

# Future Prospects. T3.

## 1. Введение

### 1.1 Постановка задачи:

реализовать android приложение для глухонемых инвалидов со следующим функционалом: мессенджер для глухонемых инвалидов на основе обработки видеопотока с последующим выявлением языковых жестов на нем с некоторой периодичностью.

### 1.2 Краткое описание:

при входе в приложение пользователь должен авторизоваться. Затем перед ним открывается список чатов. В самом чате пользователь может общаться с собеседником при помощи текстовых сообщений или же с помощью языка жестов следующим образом: на устройстве открывается камера с возможностью начать запись; когда идет запись, один раз в определенный промежуток времени (в секунду) моргает вспышка устройства, оповещающая пользователя о том, что можно показать жест. После заданного промежутка времени выбираются несколько лучших обработанных кадров со съемки и отправляются на сервер для обработки показываемого жеста. С сервера приходит ответ — сообщается о результате расшифровки и ожидается новый символ. После того, как пользователь заканчивает съемку, в поле ввода текста отображается вся сказанная информация и ожидается нажатие кнопки «Отправить».

### 1.3 Особенность идеи:

в современных приложениях, например, Вконтакте, Telegram, WhatsApp и другие, активно используется спросом такая функция, как «Голосовые сообщения», т. к. она позволяет пользователю проводить общение с собеседником(ами), похожее на живое. Но люди, не имеющие возможности разговаривать, лишены этой технологии. Данное приложение решает эту проблему.

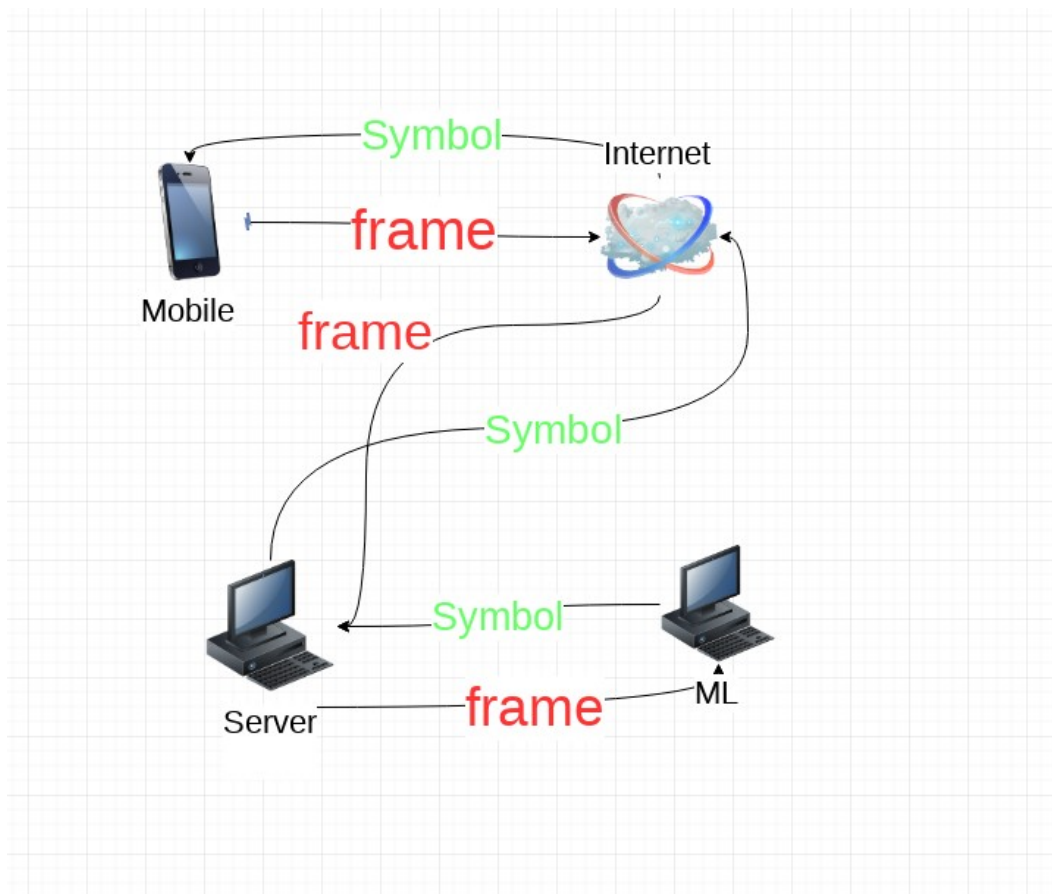
## 2. Задачи, решаемые приложением:

- текстовый мессенджер между клиентами
- замена аудиосообщений для глухонемых на эквивалентный им функционал с помощью текстовых сообщений

## 3. Требование к функционалу:

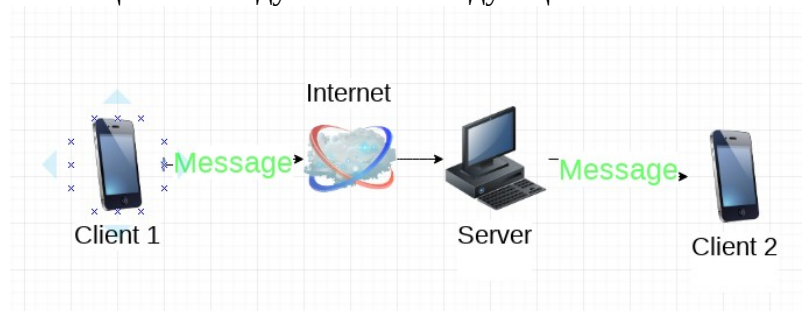
### 3.0 Приложение должно авторизовывать пользователя

### 3.1 Приложение должно определять показываемый символ по следующей схеме:



(ML — Machine Learning)

3.2 Клиенты должны общаться между собой по следующей схеме:



### 3.3 Используемые технологии

#### 3.3.1 Мобильный клиент

Приложение должно быть реализовано для работы на базе Android и использовать библиотеки из открытого доступа для работы с сервером и MVC (Model View Presenter) модели для минимизации ручного кода во избежание ошибок.

#### 3.3.2 Сервер

Реализуется на фреймворке Python Django, т. к. имеет удобное общение через REST протокол и сводит количество ошибок к минимуму. На сервере хранятся файлы с алгоритмом обработки изображения, которые вызываются при поступлении запроса на обработку.

#### 3.3.3 Machine Learning технологии

Реализуется на ЯП Python с использованием библиотек из открытого доступа.

Решается две задачи:

- определяется местоположение руки на снимке
- определяется показываемый символ

#### 4. Дальнейшие разработки

В будущем приложение должно уметь распознавать символы из любого положения камеры. Также не будет интервала для съемки, т. к. она будет происходить монопоточно, не тратя время на ожидание получения ответа с сервера