- **Language/Libraries:**
  - Java 1.8
  - Maven 3.5.4
  - H2 in-memory database
  - Maven Shade build plugin
  - Jersey 2.x
  - Hibernate 2
  - Hibernate JPA
  - Tomcat Embedded Container 8.0.23

- **Generic information:**
  - On application start up some data will be auto populated into the tables.
  - No external framework is used to implement the solution.
  - No external server or container is required to run this application. It can run as a standalone jar application and contains its own embedded container.
  - Prefilled some location data and for the sake of simplicity all the apis will operate under this limited number of locations, but it can be easily replaced with some live location APIs and the solution is expected to work.
  - Rent calculation is on a minute basis.

- **Solution approach and workflow:**
  - User registration.
  - User finds the cars.
    - Get all available cars.
    - Just like in real application this API calculates the distance between all the available cars and users current location.
    - Returns all the available cars within 1KM radius.
  - User blocks the car
    - User selects one car from all the nearest cars
    - Transaction marks start here and the current time is recorded as start time for the trip.
    - Selected car is set as unavailable to other users.
  - User ends the transaction.
    - Find difference between start time and current time in minutes, as rent specification is on minute basis.
    - (rent_per_minute * time_difference_in_minutes) gives the total rent for the trip and gets deducted from users balance.
    - Sets paid status as true and current time is marked as the end time of the trip.
    - Location of car and user is changed to the destination user selected when blocking the car.
    - Car is made available for users to block.