Truc Huynh

Purdue University

Client Sever Java Implementation

1. **Overview:**
   * Project Admin: Truc Huynh
   * Project using Java and build **without any framework**.
   * Using Apache Tomcat as the servlet.
   * Client and Server are built separately (Please refer to the UML)
2. **Test Plan:**
   1. Run Server and test if Server is function properly
   2. Simultaneously create three clients and test if they are function properly
   3. Upload the data to Server (#1)
   4. Send Request to Server (#2)
   5. Received Request from Server (#2) (#3)
3. **Implementation and Result**
   1. *Run Server and test if Server is function properly:*

Tested complete successfully.

* 1. *Simultaneously create three clients and test if they are function properly:*

Tested complete successfully.

* 1. *Upload the data to Server (#1):*

Create multithreading Server that can handle multiple clients. In Image 1, I run one server and three clients. The Server will receive a properties file (data) which upload by the client (#1), parse it, and store it on the server. (Validate data is successfully upload)

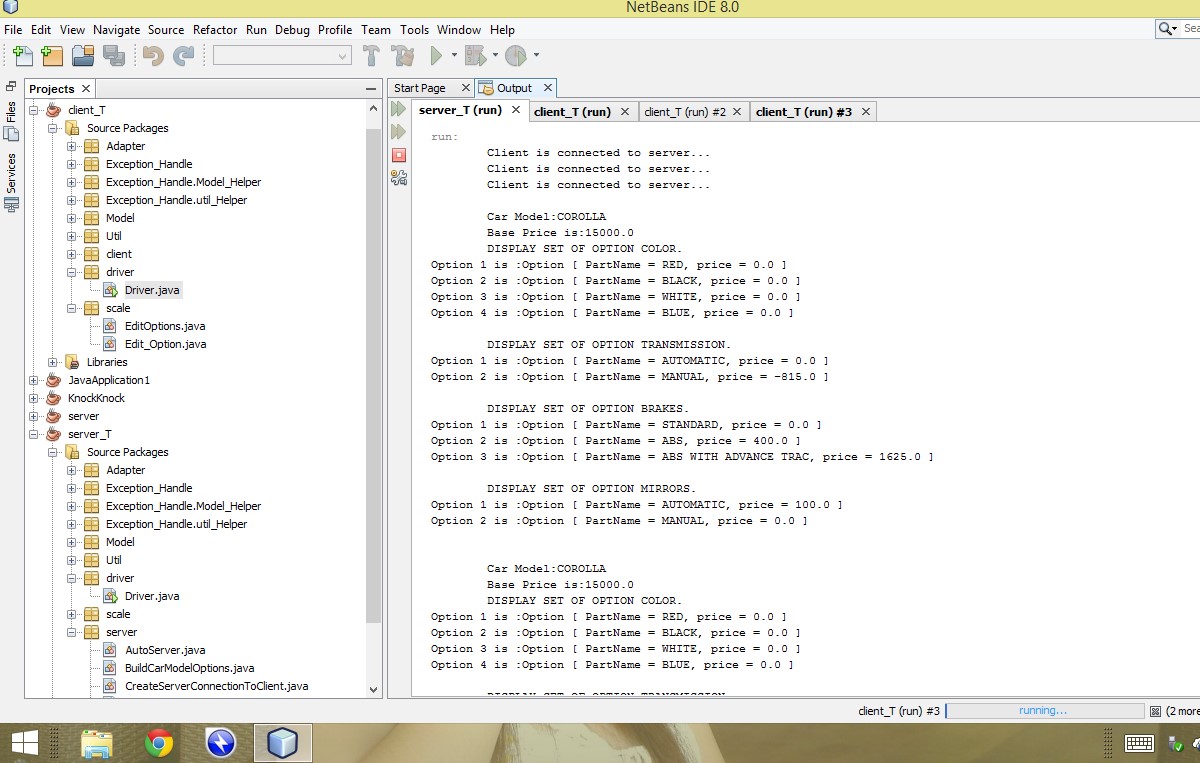


Image 1(Image by Truc Huynh, 2014)

After client\_T (#1) post (send) the text file (properties file) to the server. Server receive the properties file and parsed the text file to create an instance of Automobile and store it in the Link Hash Map (which is the application database) (Refer to Image 2).

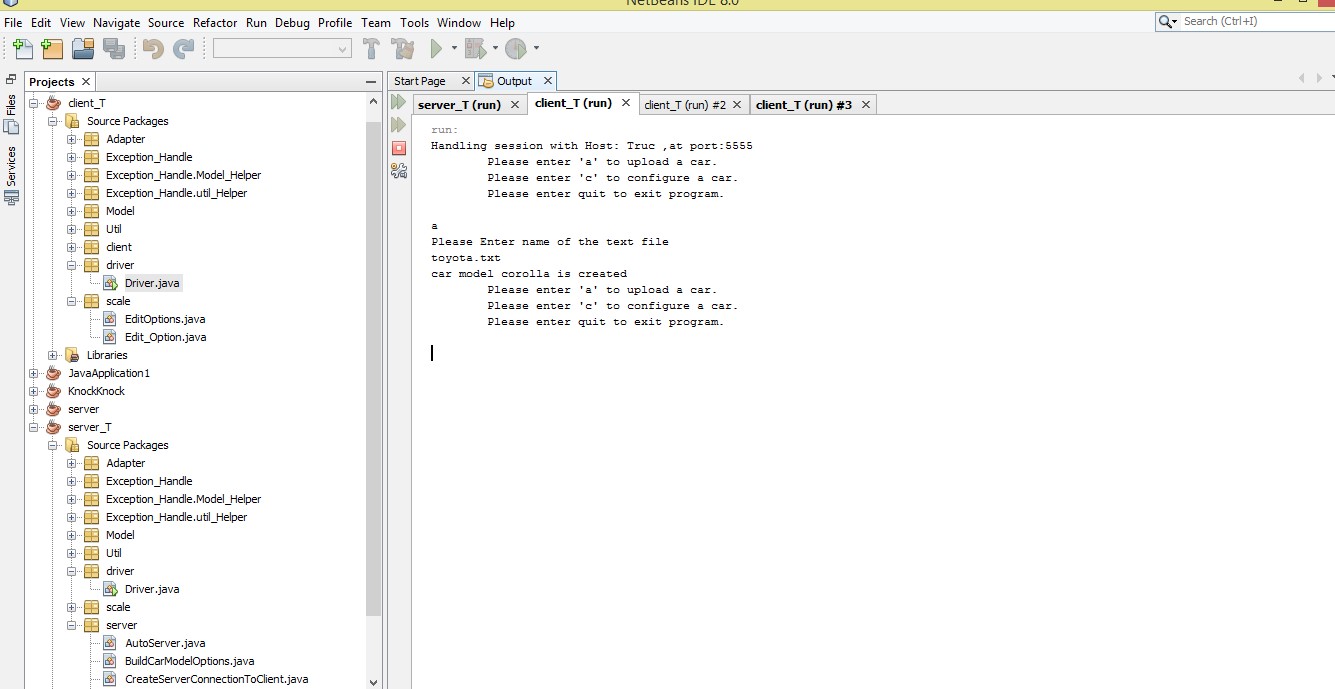


Image 2 (Image by Truc Huynh, 2014)

* 1. *Send Request to Server (#2)*

Client-T (#2) send a request to the Server for cars configuration. Then Server will send the respond back (instance of Automobile). (Image 3)

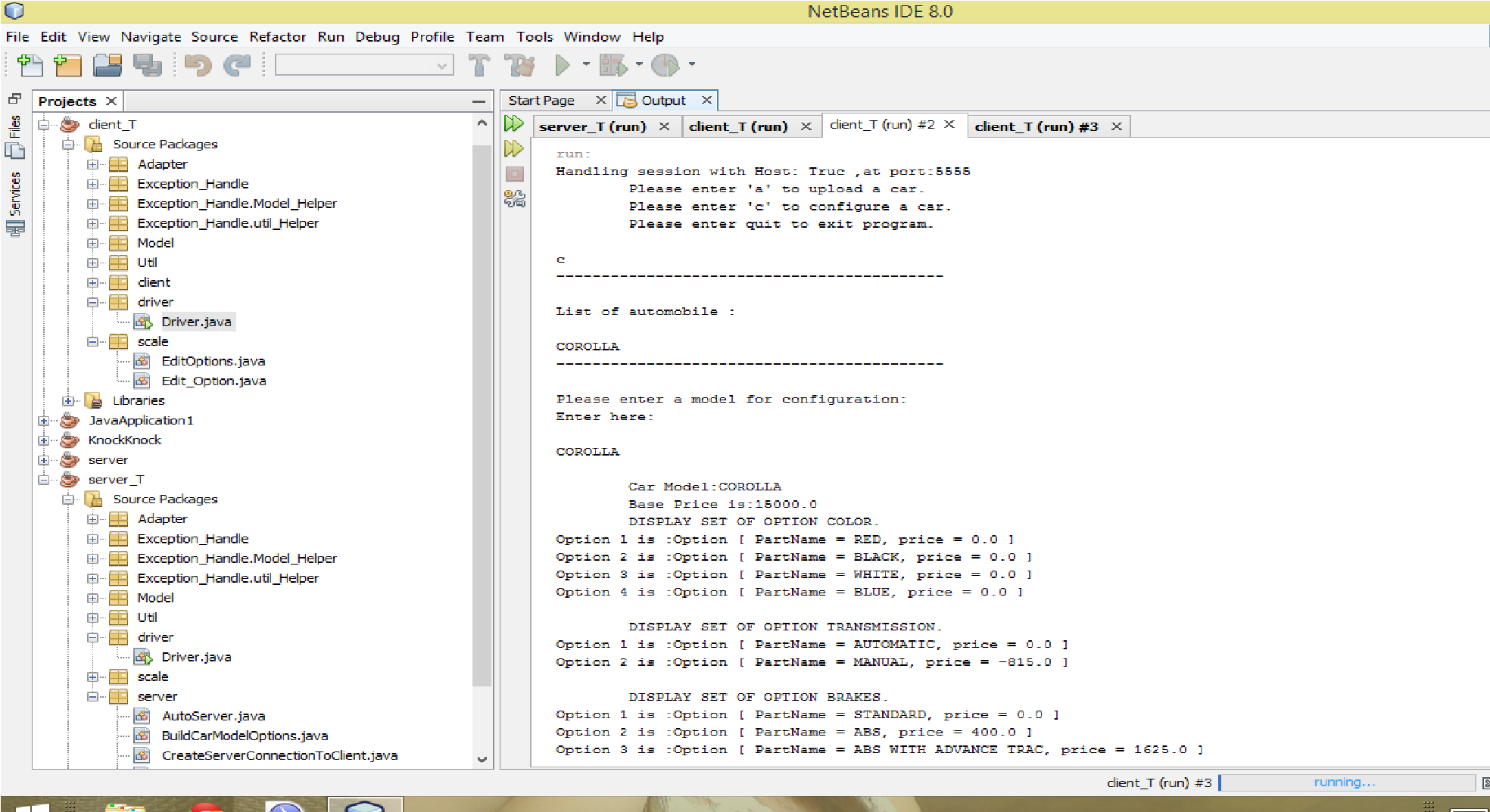


Image 3 (Image by Truc Huynh, 2014)

* 1. *Received Request from Server (#2) (#3)*

Client-T (#2) will get the choice from this instance of Automobile and store it within this Automobile. Picture below show successfully configurate the Car (Upload the data to server successfully) (Image 4)

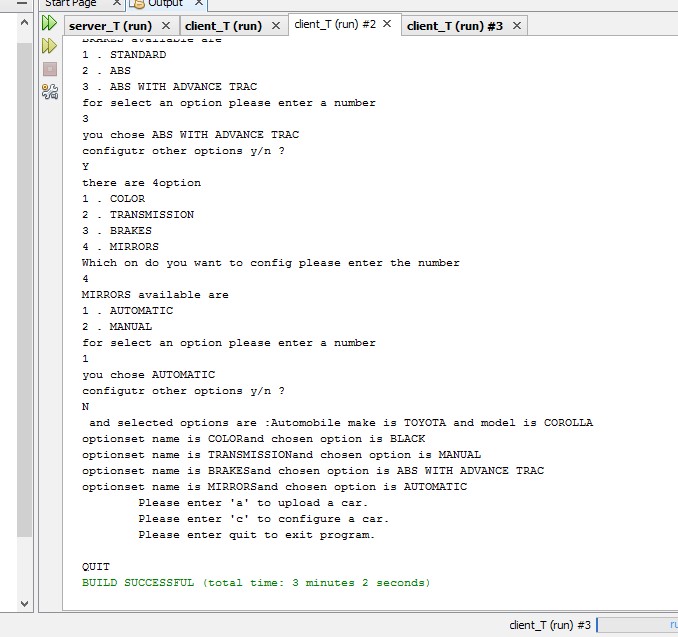


Image 4 (Image by Truc Huynh, 2014)

* 1. *Send and Received Request from Server (#3)*

Call Client-T (#3) to configurate a Car after terminate Client-T (#2) to verify if the data is successful upload to the Server. (Image 5)

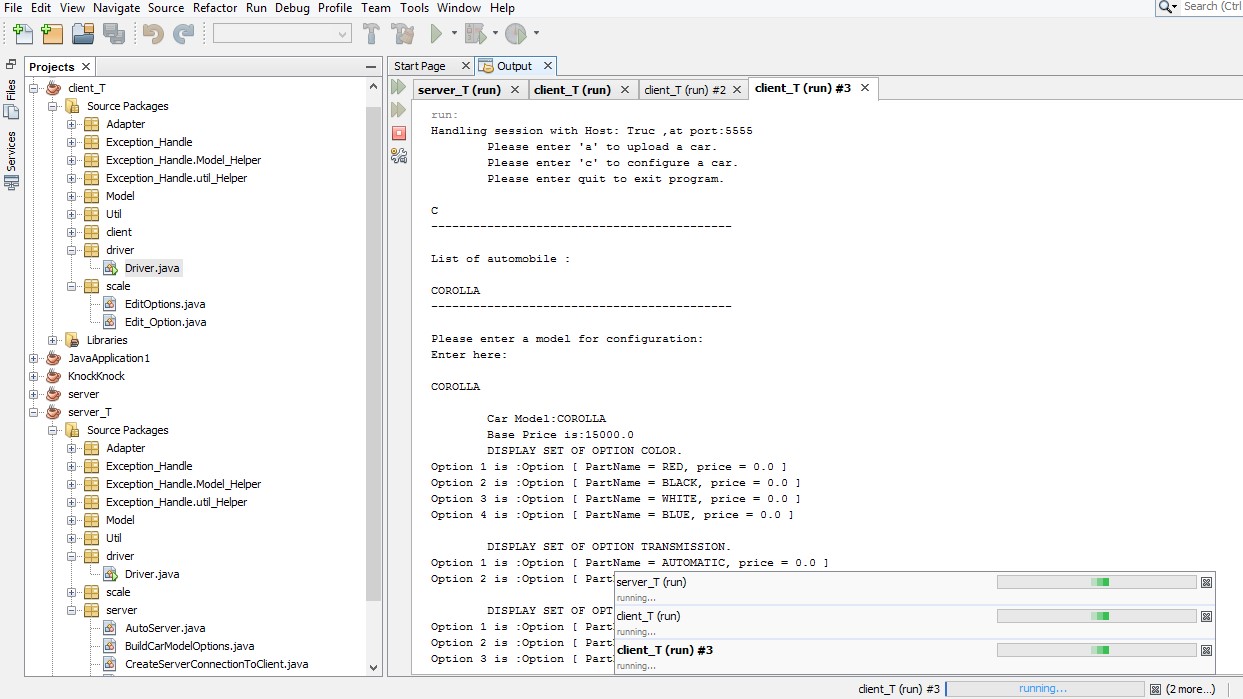
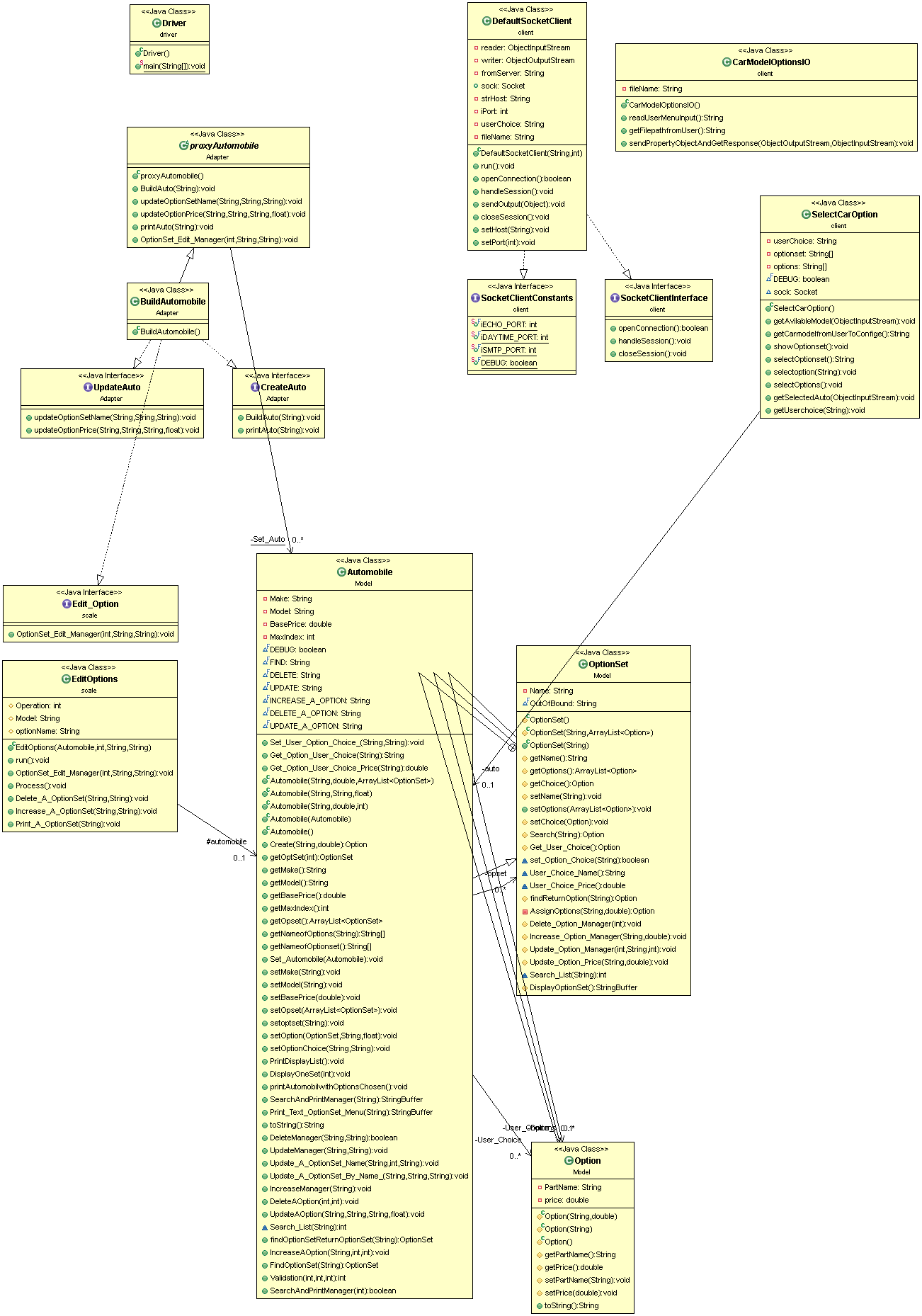


Image 5 (Image by Truc Huynh, 2014)

1. **Conclusion:**

The Server and Clients satisfy basic communication: send and receive request. Server receive upload data and store in its database (instance of Link Hash Map). The clients have ability to insert, update and delete an Automobile. No crashing (Implement Exception Handling)

1. **Architecture UML:**



Client UML, Designed by Truc Huynh

Server UML, Designed by Truc Huynh

