Sky Broadcasting Application

This document contains the design and information about the sky broadcasting application.

# Objective

Develop a JAVA service that provides an implementation of a catalogue service and a product selection web page

# Design

The application is created over Spring Boot leveraging over Maven and MySQL database. The architecture of SkyApplication can be understood by the following diagram:

## 2.1 Application Architecture Flowchart

**CustomerLocationService.java**

Calls getLocationOfCustomer ()

Returns locationID

*pass customerID*

(*from session*)

**ProductSelectionController.java**

view

success.jsp

confirm.jsp

Product.jsp

**MySQL DB**

(schema: sky\_app)

**Product**

P.K productID

F.K locationID

F.K categoryID

Stores product data

Returns list of products *locationID*

error.jsp

**CatalogueService.java**

Gets locationID

**Category**

P.K: categoryId

**Customer**

P.K customerID

F.K locationID

Stores customer data

**Location**

P.K: locationID

*SkyApplication Process Flowchart*

## 2.2 Steps of Process Flow

1. On Execution of Java Service, SkyApplication.java starts the application and the ProductSelectionController executes.
2. At first the controller looks for customerID present in session, (you can provide customer\_id parameter in URL also like? customer\_id=1, by default it is 1, and fetches the locationID from the CUSTOMER table.
3. Following this, there is CatalogueService that is called which retrieves the list of sports channels and news channels available for the location.
4. Sports Channels: on the filter of availability by is\_available column present in table PRODUCT and location\_id and category which is 1 for sports channel that can be retrieved from CATEGORY table.
5. News Channels: on the filter of availability i.e. is\_available and the category\_id.
6. After fetching the channels list, they are passed to the product.jsp and build the spring form.

The form will be displayed as follows:



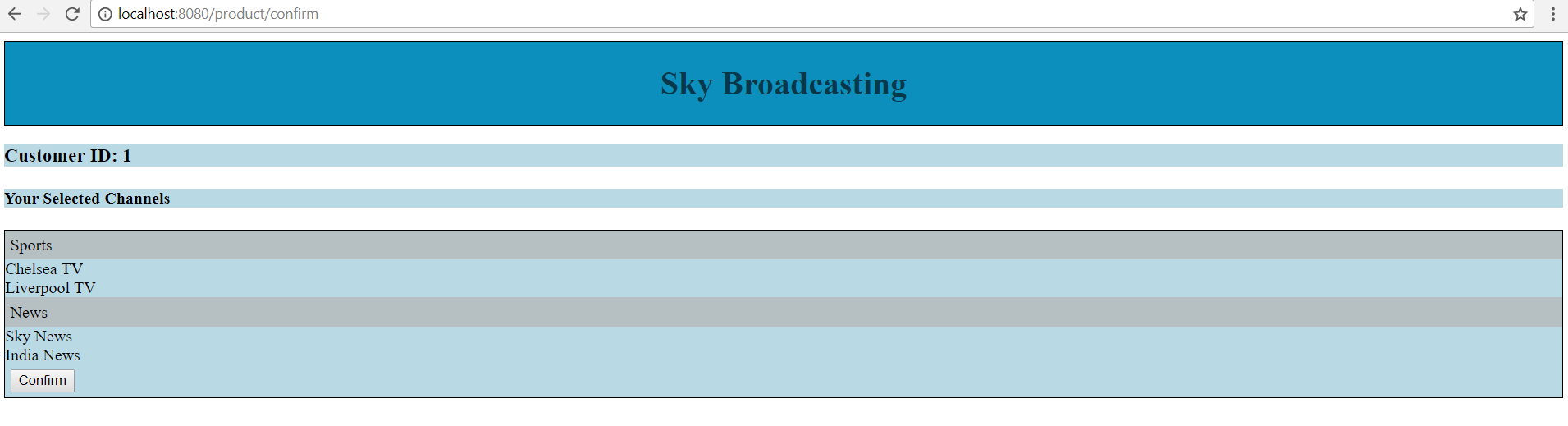
You can also provide customerID in URL as:



1. Now select/unselect the channels from [URL](localhost:8080/product) and items will be added/removed from the bucket.



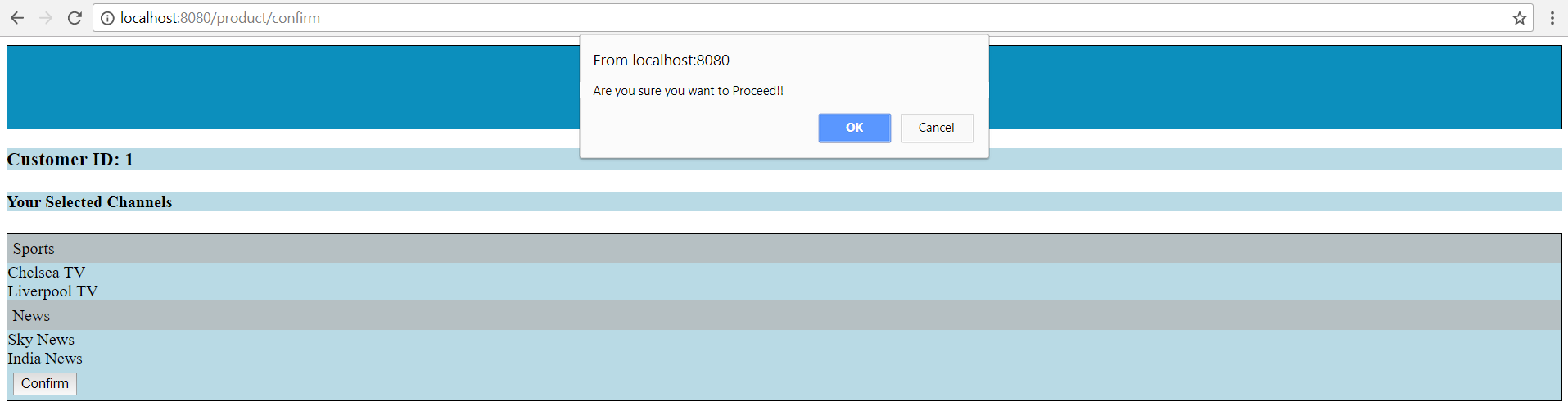
1. After adding the channels in your basket, you can click over checkout to proceed with the confirmation page.



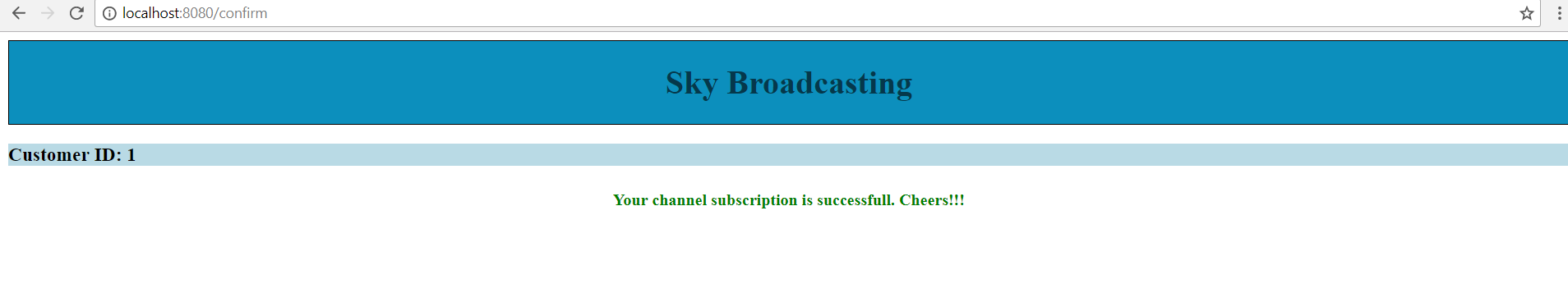
**Note: The customerID is in session and so is visible over the JSP.**

1. On clicking over confirm, you will be asked for the confirmation once again and then redirected to the status of your purchase.

Confirmation Pop-Up



Purchase Successful



## 2.3 Validation, Error Handling & Configuration

Monitoring Services should be provided to all applications with a view to reducing the time, cost and risks of development through re-use. For this, the following services within the application have been provided:

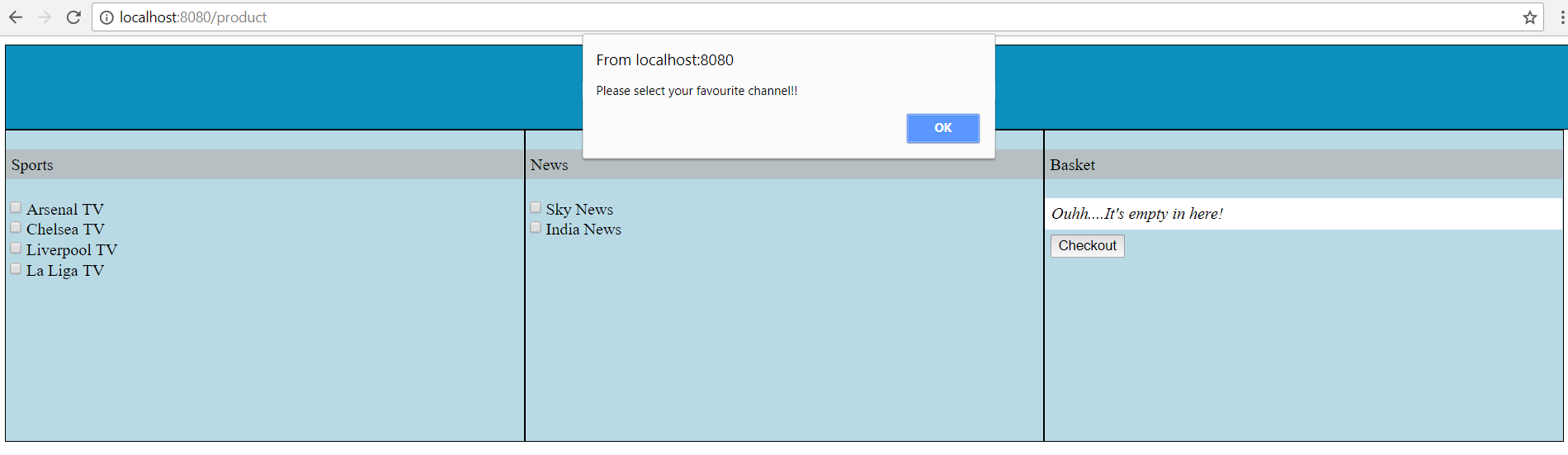
### **2.3.1 Application Configurable**

All the properties like service running frequency and the MySQL DB credentials are configurable.

* application.properties: contains DB credentials and application configurable properties.

2.3.2 Validation

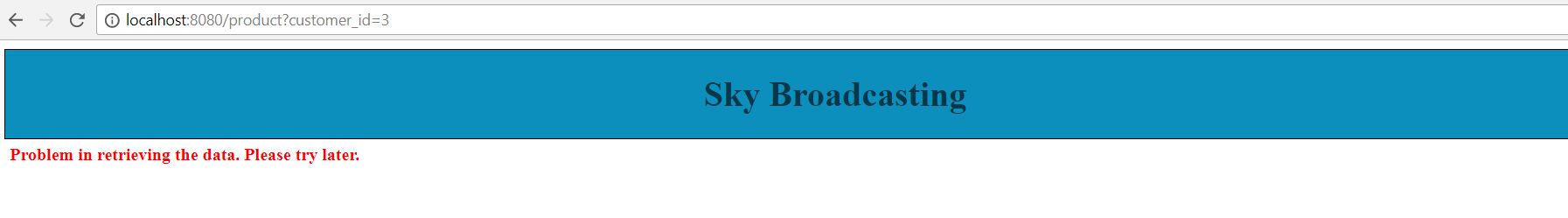
* + Validation is important for guiding the user through the application. And there is some validation check that are incorporated with in the application.
  + Over Product Selection page if user does not select any item and tries to checkout, he will receive an alert message to choose some item.



### Error Handling

There has been error handling and error logging with in the application.

In case of any error throughout the application, user will be redirected towards the error page.



## 2.4 Software development tools

The Software Development Tools used with in the application are:

1. Eclipse IDE
2. MySQL Database
3. Reference of spring.start.io to build the maven Project with Spring boot.