**Homework 2**

**CPP MAPS (OOA and OOD)**

**Bronco ID: 016861260**

**Name: Ruchitha Gande**

**Part 1 – Use Case (OOA)**

Use case divides   
\* Title.

\*Actor

\*Success stories

\*Preconditions

Here we are creating CPP maps application where need to write 5 selected use cases.

1. **Title:** Navigation

Actor: User/Driver

Success stories:

* User/Driver enters their starting location and destination location (address, landmark)
* User/Driver chooses the mode of transport (Walk, Bike, Car)
* CPP maps shows the efficient routes.
* User selects one of the listed routes and can start the route.
* CPP maps provides turn by turn directions.

Preconditions: User has internet access enabled.

User has Location enabled.

1. **Title:** Search

Actor: User

Success stories:

* User enters the name of the place or monument (“White House”)
* CPP maps app displays all the matching places along with their information (Address, Images etc.)

Preconditions:

* Enter a Valid place name to recognize and give information of place.

1. **Title:** Explore

Actor: User

Success stories:

* User chooses one of the pre-defined categories (gas, restaurant, bank)
* CPP Maps displays all the places for that category.

Preconditions:

* User GPS and internet access.

1. **Title:** Share Location

Actor: User

Success stories:

* User searches for a place and selects the share.
* Users select the apps to send location (iMessages, WhatsApp)
* CPP Maps open the selected app along with the link.

Preconditions:

* Both the user and the selected person contacts has messaging app.

1. **Title:** Share Rideshare

Actor: User

Success stories:

* User enters their starting location and destination location and clicks on rideshare.
* CPP Maps shows the available riding apps and their estimated cost with the vehicle information.

Preconditions:

* User should have internet turned on.
* User have riding app in their devices.

**Part 2 – (OOA)**

1. Highlight all the nouns to determine possible potential classes.

* **Navigation:**

Location: Represents the starting point and destination.

Directions: Provides information on how to reach the destination.

Route: The calculated optimal path to follow.

Map: The graphical representation of the navigation.

**Nouns:**

CPP Maps application

Starting point

Destination

Route

Traffic conditions

Distance

User preferences

Application

Map

Travel time

Directions

* **Search:**

Place: Refers to the locations or points of interest.

Search Query: The user's input to find specific places.

Address: Detailed information about the location.

* **Nouns:**

CPP Maps application

Place

Name

Search results

Query

* **Explore:**

Category: Represents the user's area of interest, such as restaurants or parks.

Location: Provides geographical information about places within the selected category.

Destination Details: Specific information about each place, like addresses or opening hours.

Search Results: The list of places matching the chosen category for the user to explore further.

* **Nouns:**

CPP Maps application

Category

Places

Location

Results

* **Share Location:**

User: The individual utilizing the CPP Maps application.

Messaging App: Communication platforms for sharing.

Location Message: Information shared containing the location.

* **Nouns:**

CPP Maps application

Location

Map

Link

Code

Messaging apps

Social media

* **Share Rideshare:**

User: The person initiating the sharing of the destination.

Rideshare App: The application for booking rides.

Location Request: The destination information transferred to the rideshare app.

* **Nouns:**

CPP Maps application

Destination

Rideshare app

Rideshare services

Device

Destination location

Rideshare app

1. Highlight all the verbs and verb phrases to determine possible methods and relationships.

**Navigation:**

**Verbs:**  Search, Location, Directions, Calculate, Route, Display

searchLocation(startLocation, endLocation): Finds locations (optional, if separate from Search).

getDirections(startLocation, endLocation): Calculates route and directions.

calculateRoute(startLocation, endLocation): Internal method for route calculation.

displayRoute(route, map): Displays the route on a map.

**Search:**

**Verbs:**  Search, Name, Results.

searchByName(searchQuery): Finds places matching the name.

getSearchResults(): Returns a list of matching places.

**Explore:**

**Verbs:**  Category, Location, Display

filterByCategory(category): Filters locations by a chosen category.

findLocations(category): Retrieves locations in a chosen category.

displayLocations(searchResults): Presents a list of filtered locations.

**ShareLocation:**

**Verbs:**  Search, Place, Name, Display, Location, Messaging

searchPlaceByName(placeName): Finds a specific place by name (optional, if separate from Search).

displayLocation(location): Shows the location on a map.

sendLocation(location, recipient): Shares the location with a chosen recipient.

openMessagingApp(messagingApp): Opens the chosen messaging app for sharing.

**ShareRideshare:**

**Verbs:** Location, Details, Rideshare, App

sendLocation(location): Shares the location with a rideshare app.

getRideDetails(location, rideshareApp): Retrieves ride options and details.

openRideshareApp(rideshareApp): Opens the chosen rideshare app.

requestRide(rideDetails): Initiates a ride request with the chosen app.

1. Create a rough graph showing how all the classes are connected.

A diagram of a business

Description automatically generated

**Part 3 – (OOD) NOTE**: Use word editor such as MS word/Apple Pages

1. Create CRC cards of the classes you have selected from the potential list above.

Use the following template for your CRC.

|  |  |
| --- | --- |
| User | |
| Responsibilities  \* Represents a user of the app.  \* Stores user information (name, location preferences, etc.).  \* Initiates searches, explorations, navigations, location sharing, and rideshare requests.  \* Provides input (search criteria, start/end locations).  \* Receives output (search results, route details, location information). | Collaborators  \* Search  \*Explore  \* Navigation  \* ShareLocation |

|  |  |
| --- | --- |
| Search | |
| Responsibilities  \*Performs searches based on user criteria (name, category, etc.).  \*Returns a list of matching places with relevant information (name, address, category, etc.). | Collaborators  \*User  \*Place  \*Category |

|  |  |
| --- | --- |
| Explore | |
| Responsibilities  \*Allows users to browse places by category (e.g., restaurants, gas stations).  \*Presents a list of relevant places with basic information. | Collaborators  \*User  \*Place  \*Category |

|  |  |
| --- | --- |
| Place | |
| Responsibilities  \*Represents a specific location of interest (e.g., restaurant, landmark).  \*Stores place information (name, address, category, opening hours, reviews, etc.).  \*Provides methods to access and update place details. | Collaborators  \*Search  \*Explore  \*Navigation  \*Share Location  \*Rideshare |

|  |  |
| --- | --- |
| Navigation | |
| Responsibilities  \*Calculates routes between two locations, considering factors like traffic and road closures.  \*Provides turn-by-turn navigation instructions.  \*May display the route on a map. | Collaborators  \*User  \*Place  \*Map |

|  |  |
| --- | --- |
| Share Location | |
| Responsibilities  \*Shares the user's current location or a specific location with a chosen recipient.  \*Integrates with messaging apps or other sharing methods. | Collaborators  \*User  \*Place  \*messaging App |

|  |  |
| --- | --- |
| Rideshare | |
| Responsibilities  \*Opens the chosen rideshare app with the user's location or a specific location.  \*May display estimated fares and available vehicles. | Collaborators  \*User    \*Place    \*RideshareApp |

|  |  |
| --- | --- |
| Route | |
| Responsibilities  \*Represents a planned path between two locations.  \*Stores route information (start location, end location, intermediate points, distance, estimated travel time, etc.).  \*May consider factors like traffic conditions, road closures, and user preferences.  Can be calculated by the Navigation class. | Collaborators  \*Navigation    \*User  \*Location  \*Map |

|  |  |
| --- | --- |
| Direction | |
| Responsibilities  \*Represents a specific instruction for a user to follow during navigation.  \*Stores information like turn type (e.g., left, right), distance to the next turn, and road name.  \*Provided in turn-by-turn format by the Navigation class. | Collaborators  \*Navigation  \*User  \*Route |

|  |  |
| --- | --- |
| Maps | |
| Responsibilities  \*Stores information about multiple places  \*Displays a visual representation of the geographical area.  \*Finds user selected category places near the user's location.  \*Finds places by name based on user query.  \*Shows user location. | Collaborators  Places  Navigation  Explore  Search  User |

b) Create a final graph connecting all the Class names with their Collaborators.

A diagram of a navigation system

Description automatically generated