System Design

Group: Planners

Project: planit

Instructor: Professor Dema

Teaching Assistant: Kevin Zhang

Table of Contents

CRC

*All changes have been written in red font.

Class name: AttractionsApi

Responsibilities:

- Checks if given response body is correct
- Responsible for retrieving possible attraction choices
- Responsible for validating attractions

Collaborators:

- *InsertDAO*
- *Validation*

Class name: TransportationApi

Responsibilities:

- Checks if transportation option is valid
- Responsible for sending response indicating if transportation option is valid

Collaborators:

- *Validation*

Class name: UserRegistrationApi

Responsibilities:

- Checks if given response body is correct
- Responsible for putting users into the database

Collaborators:

- *InsertDAO*

Class name: LocationValidationApi

Responsibilities:

- Responsible for receiving requests from front end
- Responsible for sending request with if the location was valid or not

Collaborators:

- *Validation*

Class name: *Validation*

Responsibilities:

- Validates the following data given from users:
    - o Transportation Method
    - o Start time
    - o End time
    - o Date
    - o Location
    - o Budget
    - o Distance
    - o Number of participants
    - o Attractions

Collaborators:

- None

Class name: *MyConnection<T>*

- Several classes don't have a reference to MyConnection<T>, but instead have a reference to the external source MyConnection<T> encapsulates. These references will also be referred to as MyConnection<T> for simplicities sake.

Responsibilities:

- Handles implementation of connection to an external source
- Handles implementation of closing a connection to an external source

Collaborators:

- None

Class name: Connector

Responsibilities:

- Calls the connect method of MyConnection<> implementations.
- Calls the close method of MyConnection<T> implementations

Collaborators:

- MyConnection<T>

Class name: JSONExchangeConverter

Responsibilities:

- Converting between an exchange object and a JSON object

Collaborators:

- None

Class name: *InsertDAO*

Responsibilities:

- Responsible for inserting things into the database.

Collaborators:

- *MyConnection<T>*

Class name: Globals

Responsibilities:

- Stores all singular global variables for the back-end use.

Collaborators:

- *MyConnection<T>*

Class name: *User*

Responsibilities:

- Knows its username, settings, and password
- Has an identifier

Collaborators:

- *Itinerary*
- *DAO*
- *Event*
- *Rating*

Class name: *Rating*

Responsibilities:

- Knows the identifier of the user that made the rating

- Knows the identifier of the event that was rated
- Responsible for getting rating information given by the user on the front end
- Has an identifier

Collaborators:

- *User*
- *Event*
- *DAO*

Class name: *Itinerary*

Responsibilities:

- Times and chosen events for those times.
- Knows what things the user chose for this itinerary
    - Number of people
    - Budget
    - Transportation options
    - Location
    - Total time
    - Maximum travel distance
    - Chosen user attractions
- Has an identifier

Collaborators:

- *User*
- *Event*
- *DAO*

Class name: *Event*

Responsibilities:

- Holds details about the Event
    - Description
    - Price point
    - Category
    - Availability (time it's possible to attend the event)
    - Contact info.
- Has an identifier

Collaborators:

- *Rating*

- *Event*
- *DAO*

Class name: *DAO*

Responsibilities:

- Has a reference to the database instance
- Putting information into the database
- Retrieving information from the database
- Deleting information from the database

Collaborators:

- *Itinerary*
- *Event*
- *Rating*
- *User*

System Architecture Diagram

Reference: https://www.edrawsoft.com/software-architecture-example.php