

Разработка базового расписания

Прототипы экранных форм:

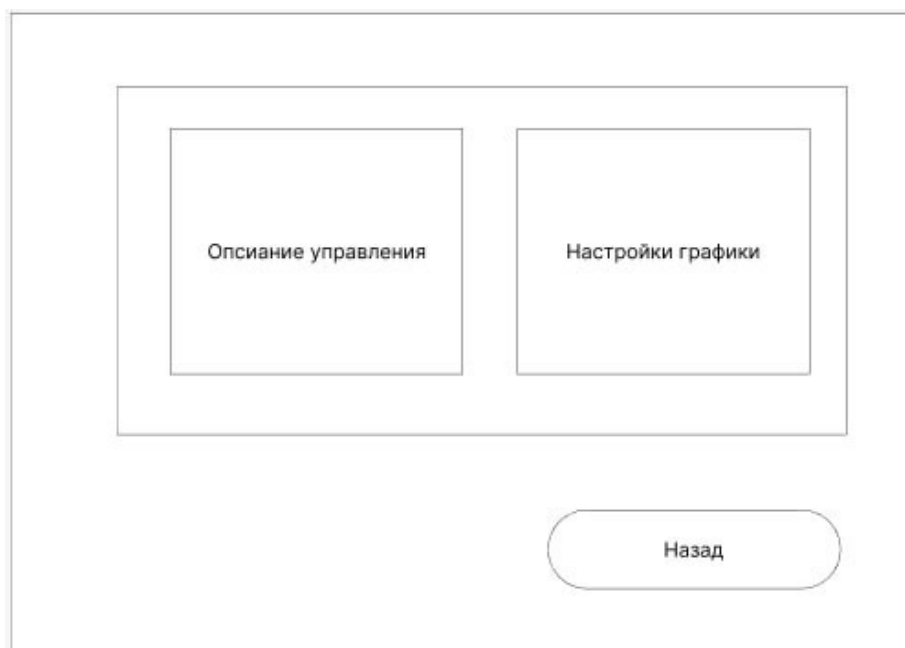
1. Прототип главного экрана: Главный экран состоит из текстовой области и группы кнопок: начать, настройки, выход.

При нажатии кнопки «Начать» открывается лабораторная работа (прототип 3), при нажатии кнопки «Настройки» открывается меню настроек (прототип 2), при нажатии кнопки «Выход» программа закрывается.



2. Прототип меню настроек: меню настроек состоит из текстовой области «описание управления», «Настройки графики» (набор кнопок для изменения настроек графики) и кнопки «Назад».

При нажатии кнопки «Назад» открывается главное меню (прототип 1).



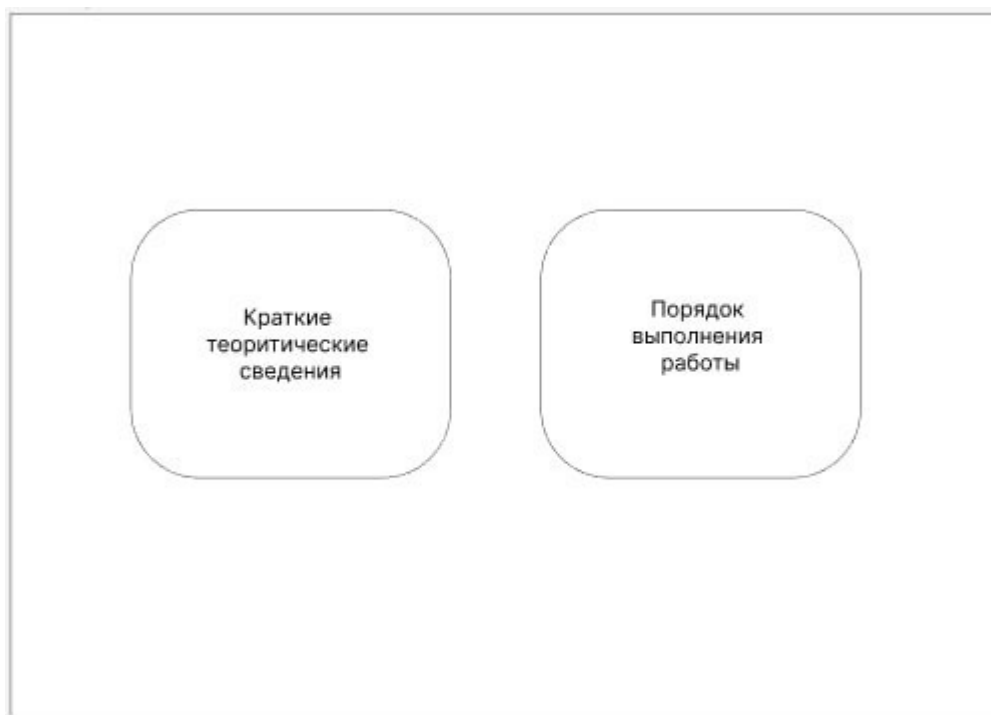
3. Прототип лабораторной работы: лабораторная работа состоит из блока текста, который нужен для обучения работе в среде, и рабочей области.

В рабочей области находится лазерная установка, таблица для ввода данных(прототип 6), графиков(прототип 7) и экрана информации (прототип 4).



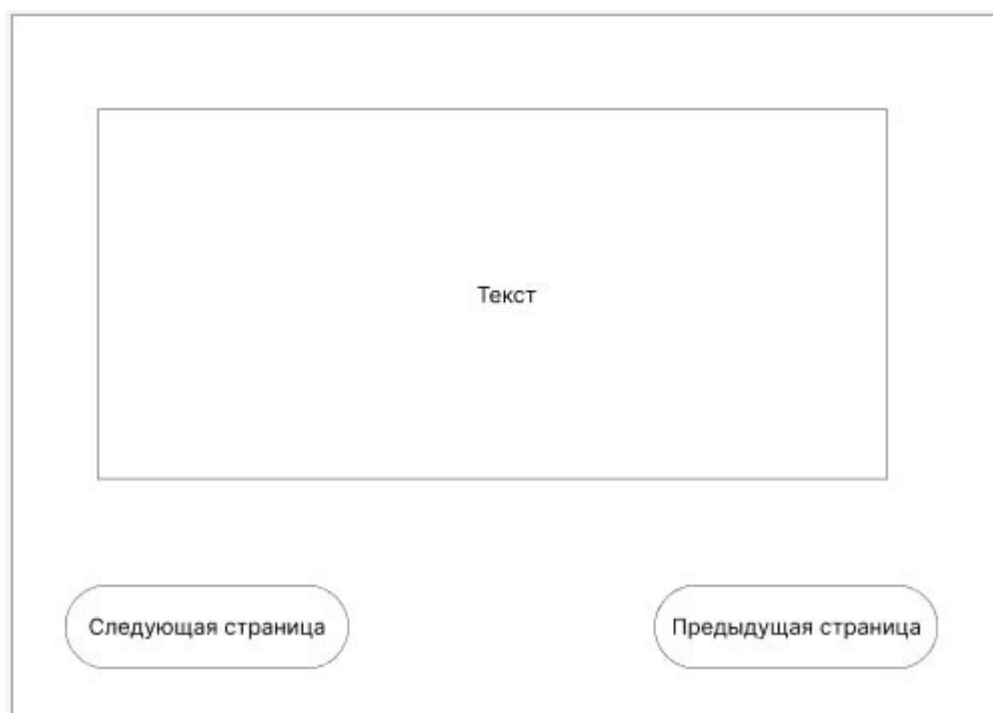
4. Прототип экрана информации: экран информации состоит из кнопок «Краткие теоретические сведения» и «Порядок выполнения работы».

При нажатии кнопок открывается окно с текстом (прототип 5).



5. Прототип окна с текстом: окно с текстом состоит из блока с текстом и кнопок «Следующая страница» и «Предыдущая страница».

При нажатии кнопки «Следующая страница» открывается следующая страница (прототип 5), при нажатии кнопки «Предыдущая страница» открывается предыдущая страница (прототип 5).



6. Прототип таблицы: таблица состоит из текстовой области и области ввода значений.

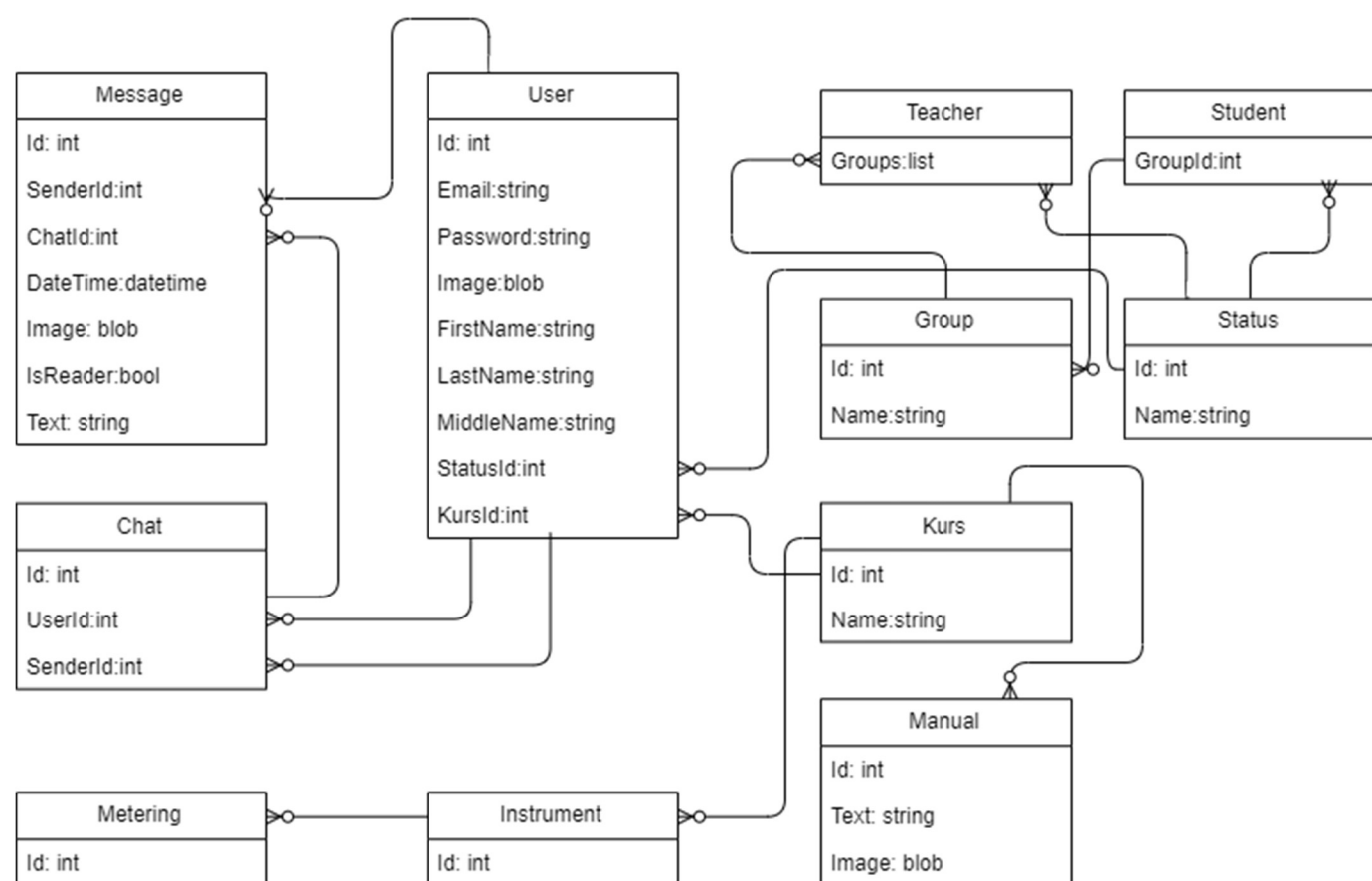


7. Прототип графиков: стена с графиками состоит из изображений.

График

График

Диаграмма сущностей:



Разработка API системы:

1. LoadData

Функция предназначена для считывания информации из базы данных и представление её в виде списка объектов. Данная функция использует Entity Framework для сопоставления информации о сущности с объектами.

Входная информация: объект String содержащий в себе информацию для подключения к базе данных.

Выходная информация: список объектов List<T>.

2. SyncData

Функция предназначена для синхронизации информации из локальной базы данных программы с базой данных, хранимой на сервере. Осуществляется данный процесс с помощью репликации базы данных.

Входная информация: массив String содержащий в себе информацию для подключения к базе данных и информацию о выбираемых таблицах для репликации.

Выходная информация: enum-статус завершения операции.

3. SendMessage

Функция предназначена для отправки сообщения от одного пользователя к другому пользователю.

Входная информация: ID – отправителя, ID – получателя, сообщение.

Выходная информация: ID – отправителя, сообщение, дата отправки сообщения.

4. UserAuthentication

Функция предназначена для аутентификации пользователя в приложении.

Входная информация: логин, пароль.

Выходная информация: enum-статус завершения операции.

5. LoadFile

Функция предназначена для загрузки пользователем файла в программу.

Входная информация: путь к файлу.

Выходная информация: enum-статус завершения операции.

6. Rotate

Функция предназначена для поворота камеры пользователя.

Входная информация: нажатие клавиши мыши.

Выходная информация: вектор поворота.

7. Conformity

Функция предназначена для показа значений на приборах. В зависимости от выбора режима, значения различаются.

Входная информация: нажатие клавиши мыши.

Выходная информация: элемент массива.

8. Update

В функции описывается движение камеры при работе с ассистентом. Положение камеры меняется в зависимости от подсказки, которую помощник дает.

Входная информация: номер элемента массива строк.

Выходная информация: вектор поворота камеры.

9. ButtonActivator

Функция предназначена для активации кнопок нажатием на нее мышью.

Входная информация: номер элемента массива кнопок.

Выходная информация: true или false.

10. IEnumerator RuningText

Функция предназначена для последовательного вывода текста в работе ассистента.

Входная информация: время задержки, имя файла.

Выходная информация: массив строк.

Иерархическая структура работ:

1. Разработка технического задания

1.1. Сбор требований;

1.2. Определение стадий и этапов разработки

1.2.1. Определение стадий разработки;

1.2.2. Определение сроков разработки;

1.3. Общее описание

1.3.1. Назначение продукта;

1.3.2. Взаимодействие продукта;

1.3.3. Допущения и ограничения продукта;

1.3.4. Определение функций продукта;

2. Разработка приложения

2.1. Backend-разработка

2.1.1. Проектирование базы данных

- 2.1.1.1. Определение структуры базы данных;
 - 2.1.1.2. Определение связей между сущностями;
 - 2.1.1.3. Определение взаимодействия с базой данных;
- 2.1.2. Разработка API приложения
 - 2.1.2.1. LoadData;
 - 2.1.2.2. SyncData;
 - 2.1.2.3. SendMessage;
 - 2.1.2.4. UserAuthentication;
 - 2.1.2.5. LoadFile;
 - 2.1.2.6. Rotate;
 - 2.1.2.7. Conformity;
 - 2.1.2.8. Update;
 - 2.1.2.9. ButtonActivator;
 - 2.1.2.10. IEnumerator RuningText.
- 2.1.3. Сетевое взаимодействие
 - 2.1.3.1. Определение протокола взаимодействия;
 - 2.1.3.2. Обеспечение защищённости соединения;
 - 2.1.3.3. Определение местоположения хранения данных;
- 2.1.4. Взаимодействие с UI
 - 2.1.4.1. Определение модели привязки данных;
 - 2.1.4.2. Создание механизма обновления данных;
- 2.2. Frontend-разработка
 - 2.2.1. Дизайн
 - 2.2.1.1. Разработка макетов страниц
 - 2.2.1.1.1. Разработка макета главного экрана;
 - 2.2.1.1.2. Разработка макета меню настроек;
 - 2.2.1.1.3. Разработка макета лабораторной работы;
 - 2.2.1.1.4. Разработка макета экрана информации;
 - 2.2.1.1.5. Разработка макета окна с текстом;
 - 2.2.1.1.6. Разработка макета таблиц;
 - 2.2.1.1.7. Разработка макета графиков;
 - 2.2.1.2. Взаимодействие с backend
 - 2.2.1.2.1. Определение модели привязки данных;
 - 2.2.1.2.2. Настройка механизма обновления данных;
 - 2.2.1.2.3. Определение динамического взаимодействия с данными;
 - 2.2.1.3. Разработка общего стиль-кода приложения
 - 2.2.1.3.1. Создание иконок элементов;
 - 2.2.1.3.2. Определение стиля и размера шрифтов;
- 3. Приемо-сдаточные испытания
 - 3.1. Подготовка и проведение демонстрации;

- 3.2. Проведение испытаний;
- 4. Размещение приложения
 - 4.1. Аренда сервера;
 - 4.2. Развёртывание приложения;
 - 4.3. Размещение в магазине приложений;
- 5. Поддержка приложения
 - 5.1. Мониторинг работоспособности;
 - 5.2. Получение и обработка обратной связи;
 - 5.3. Улучшение работы приложения
 - 5.4. Добавление новой функциональности;

Оценка времени выполнения проекта по методу PERT:

Работы	Количество	Оптимистичные затраты	Пессимистичные затраты	Наиболее вероятные затраты
Создание сущностей	8	2	6	3
Создание макетов	7	20	60	40
Создание методов API	10	10	70	20

Средние трудозатраты:

$$\text{Создание сущностей} = \frac{6 + 4 \times 3 + 2}{6} = 3.3 \text{ чел.* час.}$$

$$\text{Создание макетов} = \frac{60 + 4 \times 40 + 20}{6} = 40 \text{ чел.* час.}$$

$$\text{Создание методов API} = \frac{70 + 4 \times 20 + 10}{6} = 26,6 \text{ чел.* час.}$$

Среднеквадратичная оценка:

$$\text{Создание сущностей} = \frac{6 - 2}{6} = 0,6 \text{ чел.* час.}$$

$$\text{Создание макетов} = \frac{60 - 20}{6} = 6,6 \text{ чел.* час.}$$

$$\text{Создание методов API} = \frac{70 - 10}{6} = 10 \text{ чел.* час.}$$

$E_{\text{общ.}}$:

$$E_{\text{общ.}} = 8 * 3.3 + 7 * 40 + 10 * 26,6 = 913.3 \text{ чел.* час.}$$

СКО_{общ.}:

$$\text{СКО}_{\text{общ.}} = \sqrt{8 * 0,6^2 + 7 * 6,6^2 + 10 * 10^2} = 24.2 \text{ чел.* час.}$$

Оценка суммарной трудоёмкости проекта с вероятностью 95%:

$$E_{95\%} = 913.3 + 2 * 24.2 = 958.7 \text{ чел.* час.}$$

Базовое расписание в виде диаграммы Ганта:

Представлено в файле – «Базовое расписание.pdf»