**Implementation Details**

**Project documentation**:  include any design document: architecture, use cases, test cases, diagrams, etc.  source code, code comments.Please ask your faculty adviser if you have any question. If the source files are too large, include a link to cometSpace(box).

The project source code with comments is located under the following link: https://github.com/lfuksman/AirQualityMain

Additionally, the code had been delivered to the company sponsor.

Architecture:

Test cases:

1. In order to test that user is being added to the database upon registering:

* Clear the cache and choose to sign up into Auth0 login page
* After signing up, execute the following query:

SELECT \* from Users where Username=(email of the user registered)

* Make sure that the query returns the desired record with user role set to

1. In order to test that the admin is redirected to amin dashboard:

* Log into Auth0 login page using the following credentials:

Username: [utdproject2020@gmail.com](mailto:utdproject2020@gmail.com)

Password: testAccount06!

* Make sure that the admin is redirected to a page displaying the list of all users

Diagrams:

Authentication: Auth0 connection

**Step 1: Create Auth0 account**

Create an Auth0 account

Create Auth0 regular web application in Auth0 website dashboard

In order to be able to run the code locally:

Add <http://localhost:5000/callback> to Allowed Callback URLs

Add <http://localhost:5000> to Allowed Logout URLs

Add <http://localhost:5000> to Allowed Web Origins

Note: ensure that the port number in Python code matched the port number in Auth0 application. For example, if the port number is 3000 in the server.py, change the URL to use localhost:3000.

**Step 2: Create connection inside the code and database**

Make sure to install Python3.6.8

Make sure to install using pip install the following:

flask

python-dotenv

requests

authlib

six

pyodbc

Jinja2

urllib3

flask\_table

wtforms

flask\_cors

The connection to Auth0 api happens in server.py lines 20-23 and 35-45.

Make sure that there exists a database called Authentication with the following tables:

* Users

Username varchar(50)

UserRoleId int

* UserRoles

Id int

UserTitle varchar(64)

**Discussion on Impact and Security issues**(How this project will affect the user, community, city, etc?  have to use password protected resources? did you use encryption?, proprietary solutions?),

The project is a powerful tool which can be used among those patients who experience episodes of arrhythmic heartbeat.

Facilitate doctor: do not spa with loads of info, do not miss crucial info

Alert patient in real time

Due to security reasons, it was decided to not handle any user passwords. In fact, when the user signs up, his authorization is handled by Auth0 server and thus our application does not receive any password information. Our databases only store the email of the user and therefore there is no need to use encryption.

Both Auth0 authentication and LightWave visualization in the project were based on open source code publicly available on GitHub. However, the machine learning algorithm is patented and therefore NDA is required to gain access to the its code or to any information regarding the output. Because of that we could not get access to output of the algorithm and had to make the part of the mobile app that receives the output as very broad in order to allow future developers who can access to algorithm to customize it in order to fit the format of the output data.

**Individual assessment**

**Individual assessment** - one paragraph/page for each team member. Include one paragraph/page per team member indicating your role and how you contributed, make sure to add completed task/pending tasks including deadlines. All the paragraphs/pages should be included as part of the final report file, approved and signed by all team members, sponsor and advisor.

Lirit Fuksman

Role: wrote authentication autoecg.utdallas.edu website and was a point of contact for the team

First, I had researched the best authentication tool for the website. The team was given the choice of either use Auth0 api or find another similar service that would perform the authorization of users. Indeed, the rationale behind using a third-party service was a security concern as we did not want to store any passwords in our databases. After some research, Auth0 was picked because it satisfied all of our needs as well as was free up to a certain number of users. Next, I have created Auth0 web application and wrote a Python backend that would call the api and then have a callback to our code. As a part of this code, I have first created a local database with two tables: one holding users and associated user roles and the second holding user roles and the associated descriptions. Those tables were then given to Danielle who was working on database part of the project. In pariticular, I have first replicated the existing frontend home page of the autoecg website. Next, I have added a route that called the Auth0 server once the Login button is clicked. I have also created a user dashboard and an admin dashboard. Moreover, in order to allow the admin to alter user roles without accessing the database, I have created an editable form that contains a list of all users and allows to edit the role for the selected user. Upon saving the changes, the database is updated.

The tasks described above are summarized below with the corresponding deadlines:

|  |  |
| --- | --- |
| Task | Deadline |
| Replicate the existing front end of autoecg website. Create Auth0 web application and create a local database with tables to allow to store users and their associated user roles. Create a connection between SQL server and the python code. Allow the users to login into the website and be navigated to a page with a link to lighwave visualization page. Make sure to save user information during the entire session. | March 16, 2020 |
| Differentiate the users based on user roles and redirect the users to their respective screens. Create an admin dashboard that would display the list of all currently registered users. Add logout functionality. | April 6, 2020 |
| Add “Edit” functionality to admin dashboard. Create an editable form to allow to edit each user separately and add a save route that will make the respective changes to the database. | April 14, 2020 |

Moreover, I was the point of contact for the team, communicating with the advisor and helping to organize meetings.

* **Issues and lessons learnt.**An individual summary of issues you had to face, how you managed to overcome those issues, things you learnt,  extend the appendix to add references that you found useful, add step by step instructions to install, use, extend any feature (iff any), etc. Identify one major concept you used from your previous courses, Identify one major item you learned during this term to succeed in this course. Your thoughts on how we can improve senior design course further.

Lirit Fuksman

The biggest issue I have encountered was installation of local SQL server database and configuration of the driver to be accessible by PyCharm IDE. In fact, I have started to write a project on a mac computer; however, after trying for couple of hours to install the local database, I gave up and switched to a windows computer. I still had problems to install the database and after few more hours of research managed to create a local database in SQL server 2014. While the edition is old, it worked for my purposes as I only needed the database to access the users and update user information. Perhaps, if I was responsible for the database part of the project, I would have continued to search for another solution.

Thus, one of the things I learned was how to install a local database and to configure the driver. In fact, I also learned that the configuration of the driver depends on the local machine and thus the appropriate comment is made in the source code. I have also learned how to setup source control system as well as commit new changes. Indeed, in other classes I did not have to use source control systems while in my internships the system was already set by devops team. Thus, the project allowed mw to learn how to do it by myself. Moreover, while working on the project, I learned Python and Flask framework. Before starting to work on the project, I did not know any Python. Because the patented machine learning algorithm provided by Dr. Lakshman Tamil was written in Python, our group has decided to write the project in Python as well and thus I had to learn on the fly. While I have attempted to learn Python before starting the project, it turned out that actually writing code was much more effective way to grasp the syntax of Python and I truly enjoyed the learning curve.

The major concept from a previous course that was useful was SQL and concepts of primary and foreign keys from database course. While my part of the project did not involve the database directly, being able to quickly setup some tables allowed me to work on authentication in parallel with Danielle instead of waiting for her to create all database infrastructure.

Python was the major item which I had to learn to succeed in the project. Because I had experience with api requests in C#, I already knew the concepts and thus only had to learn the Python syntax as it related to rendering html pages, making function calls, calling api and writing inline queries to access the database.

To improve senior design project: