**CS5800 – Advanced Software Engineering**

**Cal Poly Pomona**

**Homework 2**

**Spring 2024**

**Description:**

CPP Maps

(OOA and OOD)

Name: Fidelis Prasetyo

Email: ([fprasetyo@cpp.edu](mailto:fprasetyo@cpp.edu))

BroncoID: 015765555

# Use Case (OOA)

* Navigation: The system loads the map from the database to the screen. If the user clicks on the navigate button on the screen, the system gets the starting location and destination location that the user entered. The search engine searches those locations from the database. Markers of those locations are placed on the map according to their coordinates. If the markers are not visible on the screen, zoom in/ out the map until the markers are at the center of the screen. A route generator calculates the shortest route from the starting location to destination location. From the calculated route, the system generates a list of directions from the starting location to destination location. As the navigation runs, the system will get and update the user’s current location periodically to update the directions as the user moves. The user will also be visible as a marker on the map, which will move along the map according to their location.
* Search: If the user clicks on the search button on the screen, the system gets the input query of the location that the user entered. A query identifier identifies the query whether it’s a name of location or not. If it is the name of a location, the search engine searches through the database and returns a search result which is the most relevant location to that input query. The search result consists of the location, name of the location, address information, phone number, and web address. The marker of that location is placed on the map, if the marker is not visible on the screen, zoom in/ out the map until the marker is at the center of the screen.
* Explore: If the user clicks on the search button on the screen, the system gets the input query of the location that the user entered. A query identifier identifies the query whether it’s a category or not. If it is a category, the search engine searches through the database and returns a list of search results that belong to that category. The markers of those locations are placed on the map, if the markers are not visible on the screen, zoom in/ out the map until all markers are visible.
* Share location: When the user clicks the share button, a share window will pop up on the screen to choose the desired share method. If a messaging app button is chosen, a url generator will generate a url based on the current selected location, then a messaging service will pass the url to the messaging app.
* Share rideshare: When the user clicks the share button, a share window will pop up on the screen to choose the desired share method. If a rideshare app button is chosen, a rideshare service will send the user’s current location and the selected location to the rideshare app (to calculate the ride cost).

# Use Case (OOA)

* Potential classes:
  + Map
  + Database
  + Screen
  + Location
  + Route
  + User
  + SearchEngine
  + SearchResult
  + Marker
  + Query
  + QueryIdentifier
  + SearchResultList
  + UrlGenerator
  + MessagingHandler
  + RouteGenerator
  + RideshareHandler
* Rough graph:

UrlGenerator

Location

RideshareHandler

RouteGenerator

Database

Screen

SearchResult

Route

Map

Marker

SearchResultList

SearchEngine

QueryIdentifier

User

Query

MessagingHandler

*isCategory*

*display*

*zoomIn*

*zoomOut*

*getLocation*

*isOutOfBounds*

*display*

*highlight*

*display*

# OOD

* CRC cards

|  |  |
| --- | --- |
| Map | |
| Display graphical map data | Marker |
| Display marker of the location(s) | Route |
| Display the calculated route between locations |  |
| Highlight a marker |  |
| Zoom in/ out when marker(s) is/ are out of bounds |  |

|  |  |
| --- | --- |
| Location | |
| Provide geographical position data | SearchEngine |
| Store location information (address, etc) | Marker |
| Translate user’s current/ selected position into location data | User |
|  | Route |

|  |  |
| --- | --- |
| Route | |
| Calculate the route between 2 locations | Location |
| Display the route on the map | Map |
|  |  |

|  |  |
| --- | --- |
| User | |
| Get current/ selected location | Location |
| Input search query | Query |
| Choose desired actions | MessagingHandler |
|  | RideshareHandler |

|  |  |
| --- | --- |
| SearchEngine | |
| Handle search queries | Query |
| Return the search result as location | Location |
| Return the search results as a list of location |  |

|  |  |
| --- | --- |
| Query | |
| Store search parameters from user | User |
| Identify the search query (by name/ category) | SearchEngine |
|  |  |

|  |  |
| --- | --- |
| Marker | |
| Mark certain points on the map | Map |
| Get the position of the marker according to location | Location |

|  |  |
| --- | --- |
| MessagingHandler | |
| Receive user’s selected location | User |
| Generate a url from the selected location |  |
| Handle communication with messaging app |  |

|  |  |
| --- | --- |
| RideshareHandler | |
| Receive user’s current and destination locations | User |
| Handle communication with rideshare app |  |

* Final Graph

*isCategory*

*isOutOfBounds*

*isOutOfBounds*

*displayMap*

*displayRoute*

*displayMarker*

*highlightMarker*

*zoomIn*

*zoomOut*

User

RideshareHandler

MessagingHandler

Route

Marker

Map

SearchEngine

Location

Query