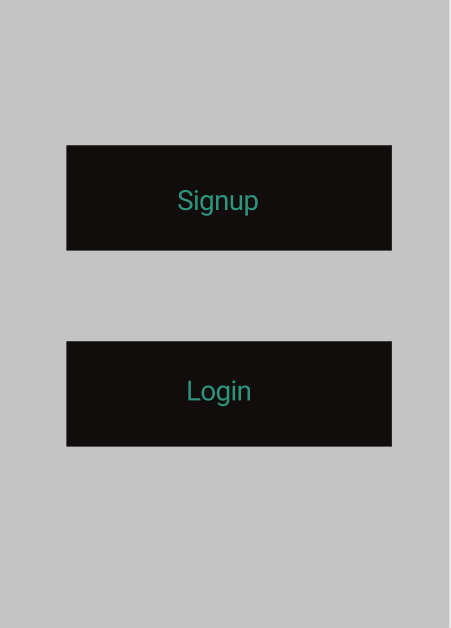
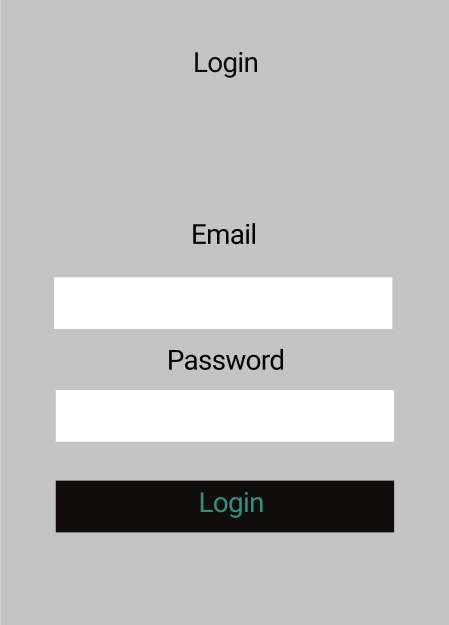
1. **Прототипы экранных форм**
   1. Вход в систему

Если пользователь уже зарегистрирован, то вводится логин и пароль и нажимается кнопка «Login». Если пользователь хочет зарегистрироваться, то нажимается кнопка «Signup».



* 1. Вход



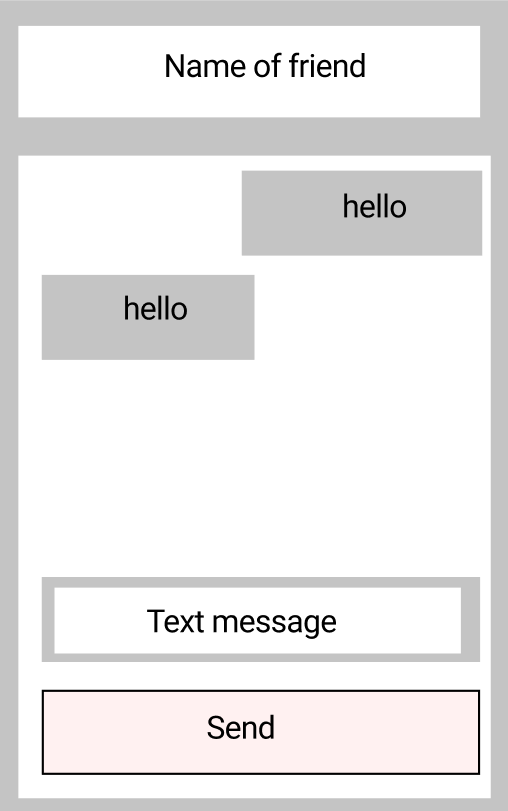
* 1. Регистрация нового пользователя



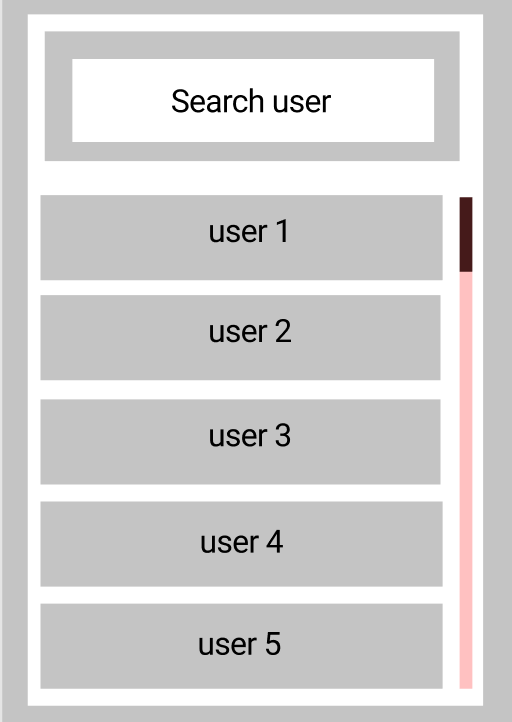
* 1. Чат

- Вверху выводится имя друга (в поле ‘Name of friend’)

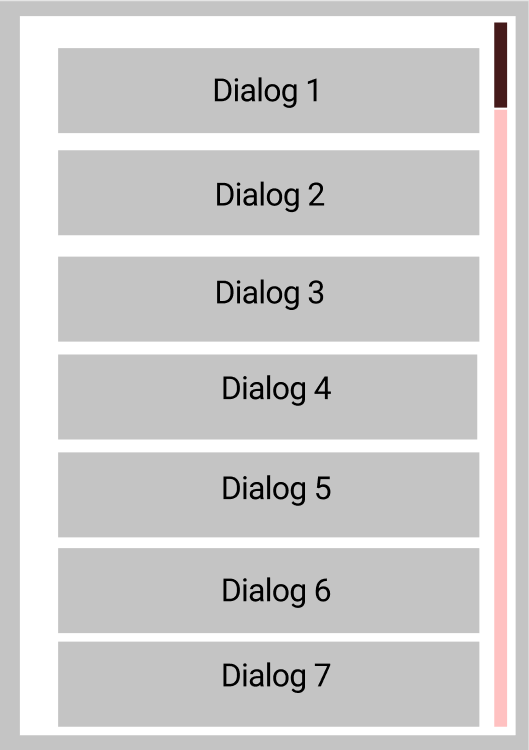
- Далее идет история сообщений  
- В поле ‘Text message’ – вводится текст  
- Для отправки текста – нужно нажать кнопку ‘Send’-



* 1. Список друзей  
     Выводится список всех друзей пользователя.  
     Можно воспользоваться поиском.

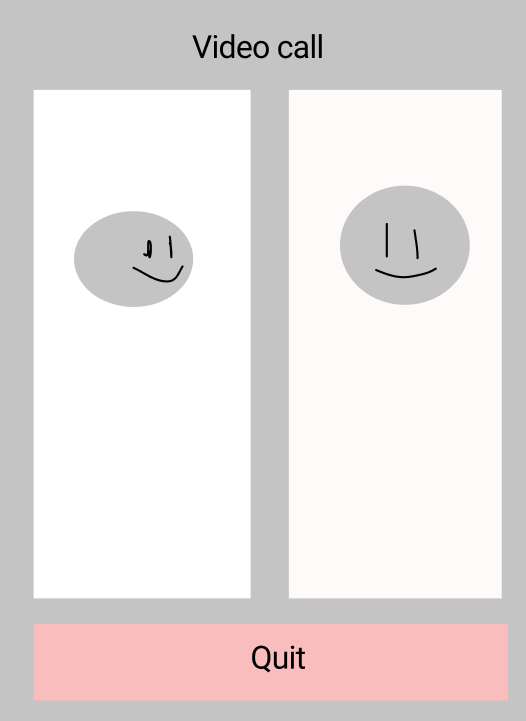


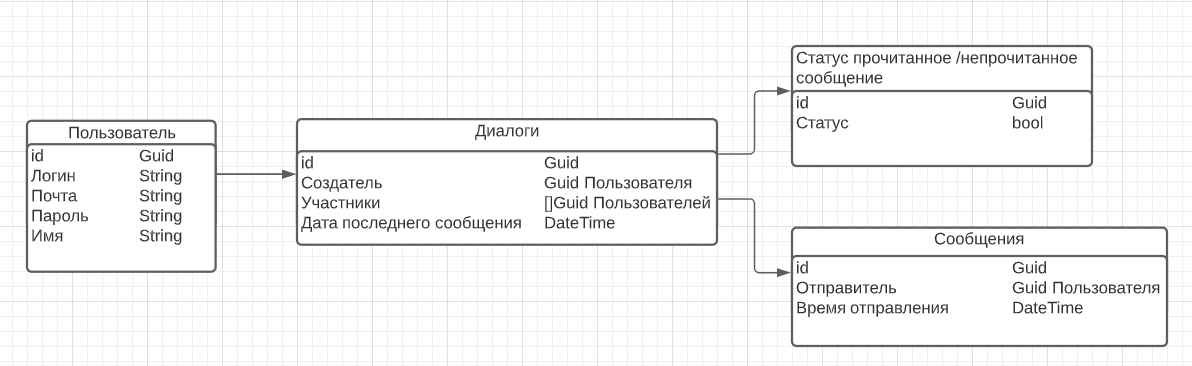
* 1. Список диалогов пользователя



* 1. Видеозвонок

Позволяет общаться с другими пользователями по видеосвязи



1. **Диаграмма сущностей** 
2. **Система API**

3.1. bool Register(User)

● Регистрирует пользователя в системе

● Входные данные: объект User, у которого указаны Email, Login и Password

● Выходные данные: результат отправки(true/false)

3.2. Token SignIn(User)

● Выполняет вход пользователя в систему

● Входные данные: объект User, у которого указаны Login и Password

● Выходные данные: токен для авторизации

3.3. bool WriteMessage(String, User)

● Отправляет сообщение указанному пользователю

● Входные данные: текст сообщения, пользователь для отправки

● Выходные данные: результат отправки(true/false)

3.4. bool DeleteMessage(Guid)

● Удаляет указанное сообщение

● Входные данные: id сообщения

● Выходные данные: результат отправки(true/false)

3.6. User GetUser(Guid)

● Получает пользователя из базы данных

● Входные данные: идентификатор пользователя

● Выходные данные: объект User

3.7. Message GetMessage(Guid)

● Получает сообщение из базы данных

● Входные данные: идентификатор сообщения

● Выходные данные: объект Message

3.8. User GetUserByPhoneNumber(string)

● Получает пользователя по номеру телефона

● Входные данные: номер телефона

● Выходные данные: объект User

3.9. void RemoveUser(User)

● Удаляет пользователя из системы

● Входные данные: пользователь

● Выходные данные: результат отправки(true/false)

3.10. User[] GetAllUser()

● Получение списка всех пользователей

● Входные данные отсутствуют

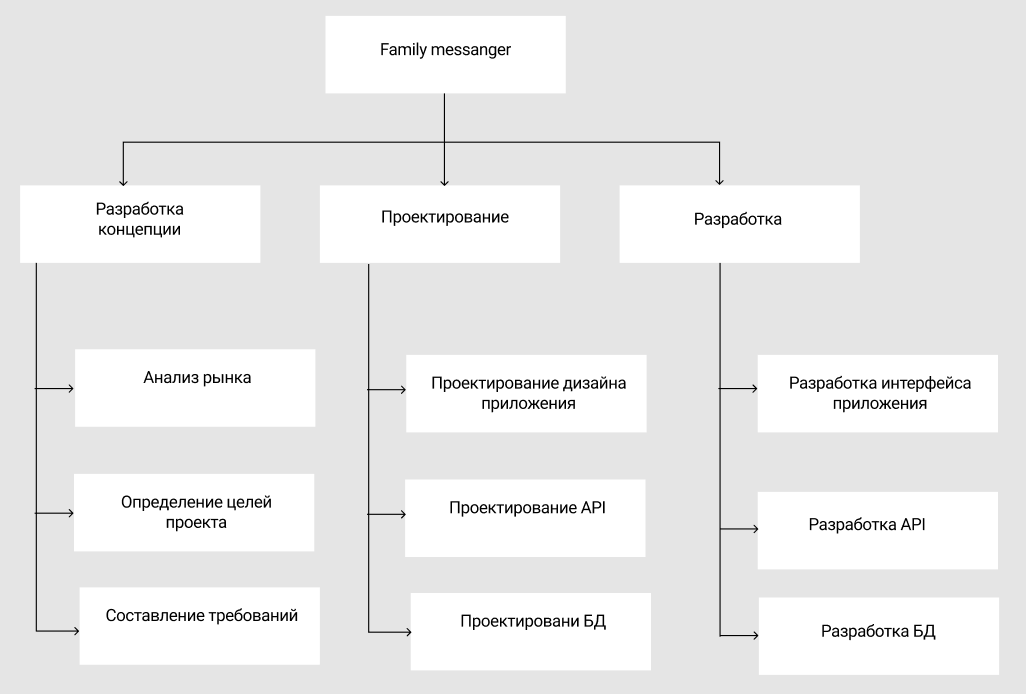
● Выходные данные: список объектов User

3.11. User[] GetUsersForCurrentUser(User)

● Получение всех пользователей, с которыми контактировал указанный пользователь

● Входные данные: пользователь

● Выходные данные: список объектов User

1. **ИСР** 
2. Этапы работ

MVP будет включать в себя следующий функционал:

* Вход и регистрация пользователей
* Возможность обмениваться сообщениями
* Родительский контроль

Финальная версия продукта будет включать в себя следующий функционал:

* Поддержка видеосвязи
* Возможность восстановить пароль от аккаунта
* Экран блокируется, пока пользователь не ответит на сообщение
* Фильтр чата

1. **Оценка времени выполнения проекта**

E – оценка средней трудоемкости

*Ei = (Pi + 4Mi + Oi)/6.*

CKO – среднеквадратичное отклонение

*CKOi = (Pi - Oi)/6.*

M – Наиболее вероятная оценка трудозатрат

O – Оптимистичная оценка трудозатрат

P – Пессимистичная оценка трудозатрат

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Работы | К(шт.) | M(ч.ч) | O(ч.ч) | P(ч.ч) | E | CKO |
| Сущности | 10 | 7 | 3 | 10 | 6,83 | 1,16 |
| Формы | 8 | 7 | 4 | 20 | 8,66 | 2,66 |
| Методы | 13 | 15 | 5 | 30 | 15,83 | 4,16 |
|  |  |  |  |  | 343 | 9 |

E = 6,83 \* 10 + 8,66 \* 8 + 15,83 \* 13 = 343

CKO = sqrt(1,16 \* 10 + 2,66 \* 8 + 4,16 \* 13) = 9

𝐸 95% = 𝐸 + 2𝐶𝐾𝑂 = 343+ 2 ∗ 9 = 361(ч.ч)

𝐸 общ = 𝐸 95% ∗ 𝑛 = 361 ∗ 3 = 1083(ч.ч)

𝑇 𝑝 = 165 ∗ 0.8 = 132(ч.ч / мес.)

𝑇 = 𝐸 общ/𝑇 𝑝 = 1083/132 = 8,2(мес.)

1. **Базовое расписание**

