

Рабочий лист №1

Дата "01" 02 2025 г.  
(заполняется оргкомитетом)

Шифр ПЧ-44  
(заполняется оргкомитетом)

Оценка работы

(таблица заполняется по итогам проверки работы членами жюри олимпиады)

№ задания	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Итого (итоговый балл, подпись председателя жюри)
Балл																97
№ задания	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Балл																93

Магистратуры

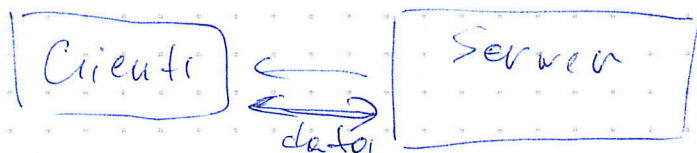
(название олимпиады, заполняется участником)

Прикладная информатика

(профиль олимпиады, заполняется участником)

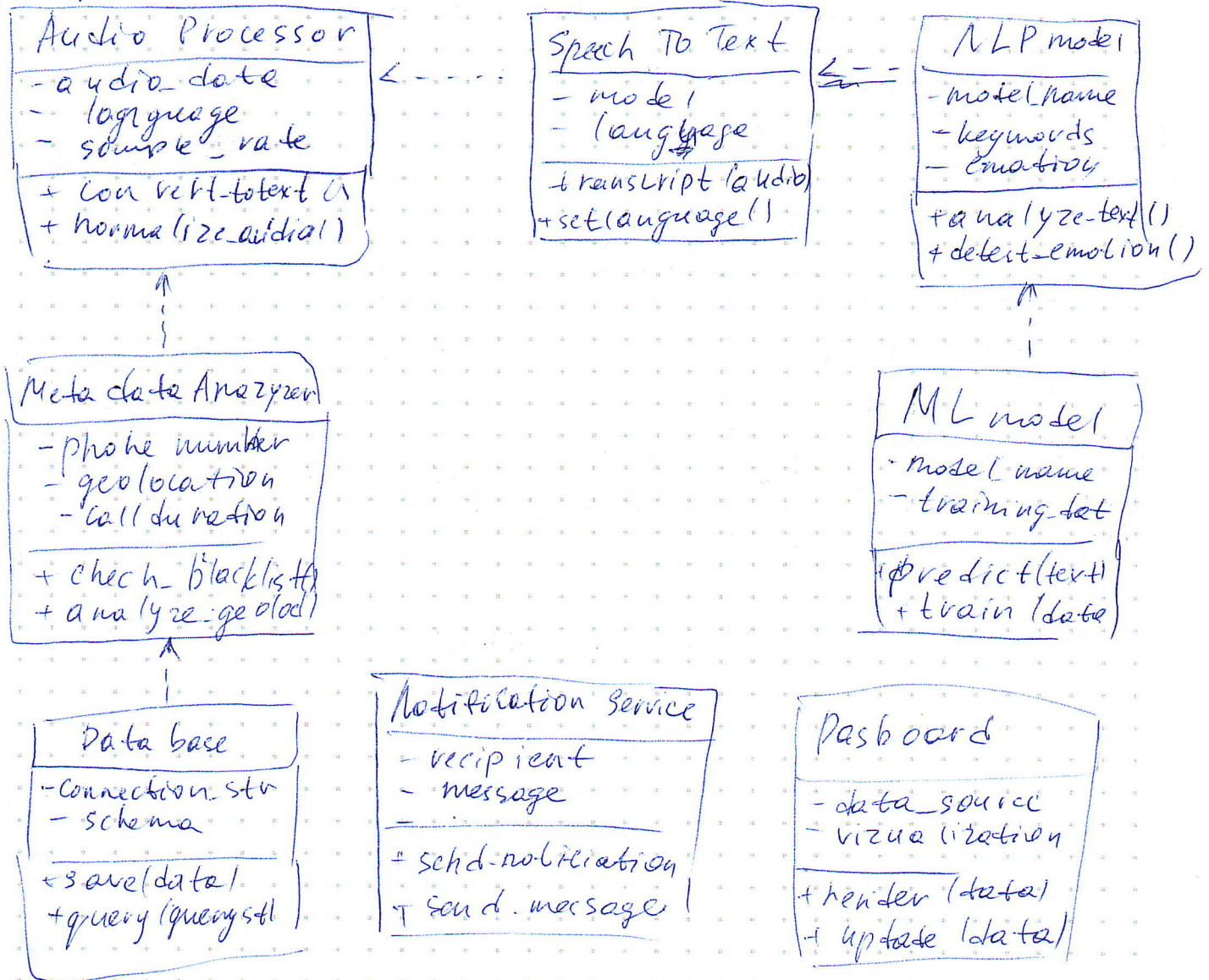
1. На мой взгляд, есть 2 варианта реализации данной системы. Первый, это приложение на мобильном устройстве, в котором все звонки пользователей будут проходить через мобильное устройство. После телефонного разговора, который осуществляется в телефонном интерфейсе, программа уведомит об этом пользователей. А дальше уже сами решат, можно ли сообщать об этом правоохранительным органам или нет. Второй вариант, это когда телефонные звонки будут проходить на телефонных станциях (только правоохранительных). В первом варианте сервер будет отправлять приложение в телефон и будет отправлять данные (аудиозаписи) на сервер, а во втором случае, приложение которое будет устанавливаться в телефонной станции (через которые будут проходить все звонки (сигнал, аудио)). Первый вариант будет разрабатываться для клиентов, а второй для государства.

Распишу более подробно про первый вариант. Пользователь устанавливает созданное приложение у себя на компьютере и дает разрешение на звонки. Все звонки пользователей сохраняются в виде аудиозаписей (формат wav, mp3...). Приложение отправляет эти данные на сервер, где они будут обрабатываться. Сервер обрабатывает эти данные и в случае необходимости на мобильное устройство уведомит пользователей.





## программа kua coll



## Описание компонентов

- API Gateway:**
  - Принимаем вызовы программ и метаданные от клиентов
  - Передаем данные в Audio Processor
- Audio Processor:**
  - Обработаем аудио (нормализация, конвертация)
  - Передаем данные в Speech To Text
- Speech to Text**
  - Преобразует аудио в текст
  - Передаем текст в NLP model и ML Model
- NLP model**
  - Анализируем текст на наличие ключевых фраз и эмоций
  - Передаем рез-ты в ML model
- ML model**
  - Классифицируем разговор как мошенничество или легитимный
  - Передаем рез-таты в Database и Notification Service
- Meta data Analyzer**
  - Анализируем метаданные
  - Передаем результаты в Database

Дополнительный рабочий лист  
(без рабочего листа №1 недействителен)

Дата " " 20 г.  
(заполняется участником)

Шифр ПУ-44  
(заполняется участником)

7. Database

Храним аудиозаписи, текстовые расшифровки и результаты анализа

8. Notification Service

- отправляем уведомление о подозрительных звонках

9. Parser

- отправляем с тапскриншоту и результаты анализов

В целом 1 и второй варианты отличаются только тем, что для мобильного телефона нужно будет создавать приложение с графическим интерфейсом, а для Телу нужно

2. Функция Вычисление вероят. (text)

~~Text~~ - obr = обработать текст (text)

~~text~~

texts

labels

↓

data = [ "Вам отключен", "молч" ]  
нужны dataset моментальных фраз

X\_train, X\_test, y\_train, y\_test = train\_test\_split(texts, labels, test\_size=0.25, random\_state=42)

model = Pipeline([ 'tfidf', TfidfVectorizer(),  
'clf', RandomForestClassifier() ])

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

print(classification\_report(y\_test, y\_pred))

def predict\_prob(text):

prob = model.predict\_proba([text])[0]

Функция вычисляет вероятность, что звонок является угрожающим, чем ML,



3, 1. Зависит от того, как устроено текстовое представление;

Если они устроены методами машинного обучения, текст скорее всего должен быть по структуре похожим на данные в рассказах. Опять все это, все зависит от того, как устроены данные модели. При разговоре, люди по-разному говорят числа. Например может быть так: один человек говорит номер паспортных данных по цифрам, а другой передает данные в первом слове после текстовой расшифровки текст будет всегда: 9 2 2 4 и т.д., через пробелы, а во втором слове 24 137...). В зависимости от того нужно писать программу (нужно учесть эти два случая)

```
import re  
def re  
passport_regex = r'\b\d{4}[ -\s]? \d{6}\b'  
snils_regex = r'\b\d{3}[ -\s]? \d{3}[ -\s]? \d{3}[ -\s]? \d{3}'  
[ -\s]? \d{23}\b'
```

Остальные регулярные выражения можно аналогично, ~~можно~~ можно переписать

```
info = re.findall(passport_regex, text)
```

↑  
Можно написать регулярные выражения, написанные выше,