

Министерство образования Российской Федерации  
Дом научной коллаборации Н. Г. Соломонова на базе государственного автономного  
образовательного учреждения высшего образования «Северо-Восточный федеральный  
университет имени М. К. Аммосова»  
Умный город и безопасность

Тема: «Мобильное приложение для мониторинга уровня воды и уведомления о  
паводках»

Автор работы:

Гоголев Вячеслав, 11 класс СУНЦ

Руководитель:

Захаров Дьулустаан Семенович

сотрудник Дома научной коллаборации Н.Г. Соломонова

Якутск 2023

## Оглавление

<b>Введение</b> .....	3
<b>Задачи:</b> .....	3
<b>2. Подробная статистика паводков за последние года</b> .....	4
<b>3. Обзор аналогов</b> .....	4
<b>4. Разработка Android-приложения</b> .....	5
<b>4.1 Разработка протипа в Фигме</b> .....	5
<b>4.2 Подключение Google maps и Firebase</b> .....	6
<b>4.3 Разработка геоданных пользователя</b> .....	7
<b>4.4 Получение данных на метку</b> .....	8
<b>4.5 Разработка функции “обращение к нам”</b> .....	9
<b>4.6 Разработка поисковика для рек и городов</b> .....	10
<b>4.7 Разработка регистрации</b> .....	11
<b>4.8 Разработка страницы для памятки</b> .....	12
<b>4.9 Разработка списка меток</b> .....	12
<b>4.10 Дизайн карты</b> .....	13
<b>5. Заключение</b> .....	14
<b>6. Список использованной литературы</b> .....	15

## Введение

**Актуальность** - Одним из главных проблем Якутии являются паводки. Ежегодно от весеннего и летнего паводков страдают несколько районов республики. Зачастую это связано с отсутствием своевременного оповещения о возможных опасных уровнях, что позволило бы организовать эвакуацию населения и материальных ценностей и существенно уменьшило бы ущерб от паводка.

**Новизна** - Предлагаемый проект — это мобильное приложение для мониторинга уровня воды и уведомления о паводках, представляет собой автоматизированную приложение, который мониторит паводковую ситуацию и уведомляет о критических уровнях воды.

**Гипотеза** - если использовать приложение для мониторинга паводковых ситуаций и оповещать уровень воды, то это поможет населению и специальным службам своевременно принять необходимые меры для снижения ущерба,

**Цель работы** - разработать приложение для мониторинга паводков и оповещения населения в районах, где присутствуют водоемы.

### Задачи:

1. Изучить статические данные и аналоги.
2. Разработать и оптимизировать базу данных в Firebase, для получения информации.
3. Разработать прототип приложения на платформе Figma
4. Изучить основы java и android studio.
5. Изучить библиотек Google maps и Firebase
6. Функции приложения
  - Разработать геоданные пользователя
  - Подключить Google maps и Firebase
  - Вывести данные на метку
  - Разработать обращение к нам
  - Разработать поисковик для рек и городов
  - Разработать функцию Регистрации, для того чтобы связать приложение и сайт.
  - Разработать функцию оповещения населения
  - Разработать страницу для памятки
  - Разработать список меток, где будут храниться все метки и избранные
  - Улучшить дизайн карты

## 2. Подробная статистика паводков за последние года

За 2019 год информации об ущербах паводков чрезвычайно мало. Это может быть связано со слабым течением паводков и малым общим ущербом за этот год. В некоторых источниках упоминается, что паводков в 2019 году и вовсе не происходило.

В 2020 году, по данным республиканского правительства, суммарный ущерб от масштабных весенних паводков составил более 1 миллиарда рублей.

В 2021 предварительный ущерб от весеннего паводка на реке Лена и малых реках Якутии оценили в 178,6 млн рублей. Пострадали 19 населенных пунктов в шести улусах и городе Якутск. Затоплено более 200 жилых домов, четыре объекта ЖКХ, пришкольный интернат.

За 2022 год, согласно данным ТАСС, ущерб, нанесенный паводками в Якутии, оценивается в более чем 480 млн. рублей. Результатом обильных осадков в Верхоянском и Кобяйском районах республики в летнее время было потоплено 579 дворовых территорий, 18 социально значимых объектов и 442 жилых дома.

## 3. Обзор аналогов

В числе аналогов нет мобильных приложений, которые позволяют мониторить паводки или уровень воды, поэтому ниже приведен обзор веб-сервисов, выполняющих аналогичные функции.

Одним из аналогов является приложение **allrivers** который предназначен для туристов-водников, рыбаков, отдыхающих на воде, гидростроителей, а также жителей прибрежных зон. Существенным недостатком данного приложения является отсутствие информации именно в нашем регионе, также отсутствие каких-либо данных о критическом уровне. Также хотелось бы добавить, что в приложении непонятный интерфейс для обычных людей.

Еще одним аналогом является проект **Meteoweb**, посвященный всему, что касается погоды и климата. На данном приложении также публикует уровень воды в реках России, представлена карта с метками. Основным недостатком является количество мониторинговых данных, по нашей республике порядка 3-5 источников мониторинга. Также функция визуального представления основана на ссылке на графический файл (png) с графиком, что является не самым удобным представлением динамики изменение уровня

воды. При этом стоит отметить отсутствие каких-либо данных о критических уровнях, но это в целом связано с направлением проекта [2].

Также отличным техническим аналогом является Народный Мониторинг. приложение представляет из себя карту с расположенными по нему датчиками, камерами и разными приборами. Приборами могут выступать так и банальные датчики влажности воздуха, так и экзотические измерители УФ-Индекса или Радиации. Датчики устанавливаются физическими лицами со всего мира, любой человек способен прописать в сайт своё устройство, главное, чтобы его одобрила администрация [3].

#### **4. Разработка Android-приложения**

Для удобства пользования населением и для уведомления людей о достижении критического уровня воды было решено разработать Android-приложение. Данные об уровне воды будут браться из базы данных Firebase. Для того чтобы разработать приложение, был использован Android studio и язык Java.

##### **4.1 Разработка протипа в Фигме**

Первым делом, было решено разработать прототип приложения в фигме, таким образом были добавлены:

1. Памятка при наводнении, в котором будут правила поведения и что необходимо делать, также номер телефона спецслужб. Памятка будет доступен даже когда отсутствует интернет связь.
2. При помощи функции “Обращение к нам” пользователь спокойно может к нам обратиться с какой-либо проблемой или сообщить, предложения для приложения.
3. Настройки, мы хотели поместить управление темой карты и включение/выключение оповещения приложения.
4. Метка, довольно важная часть приложения, в метку собираемся поместить график текущий и критический уровень воды.

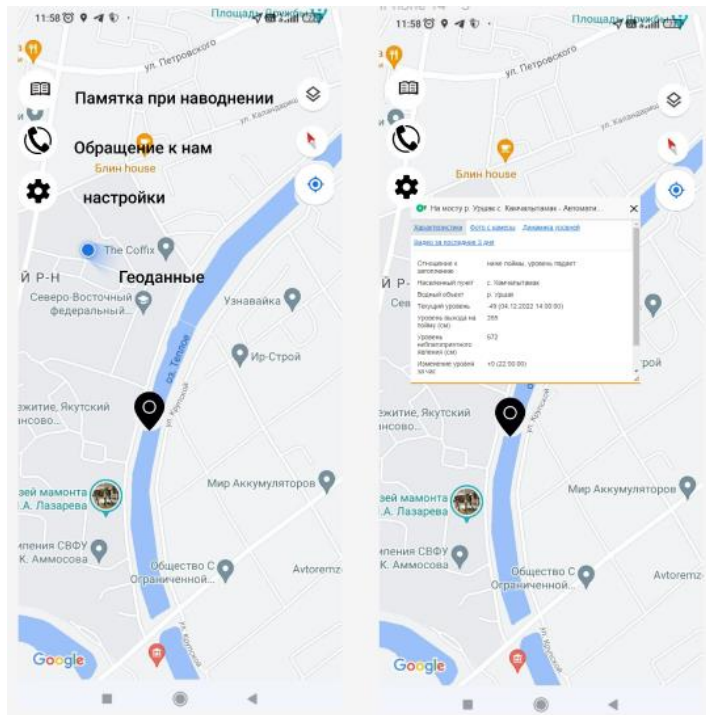


Рисунок 1. Прототип в Фигме

## 4.2 Подключение Google maps и Firebase

В качестве вывода карты в приложение был использован Google Maps SDK for Android. Maps Sdk - помогает встраивать в андроид карту и взаимодействовать с ним.

Первым делом был подключен библиотека от гугл карты и был разработан код, который выводил на весь экран карту (Рисунок 2).

```

@Override
public void onMapReady(@NonNull GoogleMap googleMap) {
    mMap = googleMap;
    mMap.setStyle(MapStyleOptions.loadRawResourceStyle(context, R.raw.mpa));
    Toast.makeText(context, "Господин, карта работает", Toast.LENGTH_SHORT).show();
    Log.d(TAG, "onMapReady: map is ready");

    if (mLocationPermissionsGranted) {
        getDeviceLocation();

        if (ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION)
            != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(context,
            Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        mMap.setMyLocationEnabled(true);
        mMap.getUiSettings().setMyLocationButtonEnabled(true);
        mMap.getUiSettings().isCompassEnabled();
        init();
    }
    if (mapView != null &&
        mapView.findViewById(Integer.parseInt("1")) != null) {

```

Рисунок 2. Подключение карты в приложение

Чтобы получать данные в реальном времени, было решено подключить Firebase. Firebase — это облачная база данных, которая позволяет пользователям хранить и получать сохраненную информацию, а также имеет удобные средства и методы взаимодействия с ней.

Структура базы данных RealTime Database приведена на рисунке 6. Для чтения данных необходимо указать ссылку на поле, в котором хранятся данные об уровне воды, местоположения, критическом уровне и так далее. Далее посредством функции библиотеки Firebase выполняется чтение с указанных ссылок.

### 4.3 Разработка геоданных пользователя

Для того чтобы получать местоположения и другие функции телефона, было получено разрешение андроид в AndroidManifest (Рисунок 3).

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
```

Рисунок 3. Разрешения в Manifest

Далее отправляем запрос на разрешение местоположения пользователя, если мы этого не сделаем, то будет плохо (Рисунок 4).

```
private void getLocationPermission(){
    Log.d(TAG, "msg: getLocationPermission: получение разрешений на местоположение");
    String[] permissions = {Manifest.permission.ACCESS_FINE_LOCATION,
        Manifest.permission.ACCESS_COARSE_LOCATION};

    if(ContextCompat.checkSelfPermission(this.getApplicationContext(),
        FINE_LOCATION) == PackageManager.PERMISSION_GRANTED){
        if(ContextCompat.checkSelfPermission(this.getApplicationContext(),
            COURSE_LOCATION) == PackageManager.PERMISSION_GRANTED){
            mLocationPermissionsGranted = true;
            initMap();
        }else{
            ActivityCompat.requestPermissions( activity: this,
                permissions,
                LOCATION_PERMISSION_REQUEST_CODE);
        }
    }else{
        ActivityCompat.requestPermissions( activity: this,
            permissions,
            LOCATION_PERMISSION_REQUEST_CODE);
    }
}
```

Рисунок 4. Запрос разрешения на местоположения

После того как получили разрешение, приступаем к определению местоположения пользователя, придаем 2 исхода действия, 1 случай, всё работает и отображается местоположения пользователя, 2 местоположение пользователя не найдено.

Если были получены местоположения пользователя, то они отправляются в `moveCamera`, чтобы направить камеру пользователя на найденные координаты. Код работоспособный, местоположение пользователя быстро определяется, также при успешном выполнении действия выскакивает маленькое окошко о том, что действие успешно выполнено (Рисунок 5).

```
private void getDeviceLocation(){
    Log.d(TAG, msg: "getDeviceLocation: получение текущего местоположения устройства");

    mFusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(activity.this);

    try{
        if(LocationPermissionsGranted){

            final Task location = mFusedLocationProviderClient.getLastLocation();
            location.addOnCompleteListener(new OnCompleteListener() {
                @Override
                public void onComplete(@NonNull Task task) {
                    if(task.isSuccessful()){
                        Log.d(TAG, msg: "Завершено: найдено местоположение!");
                        Location currentLocation = (Location) task.getResult();
                        moveCamera(new LatLng(currentLocation.getLatitude(), currentLocation.getLongitude()),
                                DEFAULT_ZOOM,
                                title: "Мое местоположение");
                    }else{
                        Log.d(TAG, msg: "onComplete: текущее местоположение равно нулю");
                        Toast.makeText(context: MainActivity.this, text: "не удалось получить текущее местоположение", Toast.LENGTH_SHORT).show();
                    }
                }
            });
        }
    }catch (SecurityException e){
        Log.e(TAG, msg: "getDeviceLocation: SecurityException: " + e.getMessage() );
    }
}

private void moveCamera(LatLng latLng, float zoom, String title){
    Log.d(TAG, msg: "moveCamera: перемещение камеры в: широту: " + latLng.latitude + ", lng: " + latLng.longitude );
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(LatLng, zoom));

    if(!title.equals("Мое местоположение")){
        MarkerOptions options = new MarkerOptions()
            .position(latLng)
            .title(title);
        mMap.addMarker(options);
    }
}
```

Рисунок 5. Код для получения местоположения

#### 4.4 Получение данных на метку

Были получены данные на метку с помощью Firebase Realtime Database, информация измерителя будут обновляться в реальном времени (Рисунок 7). Строка `User` – это сообщения от пользователей (Рисунок 6).

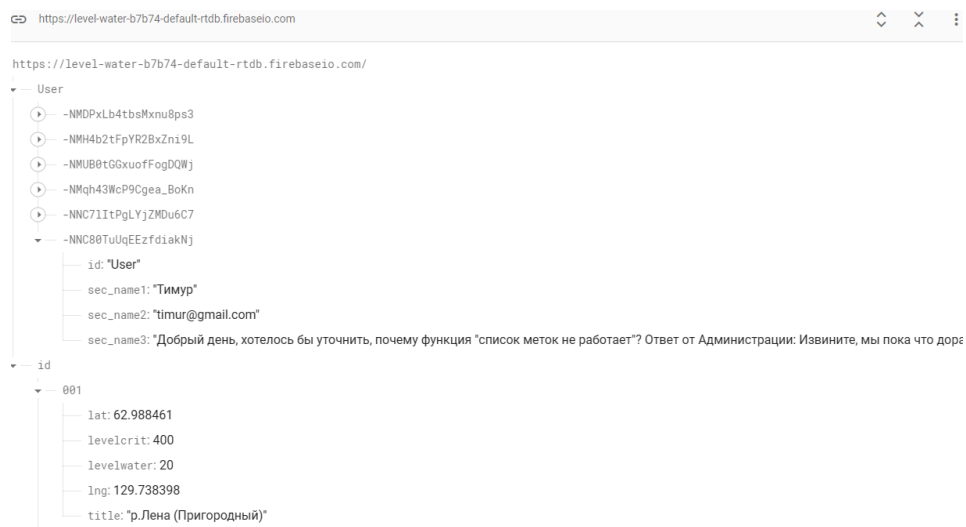


Рисунок 6. База данных



С помощью Firebase, мы можем получать данные откуда угодно, приложение не будет привязана только к одному источнику информации. Таким образом мы планируем глобализировать приложение.

Предполагается что, если подключить все измерители уровня воды и сравнивая их, можно добиться более точных данных и снизить ущерб от паводков.

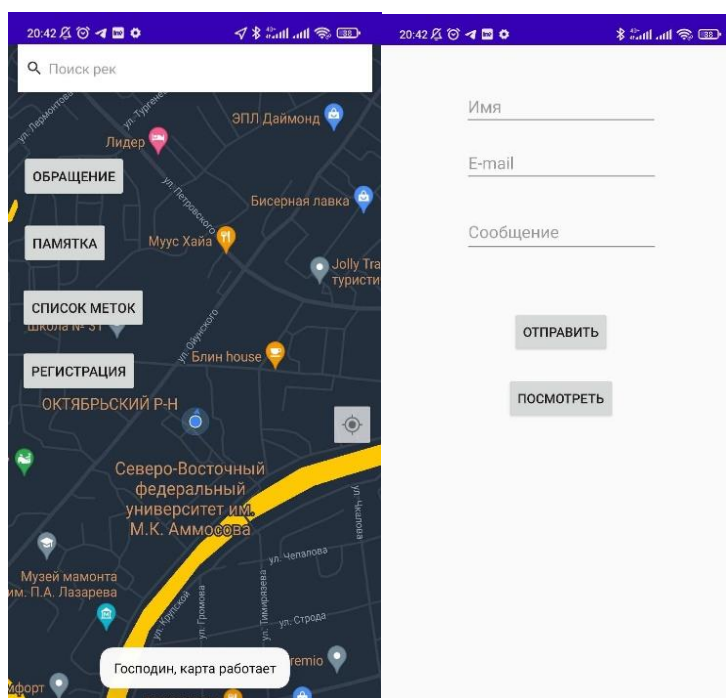
```
mesto1.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        value1 = dataSnapshot.child("levelwater").getValue().toString();
        String crit1 = dataSnapshot.child("levelcrit").getValue().toString();

        LatLng sydney1 = new LatLng( latitude: 62.988461, longitude: 129.738398);
        mMap.addMarker(new MarkerOptions()
            .position(sydney1)
            .draggable(true)
            .title("p.lena (Пригородный)")
            .snippet("Уровень воды: "+ value1 + " см"+ "\nКритический уровень воды: "+ crit1 + " см" ));
    }
    @Override
}
```

Рисунок 7. Код метки

#### 4.5 Разработка функции “обращение к нам”

Чтобы наладить связь с пользователями, был реализован функция “Обращение”, в котором можно записать что угодно, начиная с вопросов заканчивая с проблемами приложения. При нажатии на кнопку “посмотреть” пользователь может посмотреть на вопросы и на ответы от администрации.



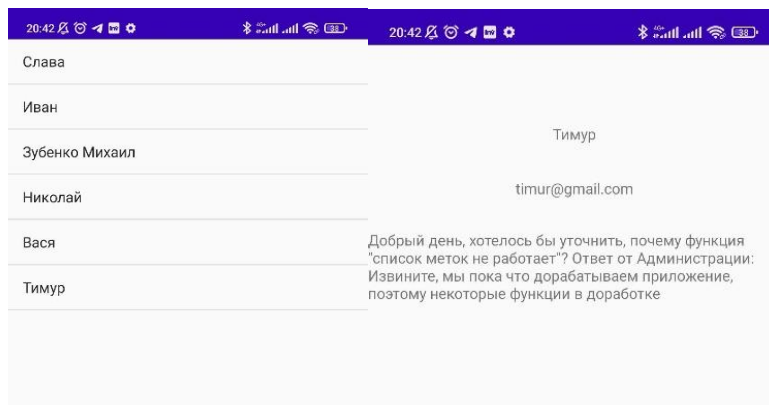


Рисунок 8. Функция "Обращение"

В базе данных есть отдельная строка "User", строка требуется для приложения, в нём хранятся обращения, которые пользователи отправляют нам. С помощью Firebase, функция "Обращение" работает (Рисунок 6).

#### 4.6 Разработка поисковика для рек и городов

Для того чтобы было удобно искать нужную метку с информацией о реке, было решено разработать функцию "Поиск рек/городов".

Для начало было решено найти координаты местоположения, объекта, который пользователь задал (Рисунок 9).

```
private void geoLocate(){ //ищет данные на карте
    Log.d(TAG, msg: "geoLocate: geoLocating");

    String searchString = mSearchText.getText().toString(); //получение данных введенных в поисковике

    Geocoder geocoder = new Geocoder( context: MainActivity.this); // выполнение действия на карте
    List<Address> list = new ArrayList<>();
    try{
        list = geocoder.getFromLocationName(searchString, maxResults: 1);
    }catch (IOException e){
        Log.e(TAG, msg: "geoLocate: IOException: " + e.getMessage() );
    }

    if(list.size() > 0){
        Address address = list.get(0);

        Log.d(TAG, msg: "geoLocate: found a location: " + address.toString());
        //Toast.makeText(this, address.toString(), Toast.LENGTH_SHORT).show();

        moveCamera(new LatLng(address.getLatitude(), address.getLongitude()), DEFAULT_ZOOM,
            address.getAddressLine( index: 0)); // получение данных ширины и долготы, потом отправляет в метку
    }
}
```

Рисунок 9. Код для поиска

Затем после того, как были найдены координаты, в то место вставляется метка, и камера пользователя направляется в эти координаты, таким образом работает функция "Поисковик" (Рисунок 10).

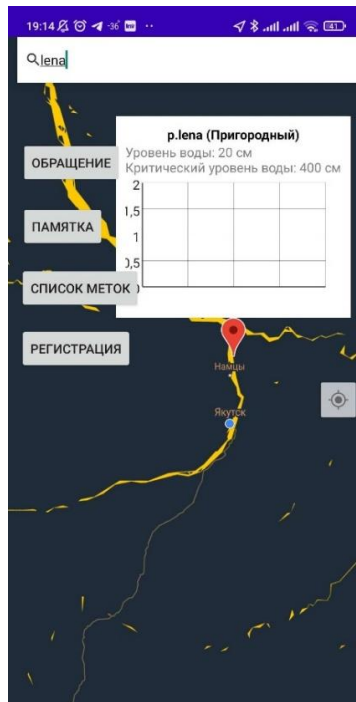


Рисунок 10. Поисковик

#### 4.7 Разработка регистрации

Немаловажная функция “регистрация”, с помощью регистрации планируем связать сайт и приложение, сохранять избранные и сможем уведомлять людей по почте (Рисунок 11).

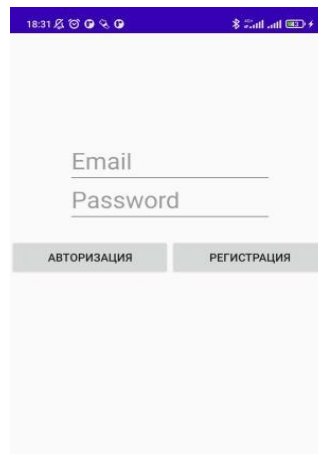


Рисунок 11. Интерфейс регистрации в приложения

С помощью Firebase Authentication был подключен функция регистрации и возможность отправлять сообщения на почту (Рисунок 12).

Identifier	Providers	Created ↓	Signed In	User UID
slava.gogolev.1989@gmail...	✉	Feb 1, 2023	Feb 1, 2023	zD7p2n5lpMa5Ww2VuakZ0kRVTK...
slava.gogolev.2005@gmail...	✉	Jan 28, 2023	Jan 28, 2023	RBpj8rj0aYaNL7JTbU03A7u01v73

Рисунок 12. Данные пользователей

Таким образом, можем соединить 2 системы в одно целое, сайт и приложение, также можно отправлять оповещения о паводках по почте, наилучшим вариантом будет если отправлять во всевозможные оповещения, так как пользователь вероятнее всего заметит об уведомлении опасности.

#### 4.8 Разработка страницы для памятки

Было принято решение поставить в приложении памятку при отсутствии интернета, которую можно посмотреть, когда угодно, даже когда отсутствует интернет. (Рисунок 13)

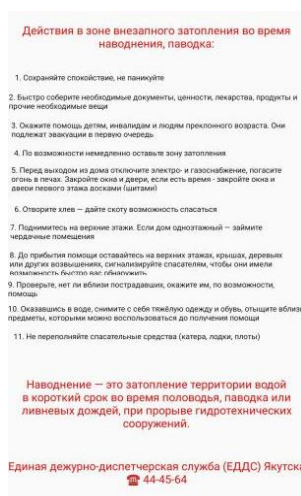


Рисунок 13. Памятка

#### 4.9 Разработка списка меток

Также хотелось бы упомянуть про “список меток”. На данный момент функция на стадии доработки и планируется добавить туда поисковик и 2 вкладки, первая - список рек, а вторая — это избранные. Если добавить в избранные определенную метку, то будете получать уведомления по этой метке. При нажатии реки будет выводиться график и вся информация реки в отдельном окне (Рисунок 14).

Лена	Уровень воды:
	Критический уровень воды:
Виллойск	Уровень воды:
	Критический уровень воды:

Рисунок 14. Список меток

#### 4.10 Дизайн карты

Также с помощью [mapstyle.withgoogle.com](http://mapstyle.withgoogle.com) был обработан сам стиль карты. Дизайн карты был изменён с целью увеличить заметность рек и убрать лишние детали на карте, по типу зданий (Рисунок 15).

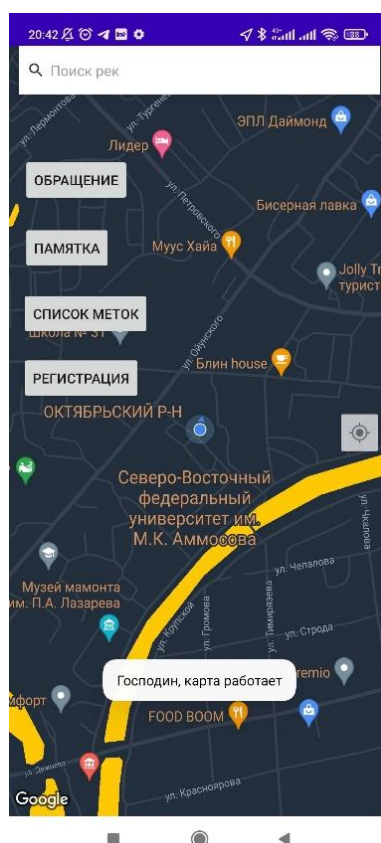


Рисунок 15. Текущий вариант приложения

Приложение находится на стадии разработки, в будущем планируется оптимизировать и добавить дополнительные функции.

## 5. Заключение

В ходе научно-практической работы были выполнены задачи:

1. Изучены статические данные и аналоги.
2. Разработан и оптимизирован база данных в Firebase, для получения информации.
3. Разработан прототип приложения на платформе Figma
4. Изучены основы java и android studio.
5. Изучена библиотеки Google maps и Firebase
6. Функционал приложения практически доработан и осталась лишь стилистическая часть

В дальнейшем планируется добавить оповещение о критической уровне воды, также доработать функционал приложение “список меток”, добавить табло информации о реках в “список меток”, также добавить график уровня воды.

## **6. Список использованной литературы**

1. Ущерб от паводков в Якутии составил почти 1 млрд руб // РИА Новости URL: <https://ria.ru/20130529/940157391.html> (дата обращения: 14.12.2022).
2. Водоучет на реках // База знаний URL: <http://www.cawater-info.net/bk/4-2-1-8-2-4.htm> (дата обращения: 25.10.2022).
3. Автоматическая система контроля уровня воды // Патент России № 2657360. 2018. Бюл. № 17. / Глядченко Ю. Г., Куликов А. Н. [и др.].
4. Общее описание метеостанции // Метеомастер URL: [http://www.meteomaster.ru/products/catalog/Gidrologicheskoe\\_oborudovanie/Gidropost/](http://www.meteomaster.ru/products/catalog/Gidrologicheskoe_oborudovanie/Gidropost/) (дата обращения: 21.10.2022).
5. Народный Мониторинг // Narodmon URL: <https://narodmon.ru/> (дата обращения: 05.10.2022).
6. Паводки 2022 // <https://www.eastrussia.ru/news/predvaritelnyy-ushcherb-ot-pavodka-podschitali-v-yakutii/>
7. Статистика по паводкам за 2022 // ТАСС URL: <https://tass.ru/proisshestiya/16627185>
8. Итоги сезона пожаров и паводков 2022 // Yakutia Daily. URL: <https://yakutia-daily.ru/v-yakutii-podveli-itogi-pozharoопасного-i-pavodkovogo-sezonov-2022-goda/>