**Mapalyze Backend Codebase Documentation**

***Overview***

This document outlines the backend architecture for the Mapalyze application, detailing the functionalities of two separate backend servers designed to manage user authentication and process architectural plans.

***Servers***

1. Mapalyze BE

- Technology: Dotnet Core

- Function: This server is dedicated to user authentication using OAuth. It facilitates secure login and signup processes by interfacing with user databases and APIs.

2. React Planner BE

- Technology: JavaScript with Node.js

- Function: Handles file uploads and the processing of architectural projects to generate 3D floor plans.

***Detailed Functionality***

**React Planner BE**

1. POST /upload-png

- Triggered By: Mapalyze Frontend (Mapalyze FE).

- Description: This endpoint specifically handles the upload of PNG files, ensuring that the uploaded file is appropriately validated and managed.

- Process:

- Validate the incoming PNG file.

- Save the file as `floor\_plan.png` in the `updates` folder.

- If an older `floor\_plan.png` exists, it is removed.

- A success message is sent back to Mapalyze FE.

- Upon successful upload, the user is redirected from Mapalyze FE to React Planner Frontend (React Planner FE).

2. GET /process-projects

- Importance: This endpoint is critical as it initiates the core functionality of the application.

- Triggered By: User action in React Planner FE, specifically the "Fetch and Generate" command.

- Description: Executes a sequence of operations to transform a 2D floor plan into a 3D model.

- Process:

- Run a Python parser script that processes `floor\_plan.png`.

- Convert the image into `output.png`.

- Execute automated drawing algorithms to create `default\_filename.json`.

- Replace the existing `default\_filename.json` in the `jsons` folder with the new file.

- The newly created `default\_filename.json` is then sent back to React Planner FE, where it is used to display a 3D floor plan.