Report Date: 11/4/2022
To: ematson@purdue.edu, ahsmith@purdue.edu and lee3450@purdue.edu
From: Chasing FOX

- Daye Kim (rlaek4793@khu.ac.kr)
- Hyun Roh (yeshyun@khu.ac.kr)
- Heewon Jeong (jhjmo0719h@chungbuk.ac.kr)
- Jihu Yang (zihooy@handong.ac.kr)
- Juwon Baek (21700340@handong.ac.kr)

Summary

- In the Back-end part, developing the API that runs python code was started using Flask. And for the API related to navigation and route, database tables were constructed.
- In the Front-end part, the Page of application has been completed step by step. And to manage state storage efficiently, the Redux setting was carried out.
- In the Data Analysis part, the population normalization formula was re-defined. And to combine the risk of the crime and the urban factors, the machine learning method was used.
- In the Algorithm part, the method to assign the risk score to the algorithm was considered.

What Chasing FOX completed this week:

- **Data analysis**
  - The population normalization formula, which is to remove the population effect from crime occurrence, was re-defined to calculate statistically logical results as follows:

$$population\ density_{zipcode} = number\ of\ people_{zipcode} \div area_{zipcode}$$

$$adjust\ crime\ density_{data\ point} = crime\ density_{data\ point} \div \frac{population\ density_{zipcode\ k}}{\sum_{k=1}^{n} population\ density_{zipcode\ k}}$$

    where n is the total number of zip code areas in Chicago, 59, and k means each zip code area. Each data point of crime density was divided by each population density ratio according to the corresponding zip code area.
  - To combine the risk of the crime and the urban factors, the machine learning method was used. The model was developed to predict crime occurrence in 2019, using crime occurrence data and urban factors in 2018. Through this model, the weight of crime and urban factors can be found in training, and the risk of crime and urban factors can combine to the output value.

- **Algorithm**
  - It was confirmed whether detailed road shape information could draw through the edge and node.
  - The method to assign the risk score to the algorithm was considered. The risk score is calculated based on each node, and the risk of the edge is calculated based on the average of the connected nodes.

- **Back-end**
    - To construct an API that runs python code, the flask was used. The previous flask, apache environment setting was done [1].
    - The user, location, navigation, route MySql database tables were constructed in the API related to navigation and route.
- **Front-end**
    - The Login page has been developed that contains crimes you want to avoid and account information. The login page UX was also redesigned.
    - The environment setting of the Redux was carried out to manage state storage efficiently [2].

Things to do by next week

- The Signup page will be developed with the Back-end part.
- The test of the Front-end server will be carried out with the Back-end part.
- In the Back-end part, the API that runs python code will be developed to request json including original, destination coordinates and user's KDE value, and response json including 9 way points.
- After confirming the machine learning results, a methodology for combining each risk score will be established.
- In the algorithm, the Risk score will be calculated and assigned for each node and edge.

Problems or challenges:

- Although There was a concern about whether to implement the login function or not. The login function has been decided to be implemented.
- When recalling data analysis results, there was a problem that took too long.

References

[1] "mod_wsgi (Apache)." Pallets. https://flask.palletsprojects.com/en/2.0.x/deploying/mod_wsgi/ (accessed Nov 03, 2022).

[2] "Usage Guides." Redux. https://redux.js.org/usage/ (accessed Nov 03, 2022).