**LA Parking Ticket Wizard Project Description**

**Team Members: Weiwen Zhao, Chan Woo Yang, Jin-Uk Luke Shin**

**GitHub Link: https://github.com/UPenn-CIT599/final-project-la-parking-ticket-wizard.git**

**Introduction:** This program is based on insights generated from a dataset published on Kaggle about historical City of Los Angeles parking ticket citations. The data contains more than 9 million citations and lively updated. We analyzed this big data set of parking tickets by area, time of day, and day of week to see a trend and generate visual charts. Then the user interface asks user’s location, day and time information and predicts likelihood of getting tickets based on historical data analysis.



Figure 1. Overall Program Flow

The overall flow of our project is shown in Figure 1. The Parking Wizard software can be divided into three parts: Data Cleaning part, Data Analysis & Visualization part, and GUI part for the user interaction. Big data analysis always starts from the data cleaning part. The raw data contains many forms of invalid data (e.g. issue date out of range, empty issue date/issue time), so such data needs to be removed before the data analysis. *FileHandler* class is used to collect all raw data from the .csv file and store them in the HashMap format. Then, this raw data is passed to *DataCleaner* class to remove any invalid data set and saves in another HashMap format. Data analysis & visualization part takes this cleaned data from the new .csv file and can be used in two different cases. The first case is running the data analysis & visualization over the entire cleaned dataset to see the analysis of parking tickets in the city of Los Angeles (we call it as Big Data Analysis). *ParkingTicketDataProcessor* class runs data analytics to understand ticket issuing patterns. It analyzes tickets by hourly, daily and each day's hourly ticket patterns. It also analyzes 10 most commonly issued parking tickets in LA and its corresponding fines. Then, using JFreeChart API, four charts are generated for visualization of these data (two bar charts and two pie charts). These charts are accessible from the GUI interface button as well. The other case is running the data analysis & visualization over the specific area around the end user's location. Using JavaFX GUI interface, we ask user’s location, time and day of the week. Then, we analyze historical ticket information based on this specific region using *Location* and processed by *ParkingTicketDataProcessor* class. Then, *LikelyhoodPredictor* class predicts likelihood of getting the ticket and provide the information to the user. Also, *HeatMap* class generates heatmap information based on zoning of LA city to identify which zone contains more issued tickets. *ParkingTicketWizard* combines *DataCleaningRunner* class and *ParkingTicketRunner* class to coordinate the workflow of the backend side of the program. And then, the workflow of the frontend side and the user interaction part are managed by the GUI class.

Figure 2 shows the techniques deployed for this project to fulfill the requirements.



Figure 2. Techniques Deployed

Responsibilities:

* Data processing and clean CSV generation: Chanwoo
* Big data analysis: Jinuk Luke, Chanwoo
* Graph generation using JFreeChart API: Jinuk Luke
* Runner and console-based user interface: Chanwoo, Jinuk Luke
* Zone based analysis including HeatMap: Weiwen
* Likelihood prediction and ParkingDataProcessor interaction: Weiwen, Chanwoo
* JavaFX API User Interface: Weiwen