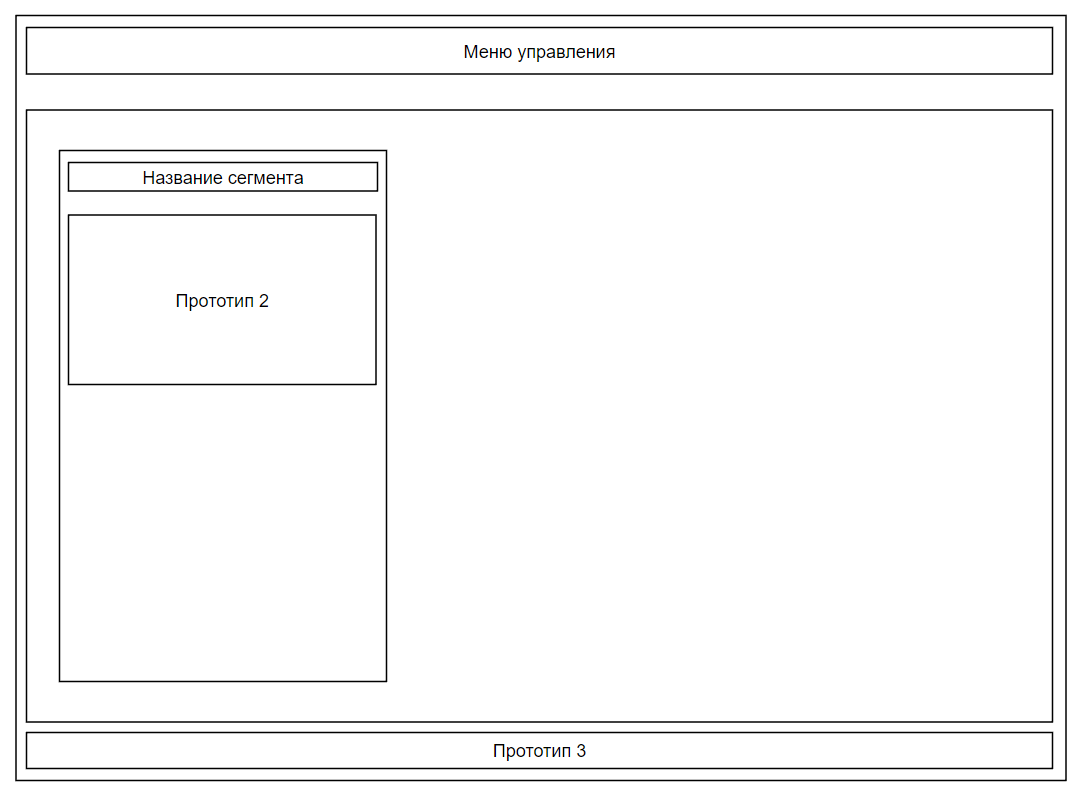
**Разработка базового расписания**

**Прототипы экранных форм:**

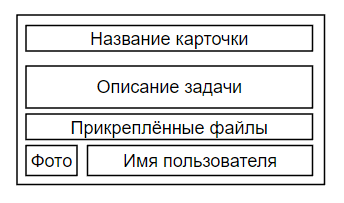
1. **Прототип главного окна**

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).



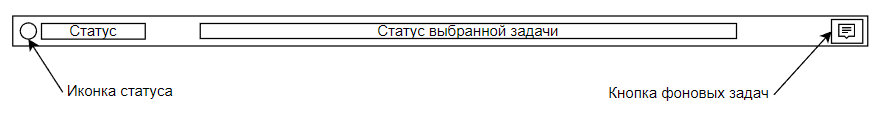
1. **Прототип карточки подзадачи**

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



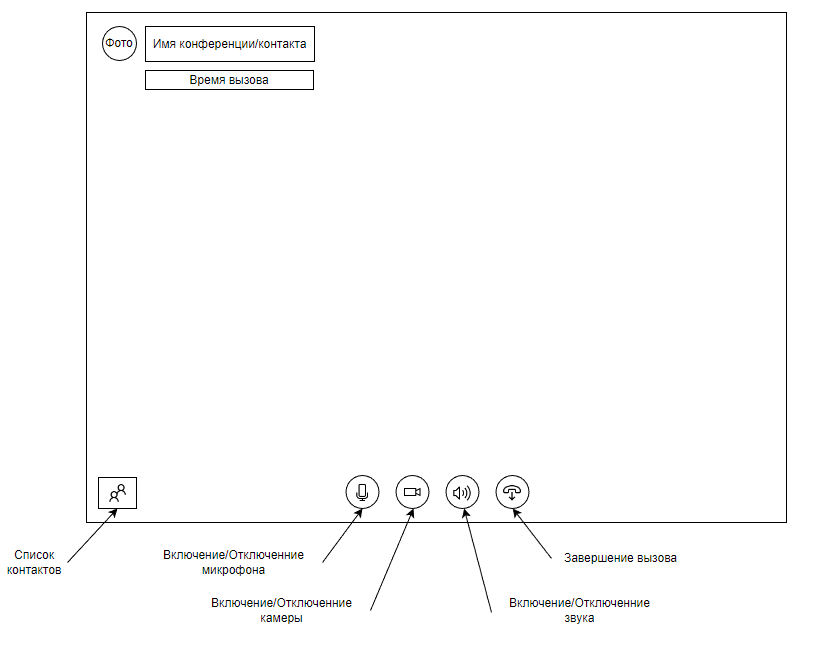
1. **Прототип статусбара**

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.



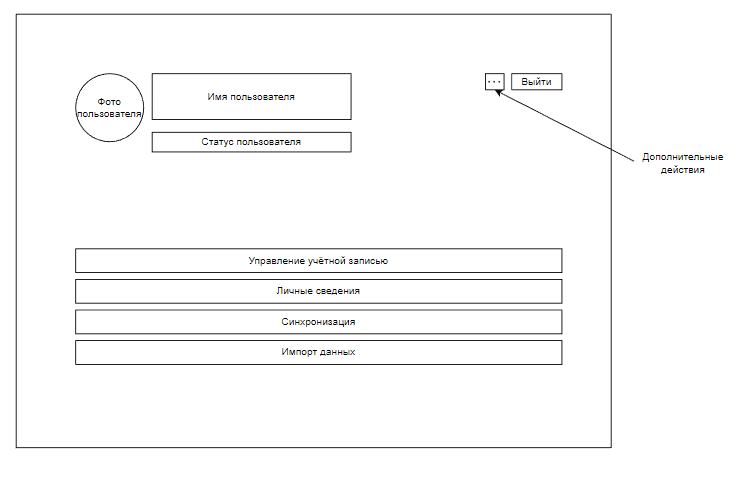
1. **Прототип окна аудио/видео вызова**

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.



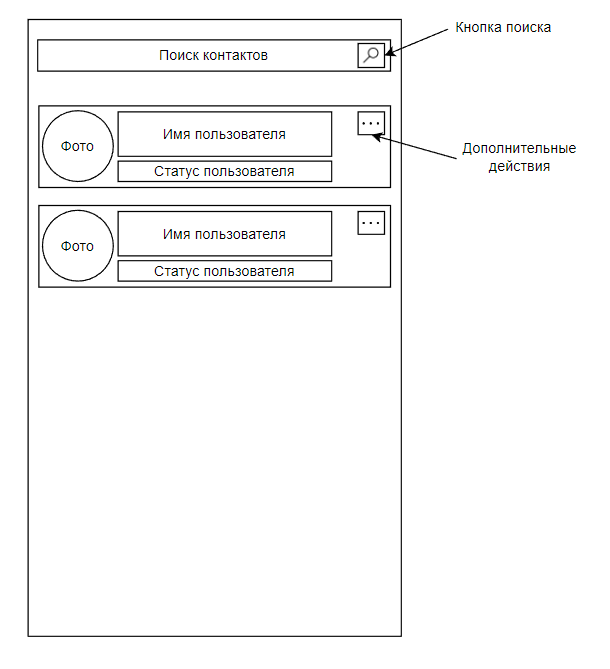
1. **Прототип окна аккаунта**

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.



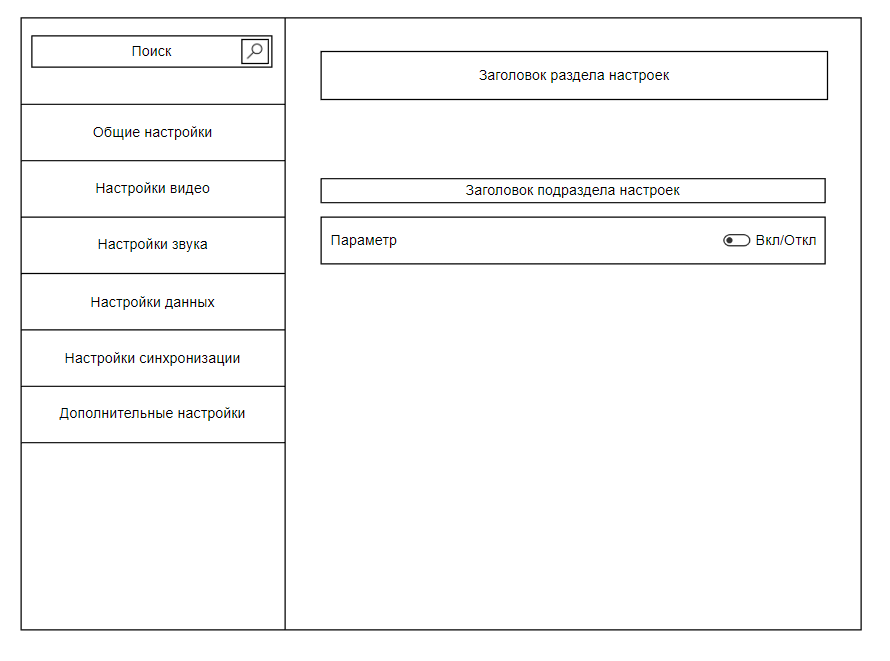
1. **Прототип всплывающего окна списка контактов**

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.

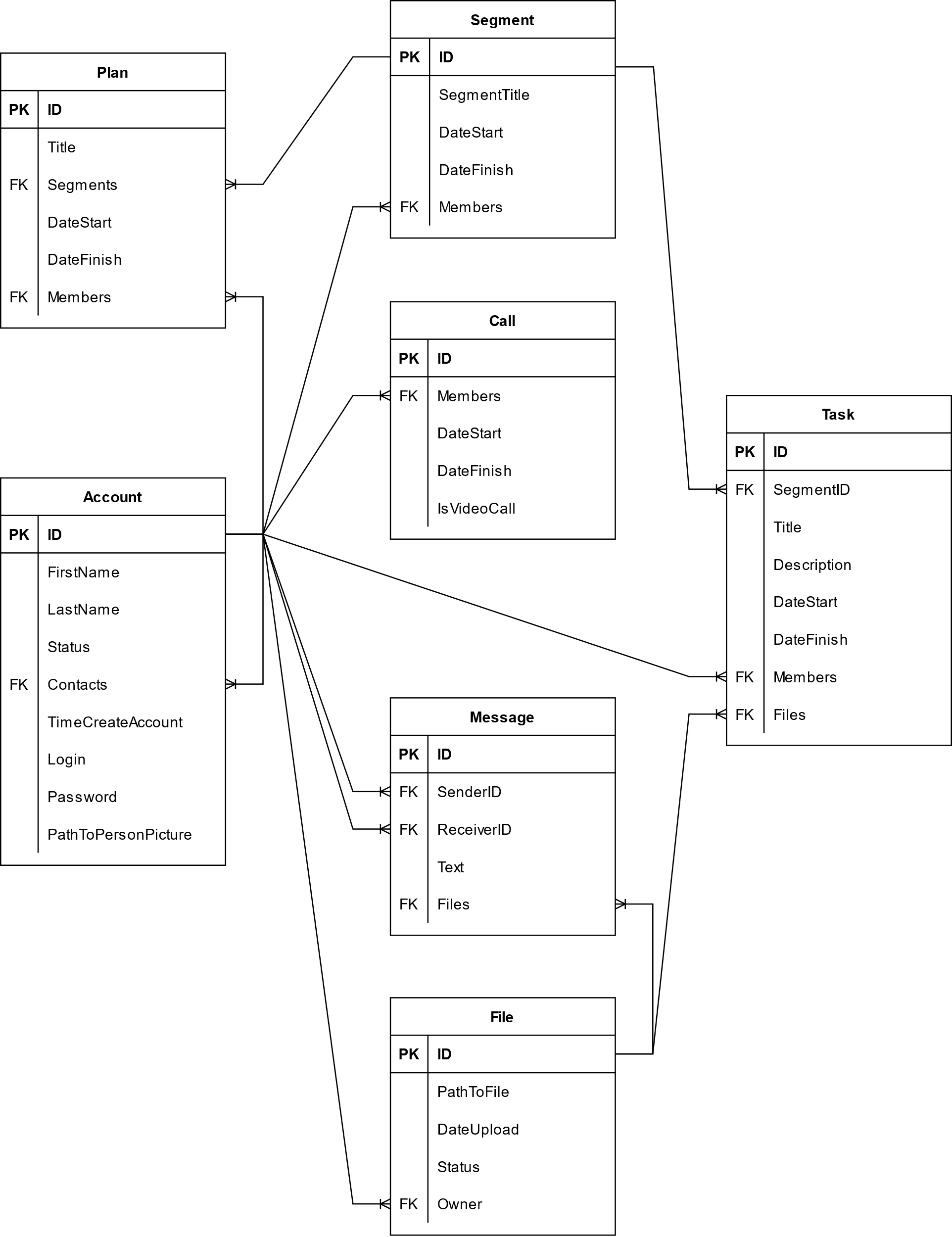


1. **Прототип окна настройки приложения**

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.



**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>.

1. **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.

1. **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.

1. **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.

1. **SerializationObject**

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

Входная информация: сериализуемый объект.

Выходная информация: дескриптор json-файла.

1. **PageNavigation**

Функция предназначена для навигации пользователя по страницам внутри приложения.

Входная информация: элемент из списка навигации.

Выходная информация: UI страница.

1. **LoadApplication**

Функция предназначена для загрузки всех необходимых для старта окон приложения.

Входная информация: дескриптор окна приложения.

Выходная информация: UI.

1. **UserAuthentication**

Функция предназначена для аутентификации пользователя в приложении.

Входная информация: логин, пароль.

Выходная информация: enum-статус завершения операции.

1. **SearchData**

Функция предназначена для поиска информации в данных пользователя.

Входная информация: строковый запрос.

Выходная информация: объект запроса.

1. **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. Сбор требований;
   2. Определение стадий и этапов разработки
      1. Определение стадий разработки;
      2. Определение сроков разработки;
   3. Общее описание
      1. Назначение продукта;
      2. Взаимодействие продукта;
      3. Допущения и ограничения продукта;
      4. Определение функций продукта;
2. Разработка приложения
   1. Backend-разработка
      1. Проектирование базы данных
         1. Определение структуры базы данных;
         2. Определение связей между сущностями;
         3. Определение взаимодействия с базой данных;
      2. Разработка API приложения
         1. LoadData;
         2. SyncData;
         3. SendMessage;
         4. MakeCall;
         5. SirializationObject;
         6. PageNavigation;
         7. LoadApplication;
         8. UserAuthentication;
         9. SearchData;
         10. LoadFile;
      3. Сетевое взаимодействие
         1. Определение протокола взаимодействия;
         2. Обеспечение защищённости соединения;
         3. Определение местоположения хранения данных;
      4. Взаимодействие с UI
         1. Определение модели привязки данных;
         2. Создание механизма обновления данных;
   2. Frontend-разработка
      1. Дизайн
         1. Разработка макетов страниц
            1. Разработка макета главной страницы;
            2. Разработка макета страницы календаря;
            3. Разработка макета страницы вызова;
            4. Разработка макета страницы настроек;
            5. Разработка макета страницы аккаунта;
            6. Разработка макета страницы помощи;
            7. Разработка страницы контактов;
         2. Взаимодействие с backend
            1. Определение модели привязки данных;
            2. Настройка механизма обновления данных;
            3. Определение динамического взаимодействия с данными;
         3. Разработка общего стиль-кода приложения
            1. Создание логотипа приложения;
            2. Создание иконок элементов;
            3. Определение стиля и размера шрифтов;
3. Приемо-сдаточные испытания
   1. Подготовка и проведение демонстрации;
   2. Проведение испытаний;
4. Размещение приложения
   1. Аренда сервера;
   2. Развёртывание приложения;
   3. Размещение в магазине приложений;
5. Поддержка приложения
   1. Мониторинг работоспособности;
   2. Получение и обработка обратной связи;
   3. Улучшение работы приложения
   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:

Посчитаем среднеквадратичную оценку:

Посчитаем :

Посчитаем :

Оценка суммарной трудоёмкости проекта с вероятностью 95%:

**Базовое расписание в виде диаграммы Ганта:**

Представлено в отдельном файле – «Базовое расписание.xlsx»