Server-Side RAD

Shipping System

[1 Description 3](#_Toc508109181)

[2 Requirements 3](#_Toc508109182)

[3 Marks 3](#_Toc508109183)

[4 Submission 3](#_Toc508109184)

[5 Skeleton 4](#_Toc508109185)

[6 Functionality 5](#_Toc508109186)

[6.1 Main Screen 5](#_Toc508109187)

[6.2 List Ships 6](#_Toc508109188)

[6.3 Add Ship 7](#_Toc508109189)

[6.3.1 Success Conditions 7](#_Toc508109190)

[6.3.2 Invalid user input conditions 8](#_Toc508109191)

[6.4 List Shipping Companies 9](#_Toc508109192)

[6.5 Add Shipping Company 10](#_Toc508109193)

[6.5.1 Success Conditions 10](#_Toc508109194)

[6.5.2 Invalid user input conditions 11](#_Toc508109195)

[6.6 List Orders 12](#_Toc508109196)

[6.7 Create Order 13](#_Toc508109197)

[6.7.1 Success Conditions 13](#_Toc508109198)

[6.7.2 Business Logic for Order Creation: 15](#_Toc508109199)

[6.7.3 Invalid user input conditions 18](#_Toc508109200)

[6.8 Security 19](#_Toc508109201)

[6.8.1 Logout 19](#_Toc508109202)

# Description

Write a Spring Boot MVC application that allows a user to manage Ships, Shipping Companies and Orders.

# Requirements

The application should be written in Spring Boot based on the skeleton application available on Moodle and should use the Spring MVC n-tier architecture – Controllers, Services, Repositories, Views, with the appropriate data/logic in each tier.

The Model for the application consists of three objects:

* Ship
  + Ship ID
  + Name
  + Number of Passengers
  + Cost
  + Size (metres)
  + Associated Shipping Company
* Shipping Company
  + Shipping Company ID
  + Name
  + Home Port
  + Company Balance
  + Associated Ships
  + Associated Orders
* OrderInfo
  + Order ID
  + Associated Shipping Company
  + Associated Ship
  + Order Date

These models have been given in the skeleton code for the project and their dependencies already set up.

# Marks

This project is worth 50% of the marks for the module.

# Submission

A zipped file containing the source code of the entire application should be uploaded to Moodle before 5:00pm on Friday April 27th 2018.

# Skeleton

The application should be created by updating the skeleton application available on Moodle.

# Functionality

## Main Screen

When the application starts, the user should see the following screen:

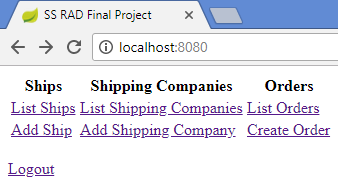


Figure Main Screen

## List Ships

This screen shows the list of Ships in the shipping database:

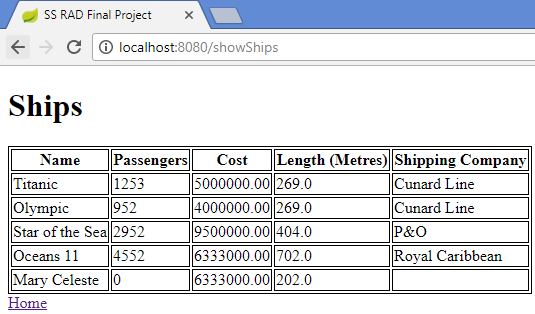


Figure List Ships

## Add Ship

This screen allows the user to add a new Ship.

### Success Conditions

If all information is entered correctly, the shipping database is updated with the new Ship information:

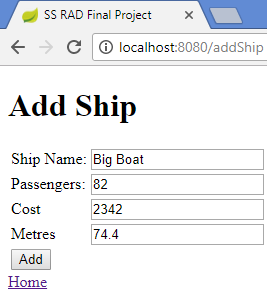


Figure Add Ship – OK

And the user is immediately brought to the List Ships page, where the Ship just entered is shown:

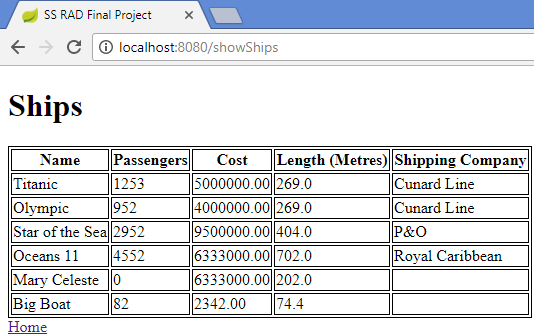


Figure List of Ships after Ship added

### Invalid user input conditions

If any of the following errors exist:

* The Ship name is blank
* The number of passengers is less than 0
* The cost is less than 1
* The cost is blank
* The size is less than 1

Appropriate error messages should be shown, and no Ship should be added to the shipping database.

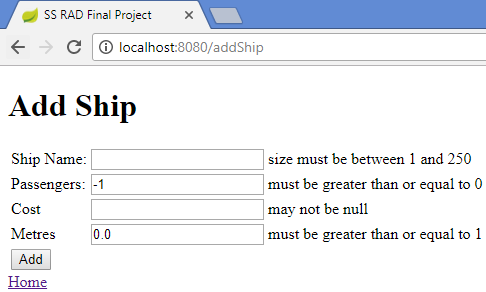


Figure Add Ship – NOK

## List Shipping Companies

This screen shows the list of Shipping Companies in the shipping database.

The **Ships** column lists the ship(s) associated with each Shipping Company.

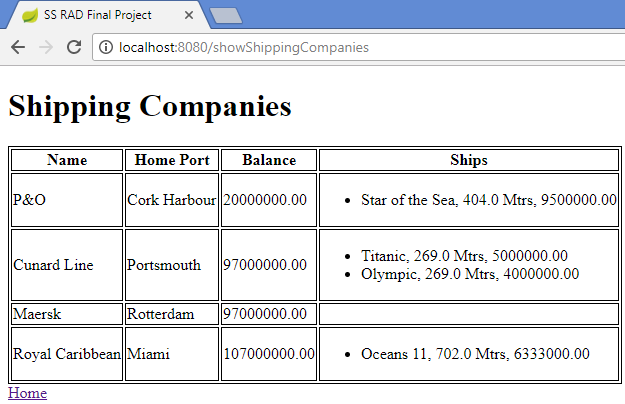


Figure List of Shipping Companies

## Add Shipping Company

This screen allows the user to add a Shipping Company.

### Success Conditions

If all information is entered correctly, the shipping database is updated with the new Shipping Company:

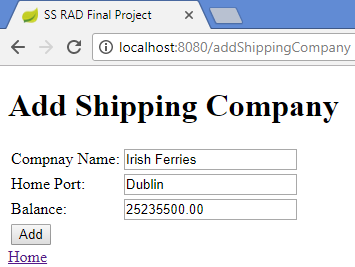


Figure Add Shipping Company – OK

And the user is immediately brought to the List Shipping Companies page, where the Shipping Company just entered is shown:

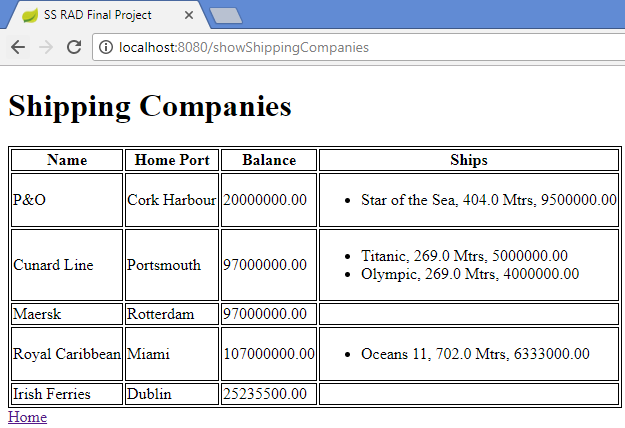


Figure List of Shipping Companies after new Shipping Company added

### Invalid user input conditions

If any of the following errors exist:

* The Shipping Company name is blank
* The Home Port is blank
* The cost is less than 1
* The cost is blank

Appropriate error messages should be shown, and no Shipping Company should be added to the shipping database.

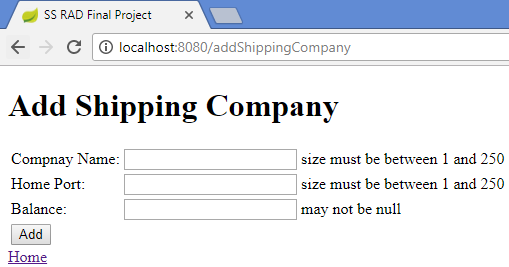


Figure Add Shipping Company – NOK

## List Orders

This screen shows the list of Orders in the shipping database, indicating the Shipping Company name and Ship name.

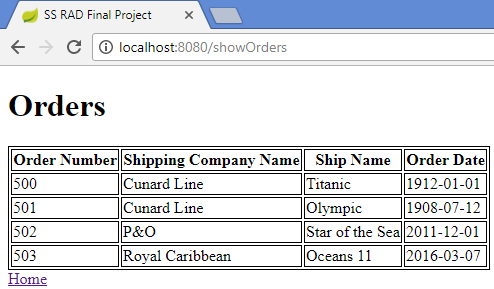


Figure List of Orders

## Create Order

This screen allows the user to create an Order.

It contains two fields:

* Ship Name – This is a drop-down list of all ships that are not currently associated with any Shipping Company. Also displayed is the cost of the ship.
* Shipping Company – This is a drop-down list of all Shipping companies. Also displayed is the Shipping company balance.

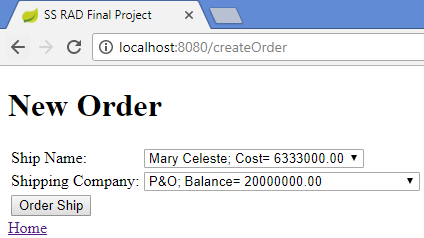


Figure New Order

### 

### Success Conditions

If both a Ship and a Shipping Company are successfully selected, an order may be placed, and if the order is valid, the user is immediately brought to the List Orders page, where the new Order is shown:

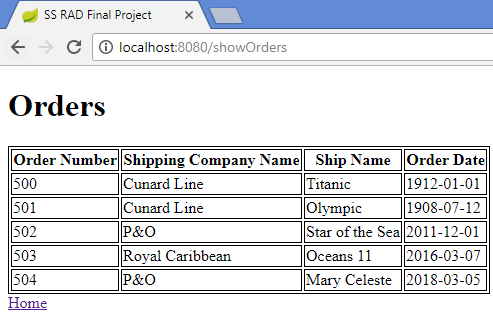


Figure List of Orders after new order added

### Business Logic for Order Creation:

* The Cost of a Ship should not be greater than the balance of the Shipping Company attempting to order a Ship, if so an error page should be shown:

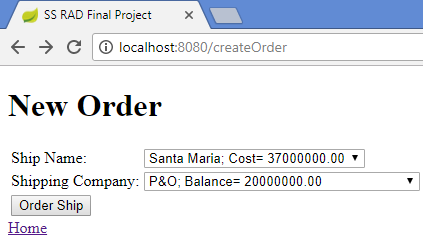


Figure Attempting to order a Ship when Cost > Shipping Company Balance

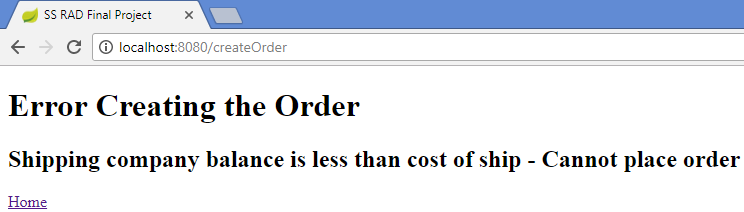


Figure Error after attempting to order a Ship when Cost > Shipping Company Balance

* If the cost of a Ship is not greater than the Shipping Company balance a new order should be created specifying the Ship ID and the Shipping Company ID.

Also, the Order date (the current date) should be automatically inserted by the application.

After successful order creation, the user is returned to the List Orders page.

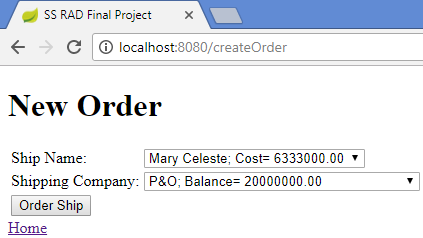


Figure Ordering a Ship where Cost <= Shipping Company Balance

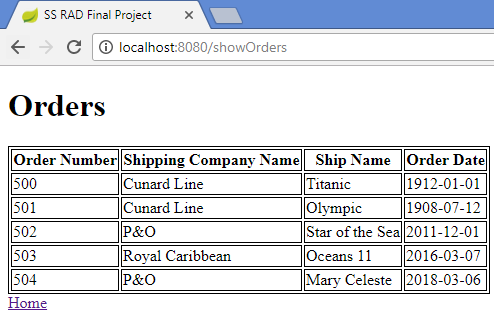


Figure List of Orders with newly created Order

* As part of creating a new order, the Ship table should be updated to reflect the fact that the Ship being ordered in now associated with the Shipping Company.

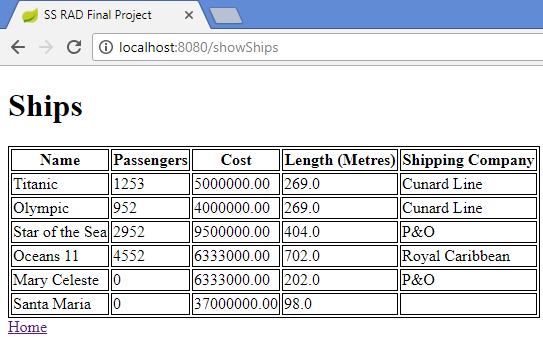


Figure Ship associated with Shipping Company

* Also, as part of creating a new order, the Shipping Company table should be updated to reflect the new balance (previous balance – cost of Ship).

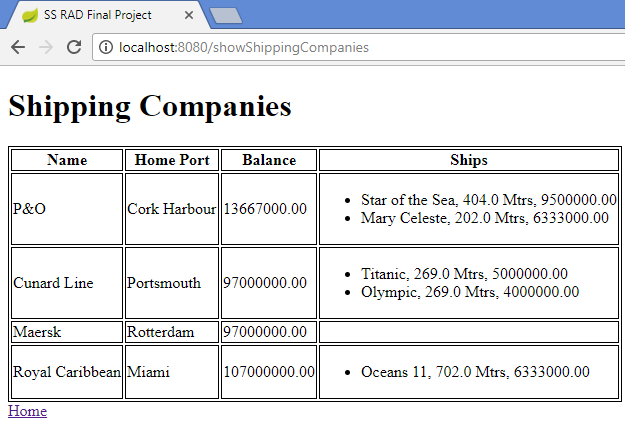


Figure Shipping Company with new Balance

### Invalid user input conditions

If no Ship and/or Shipping Company is/are selected, an error page should be displayed informing the user of this.

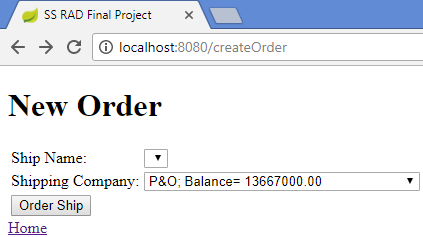


Figure Attempting to create an order without selecting a Ship

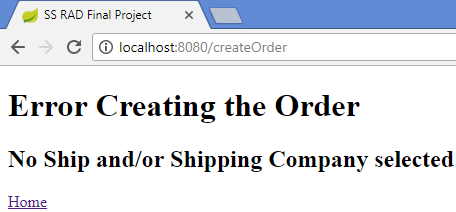


Figure Attempting to create an order without selecting a Ship - Error Page

## Security

The application should implement security on the following pages:

* Add Ship
* Add Shipping Company
* Create Order

Only logged in users should be able to access these pages.

There should be one user (user) with password user.

### Logout

There should be a logout option on the Main Screen.

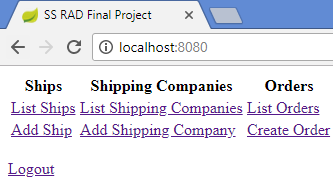


Figure Logout option on Main Screen