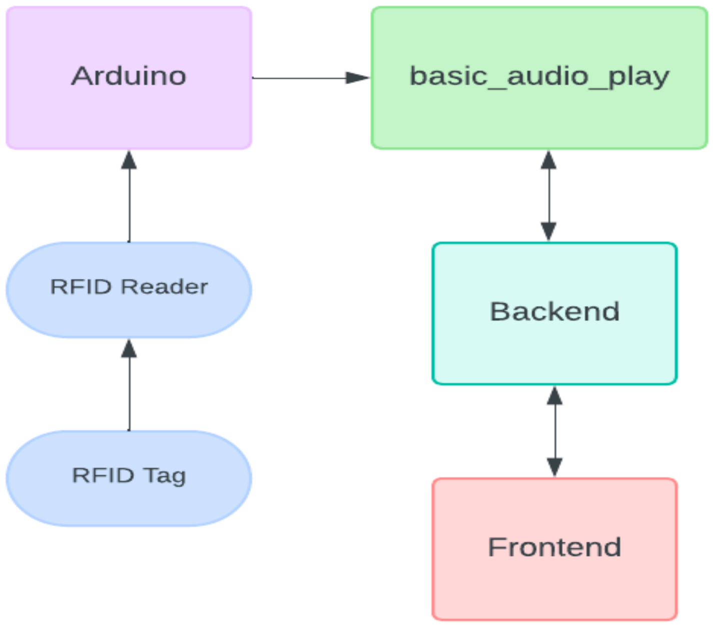
StoryQuest: Technical Notes

The Architecture



Arduino Integration with RFID Readers

When an Arduino file is uploaded to the RFID readers, it enables the reader to write the identifier of the RFID tag that is tapped into the serial port.

Basic Audio Playback Functionality

The basic_audio_play program operates by reading the serial port. It then prompts the user to register a character. If the character is registered, the program plays the associated sound file (if available) corresponding to the tag.

Backend Implementation using Flask

The backend, developed with Flask, serves as the intermediary between basic_audio_play and the frontend. It facilitates various HTTP requests such as GET, PUT, PATCH, and DELETE. Additionally, it manages data storage, including serial ports and associations between character names, readers, tags, and sound files within the database.

The endpoints exposed for frontend interaction include:

To get all character names
Get Request
http://127.0.0.1:5000/getAllChar

Returns [] if there is no tags registered yet

Otherwise, will return a list of all character names, something like this

[{'name': Bob'}, {'name': 'Mary'}]

To add a new character
Patch Request
http://127.0.0.1:5000/modifyChar

Data to pass in = {"name": name , "tag": tag, "reader": port_name}

To get a tag that doesn't have a name
Get Request
http://127.0.0.1:5000/getRecentlyScannedChar

Returns [] if all tags have a name or there is no tags registered yet

If there is will return a list with 1 item, something like [{'tag': tag,   'reader': string}]

To Update a character with sound file, new name and reader association
Patch
http://127.0.0.1:5000/updateChar

Data to pass in = {"prevName": previous name , "reader": reader, "newName": new name, "soundFile": path to sound file}

To delete a character

Delete

http://127.0.0.1:5000/modifyChar

Data to pass in = {“name”: name of character to delete}

To add a serial port

Put

http://127.0.0.1:5000/ports

Data to pass in = {"reader1": serial port, "reader2": serial port, "reader3": serial port, "reader4": serial port}

Serial port is of the form: "/dev/cu.usbmodem142101"

To get serial port

Get

http://127.0.0.1:5000/ports

Returns [{port: serial port}, {port: serial port}, {port: serial port}, {port: serial port}]

To get character name, reader and sound file associations

Get

http://127.0.0.1:5000/updateChar

Data to pass = {name: character's name}

Returns [{name: character's name, reader: reader associated with character, soundfile: path to sound file}]

Frontend Development with React

The frontend, built using React, interacts with users by receiving input and sending HTTP requests to the backend. There are three pages: Arduino, Character, and Sound registration. To populate these pages, GET requests are sent to gain previously saved information, and a series of PATCH, PUT and DELETE requests send user information to be saved.