию участников соревнований.
* Окно по добавлению/редактированию расписания.
* Окно по добавлению/редактированию спортсмена.
* Окно по добавлению/редактированию инвентаря.
* Окно по добавлению/редактированию тренеров.

База данных реализована на языке SQLite, взаимодействующим с функционалом библиотеки QSql. Всего создано 10 таблиц базы данных.

Навигатор помощи реализован с помощью файлов .html, содержимое которых выводится в элемент класса QTextBrowser.

## Скриншоты работы программы

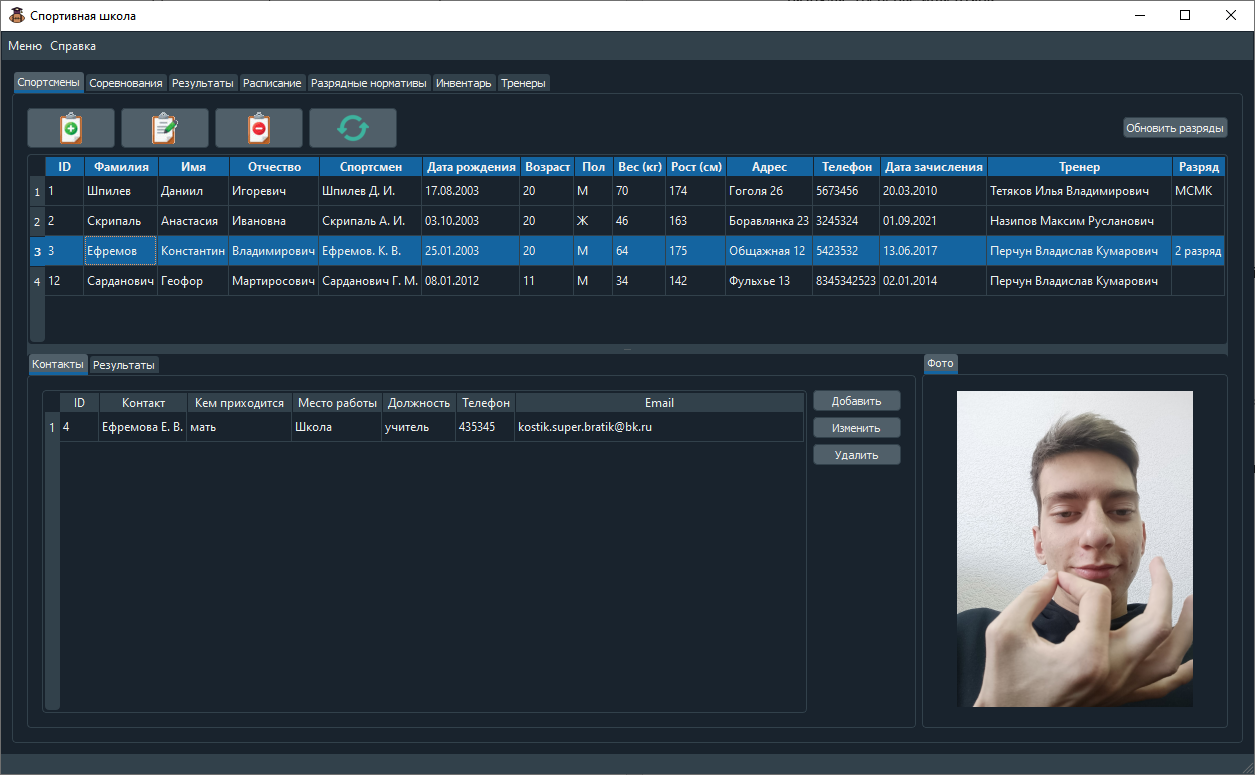


Рисунок 11. Начальное окно со спортсменами

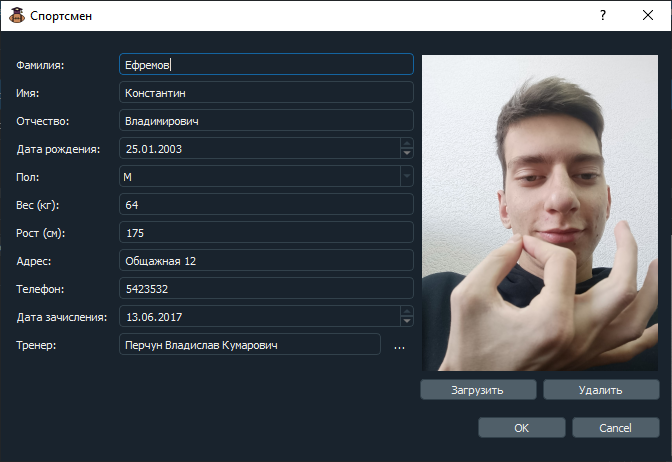


Рисунок 12. Редактирование спортсмена

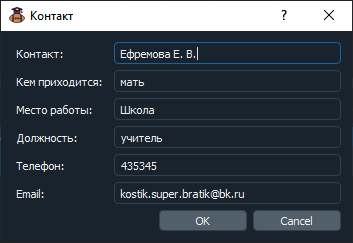


Рисунок 13. Редактирование контактов спортсмена

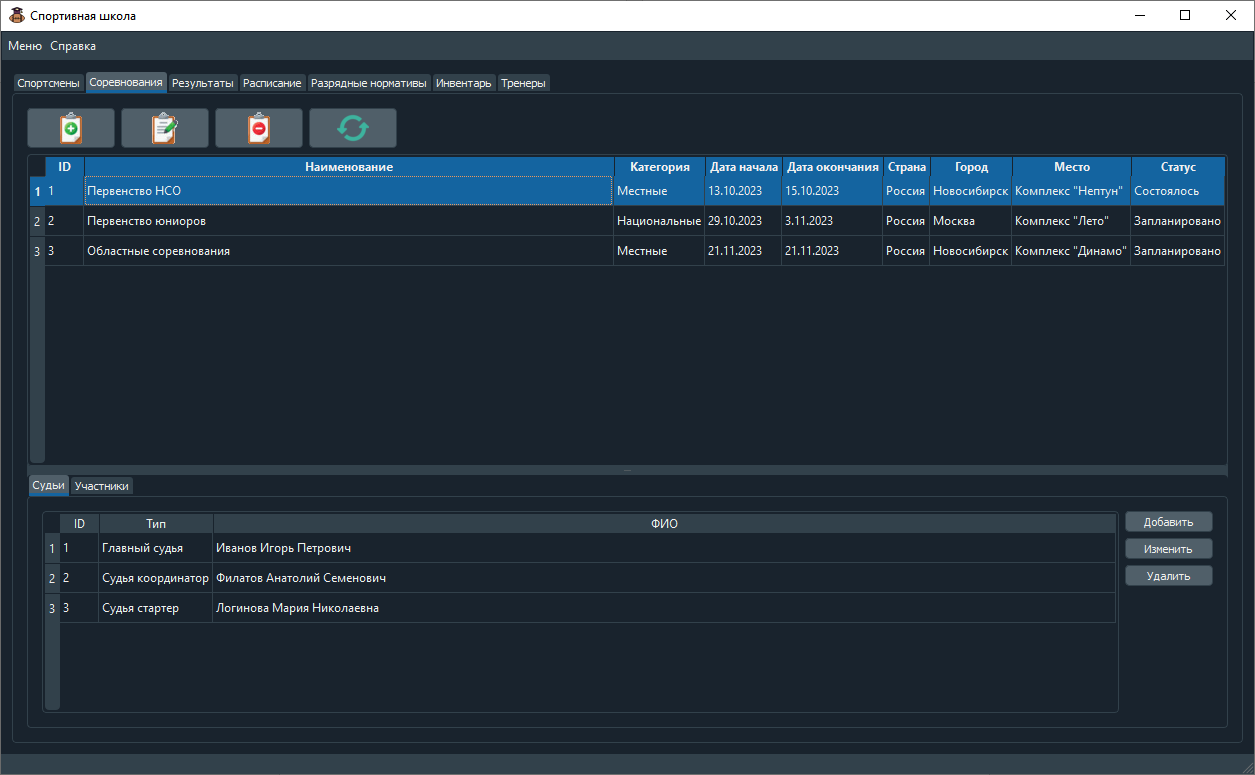


Рисунок 14. Вкладка «Соревнования»

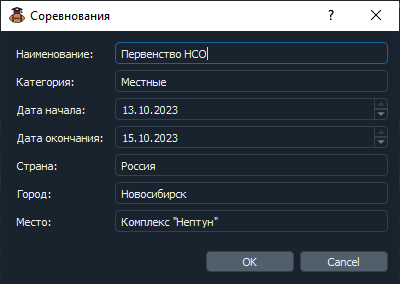


Рисунок 15. Изменение соревнования

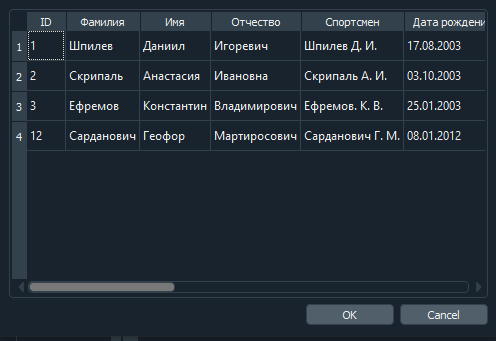


Рисунок 16. Добавление спортсмена

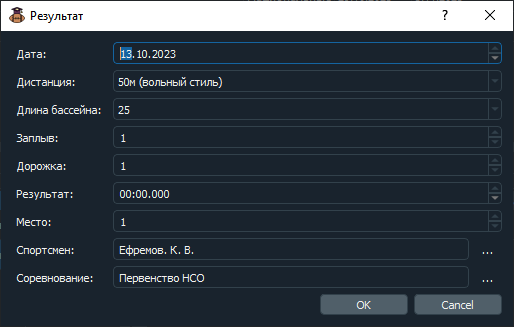


Рисунок 17. Добавление результата спортсмену

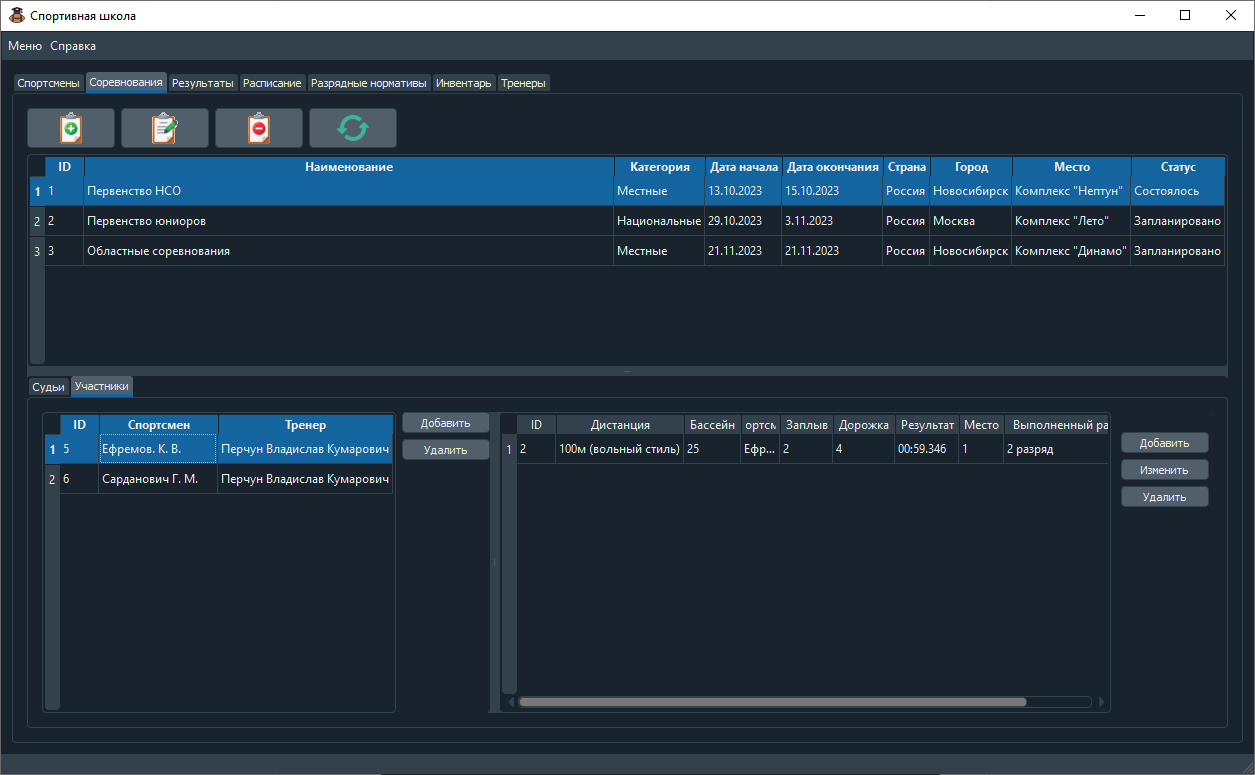


Рисунок 18. Участники соревнований

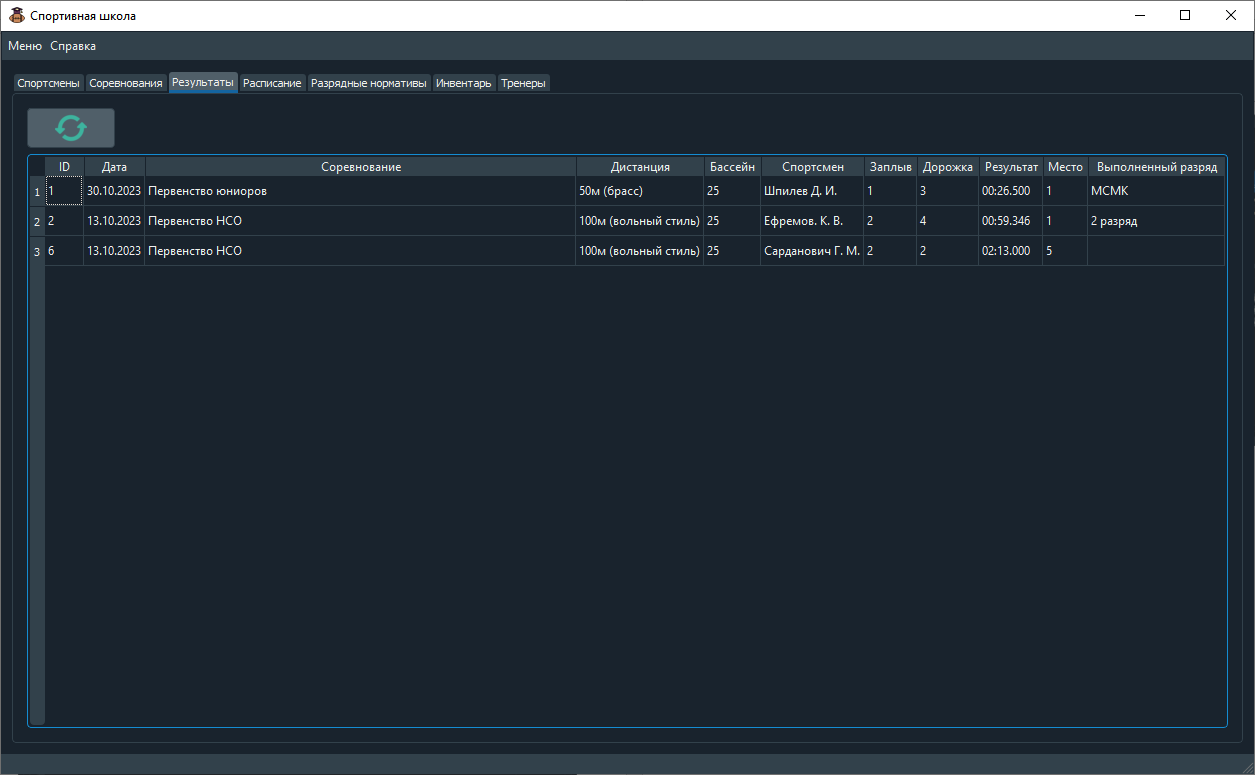


Рисунок 19. Общие результаты всех спортсменов и соревнований

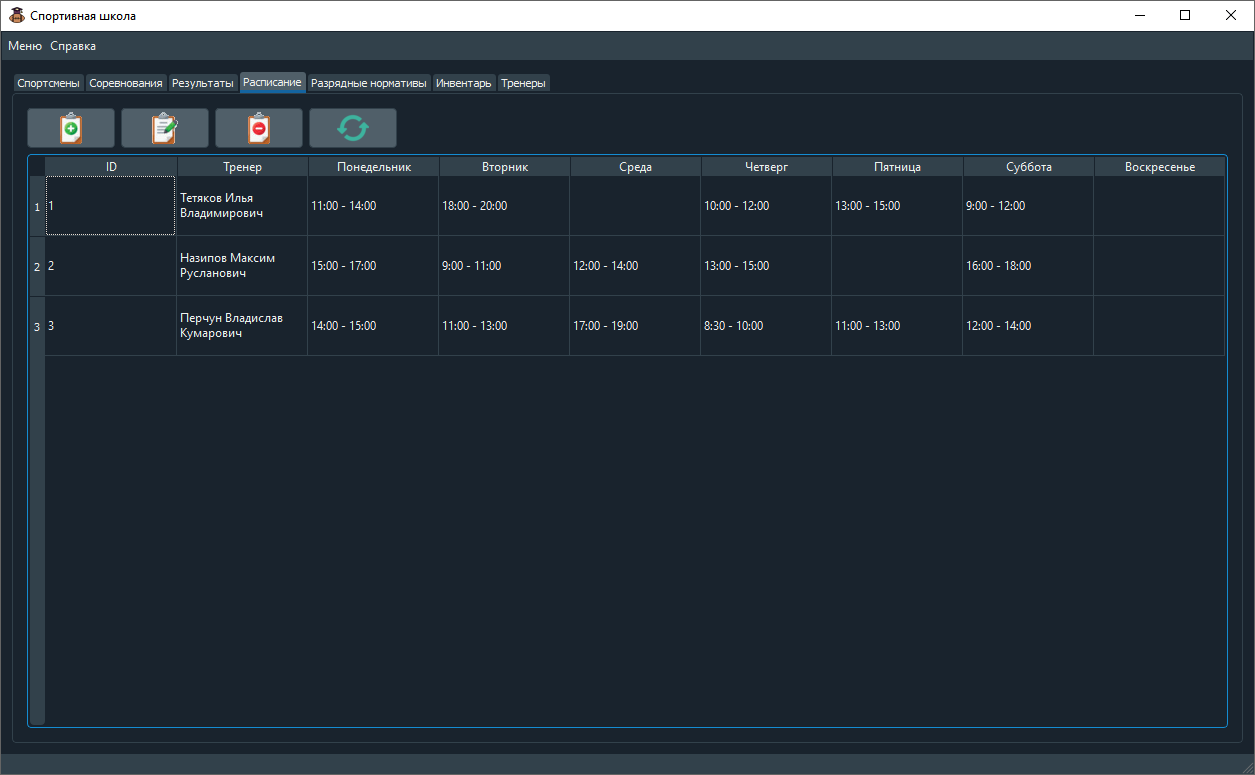


Рисунок 20. Расписание занятий

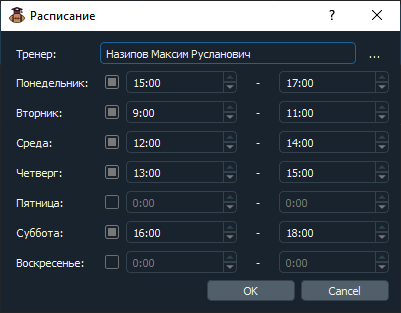


Рисунок 21. Редактирование расписания

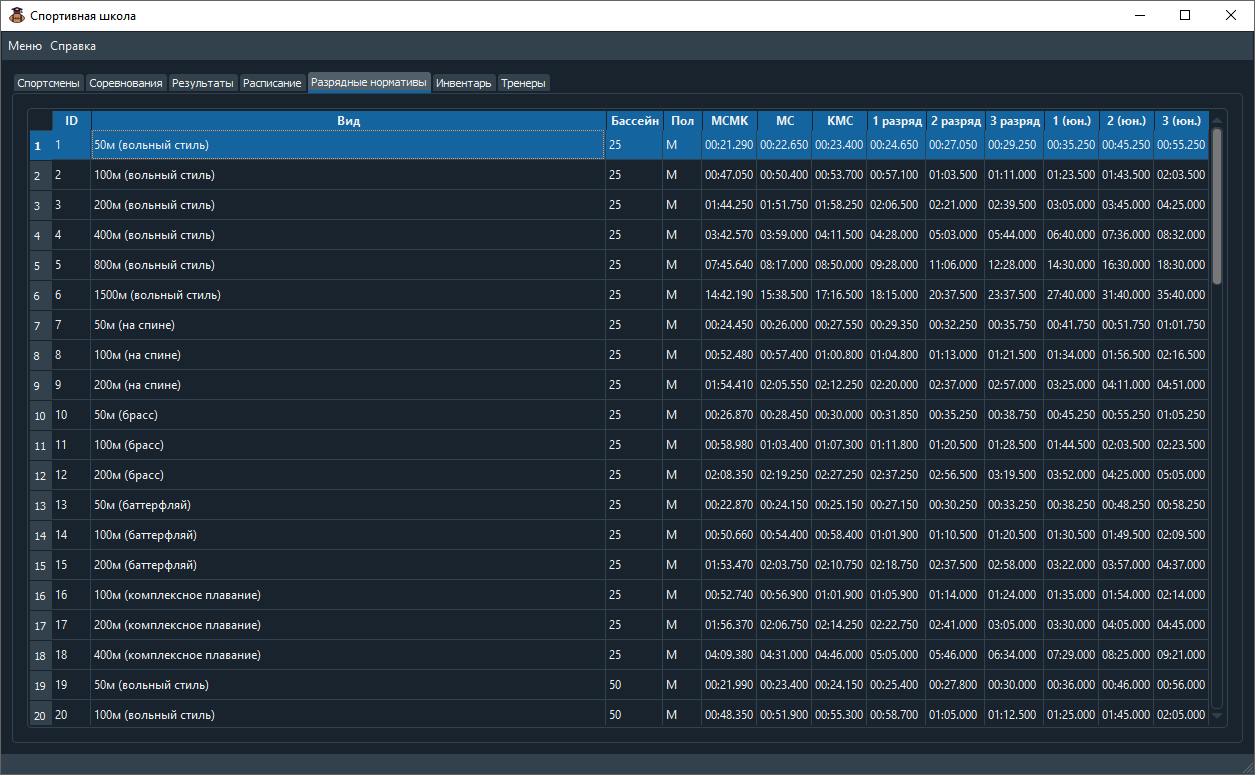


Рисунок 22. Таблица разрядов по разным видам

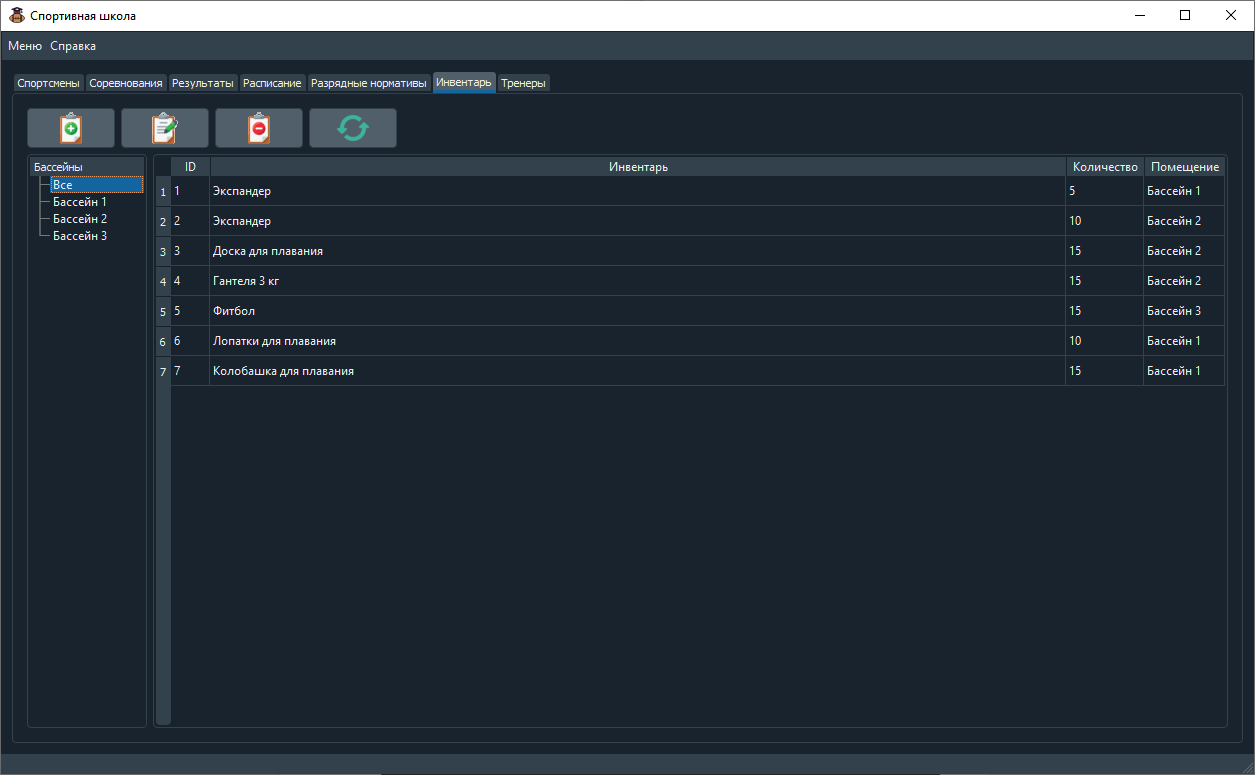


Рисунок 23. Инвентарь

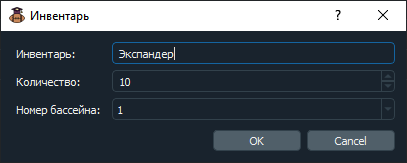


Рисунок 24. Редактирование инвенторя

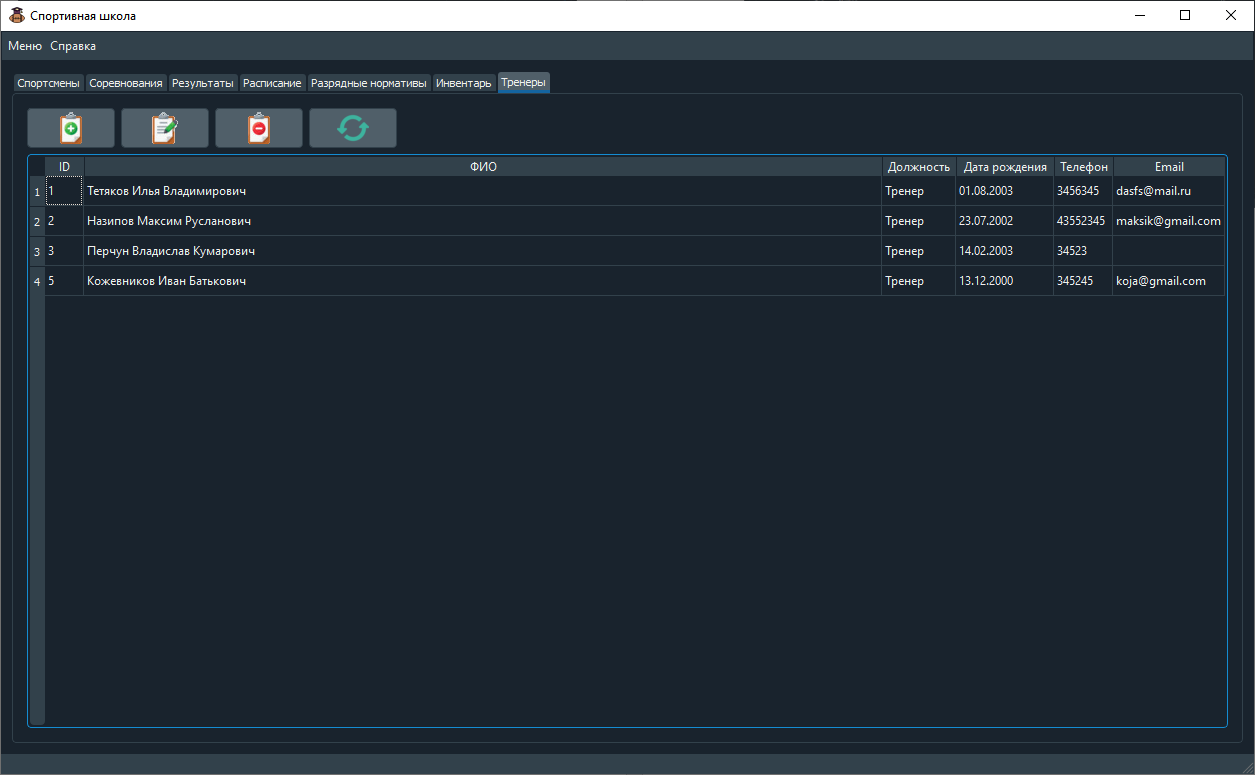


Рисунок 25. Тренеры

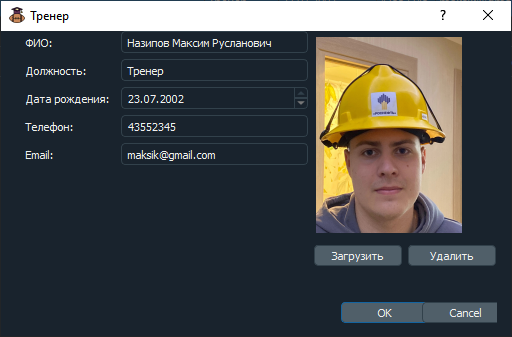


Рисунок 26. Редактирование тренеров

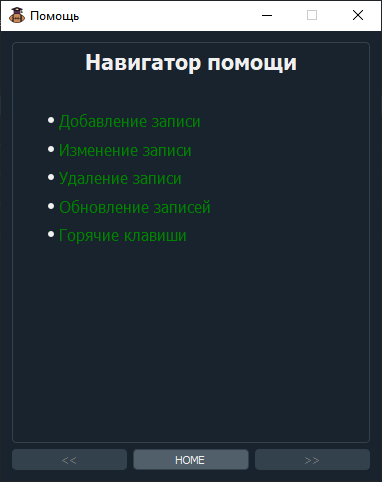
\

Рисунок 27. Навигатор помощи

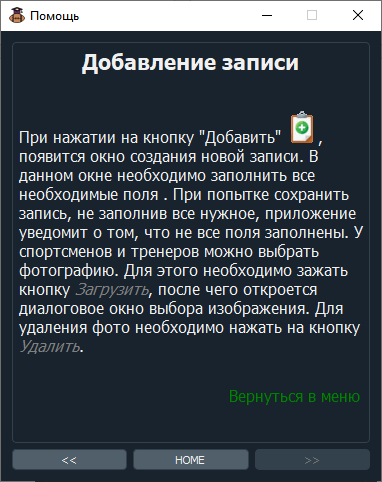


Рисунок 28. Элемент навигатора помощи

## CWT-анализ

**Задача 1: Добавление нового спортсмена без фото**

*Список действий:*

1. Запустить приложение.
2. Нажать на кнопку «Добавить спортсмена».
3. Заполнение всех необходимых полей.
4. Нажать на кнопку «OK».

*Анализ действий задачи 1:*

С первым действием у пользователя не должно возникнуть затруднений, замечаний нет.

Во втором пункте у пользователя могут возникнуть трудности с поиском кнопки «Добавить спортсмена». Для решения этой проблемы были использованы интуитивно понятные иконки с привычным для пользователя плюсом, а также добавлена подсказка, которая при на ведении на кнопки показывает, какая из кнопок за что отвечает.

С заполнением всех нужных полей у пользователя так же может возникнуть проблема с выбором всех нужных полей. Если не все нужные поля были заполнены, открывается окно, которое говорит, что нужно заполнить все поля. Пустыми полями могут оставаться лишь поле «Отчество» и «Email».

С нажатием на «OK» трудностей вызвать не должно.

Так же, как дополнительная помощь людям для этой задачи есть навигатор помощи, где они могут найти ответ на свой вопрос.

**Задача 2: Удаление спортсмена**

*Список действий:*

1. Запустить приложение.
2. Выбрать спортсмена, которого необходимо удалить.
3. Нажать на кнопку «Удалить спортсмена».

*Анализ действий задачи 1:*

С первым действием у пользователя не должно возникнуть затруднений, замечаний нет.

Во втором пункте у пользователя так же не должно возникнуть никаких проблем.

Проблема может возникнуть с поиском кнопки «Удалить спортсмена». Для этого, как и в 1 задаче были добавлены интуитивно понятные иконки и подсказки что делает данная кнопка.

GOMS-анализ

**Описание анализа**

Практически все интерфейсные взаимодействия в можно описать следующими операциями:

K – нажатие клавиши;

B – клик кнопкой мыши;

P – наведение указателя мыши;

R – ожидание ответной реакции компьютера;

H – перенос руки с клавиатуры на мышь или наоборот

D – проведение с помощью мыши прямой линии (например, выделение или прокрутка текста);

M – мыслительная подготовка (к осуществлению одной из перечисленных операций).

Разные пользователи выполняют указанные операции за разное время. Однако, GOMS исследует работу опытного пользователя. Многочисленные исследования выявили средние значения времени операций, выполняемых опытными пользователями.

K 0.2 с

B 0.2 с

P 1.1 с

H 0.4 с

M 1.35 с

**Цель №1:** В таблице «Инвентарь» изменить количество «Доска для плавания» с 15 до 20.

Для выполнения цели сформулируем подцели:

1. Выбрать интересующую запись из таблицы
2. Изменить количество
3. Сохранить изменения

Теперь опишем методы для каждой подцели и распишем каждый метод с точностью до операции:

1. Выбрать интересующую запись из таблицы:
   1. Открыть вкладку «Инвентарь».

M PB (перемещение указателя и выбор вкладки)

* 1. Нажать на интересующую запись.

M PB (перемещение указателя и выбор записи)

1. Изменить количество:
   1. Нажать на кнопку «Изменить».

M PB (перемещение указателя и нажатие на кнопку)

* 1. Нажать на поле «Количество».

M PB (перемещение указателя и нажатие на поле)

* 1. Стереть число 15 и написать 20.

H KKKK (перемещение руки с мыши на клавиатуру, стирание 2 символов и печатание 2)

1. Сохранить изменения:
   1. Нажать на кнопку «OK».

M H PB (перемещение руки с клавиатуры на мышь и нажатие на кнопку)

5М = 6.75 5P=5.5 5B = 1 2H = 0.8 4K = 0.8

Итог: **14.85 с**

Для улучшения результатов можно не искать и не нажимать на кнопку изменить, а дважды нажать на запись M PBB вместо M PB M PB

Результат: **12.4 с**

**Цель №2:** Удаление необходимого элемента из вкладки «Инвентарь».

Для выполнения цели сформулируем подцели:

1. Выбрать нужную запись.
2. Нажать на кнопку удалить.

Теперь опишем методы для каждой подцели и распишем каждый метод с точностью до операции:

1. Выбрать нужную запись.
   1. Открыть вкладку «Инвентарь».

M PB (перемещение указателя и выбор вкладки)

* 1. Нажать на интересующую запись.

M PB (перемещение указателя и выбор записи)

1. Изменить количество:
   1. Нажать на кнопку «Удалить».

M PB (перемещение указателя и нажатие на кнопку)

3М = 4.05 3P=3.3 3B = 0.6

Итог: **7.95 с**

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

* Учет клиентов – программа, с которой был позаимствован интерфейс;

URL:  [https://prostoysoft.ru/SportSchool.htm](https://sqlitebrowser.org/)

* Qt Documentation – официальная документация по разработке в среде Qt;

URL: <https://doc.qt.io/>

# ПРИЛОЖЕНИЕ

Основной код:

settings = new QSettings(this); //создание указателя на QSettings

db = QSqlDatabase::addDatabase("QSQLITE", "db"); //долавлаение DB

db.setDatabaseName("./../CourseWork/database.db");

if(db.open()){ //открытие DB

ui->statusBar->showMessage("Database open", 3000);

}

else{

ui->statusBar->showMessage("Database not open", 3000);

}

query = QSqlQuery(db); //инициализация переменной класса QSqlQuery

Связь элементов меню с функциями:

connect(ui->newDataBase, SIGNAL(triggered()), this, SLOT(on\_newDB\_triggered()));

connect(ui->exportDataBase, SIGNAL(triggered()), this, SLOT(on\_exportDB\_triggered()));

connect(ui->importDataBase, SIGNAL(triggered()), this, SLOT(on\_importDB\_triggered()));

connect(ui->help, SIGNAL(triggered()), this, SLOT(slotInfo()));

Таймер обновления состояния соревнований:

timer = new QTimer();

connect(timer, SIGNAL(timeout()), this, SLOT(timerUpdate()));

timer->start(60000);

Подгрузка .qss файла (.css):

QFile file(":/resourses/style/style.qss");

file.*open*(QFile::ReadOnly);

a.setStyleSheet(file.readAll());

Сохранение и загрузка данных QSettings:

void MainWindow::**saveSettings**()

{

settings->setValue("title", windowTitle());

settings->setValue("geometry", geometry());

settings->setValue("page", ui->scheduleTable->currentIndex());

}

void MainWindow::**loadSettings**()

{

setWindowTitle(settings->value("title", "Спортивная школа").toString());

setGeometry(settings->value("geometry", QRect(200,200,300,300)).toRect());

ui->scheduleTable->setCurrentIndex(settings->value("page", 0).toInt());

}

Пример запроса для выгрузки спортсменов из базы данных:

if (!query.exec("SELECT a.ID, a.Surname, a.Name, a.Patronymic, a.Athlete, a.dateOfBirth, a.Gender, a.Weight, a.Height, a.Address, a.Telephone, a.DateOfEnrollment, b.Name, a.Category "

"FROM Athletes AS a, Coaches AS b "

"WHERE a.CoachID = b.ID")){

ui->statusBar->showMessage("Database table 'Athletes' not open", 3000);

return;

}

Реализация навигатора справки:

connect(ui->textBrowser, SIGNAL(anchorClicked(QUrl)), this, SLOT(on\_textBrowser\_anchorClicked(QUrl)));

connect(ui->nextButton, SIGNAL(clicked()), ui->textBrowser, SLOT(forward()));

connect(ui->backButton, SIGNAL(clicked()), ui->textBrowser, SLOT(backward()));

connect(ui->homeButton, SIGNAL(clicked()), ui->textBrowser, SLOT(home()));

connect(ui->textBrowser, SIGNAL(forwardAvailable(bool)), ui->nextButton, SLOT(setEnabled(bool)));

connect(ui->textBrowser, SIGNAL(backwaЫrdAvailable(bool)), ui->backButton, SLOT(setEnabled(bool)));

ui->textBrowser->*setSource*(QUrl::fromLocalFile("./../CourseWork/helpInformation/index.html"));

Полный код проекта (включая .qss файл) находится на github:

<https://github.com/Kaspener/SportSchedule>