I listen my speaking agent reading book fragments as I walk by

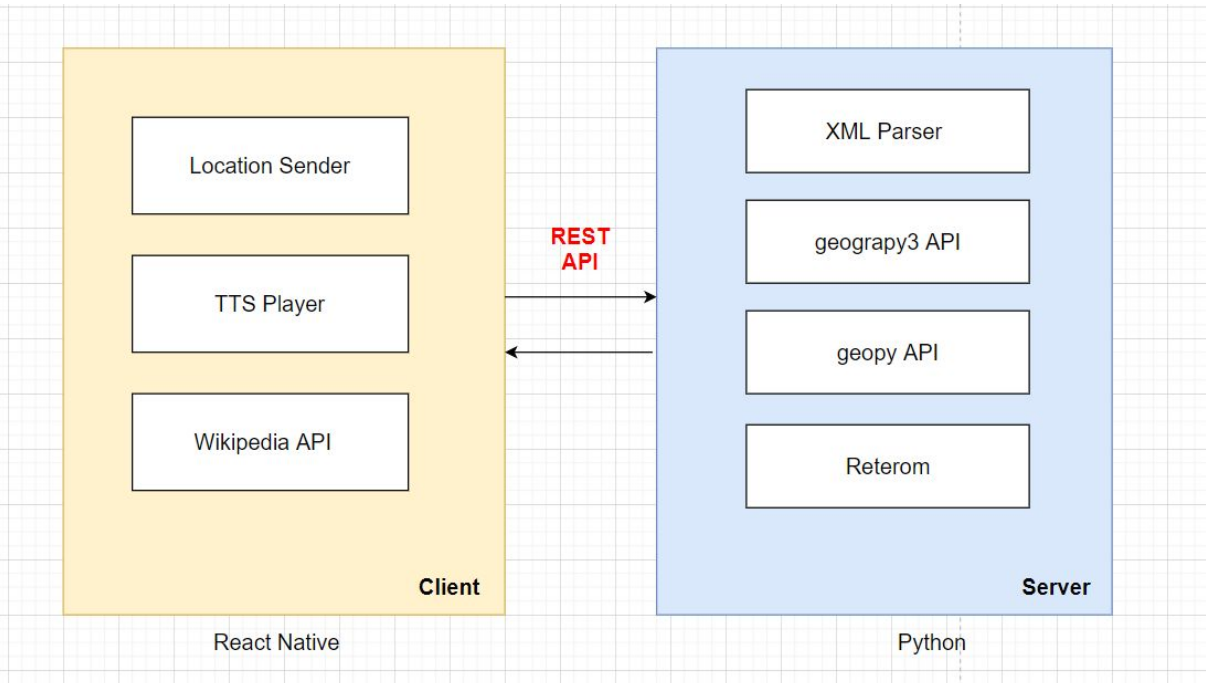
READING ASSISTANT

Chiperi Andrei (A1), Stoica Ioana Dana (A1)

Project’s short description – project’s goal*:*

The project has an .txt file as input with some geographical locations and based on the actual location of the user’s phone, we search the nearby locations wich can be found in the file, and the mobile application will read some informations about that place (informations taken from google).

Architecture:



The application is structured on client-server architecture:

The client – is the mobile app which send the location of phone to the server, and receive information about the location sent using REST API.

The server - here takes place the reading of the file, finding the locations from the text (using geograpy3 API), the transformation of the locations (words) into latitude and longitude (using geopy API) as well as calculating the nearby location from the actual location of the user, received from the the client module.

Project’s details

Motivation: We chose this project because it has an interesting idea and can be used in a cultural/ educational way. With the use of this app, the user can experience more deeply the walk in the places he/she is visiting, without the help of a travel guide.

Challenges: using of Reterom project and location finder;

State of The Art: we found useful the following articles:

- *MappingBooks*
- *https://towardsdatascience.com/geoparsing-with-python-c8f4c9f78940*

Objectives: to create an app that can be used in a cultural purpose.

Project management:

- ➔ Client side part & team leader – Andrei Chiperi
- ➔ Server side part – Ioana Dana Stoica

The client module has several sub-modules:

- Play audio file module - play in background audio files, avoid the sound stop due to application close;
- Call Google Text To Speech – call Google Translate API “to play” some dummy text;
- Design module – styles of application components;
- Send location module – send current location, speed, altitude of the mobile phone(we should use speed to avoid to use our application by using a car a bike, or something else);

The whole application is written in React Native, and it should be render for Android OS, and IOS .

The server module has several sub-modules:

- XML parsing - (in the beginning);
- finding the locations from the generated text (using geograpy3 API);
- the transformation of the locations (words) into latitude and longitude (using geopy API);
- returning the nearby locations from the actual location of the user, received from the the client module;
- Reterom - transform text into speech;

- I have used ngrok for an instant, secure URL to my localhost server through any NAT or firewall;

The server module will be written in Python.

Detailed deadlines and milestones:

27th February 2020: State of The Art (the articles that we found most useful).

5th March 2020: Project's Architecture, Project Description.

Milestone 1: State of The Art.

12th March 2020: Client module implemented.

19th March 2020: parsing the XML file

26th March 2020: finding the locations from the generated text

2th April 2020: calculate distance between two locations (latitude/ longitude)

9th April 2020: find the nearby locations from the current location

16th April 2020: Server module implemented.

Milestone 2: Client and server modules implemented.

23th April 2020: using of a server to send the locations to the client module

30th April 2020: Client and server modules integrated.

14th May 2020: Further tests. Writing the documentation.

28th May 2020: Writing the documentation.

Milestone 3: Client and server modules integrated. Writing the documentation.

Qualitative measures:

Ability to extract places from both long and short sized texts just as well

TEST REPORT

Client Module:

1. Some information in the backend was too cluttered, so I took the following aspect as an optimization measure: at the beginning of accessing each available API, I save the city in a variable, and I sanitize the text received as input - I remove everything that means digits> 5 (reducing the possibility of a postal code in the text, which will disrupt the API for collecting information about where we are);
2. Since RETEROM microservice generates audio files and stores them for a while in the file system, there was no need to save them locally on the phone, so I solved the problem with excessive consumption of the device's storage memory;
3. The collection of information is done using their extraction from a simple google search. However, given that we are trying to make the application only for the Romanian language, so places in Romania, this collection of information may be possible at present, using the google search algorithm, which returns results using more information from our browser (distance from that place, the country we are from, the city). Therefore, the persistence of microservices cannot be ensured by any cloud type service, because they are kept in the cloud, somewhere in servers outside Romania. I tried GOOGLE and HEROKU as cloud services.

Server Module:

1. I don't know if that was a problem, but in the beginning of the project I have parsed in Python the xml file presented on the site; after that I have changed that with reading a text from a .txt file;
2. I had a problem for extracting locations for a text which is larger; the API would return to me the name of the nearBy city and not of the nearBy region/place.
   Solution: I have replaced geopy API with a Python file that calls Google Maps API and returns latitude and longitude;
3. Another problem I have struggled with was deploying my service on Google Cloud Platform.

Current application usage scenarios:

This app can be used if you are a tourist in a foreign city or country and you do not have a travel guide or a friend to guide you or to present you essential informations about the places you walk by.

Future use scenarios:

For the moment, the informations of the nearby place are taken from Wikipedia. It can be improved in the future by taking this information from some texts or books.

Also, it can also be added some AR/VR functionalities such as, when you fix the phone on a specific monument/ place, you can see on the screen of your phone the name of that monument as well as listening to specific information of that monument or place.

In fact, this application can be reduced to a single module that can be incorporated into an AR / VR application. I worked on a project in the 2nd year of college in the course of Programming Engineering, which resulted in an application that, by scanning some objects, had the possibility to provide information about the object in question. However, the application was managed by someone who adds photos, information, audio files, GPS coordinates. Here comes the Reading-Assistant application, which, through its functionalities, brings this set of information.

Another usage scenario is when a group of children go on a historical trip, and based on this application they can create their lesson in a PDF format based on the places visited.