

# Разработка базового расписания

## Прототипы экранных форм:

### 1. Прототип главного окна

Главное окно, на котором представлены списки задач. Необходимо для отслеживания задач. Состоит из меню управления - набор кнопок для редактирования задач (основные функции по добавлению элементов, сохранению, редактированию), сегмента (список подзадач), названия сегмента - текстовое поле описывающее название сегмента, определяемое пользователем, карточки подзадачи - форма-кнопка, описывающая подзадачу (Прототип 2. Описание подзадачи) и статусбара - форма, описывающая статус выбранной подзадачи (Прототип 3).

The diagram illustrates the layout of the main window. It features a top bar labeled "Меню управления" (Management Menu). Below this, on the left side, is a vertical container. Inside this container, at the top, is a text input field labeled "Название сегмента" (Segment Name). Below the input field is a large rectangular area labeled "Прототип 2" (Prototip 2), which represents a task card. At the bottom of the main window is a horizontal bar labeled "Прототип 3" (Prototip 3), representing a status bar.



## 2. Прототип карточки подзадачи

Форма-кнопка включающая в себя поля: название карточки – текстовое поле заголовка подзадачи, описание карточки – текстовое поле описания подзадачи, прикрепленные файлы – список состоящий из названий и иконок прикрепленных файлов к выбранной подзадачи, фото – изображение пользователя создавшего подзадачу, имя пользователя – текстовое поле отображающее имя пользователя создавшего подзадачу. Служит отображением подзадачи.

Название карточки	
Описание задачи	
Прикрепленные файлы	
Фото	Имя пользователя

## 3. Прототип статусбара

Форма, описывающая статус выбранной подзадачи, состоит из: иконки статуса – круг, заполняющийся цветом, описывающим статус, статус – текстовое поле, определяющее статус подзадачи, статус выбранной подзадачи – заполняемая шкала прогресса выполнения задачи, кнопка фоновых задач – кнопка, открывающая раздел фоновых задач в приложении.

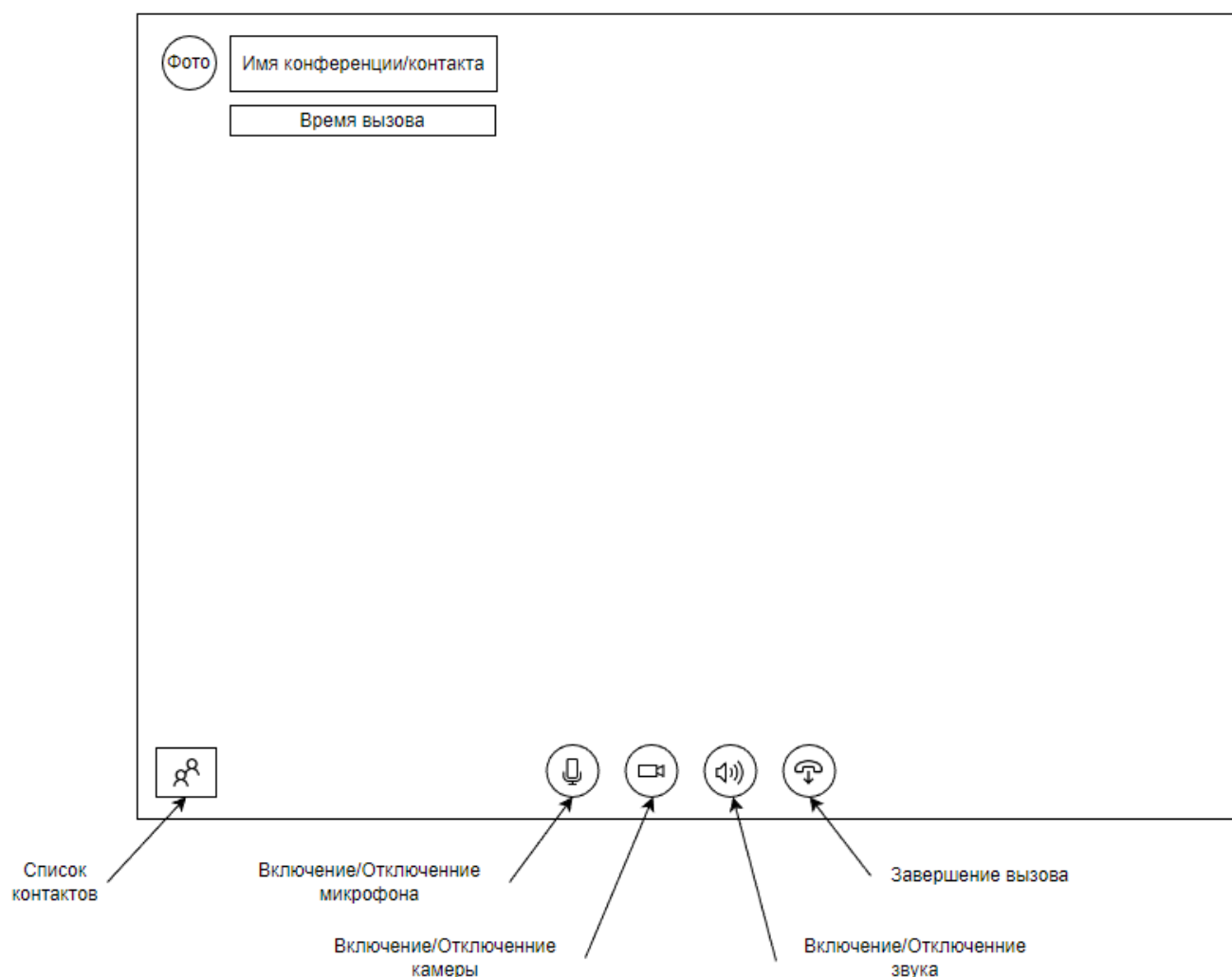
	Статус	Статус выбранной задачи	
---	--------	-------------------------	---

Иконка статуса

Кнопка фоновых задач

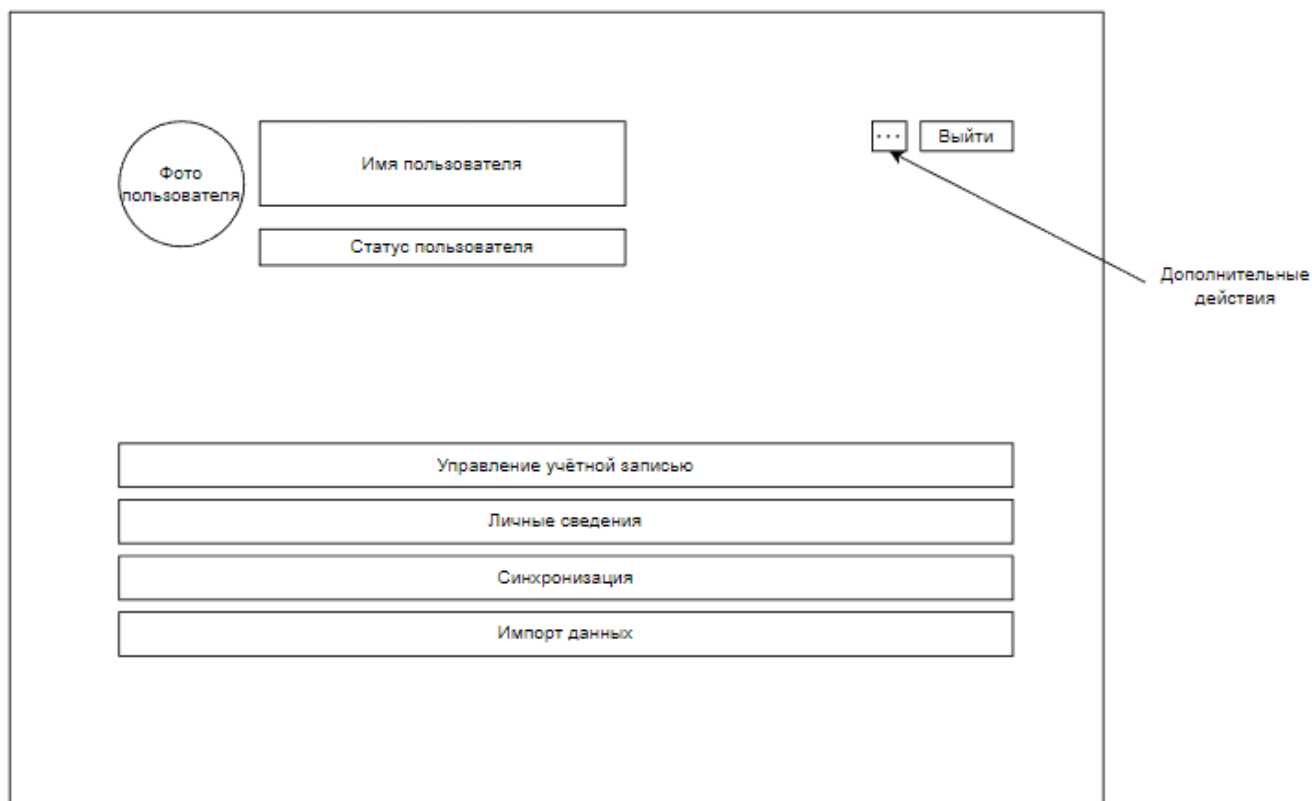
#### 4. Прототип окна аудио/видео вызова

Окно отображает пользователя или группу пользователей при совершении вызова. Состоит из: фото – изображение контакта или конференции, имя конференции/контакта – текстовое поле, время вызова – текстовое поле, отображающее продолжительность вызова, список контактов – кнопка, открывающая всплывающее окно со списком контактов пользователя, кнопка включения/отключения микрофона во время вызова, кнопка включение/отключение камеры во время вызова, кнопка включения/отключения звука при совершении вызова и кнопка завершения вызова.



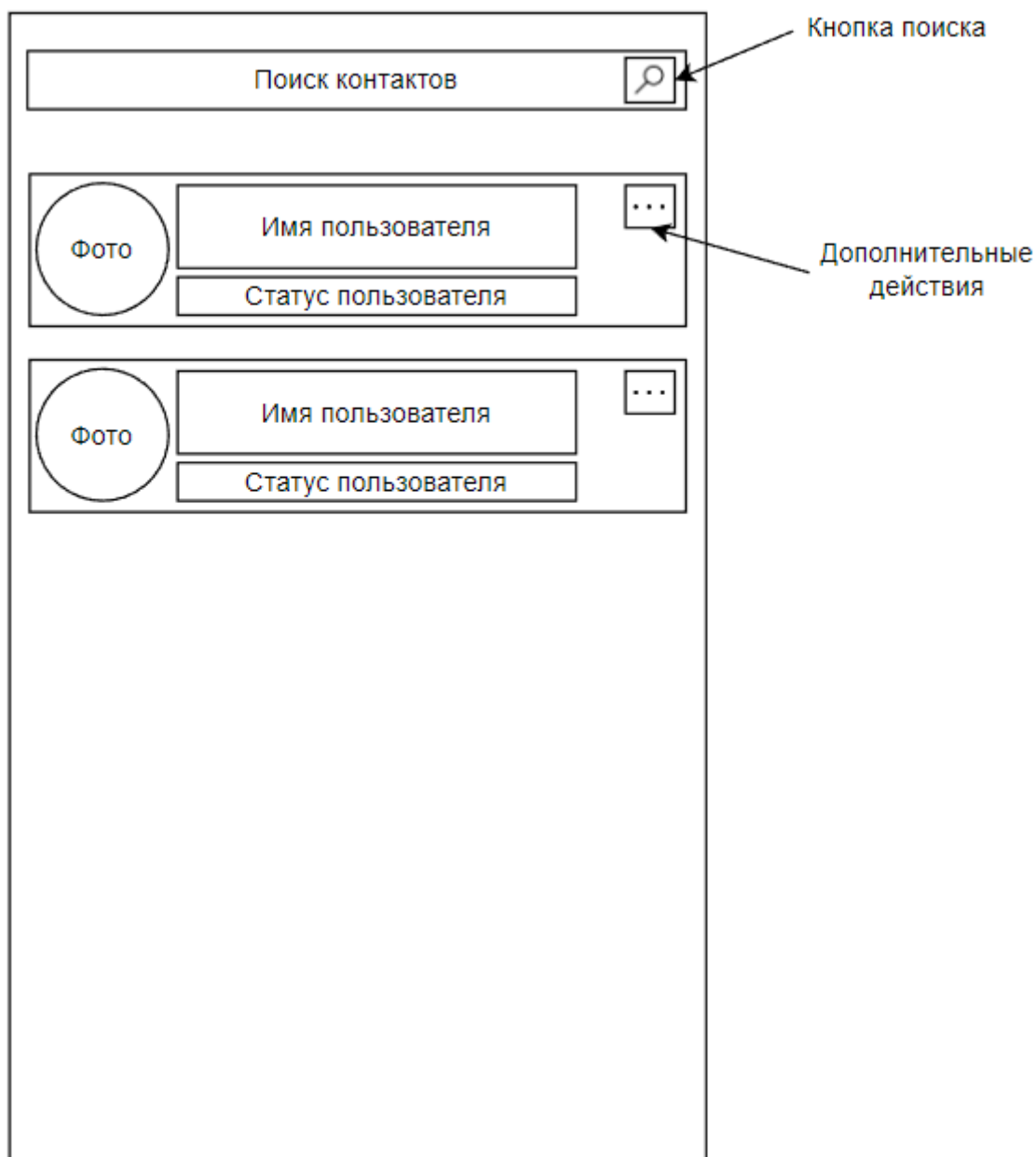
## 5. Прототип окна аккаунта

Окно необходимо для отображения информации о пользователе и его учётной записи. Состоит из: фото пользователя – изображение профиля пользователя, имя пользователя – текстовое поле, статус пользователя – текстовое поле, кнопка дополнительные действия – выпадающий список дополнительных действий с аккаунтом пользователя, выйти – кнопка, осуществляющая выход пользователя из аккаунта, элементы управления аккаунтом: управление учетной записью – кнопка, личные сведения – кнопка, синхронизация – кнопка и импорт данных кнопка.



## 6. Прототип всплывающего окна списка контактов

Данное окно предназначено для отображения списков контакта пользователя. Состоит из: поиск контактов – текстовое поле ввода, кнопка поиска, фото – изображение контакта пользователя, имя пользователя – текстовое поле, статус пользователя – текстовое поле, дополнительные действия – кнопка с дополнительными возможностями над контактом пользователя.



## 7. Прототип окна настройки приложения

Это окно предназначено для отображения пользователю настроек приложения. Состоит из: поиск – текстовое поле ввода, кнопка поиска, разделы настроек – список элементов, заголовок раздела настроек – текстовое поле, заголовок подраздела настроек – текстовое поле, параметр – текстовое поле, переключатель – включение/отключение параметра настроек.


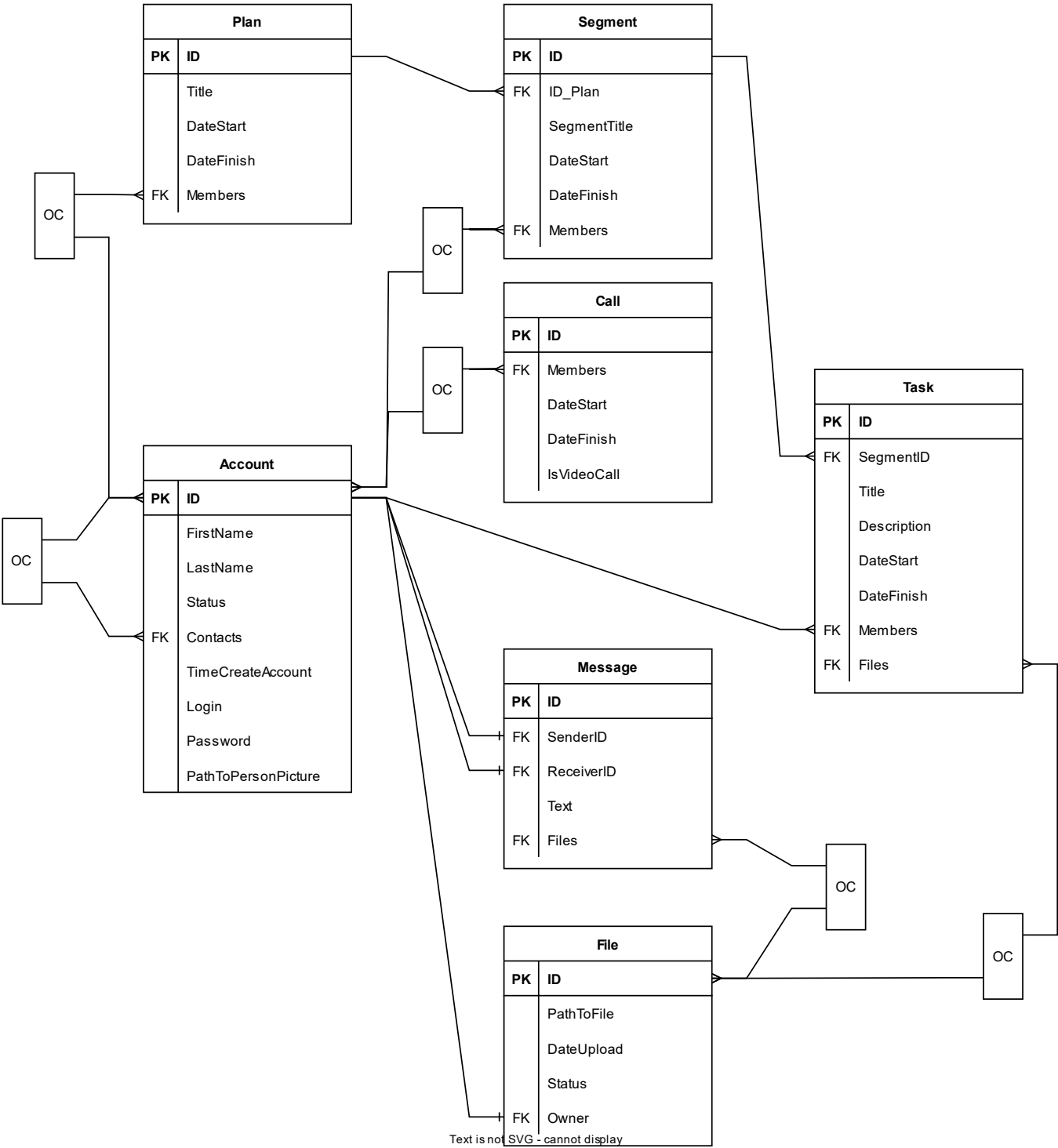
<div>Поиск </div>	<div>Заголовок раздела настроек</div>
<div>Общие настройки</div>	
<div>Настройки видео</div>	<div>Заголовок подраздела настроек</div>
<div>Настройки звука</div>	<div>Параметр <span><input type="checkbox"/> Вкл/Откл</span></div>
<div>Настройки данных</div>	
<div>Настройки синхронизации</div>	
<div>Дополнительные настройки</div>	

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. MakeCall

Функция предназначена для установления связи между двумя пользователями и совершения ими звонка.

Входная информация: ID – звонящего, ID – вызываемого, bool переменная, определяющая тип звонка (аудио или видеозвонок).

Выходная информация: поток аудио или видеоданных.



## **5. SerializationObject**

Функция предназначена для сериализации объекта и представление его в виде json-файла.

Входная информация: сериализуемый объект.

Выходная информация: дескриптор json-файла.

## **6. PageNavigation**

Функция предназначена для навигации пользователя по страницам внутри приложения.

Входная информация: элемент из списка навигации.

Выходная информация: UI страница.

## **7. LoadApplication**

Функция предназначена для загрузки всех необходимых для старта окон приложения.

Входная информация: дескриптор окна приложения.

Выходная информация: UI.

## **8. UserAuthentication**

Функция предназначена для аутентификации пользователя в приложении.

Входная информация: логин, пароль.

Выходная информация: enum-статус завершения операции.

## **9. SearchData**

Функция предназначена для поиска информации в данных пользователя.

Входная информация: строковый запрос.

Выходная информация: объект запроса.

## **10. LoadFile**

Функция предназначена для загрузки пользователем файла в программу.

Входная информация: путь к файлу.

Выходная информация: enum-статус завершения операции.

## 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. MakeCall;
      - 2.1.2.5. SirializationObject;
      - 2.1.2.6. PageNavigation;
      - 2.1.2.7. LoadApplication;
      - 2.1.2.8. UserAuthentication;
      - 2.1.2.9. SearchData;
      - 2.1.2.10. LoadFile;
    - 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 главной страницы;
- 2.2.1.1.2. Разработка макета страницы календаря;
- 2.2.1.1.3. Разработка макета страницы вызова;
- 2.2.1.1.4. Разработка макета страницы настроек;
- 2.2.1.1.5. Разработка макета страницы аккаунта;
- 2.2.1.1.6. Разработка макета страницы помощи;
- 2.2.1.1.7. Разработка страницы контактов;
- 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 логотипа приложения;
  - 2.2.1.3.2. Создание иконок элементов;
  - 2.2.1.3.3. 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:

Произведем вычисления для определения временных затрат на реализацию проекта. Для этого определим состав работ их оптимистичные, пессимистичные и средние трудозатраты в часах:

Работы	Количество	Оптимистичные трудозатраты	Пессимистичные трудозатраты	Наиболее вероятные трудозатраты
Создание сущностей	7	2	8	3
Создание макетов	7	30	60	50
Создание методов API	10	40	70	55

Посчитаем средние трудозатраты по каждой работе:

$$\text{Создание сущностей} = \frac{8 + 4 \times 3 + 2}{6} = 3.6 \text{ чел.* час.}$$

$$\text{Создание макетов} = \frac{60 + 4 \times 50 + 30}{6} = 48.3 \text{ чел.* час.}$$

$$\text{Создание методов API} = \frac{70 + 4 \times 55 + 40}{6} = 55 \text{ чел.* час.}$$

Посчитаем среднеквадратичную оценку:

$$\text{Создание сущностей} = \frac{8 - 2}{6} = 1 \text{ чел.* час.}$$

$$\text{Создание макетов} = \frac{60 - 30}{6} = 5 \text{ чел.* час.}$$

$$\text{Создание методов API} = \frac{70 - 40}{6} = 5 \text{ чел.* час.}$$

Посчитаем  $E_{\text{общ.}}$ :

$$E_{\text{общ.}} = 7 * 3.6 + 7 * 48.3 + 10 * 55 = 913.3 \text{ чел.* час.}$$

Посчитаем  $SKO_{\text{общ.}}$ :

$$SKO_{\text{общ.}} = \sqrt{7 * 1^2 + 7 * 5^2 + 10 * 5^2} = 20.8 \text{ чел.* час.}$$

Оценка суммарной трудоёмкости проекта с вероятностью 95%:

$$E_{95\%} = 913.3 + 2 * 20.8 = 954.9 \text{ чел.* час.}$$

**Базовое расписание в виде диаграммы Ганта:**

Представлено в отдельном файле – «Базовое расписание.xlsx»