# **Environment Description**

Operating System: Microsoft Windows 10

Integrated Development Environment: NetBeans IDE version 8.2

Application Server: GlassFish Server version 4.1.1

Database server: JavaDB (Apache Derby 10.11.1.2)

Web Browser: Chrome

# **Instructions how to set up and run your project.**

1. Obtain NetBeans and configure it

* Go to <https://netbeans.org/downloads/> and download the Java EE version of NetBeans
* The downloaded file is called: netbeans-8.2-javaee-windows.exe. Run this file.
* During the installation, make sure to install GlassFish Server 4.1.1 and Apache Tomcat 8.0.27 as well

1. Load the javatutorials from the GlassFish distribution

* Go to <https://javaee.github.io/glassfish/download> and download glassfish: “[GlassFish 4.1.2 - Full Platform](http://download.java.net/glassfish/4.1.2/release/glassfish-4.1.2.zip)”
* Unarchive the downloaded file and copy it to a folder of your liking, let’s call it: <**downloaded GlassFish**>
* Navigate to <**downloaded GlassFish**>\glassfish4\docs\javaee-tutorial\examples\ and copy the contents of this folder to a work folder of your linking, let’s call it <**work folder**>.

1. Load the modified **hello1** code into NetBeans

* Navigate to <**work folder**>\web\jsf\ and unarchive here the provided **hello1.zip**
* Override all contents
* Start-up your NetBeans
* Go to File->New project->Maven->double-click Project with existing POM->Finish
* Navigate to <**work folder**> and click Load Project
* In the list of projects in NetBeans now we have a javaeetutorial group. Expand it until you find the **hello1** module:

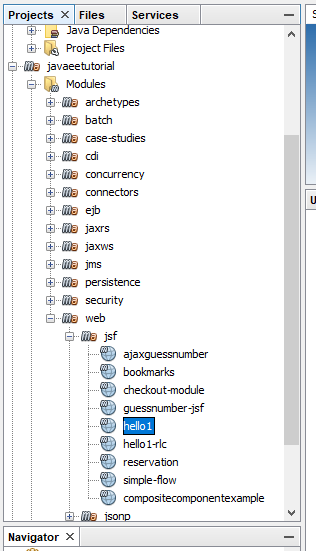


Figure 1 hello1 project in the javaeetutorials

* Double click the **hello1.** This loads the modified **hello1** project. We are going to use this tutorial project as base for our code modifications

1. Set-up the database

* Go to the Services tab in Eclipse and find the Databases -> Java DB.
* Right click and select “Create Database…”

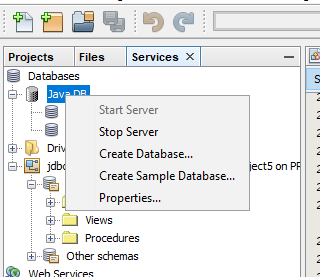


Figure Create new database

* Configure the database so that it fits the already existing configuration in the application:

Database Name: Project5

User Name: project5

Password: project5

Database Location: Click “Properties…”. Here make sure to select

Java DB Installation: The Java DB installation can usually be found in the JDK installation folder, as displayed in the picture below

Database Location: Any location, preferably in the current project folder

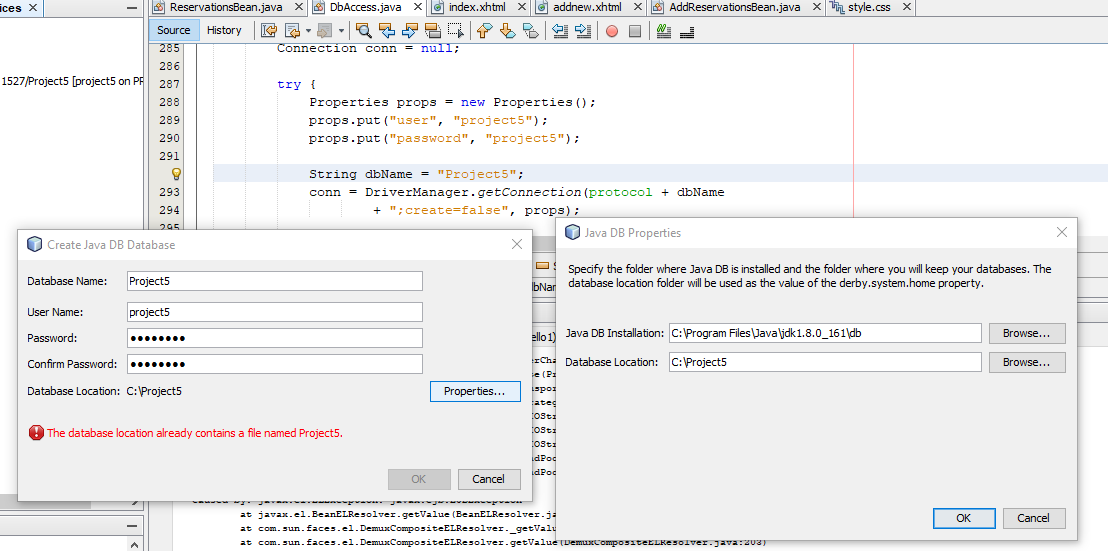


Figure Database configuration

1. Create the RESERVATIONS table:

* In NetBeans, right click the new database and click “Connect”
* A new jdbc connection is created
* Navigate to the Project5 database in this connection and right-click the “Tables” group. Select “Execute command”

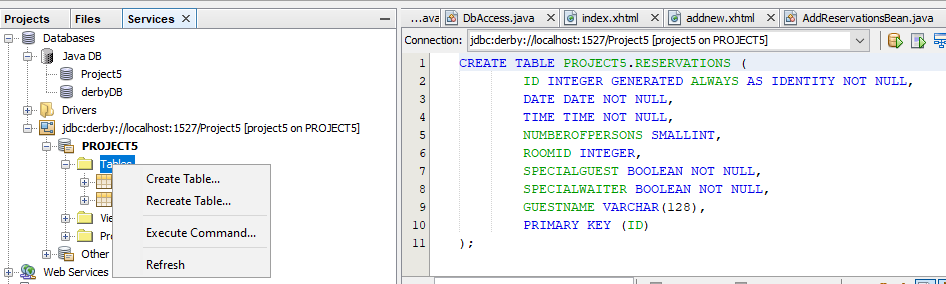


Figure Create new RESERVATIONS table

* Here is the sql used as create command:

CREATE TABLE PROJECT5.RESERVATIONS (

ID INTEGER GENERATED ALWAYS AS IDENTITY NOT NULL,

DATE DATE NOT NULL,

TIME TIME NOT NULL,

NUMBEROFPERSONS SMALLINT,

ROOMID INTEGER,

SPECIALGUEST BOOLEAN(1) NOT NULL,

SPECIALWAITER BOOLEAN(1) NOT NULL,

GUESTNAME VARCHAR(128),

PRIMARY KEY (ID)

);

1. Create the rooms table

* In the Project5 database in this connection and right-click the “Tables” group. Select “Execute command”

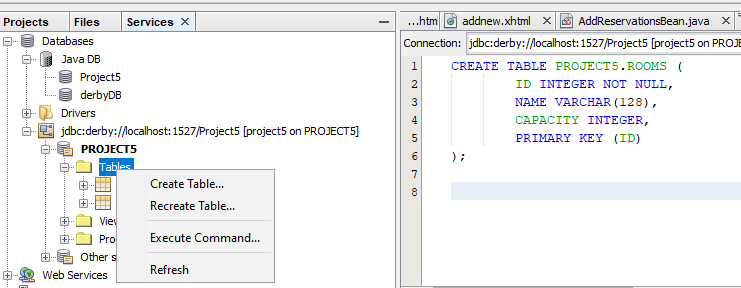


Figure Create the ROOMS table

* Here is the sql used as create command:

CREATE TABLE PROJECT5.ROOMS (

ID INTEGER NOT NULL,

NAME VARCHAR(128),

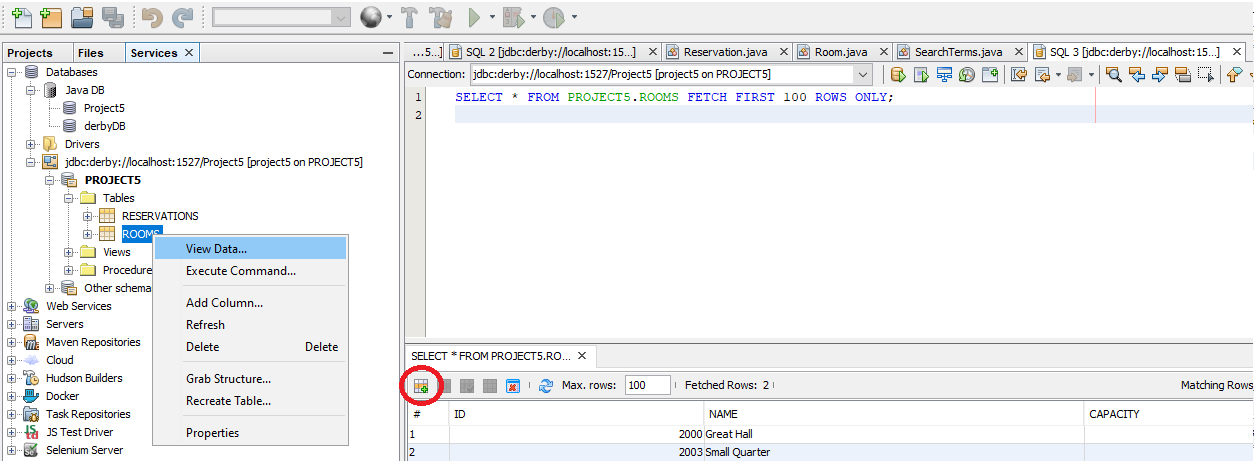
CAPACITY INTEGER,

PRIMARY KEY (ID)

);

1. Add some data to the ROOMS table

* Right-click the ROOMS table and select “View Data”.
* A table is displayed showing the contents of the ROMS table. Initially empty.
* Click the “Insert records button” to add a new room. Fill in the data, and then press “Add row”
* When you added all the rooms you need, press “OK’



1. Run the application

* Right-click the loaded project and click Run
* In the pop-up, select the GlassFish server and select “Remember in Current IDE Session”:

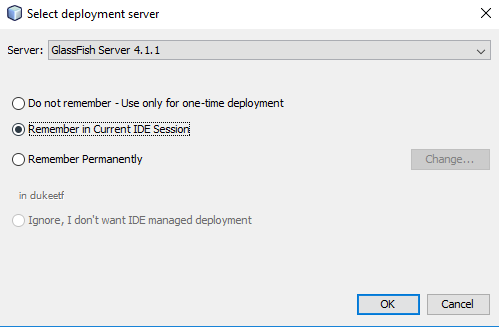


Figure 6 The server selection pop-up

# **Description of all the changes made to the hello1 project**

1. Overview of the project

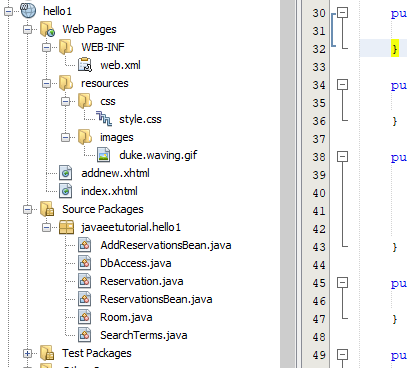


Figure Hello1 project overview

1. Added 1 css resource file to the project

For this, in NetBeans, I navigated to the hello1\Web Pages\resources\ and added there a new folder called “css”. Inside this folder I added the style.css file containing styles for our index page.

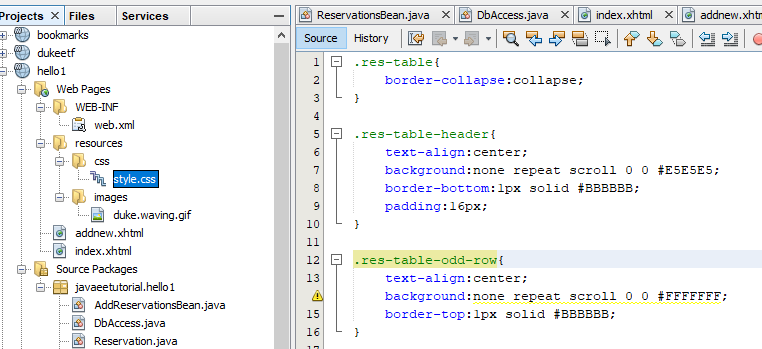


Figure 8 The style.css

1. Updated index.xhtml to contain a restaurant reservations page including a table displaying all reservations of the hotel, a button to go the another page where a new reservation can be created and an area where I can search by restaurant room name or guest name.

* Added 1 button to redirect to another page where a new reservation can be created

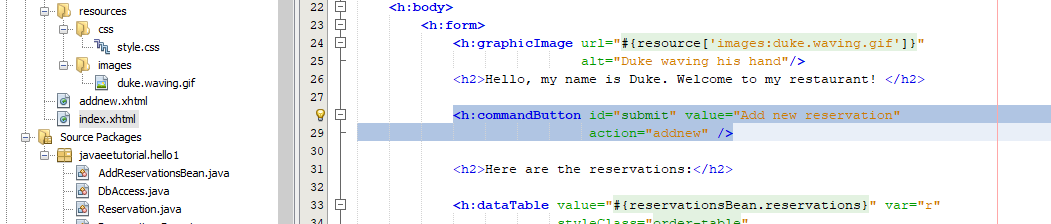


Figure Add new reservation button

* Added a new table to display all existing reservations

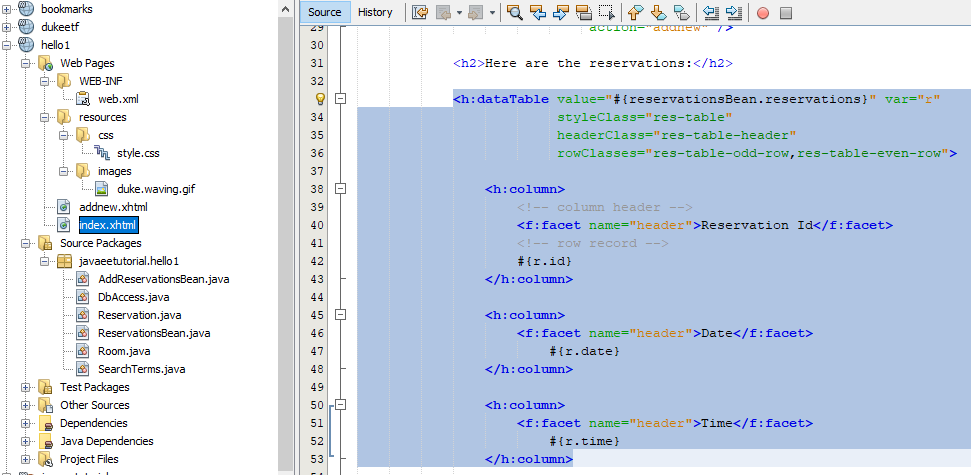


Figure Table to display all reservations

* Added new section to allow searches



Figure Search area

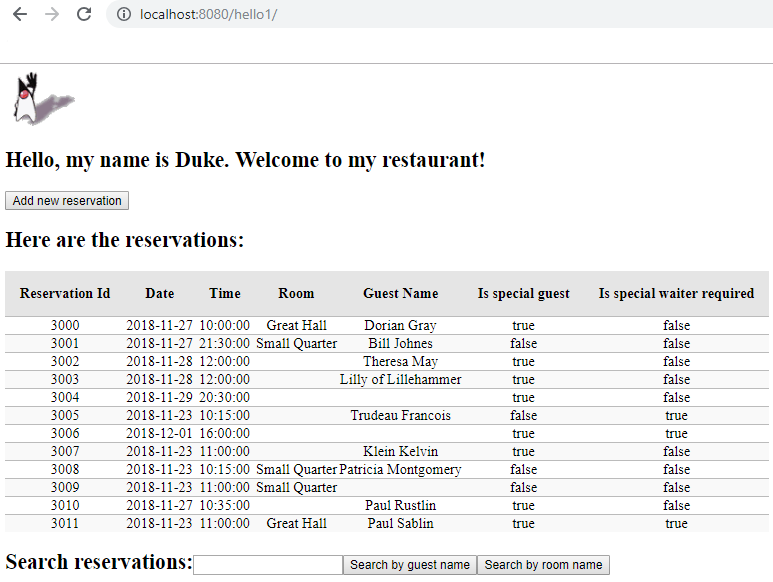


Figure The index page in the browser – example

1. Added addNew.xhtml page, where new reservations can be created. Here we have input boxes for all attributes of a reservation: date, time, number of guests (Number of persons), name of the person making the reservation (Guest name), whether this person is a special guest who should receive special treatment (Special guest), whether this reservation should receive a special experienced waiter (Requires special waiter), the restaurant room – in case the restaurant has several rooms where it can host guests:

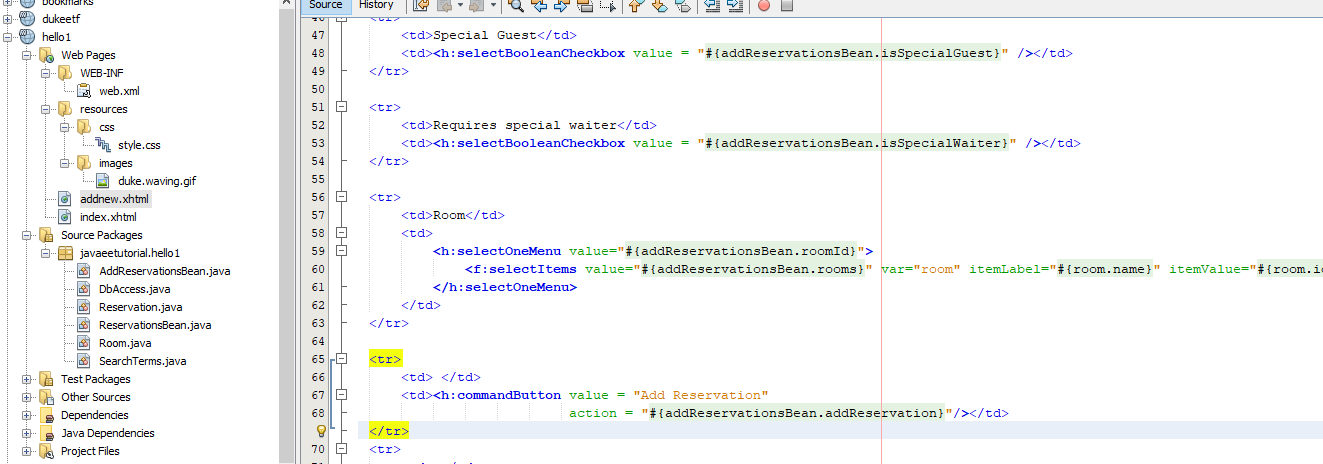


Figure Add new reservation page

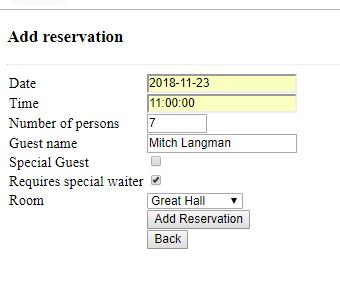


Figure Add new reservation page in browser

* Notice the formats necessary for date (yyyy-mm-dd) and time (hh:mm:dd)
* Notice that the selection of rooms is a drop down containing all rooms of the restaurant. In the current version of this application, the list is not editable, but the user can select any of the existing rooms for this reservation

1. Added new Reservation.java class to map the Reservation entity

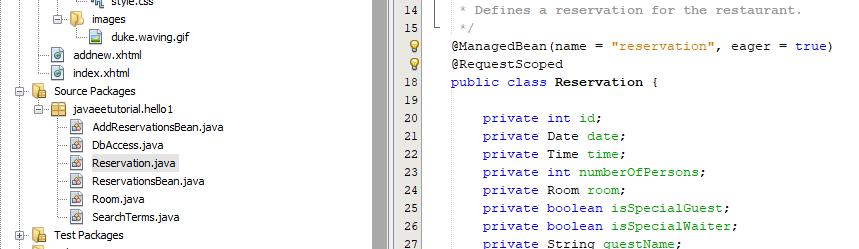


Figure Reservation.java

1. Added new Room.java class to map the Room entity

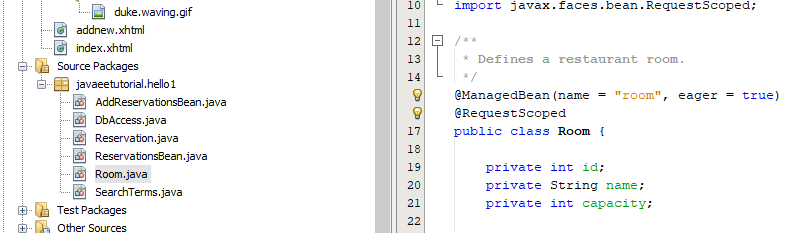


Figure Room.java

1. Added new ReservationsBean.java as backing bean for the index.xhtml page



Figure ReservationsBean.java

1. Added new AddReservationBean.java as backing bean for the addNew.xhtml page

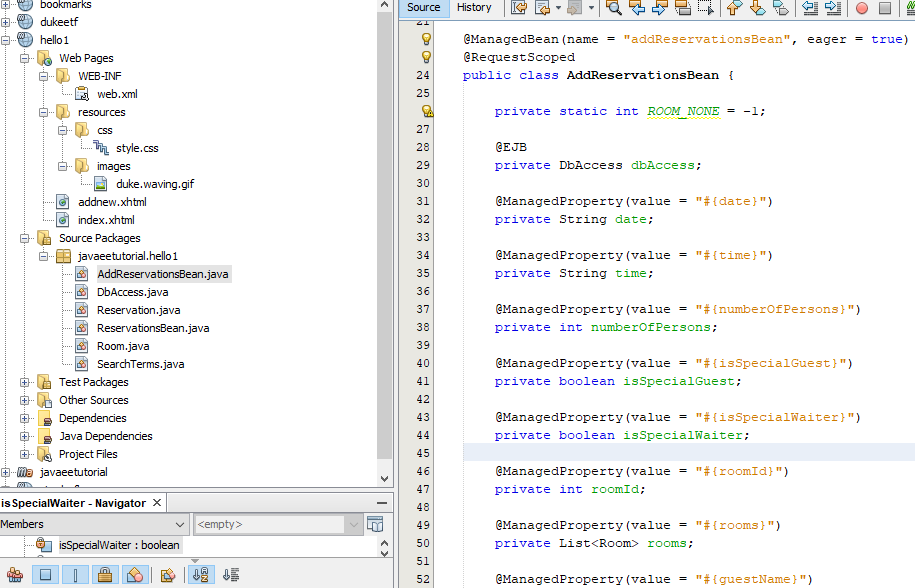


Figure AddReservationBean.java

1. Added SearchTerms.java to store information of the possible search terms

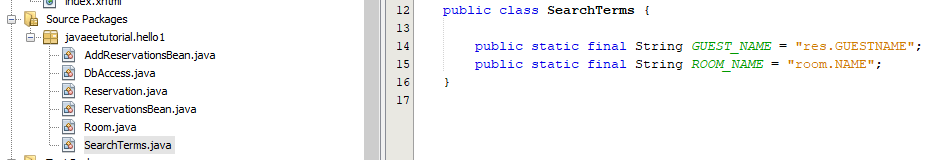


Figure SearchTerms.java

1. Added the DbAccess.java class to make the connection to the database. Here we have methods such as “get all reservations from the database”, “search reservations in the database filtering by guest name or room name”, “insert a new reservation into the database”, “Get all rooms”, etc.

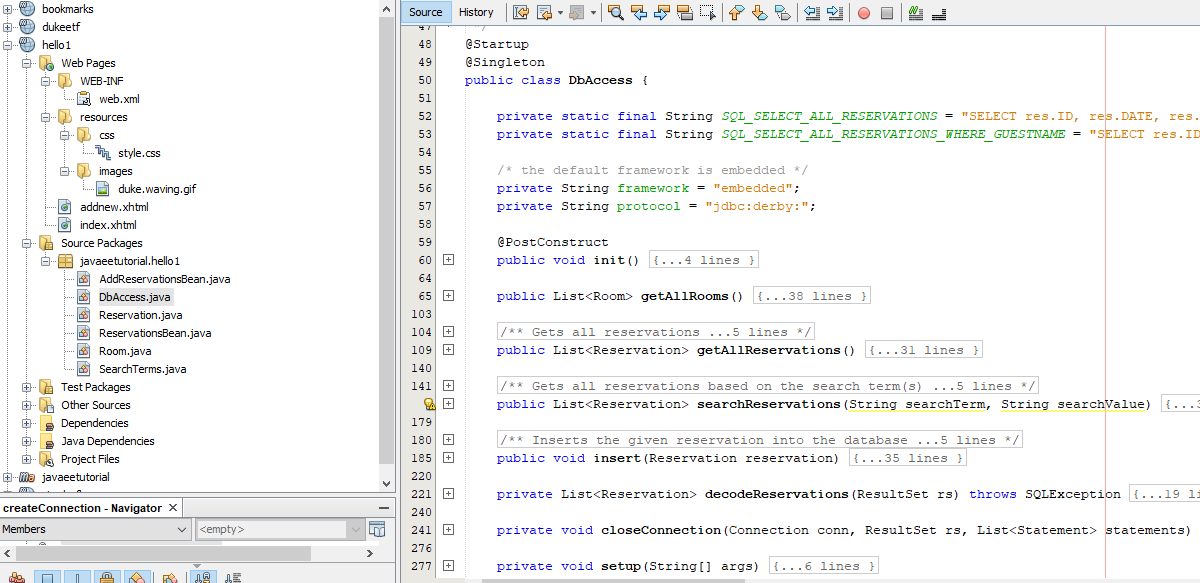


Figure DbAccess.java

# **Screen capture demonstrating the successful compilation and running of the project on your platform.**

1. Run the app

* In NetBeans, find the index.xhtml, right-click and “Run file”. This will deploy the application to GlassFish and run it.
* You can see the user interface by accessing: <http://localhost:8080/hello1/index.xhtml>
* Load this url in your browser

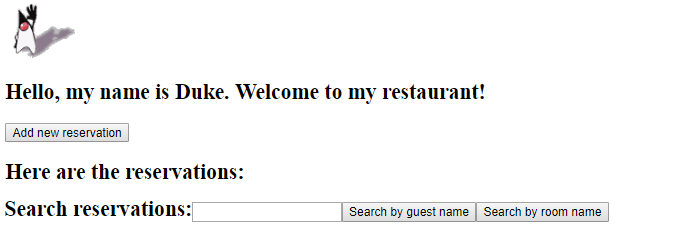


Figure 21 Initial index page

* Initially we have no reservations. So let’s add one
* Click the “Add new reservation” button
* In the new page, insert some information and click “Add reservation”

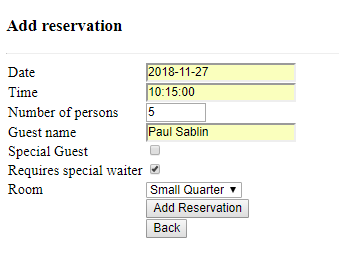


Figure 22 Add new reservation page

* The index page is loaded again and the list is refreshed

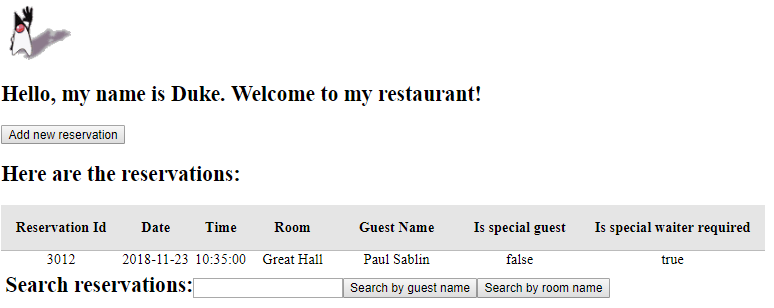


Figure Refreshed index page

* In a similar fashion, add a few more reservations. The index page might look like this



Figure Updated index page

* Write a search term in the Search reservations text box, for example “Paul” and click “Search by guest name”. The search is case sensitive.

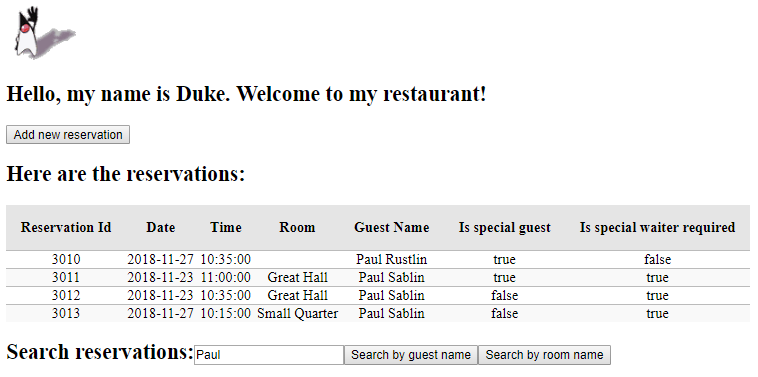


Figure Search by guest name

* Write a search term in the Search reservations text box, for example “Great” and click “Search by room name”. The search is case sensitive.

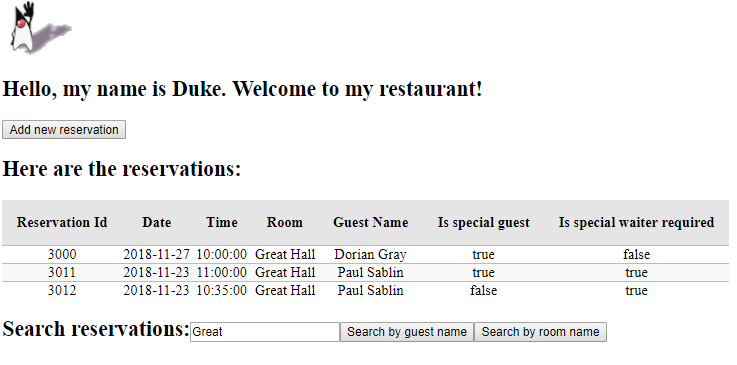


Figure Search by room name

# **Lessons Learned**

1. I learned JSF navigation
2. I learned to connect to the database with jdbc and the NetBeans services tool
3. I learned to create a data access class with several sql statements