**Электронный дневник студента**

**Техническое задание**

**Проект выполнили:**

**Разработчик: Айдашев Наиль.**

**Тестировщик: Замилов Тимур.**

**Аналитик: Некрасов Сергей.**

**Содержание**

1. [Введение.](#Введение)
   1. [Цель](#Цель).
   2. [Краткая сводка возможностей](#Краткаясводкавозможностей).
   3. [Определения, акронимы и сокращения](#Определенияакронимысокращения).
   4. [Краткое содержание](#Краткоесодержание).
2. [Обзор системы.](#Обзорсистемы)
   1. [Определение типов пользователей системы.](#Определениетиповпользователейсистемы)
   2. [Окружение системы](#Окружениесистемы).
   3. [Обзор вариантов использований](#Обзорвариантовиспользований).
   4. [Нефункциональные требования](#Нефункциональныетребования).
3. [Детальные требования](#Детальныетребования).
   1. [Функциональные требования.](#Функциональныетребования)
   2. [Требования к пользовательским интерфейсам.](#Требованиякпользовательскиминтерфейсам)
   3. [Требования к логической структуре БД.](#ТребованияклогическойструктуреБД)

Введение

Цель

Цель документа - определить и описать функциональные и нефункциональные требования к разработке СЭД студента с целью обеспечить студентам и администраторам удобный и эффективный инструмент для ведения учебного процесса, управления информацией об учебных предметах, оценках, заданиях, расписании и личных данных.

Краткая сводка возможностей

1. Авторизация и профиль студента: Возможность входа в СЭД с уникальной учетной записью студента и настройка личного профиля с контактной информацией, фотографией и другими личными данными.

2. Просмотр расписания занятий: Возможность просмотра расписания занятий на день, неделю или месяц для планирования учебного времени.

3. Управление учебными предметами и оценками: Просмотр оценок по учебным предметам, домашних заданий, сдачи работ для отслеживания учебного прогресса.

4. Уведомления и события: Получение уведомлений о ближайших заданиях, контрольных работах, изменениях в расписании и других важных событиях.

5. Ролевая структура: Разделение пользователей на студентов и администраторов с соответствующими правами доступа и функционалом.

Определения, акронимы и сокращения

Use case – сценарный план взаимодействия пользователя с программным продуктом, в котором чётко прописаны шаги для достижения того или иного результата.

User story – способ описания требований к разрабатываемой системе, сформулированных как одно или более предложений на повседневном или деловом языке пользователя.

C# - современный объектно-ориентированный язык программирования общего назначения.

Microsoft Visual Studio – лучшая интегрированная среда разработки для создания многофункциональных привлекательных кроссплатформенных приложений для Windows, Mac, Linux, IOS и Android.

MySQL – свободная реляционная система управления базами данных.

MySQL Workbench – это кроссплатформенный инструмент проектирования реляционных баз данных с открытым кодом, который добавляет функциональность и упрощает разработку MySQL и SQL.

СЭД – система электронного дневника.

Краткое содержание

Глава 2: происходит обзор типов системы, их определение, а также их возможности в приложении. Расписываются базовые user stories пользователей, а затем идёт перечисление программ и инструментов, при помощи которых создаётся приложение.

Глава 3: расписываются детальные требования к приложению, даются примеры use cases на основе user stories, описываются макеты приложения, а также приводится схема базы данных.

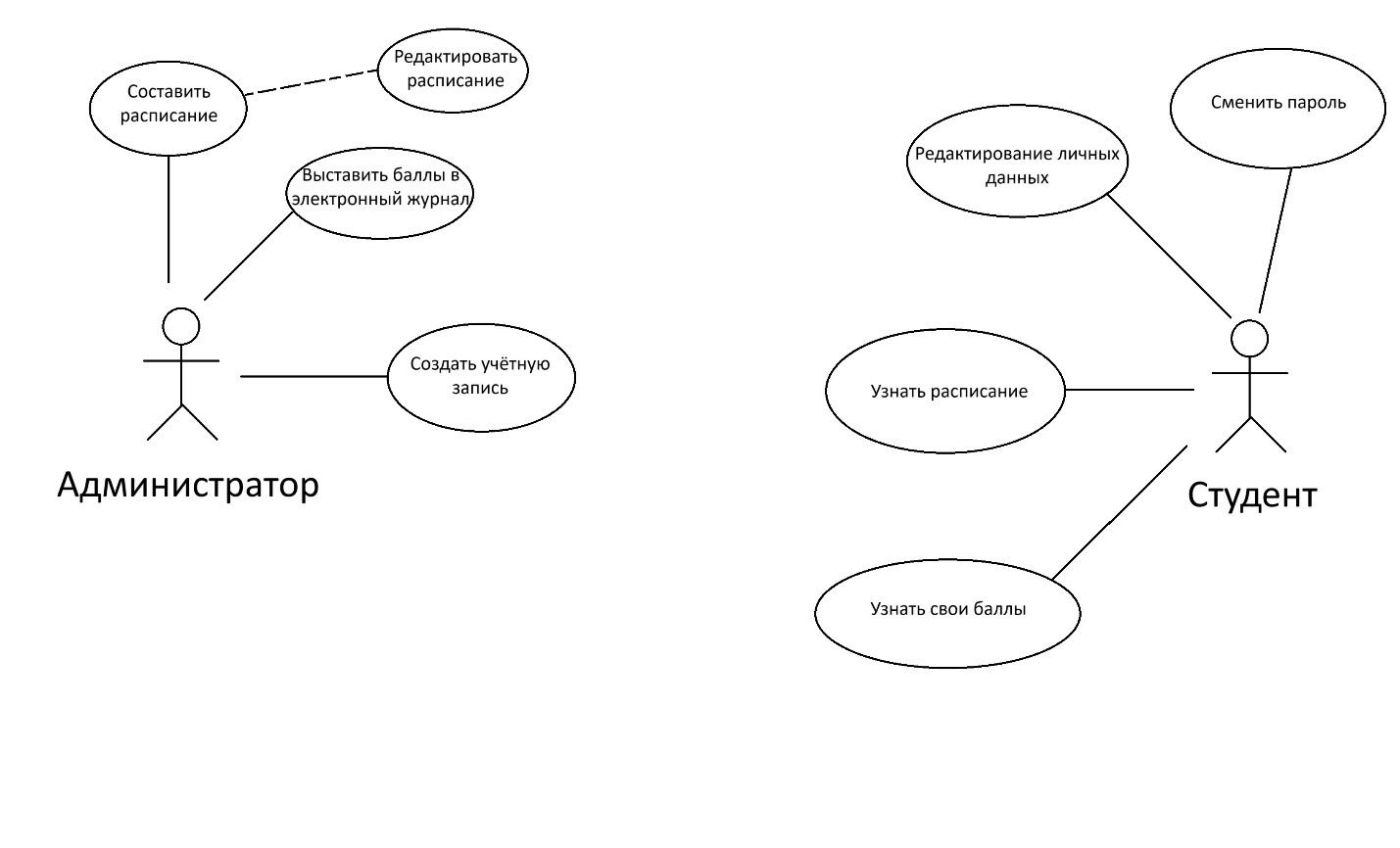
Обзор системы

Определение типов пользователей системы

Студент - пользователь системы электронного дневника, который имеет доступ к своему расписанию, информации об учебных предметах, оценкам, домашним заданиям и личным данным, а также может просматривать уведомления о предстоящих заданиях.

Администратор - пользователь с расширенными правами, который отвечает за управление системой электронного дневника. Он контролирует учетные записи пользователей и управление данными, безопасностью и работой системы. Обладает доступом к оценкам и отметкам о посещаемости студентов, а также к управлению учебным процессом через систему.

Окружение системы



Обзор вариантов использований

Варианты историй студента:

1. Как студент, я хочу иметь возможность авторизоваться в системе с использованием уникальной учетной записи, чтобы получить доступ к своему электронному дневнику.

2. Как студент, я хочу просматривать расписание занятий, чтобы быть в курсе своего учебного графика.

3. Как студент, я хочу иметь возможность просматривать и редактировать свои личные данные (контактную информацию и фотографию), чтобы они всегда были актуальными.

4. Как студент, я хочу иметь доступ к списку учебных предметов, просматривать оценки и домашние задания по каждому предмету, чтобы следить за своим учебным прогрессом.

5. Как студент, я хотел бы получать уведомления о ближайших заданиях, контрольных работах, предстоящих событиях и изменениях в расписании, чтобы не пропустить важную информацию.

Варианты историй администратора:

1. Как администратор, я хотел бы иметь доступ к созданию новых учебных записей

2. Как администратор, я хотел бы иметь возможность оценивать и отмечать посещаемость студентов, чтобы следить за их успехами в учебе.

Нефункциональные требования

Язык программирования: C#

Cреда разработки: Microsoft Visual Studio, MySQL Workbench

Программы, используемые для создания СЭД: MySQL

Детальные требования

Функциональные требования.

1) User story: Как студент, я хочу иметь возможность авторизоваться в системе с использованием уникальной учетной записи, чтобы получить доступ к своему электронному дневнику.

Use case: 1. Пользователь входит в систему электронного дневника.

2) User story: Как студент, я хочу просматривать расписание занятий, чтобы быть в курсе своего учебного графика.

Use case: 1. Пользователь входит в СЭД.

2. Переходит на окно “Профиль пользователя”.

3. Выбирает нужную неделю.

4. Производит просмотр расписания по дням.

3) User stories: Как студент, я хочу иметь возможность просматривать и редактировать свои личные данные (контактную информацию и фотографию), чтобы они всегда были актуальными.

Use case: 1. Пользователь входит в СЭД.

2. Переходит в окно “Профиль пользователя”.

3. Переходит в окно “Редактирование личных данных”.

4. Производит редактирование своих личных данных.

4) User story: Как администратор, я хотел бы иметь доступ к созданию новых учебных записей

Use case: 1. Пользователь входит в СЭД.

2. Переходит в окно “Панель админа”.

3. Переходит в окно “Создание пользователя”.

4. Производит создание новой учебной записи.

5) User story: Как администратор, я хотел бы иметь возможность оценивать и отмечать посещаемость студентов, чтобы следить за их успехами в учебе.

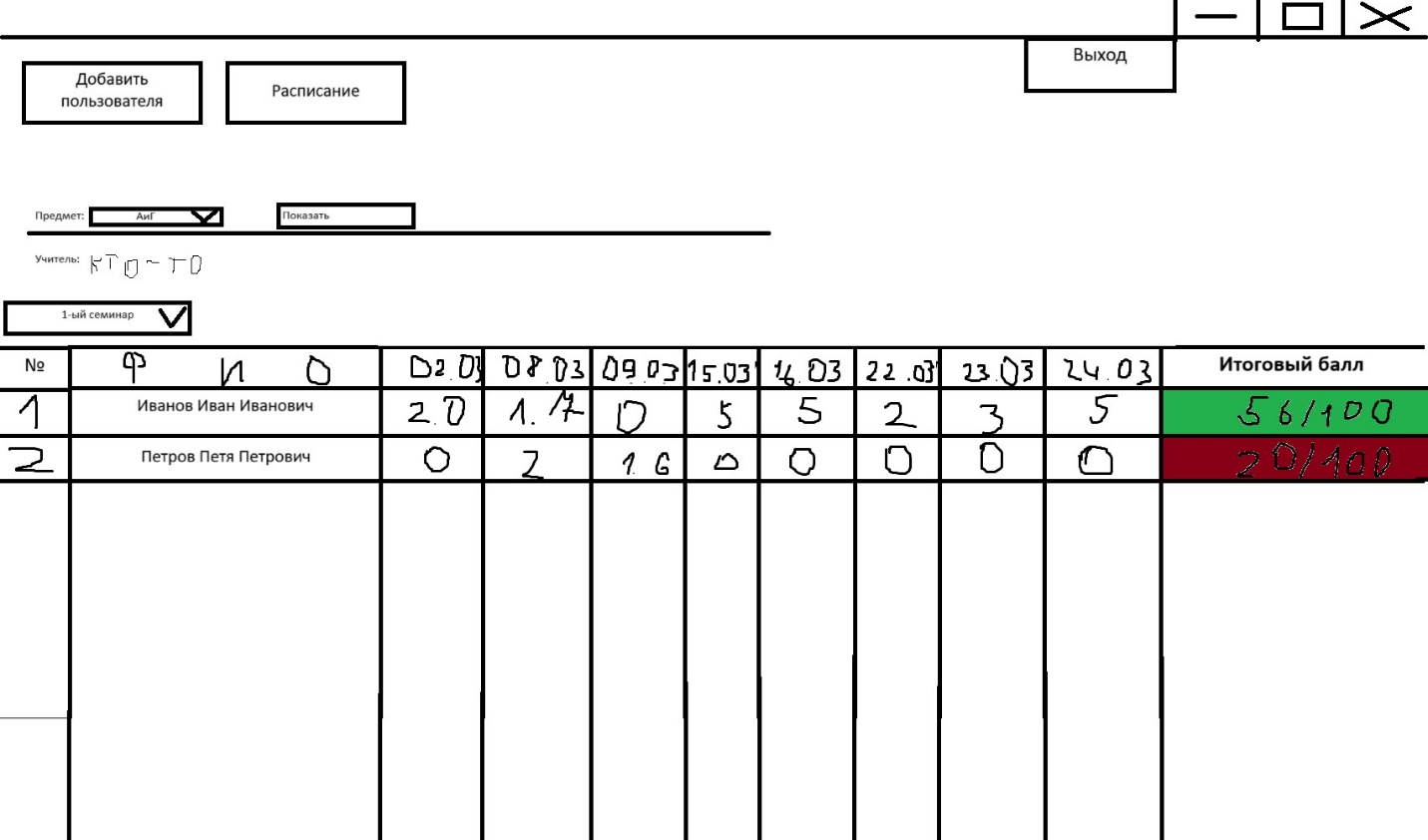
Use case: 1. Пользователь входит в СЭД.

2. Переходит в окно “Панель админа”.

3. Производит оценку и отметку посещаемости студента.

Требования к пользовательским интерфейсам.

1. Окно “Панель админа”.



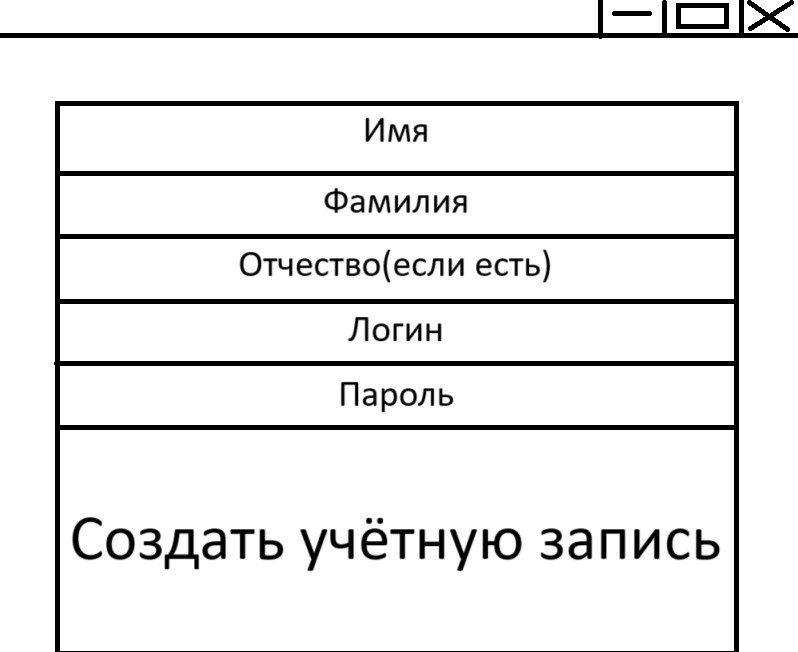
На данном макете показана начальное окно “Панель админа” после вхождения в учётную запись.

На окне показана таблица со студентами группы. В таблице указаны их ФИО, даты занятий, совпадающие с расписанием определённого предмета, проставленные баллы за проделанную работу, а также итоговый балл за семестр. Над таблицей также есть кнопка, с возможностью выбрать нужный семинар.

Далее у админа есть кнопка “Создагие пользователя” – возможность добавления студента в таблицу, кнопка “Расписание” – возможность перейти в окно “Расписание”, а также кнопка “Выход” – возможность выйти из учётной записи.

Также, чтобы админ мог выставлять оценки отдельно на каждый предмет, на окне существует кнопка с выпадающим списком предметов, в которой можно выбрать нужный предмет, а в таблице даты занятий будет автоматически меняться в зависимости от выбранного предмета .

1. Окно “Создание пользователя”

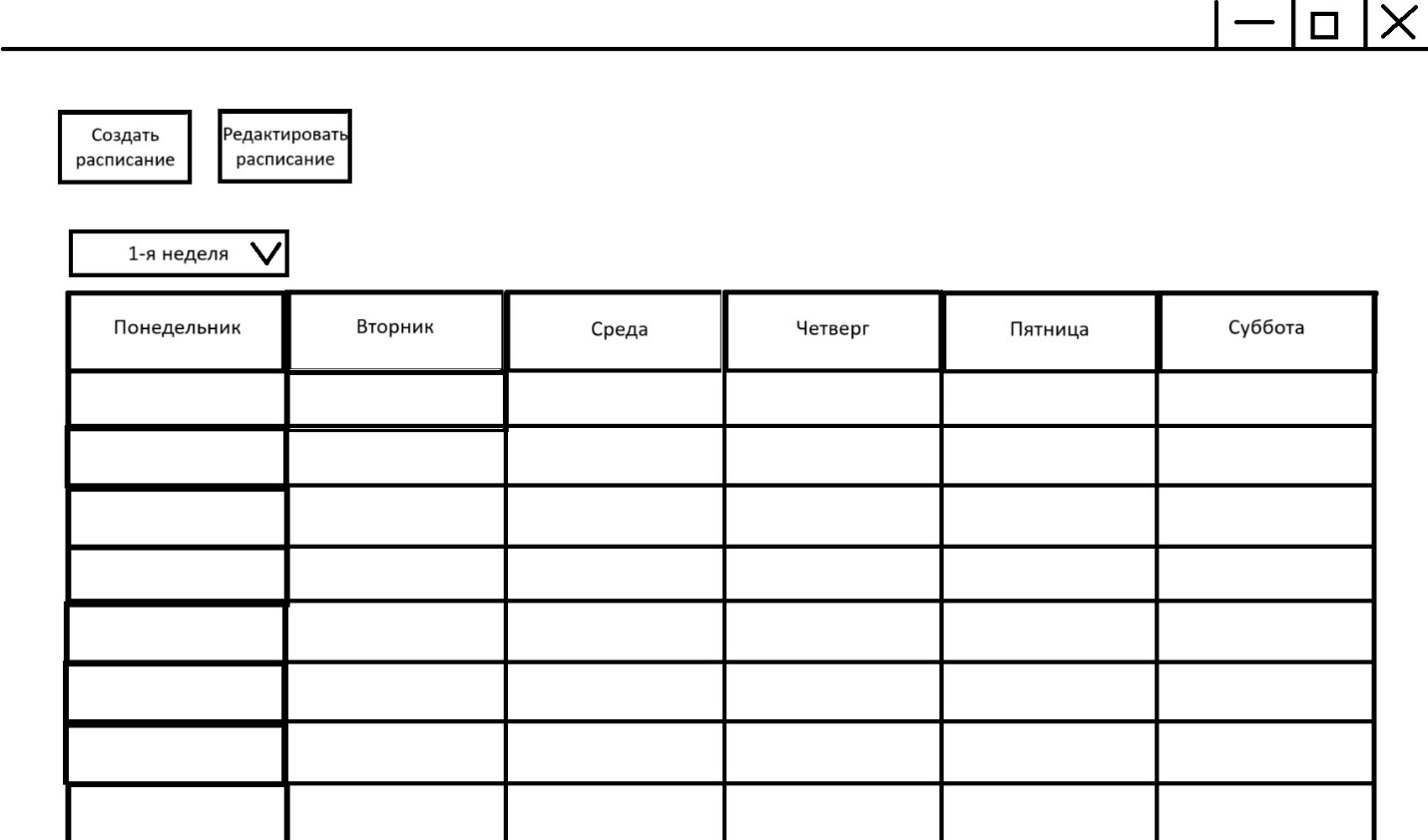


На данном макете показано окно “Создание пользователя”, которое появится, если админ прожмёт кнопку “Создание пользователя” в окне “Панель админа”.

Далее админу нужно заполнить личные данные студента, а также придумать логин и пароль для создания учётной записи.

Чтобы завершить создание учётной записи, нужно прожать кнопку “Создать учётную запись”.

1. Окно “Расписание”.



На данном макете показано окно “Расписание”, которое появится, если админ прожмёт кнопку “Расписание” в окне “Панель админа”.

Перед админом появится окно с таблицей с расписанием, кнопка “Создать расписание” с возможностью создать расписание, кнопка “Редактировать расписание”, а кнопка с выбором недели для просмотра расписания.

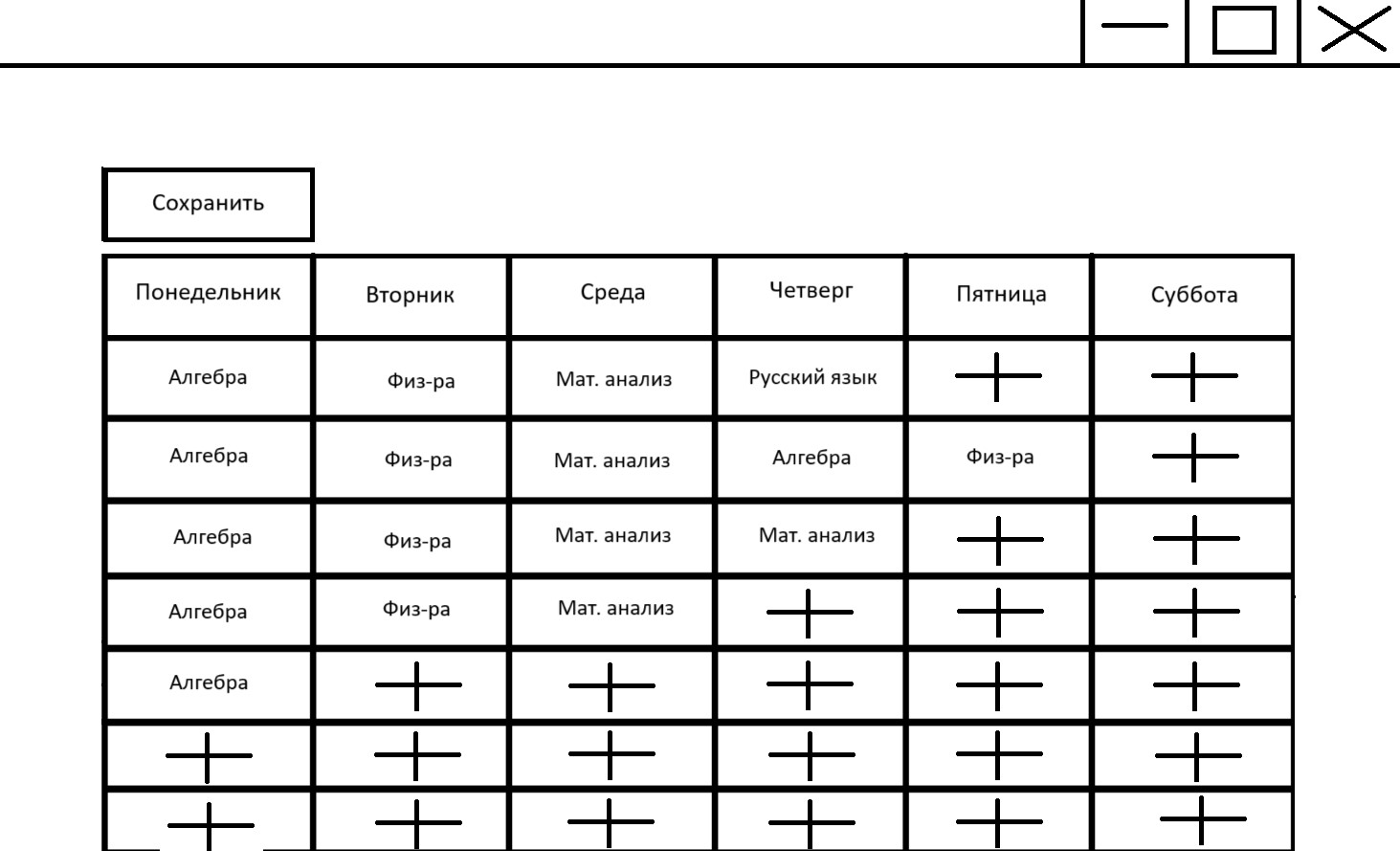
4) Окно “Понедельник 1-ой недели”.



На данном макете показано окно “Понедельник 1-ой недели”, которая появится, если администратор нажмёт кнопку “Создать расписание” в окне “Расписание”.

Перед администратором появится окошко с формой для заполнения даты 1-го понедельника, от которого расписание должно начаться, а также появится кнопка “Создать”, чтобы перейти к окну “Создание расписания”.

5) Окно “Создание расписания”.

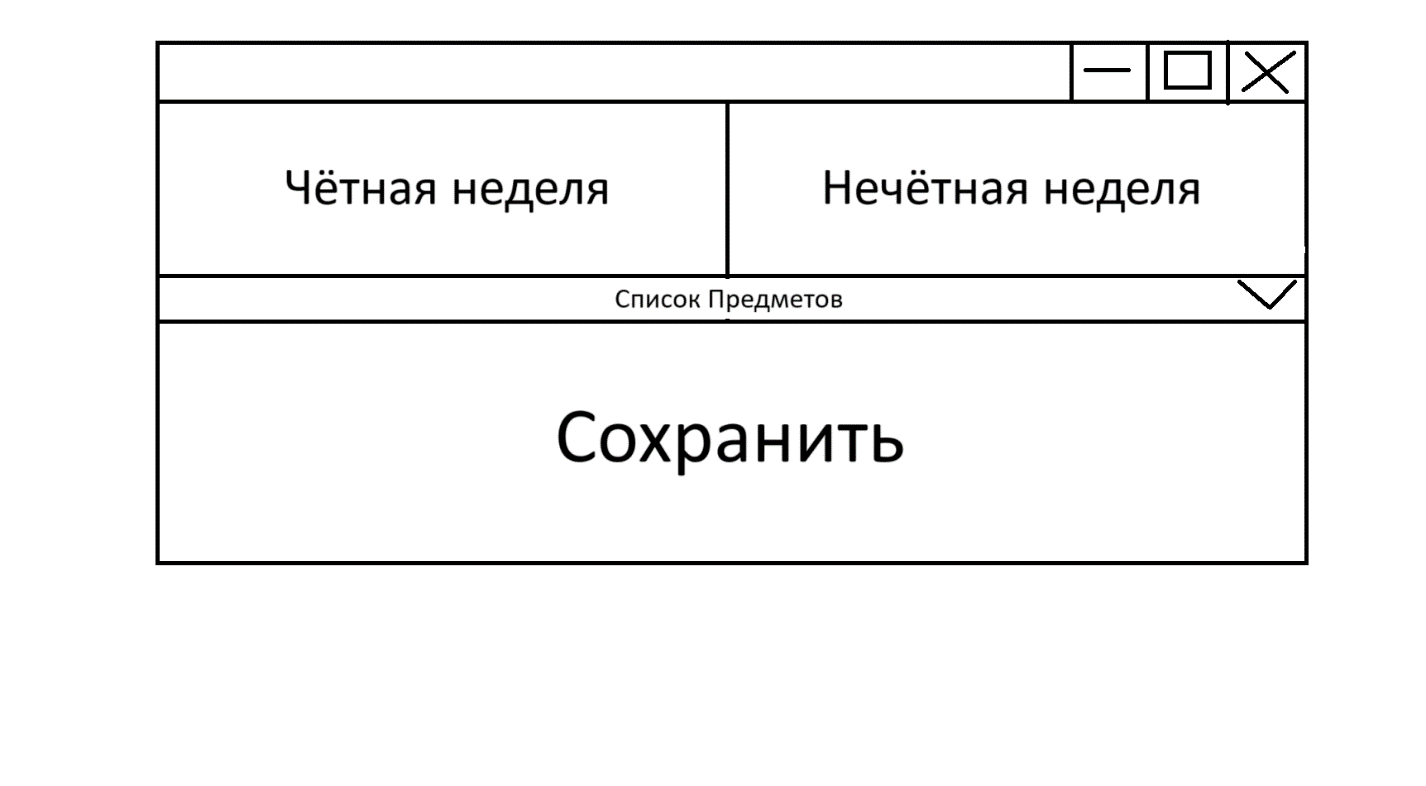


На данном макете показано окно “Создание расписания”, которая появится, если администратор нажмёт кнопку “Создать” в окне “Понедельник 1-ой недели”.

Перед админом появится окно с таблицей, первая строка которой будет заполнена днями недели, чтобы администратору было удобнее заполнять расписание. Также над таблицей будет кнопка “Сохранить” с возможностью сохранить учебное расписание.

Чтобы администратору добавить нужный предмет, ему нужно будет нажать на плюсик в ячейки таблицы, чтобы перейти в окно “Учебная пара”.

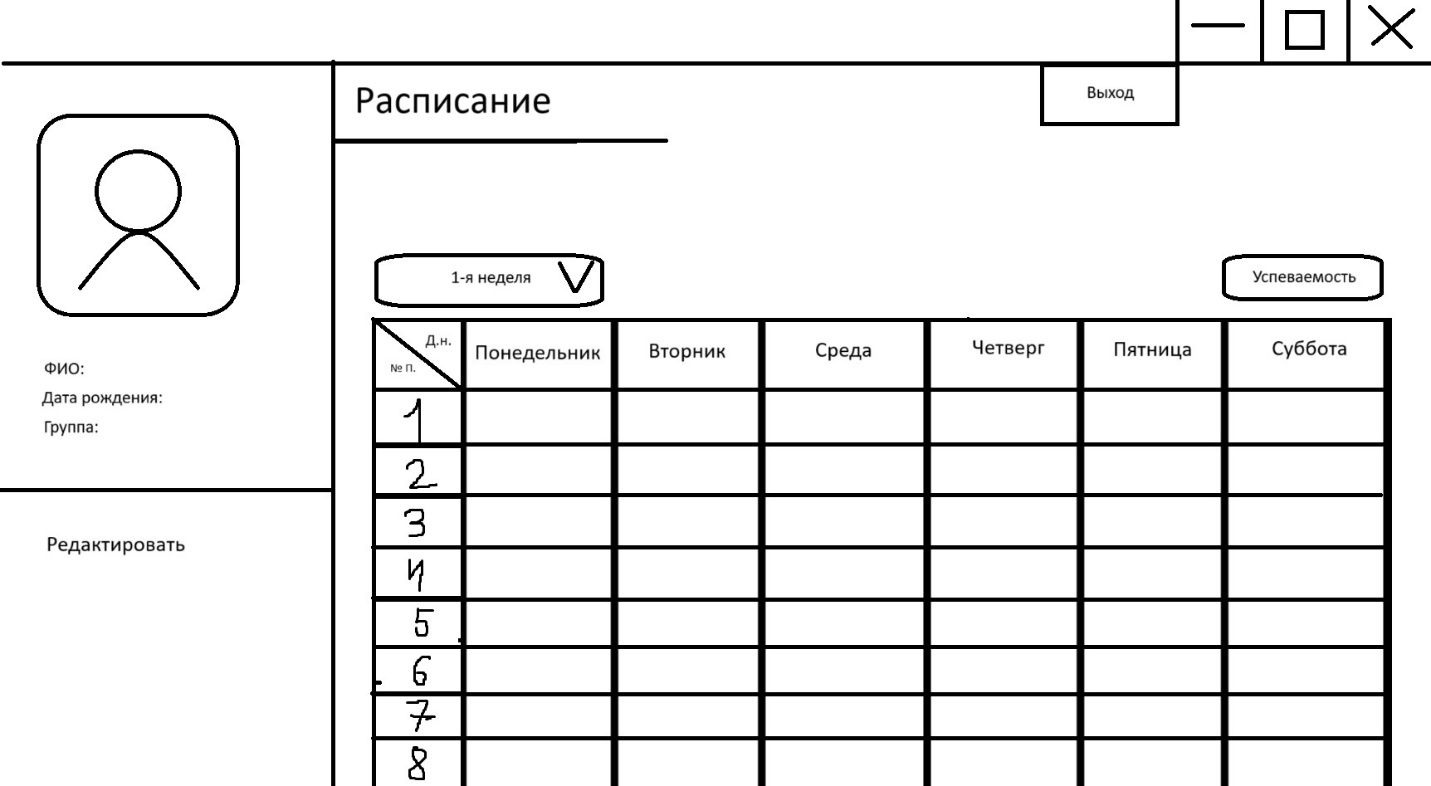
6) Окно “Создание учебной пары”.



На данном макете показано окно “Создание учебной пары”, которая появится, если администратор нажмёт на плюсик в таблице в окне “Создание расписания”.

Перед админом появится 2 кнопки с разной чётностью недели, а также кнопка “Список Предметов” с раскрывающимся списком учебных предметов. Чтобы сохранить введённые данные, нужно нажать кнопку “Сохранить”.

7) Окно “Профиль пользователя”.



На данном макете изображено окно “Профиль пользователя”, которая появится, когда студент войдёт в учётную запись.

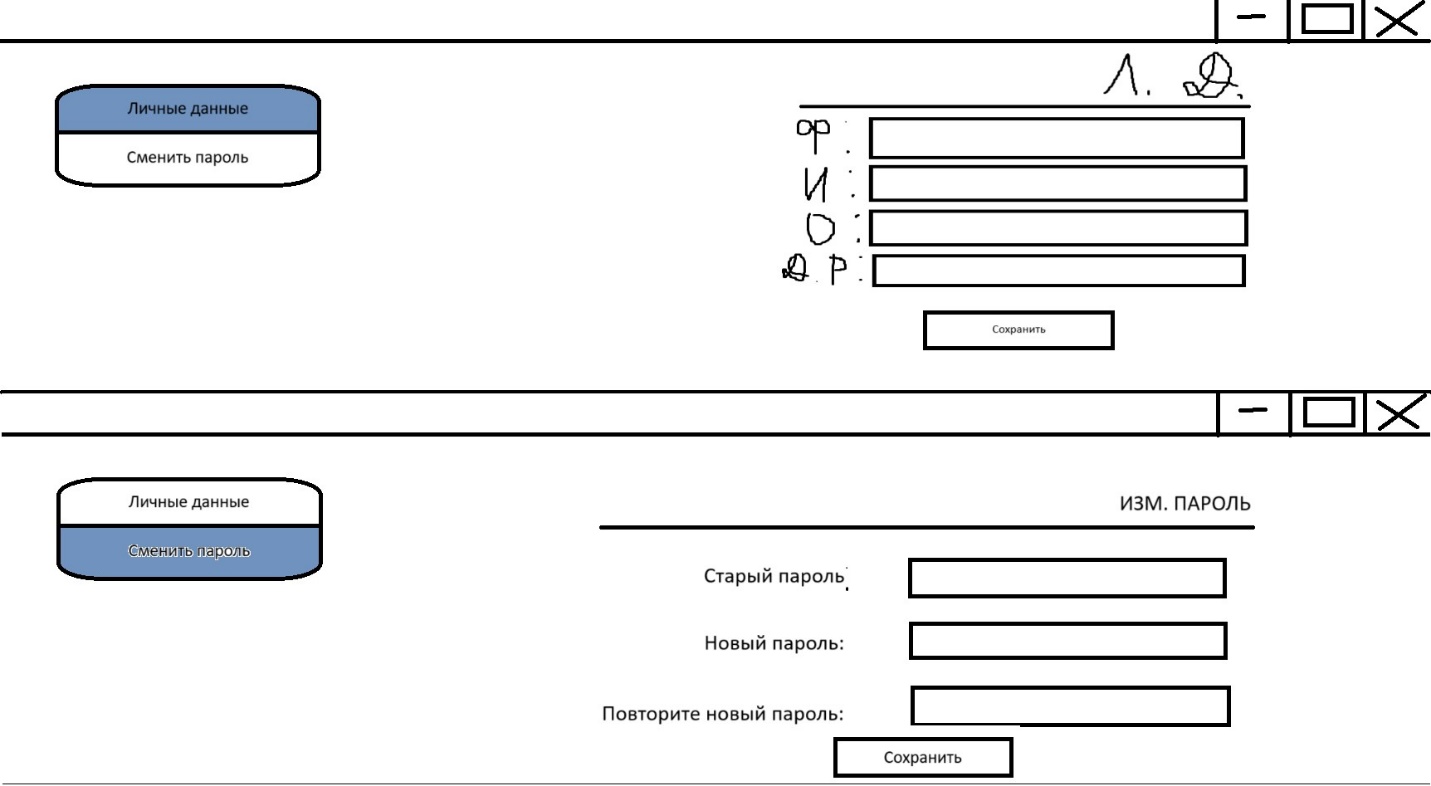
В окне находится фотография лица учащегося, его личные данные, а также таблица с расписанием занятий.

У учащегося есть возможность редактировать личные данные, для этого ему нужно нажать на “Редактирование личных данных”.

Над таблицей находится кнопка с выбором нужной неделю, а также кнопка “Успеваемость”, если студенту нужно перейти в окно “Успеваемость”.

Также у учащегося есть возможность выйти с учётной записи, для этого ему нужно нажать на кнопку “Выход”.

8) Окно “Редактирование личных данных”.



На данном макете изображено окно “Редактирование личных данных”, которая появится, если студент нажмёт на кнопку “Редактировать” в окне “Профиль пользователя”.

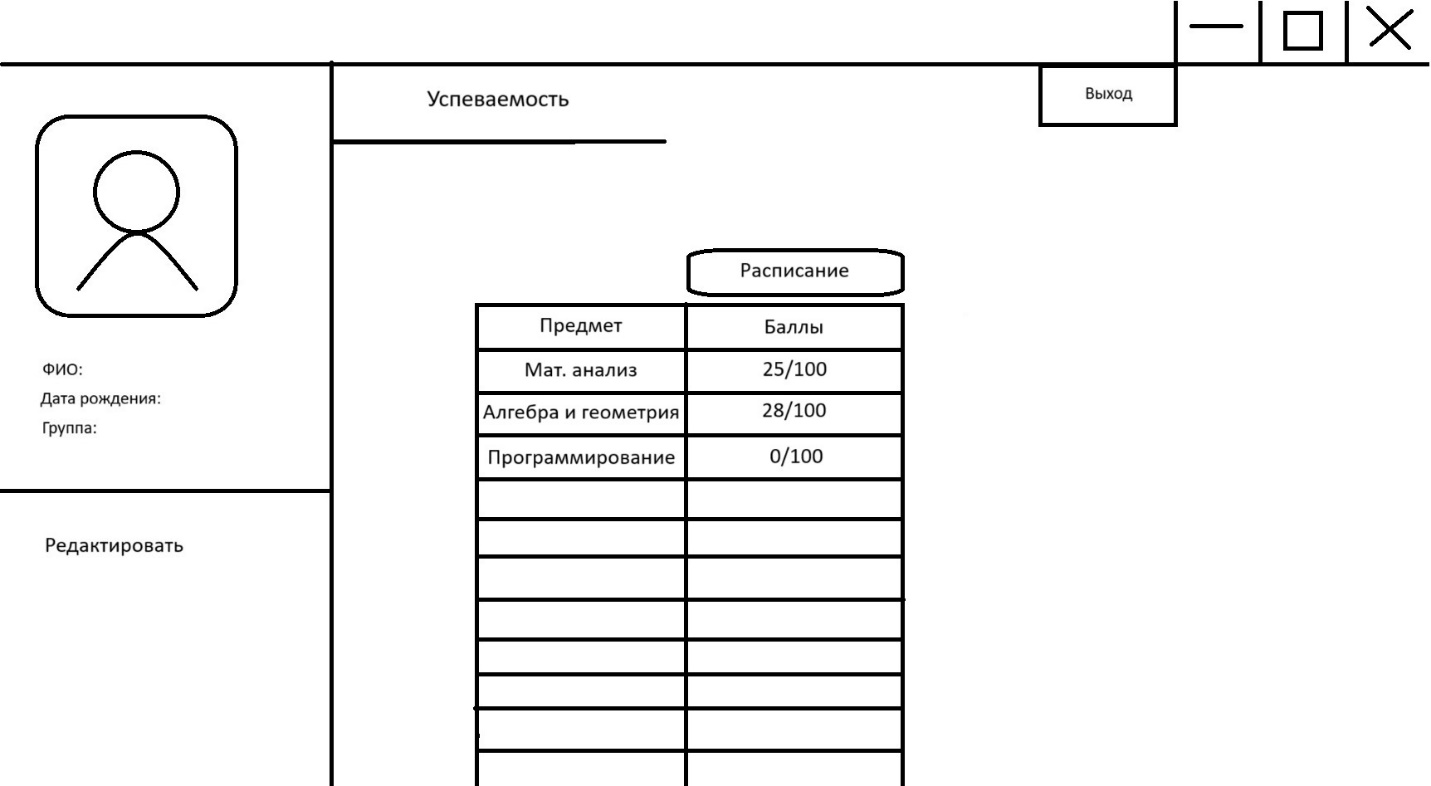
В окне находится 2 вкладки: “Личные данные” и “Cменить пароль”. В зависимости от их выбора формы заполнения в окне будут меняться.

Во вкладке “Личные данные” студент может поменять свою фамилию, имя, отчество и дату рождения.

Во вкладке “Сменить пароль” студент может поменять свой пароль. Для этого ему нужно вписать свой старый пароль, затем написать свой новый пароль, а в конце повторить его для подтверждения.

Чтобы сохранить внесённые изменения, учащийся должен нажать кнопку “Сохранить”.

9) Окно “Успеваемость”.



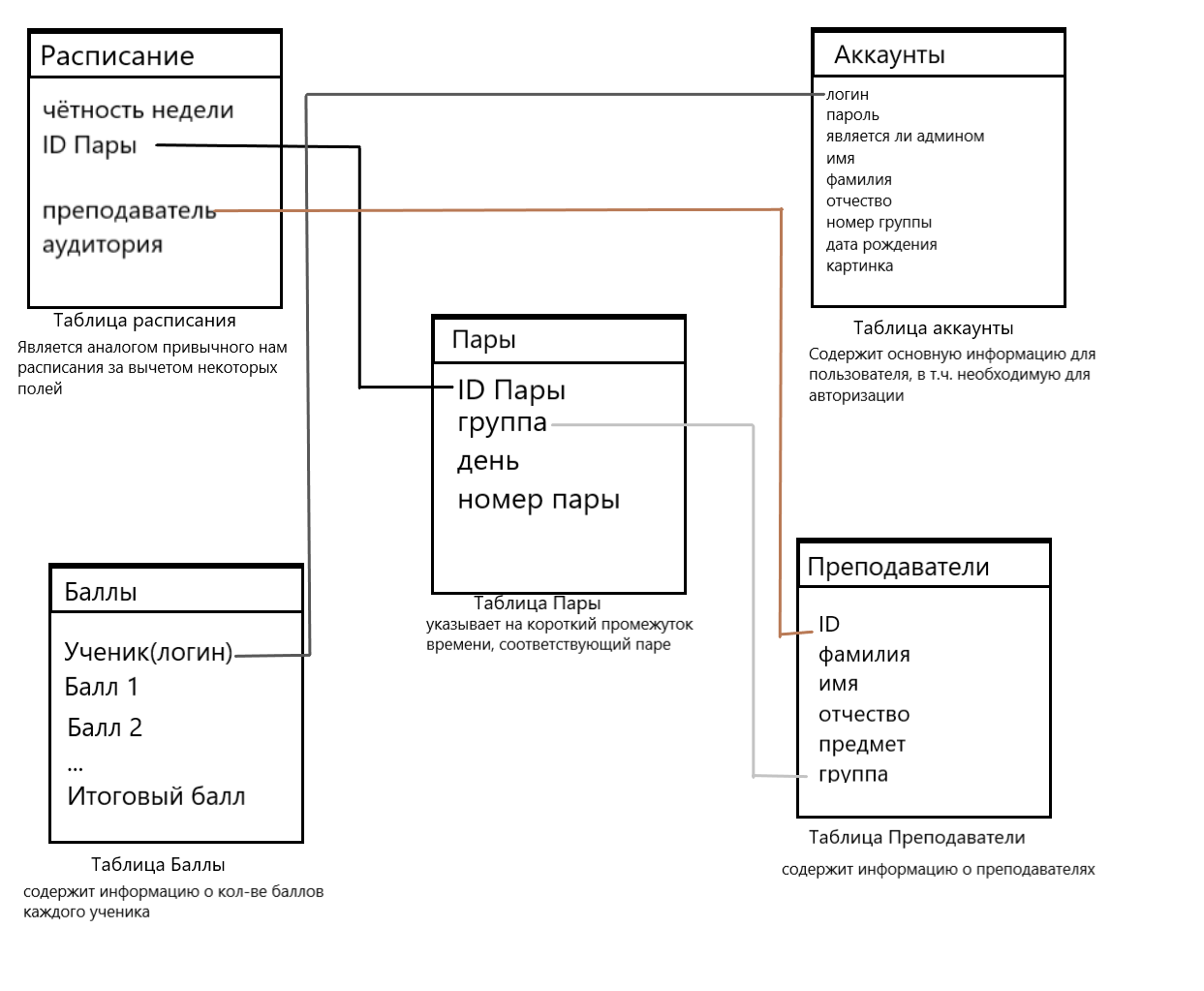
На данном макете изображено окно “Успеваемость”, которая появляется, когда пользователь нажимает на кнопку “Успеваемость” в окне “Профиль пользователя”.

В окне находится фотография лица учащегося, его личные данные, а также таблица с текущами баллами по предметам.

У учащегося есть возможность выйти с учётной записи, для этого ему нужно нажать на кнопку “Выход”.

Чтобы вернуться в окно “Профиль пользователя”, над таблицей с баллами находится кнопка “Расписание”.

Требования к логической структуре БД

****