**Movie Ticket Booking System**

**Main Idea :**

Movie ticket booking system is a SOAP Web-service, which has below functionalities -

1.  View available shows and tickets available.

2. Book ticket.

In both the operations, Interaction with backend database is established for updating the tickets count and displaying updated movie info.

For co-ordination of operations, I am handling that at server side messageHandler. User can perform book the ticket operating only after viewing available shows and tickets operation.

**Technologies used:**

Java, Netbeans, GlassFish server, JAX-WS, MySQL database, JDBC-ODBC.

**MySQL database design :**

1. CREATE TABLE `moviesInfo` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`movie\_name` varchar(45) DEFAULT NULL,

`available\_tickets` int(11) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=latin1;

MoviesInfo table stores the movies name and available tickets count for that movie.

1. CREATE TABLE `login` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`username\_session` varchar(45) NOT NULL,

`operation` varchar(45) NOT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=60 DEFAULT CHARSET=latin1;

login table stores transaction information in the form of username\_session and operation name.

**Server Side:**

**Web services Operations:**

1. getMoviesInfo()

This operation basically gathers the movie-names and corresponding ticket counts by querying the moviesInfo table with simple JDBC-ODBC connection and ‘Select’ SQL statement. We are using complex data-structure that is MovieInfo which is a POJO. The instance of this class will be populated and sent as an response to the client for this operation

1. bookTicket()

This function takes movie name and ticket count as an input and books the ticket if invoked after getMoviesInfo() operation.

Upon invocation, this method updates the moviesInfo table by decrementing the ticket count for a movie chosen by a user for booking.

**Server Message handler:**

Message handler does following steps for co-ordination and authentication –

1. It fetches the Username from the client request header and generates session id, it fetches operation name from the SOAP body. All this information is stored login table.
2. If bookTicket() function is invoked and data entry in the login table is null then SOAPFaultException will be thrown to the client.
3. If bookTicket() function is invoked and last operation name entry in the table is getMoviesInfo() then the operation is allowed to execute successfully, else if the last entry in the database is bookTicket() operation then SOAPFaultException is thrown.

**Client Side:**

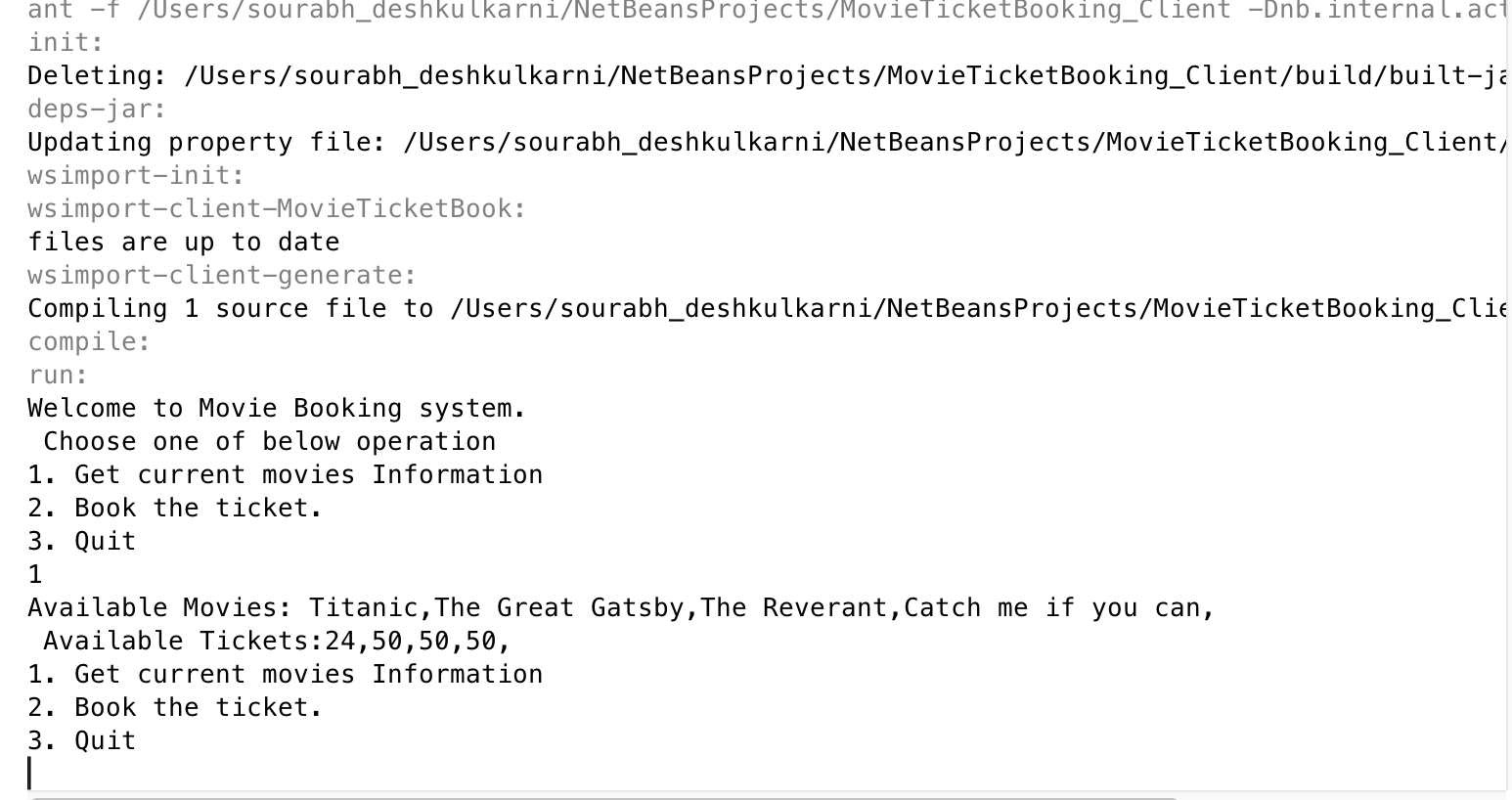
1. We import the WSDL file of our webservice into the client and access the available operations.
2. At client side simple interface has been provided to use the web-service. User will be prompted to choose the operation of his choice and based on his selection the flow will be decided.
3. In case of SOAPFaultException from server side, we catch the exception and print the error message.

**Client Message handler:**

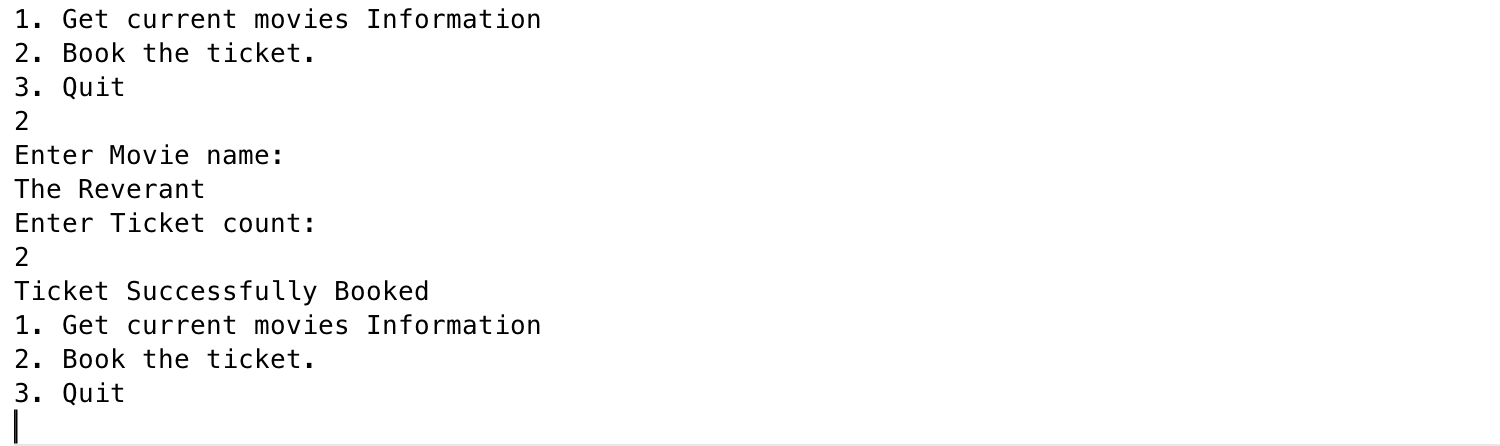
1. Here client username is embedded in the header of the request and sent to the server.

**Demonstration:**

1. **Initial screen – Select the option. If option -1 is selected then the available movies and ticket count will be displayed.**



1. **If the book ticket operation is selected, the ticket gets booked and we get the successful acknowledgement.**

****

1. **If the same operation, Book the ticket is invoked,we get the below error -**

