Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Project Final Report

Project Overview

Group 6 is building a web application that focuses on business-to-customer (B2C) e-commerce and applies the typical scenario of a small retail store seeking to create a website enabling customers to shop online. The software consists of two components – a storefront and an administration console. The store front is accessed by customers and this is their avenue of purchasing products over the internet. The administration console is a management console that is restricted to a select few users from the business who are responsible for creating, updating and deleting products, discounts, shipping information and other information relevant to day to day operations of the business. Building this simple yet effective e-commerce application would provide the team an opportunity to explore and demonstrate the various important features of Java web and EE development and familiarize themselves with working on the NetBeans IDE.

Project Goals and Scope

Through this project the affable bean (customer) is looking to reach out its customers who would prefer ordering goods through the internet. It is looking to establish an alternate channel of sales to supplement existing sales. A recent survey has indicated that 90% of the affable beans regular clientele has continuous Internet access, and 65% percent would be interested in using this service. The primary goal of the organization is to create a website that will enable their customers to shop online. Group 6 goals are to build an end-to-end 3 tier e-commerce application that would satisfy the needs of the affable bean.

A brief scope is outlined below:

1. An online representation of the products that are sold in the physical store.

2. There are four categories
    - Dairy,
    - Meats,
    - Bakery,
    - Fruits and Vegetables,

3. There are four products for each category, which online shoppers can browse.
    - Details are provided for each product (i.e., name, image, description, and price).

4. Shopping cart functionality, which includes the ability to:
    - Add items to a virtual shopping cart.
    - Remove items from the shopping cart.
    - Update item quantities in the shopping cart.
    - View a summary of all items and quantities in the shopping cart.
    - Place an order and make payment through a secure checkout process.

5. An administration console, enabling staff to view customer orders.

6. Security, in the form of protecting sensitive customer data while it is transferred over the Internet, and preventing unauthorized access to the administration console.

7. Language support for both English and Czech. (Website only)

Stakeholders :

The various stakeholders identified for this project are as follows:

- Affable bean,
- Shoppers and other end customers of the affable bean,
- Group 6,
- Ingrid and Apurva.

The affable bean is the customer who would utilize this e-commerce application to enhance their business experience. The shoppers and end consumers who buy from the affable bean are our end user as they would buy products from the affable bean via our web application.

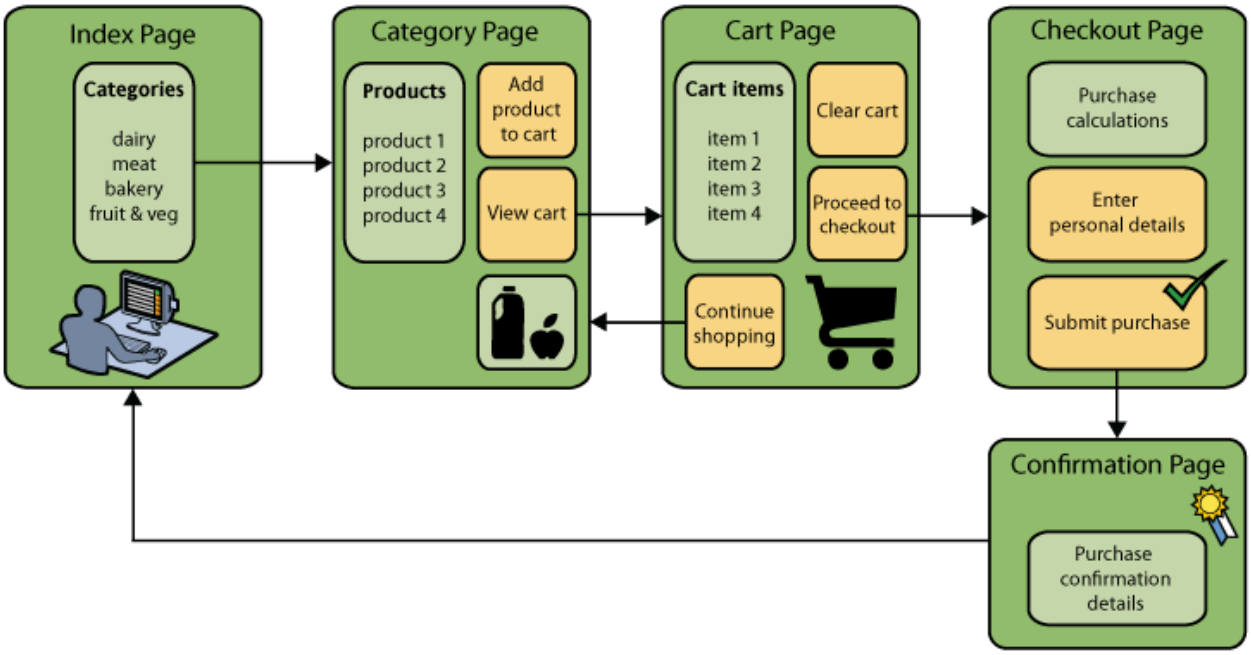
Group 6 team members are crucial stakeholders as they would like to build a quality web application that satisfies customer needs and meets the requirement for the Software Development course curriculum.

Prof.Ingrid and TA Apurva are stakeholders as they would like to ensure that the team learnt the various technologies involved and could build a successful project.

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Business Process Flow and Overall Use-Case

Customer visits the welcome page and selects a product category. Customer browses products within the selected category page, then adds a product to his or her shopping cart. Customer continues shopping and selects a different category. Customer adds several products from this category to shopping cart. Customer selects 'view cart' option and updates quantities for cart products in the cart page. Customer verifies shopping cart contents and proceeds to checkout. In the checkout page, customer views the cost of the order and other information, fills in personal data, then submits his or her details. The order is processed and customer is taken to a confirmation page. The confirmation page provides a unique reference number for tracking the customer order, as well as a summary of the order.



Functional Requirements:

Online Catalog – Homepage

Description and Priority

The homepage is the entry point for the application. It introduces the business and the service to the customers and allows the users to shop through this online representation of the product categories that are available in the physical store. There are four categories and they are Dairy, Meats, Bakery, Fruits and Vegetables. The user can browse through the products available in a category by clicking on the category image in this catalog page. This feature is of high priority and the user cannot browse products to shop without navigating through the category. Not able to navigate to the product page is high risk as it hampers the completion of the base use case of shopping for products.

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Stimulus/Response Sequences

Typing in the web address of the affable bean brings the user to this page. The user can select a category by clicking on the image of the category or the name of the category in the catalog section. The user is navigated to the product page once the category is selected.

Functional Requirements

REQ-1: Load Category name and relevant category images in the category catalog
REQ-2: Allows the user to browse through categories in the catalog
REQ-3: Navigate to the product catalog for the selected category
REQ-4: Clicking on the Logo image should reload the homepage.
REQ-5: Language support for both English and Czech.

User Interface Specifications:

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Online Catalog – Category View

Description and Priority

The category page populates the listing of products available in the store that belong the selected category. The user can view all product information in this page and can add items into their shopping cart. The user can also navigate to another category and view the listed products in that category. The use can shop products from multiple categories in one session. This feature is also of high priority and the user cannot shop products without being able to browse through and add to cart. As in the case of the homepage, not able to navigate to the product page and adding to cart is of high risk as it hampers the completion of the base use case of shopping for products.

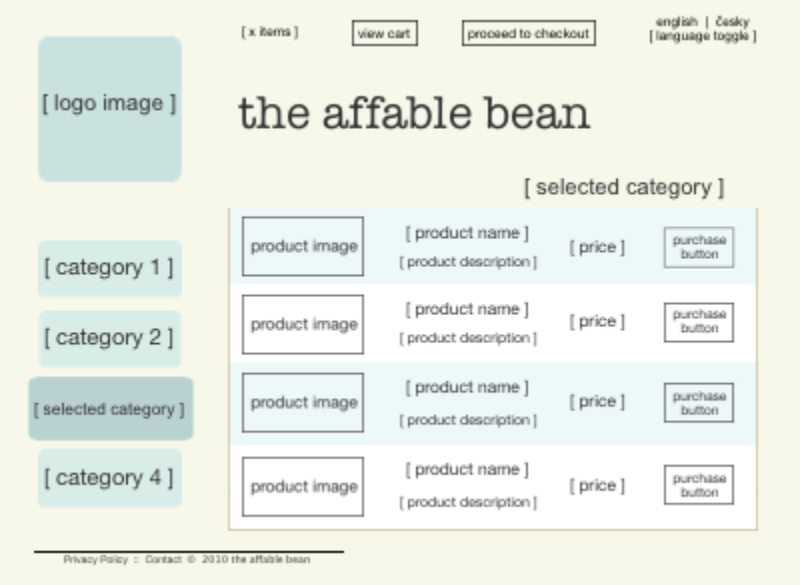
Stimulus/Response Sequences

Clicking on a category in the homepage navigates the user to this page. Selecting a category from the category tab, loads the relevant product listing. Clicking on the “add to cart” button adds the selected product to the cart and enables the view cart and proceed to check out buttons. The user can continue navigating through categories and adding products to the cart or can proceed to view cart or checkout.

Functional Requirements

REQ-1: List product information and relevant information of the selected category
REQ-2: Add selected product to cart
REQ-3: Enable view cart and proceed to check out when cart isn’t empty
REQ-4: Clicking on the Logo image should reload the homepage.
REQ-5: Language support for both English and Czech.

User Interface Specifications:



Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Cart Page

Description and Priority

The cart page lists all items held in the user's shopping cart. It displays product details for each item, and tallies the subtotal for the items in the cart. The user could clear all items in the cart which would disable the view cart and proceed checkout button, update the quantity of items in the cart or remove a product by setting quantity to 0, continue shopping by returning to the catalog or proceed to checkout. The cart is also a feature of high priority as it supports the base case.

Stimulus/Response Sequences

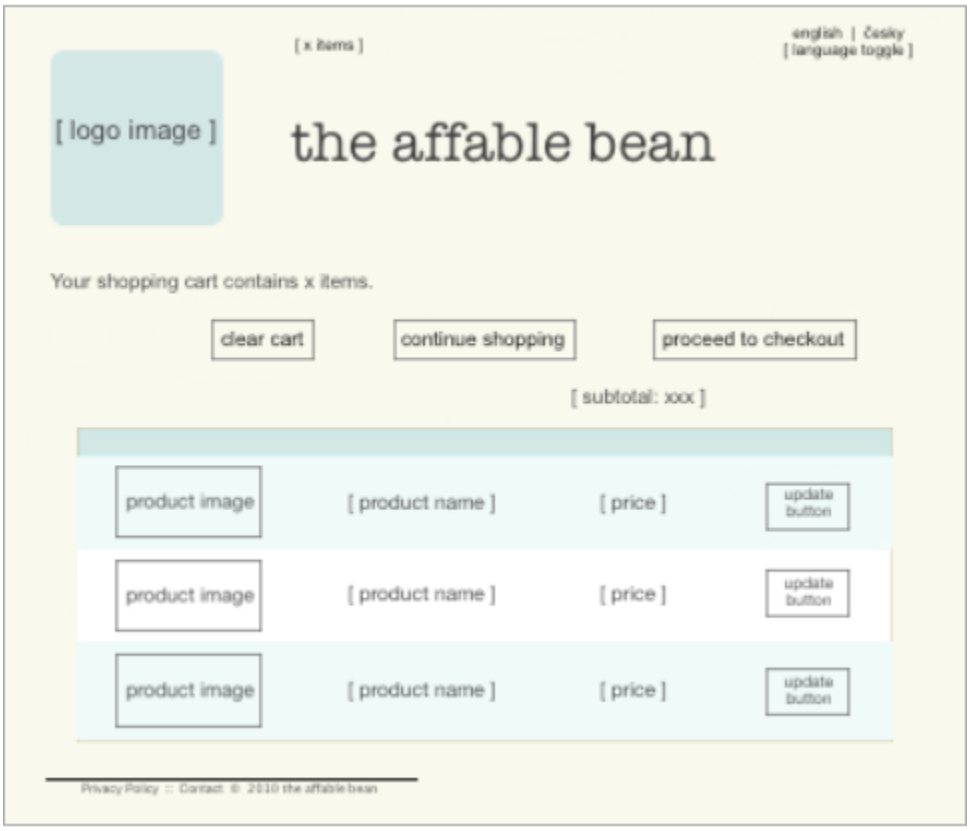
- Clicking on the view cart button from the category view navigates the user to this page.
- Clicking clear cart, removes all items in the cart. The 'proceed to checkout' buttons and shopping cart table to disappear.
- On updating the quantity for any listed item, the price and quantity are updated and the subtotal is recalculated. If user sets quantity to '0', the product table row is removed.
- Clicking on 'continue shopping' navigates the user back to the previous category that they were in before they viewed the cart.
- Clicking on ‘Proceed to checkout’ navigates the user to the checkout page

Functional Requirements

REQ-1: Populate cart with added items and prices of items
REQ-2: Allow quantity updates and product removal
REQ-3: recalculate subtotal on edits to cart
REQ-4: Navigate the user back to previous category on click on ‘continue shopping’ button
REQ-5: Navigate the user to the checkout page on click of ‘proceed to checkout’
REQ-6: Clicking on the Logo image should reload the homepage.
REQ-7: Language support for both English and Czech.

User Interface Specifications:

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu



Checkout Page

Description and Priority:

The checkout page collects information from the customer using a form. This page also displays purchase conditions, and summarizes the order by providing calculations for the total cost.The user is able to send personal details over a secure channel.

Stimulus/Response Sequences:

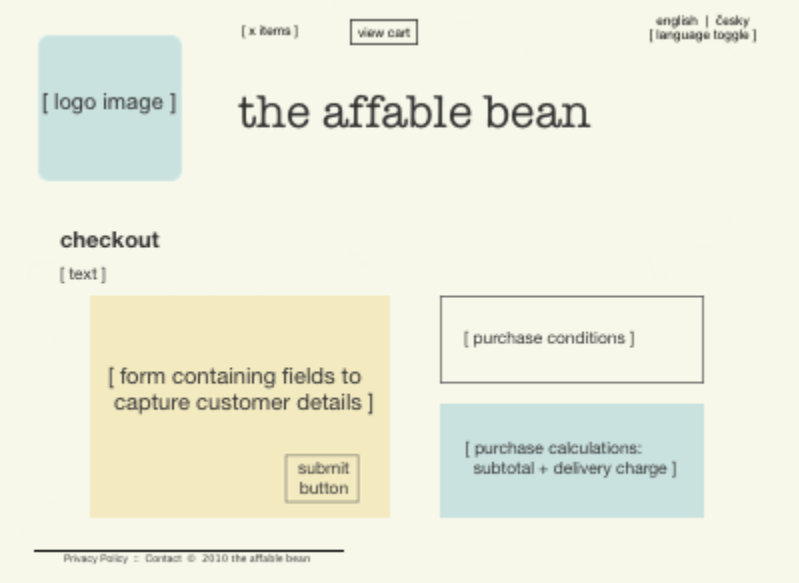
Clicking on the ‘procced to checkout’ button from the cart page or from the category page navigates the user to this page. The user enters his or her details to complete the order. Details such as name, address, phone number and credit card details are collected here.

Functional Requirements:

REQ-1: Gather customer details in a form
REQ-2: Calculate subtotal and delivery charge and display the same
REQ-3: Validate field information such as credit card number
REQ-4: Allow user to submit the purchase

REQ-5: Clicking on the Logo image should reload the homepage.
REQ-6: Language support for both English and Czech.

User Interface Specifications:



Confirmation Page

Description and Priority:

The confirmation page returns a message to the customer confirming that the order was successfully recorded. An order reference number is provided to the customer, as well as a summary listing order details. Order summary and customer personal details are returned over a secure channel.

Stimulus/Response Sequences:

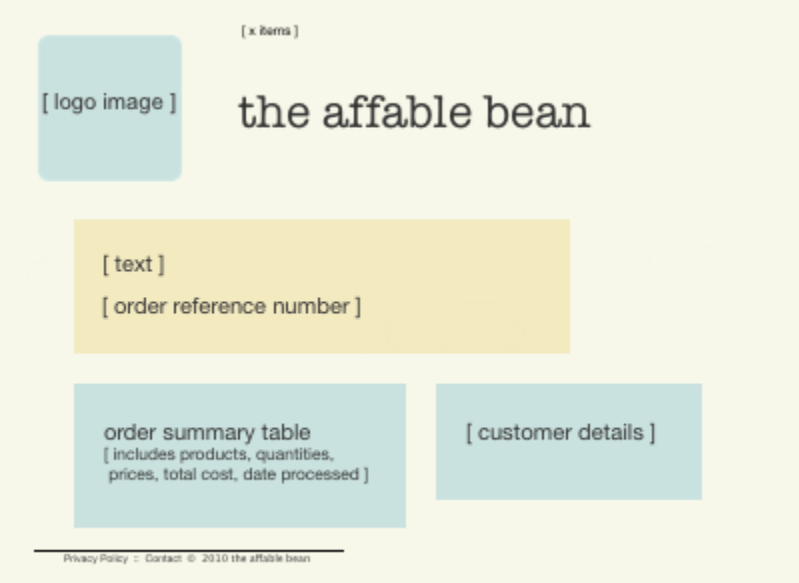
Clicking on the confirm order button in the checkout page navigates the user to this page.This page is just a view and has not task associated but clicking on the image icon to navigate to the homepage.

Functional Requirements:

REQ-1: Display order summary
REQ-2: Display customer details that were gathered in the form (previous page)
REQ-3: Clicking on the Logo image should reload the homepage.

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

User Interface Specifications:



General Business Rules:

These general rules apply to multiple pages:

- The user is able to proceed to checkout from any page, provided that:
    - The shopping cart is not empty
    - The user is not already on the checkout page
    - The user has not already checked out (i.e., is on the confirmation page)
- From all pages, the user is able to:
    - View the status of his or her shopping cart (if it is not empty)
    - Return to the welcome page by clicking the logo image
- The user is able to select the language (English or Czech) to view the page in for all pages except the confirmation page.

Project Management:

The project organization:

For the course of this project the team is organized as outlined below:
Management Team – Ingrid and Apurva
Project Owner – Jai Wadhwani
Technical Team – Mithun Nallu - UI, Front end developer, Testing
Nachiket Barve – Backend Developer, Testing
Jai Wadhwani – Infrastructure and Testing

The Agile methodology will be utilized. Group 6 plans to adopt the Unified Software Development Process or Unified Process for developing this product. The unified process is a popular iterative and incremental software development process framework and consists of 4 major phases which are
Inception,
Elaboration,
Construction and
Transition.
The technologies utilized are

- HTML, CSS, and JavaScript technologies
- Servlet and JavaServer Pages (JSP) technologies
- Enterprise JavaBeans (EJB) technology
- Java Persistence API (JPA)
- The JavaServer Pages Standard Tag Library (JSTL)
- Java Database Connectivity (JDBC)

The Development tools that will be utilized are as follows:

- NetBeans IDE
- GlassFish, a Java EE application server
- MySQL, a relational database management server (RDBMS)
- MySQL WorkBench, a visual database design tool

The Deliverables:

i. The final product

The final product that we plan to deliver is a Netbeans project that contains all the source code required to run our 3 tier e-commerce application

ii. The product artifacts during each cycle of the development

As part of the Requirements gathering phase, we plan to deliver the Use cases as one artifact. Using these requirements we plan to deliver some mockups as to how our User interfaces will look like. Once the user expectations are decided, we will do the high level design where we will come up with the class diagram and sequence diagrams for some complex use cases. Using these specifications we plan to develop this project using Netbeans, Java JDK 1.6, GlassFish server version 3 and MySql database server version 5.1, in the Construction phase.

During the Construction phase, we will also test our application and will deliver the test cases we used to test our application. During the transition phase, we will deliver our source code, and a software configuration document and instructions on deploying the application.

The Project Schedule:

As discussed in the Project management section, we plan to use Unified process for developing this product. It has 4 major phases Inception, Elaboration, Construction and Transition. The tentative project schedule is given below

Inception (11/12/2015 - 11/16/2015)

- Understand the project requirements
- Understand the Project stakeholders

Elaboration (11/16/2015 - 11/20/2015)

- Finalize the project requirements
- Prepare and deliver the Use case document
- Prepare and deliver the mock ups to be used for development
- Design the data model by understanding the use case document

Construction (11/20/2015 - 12/05/2015)
Iteration 1 (11/20/2015 - 11/25/2015)

- Design the class diagram for the whole application
- Prepare the Page views and Controller servlet
- Connect the application to the database
- Add Entity classes and Session beans to the application

Iteration 2 (11/30/2015 - 12/05/2015)

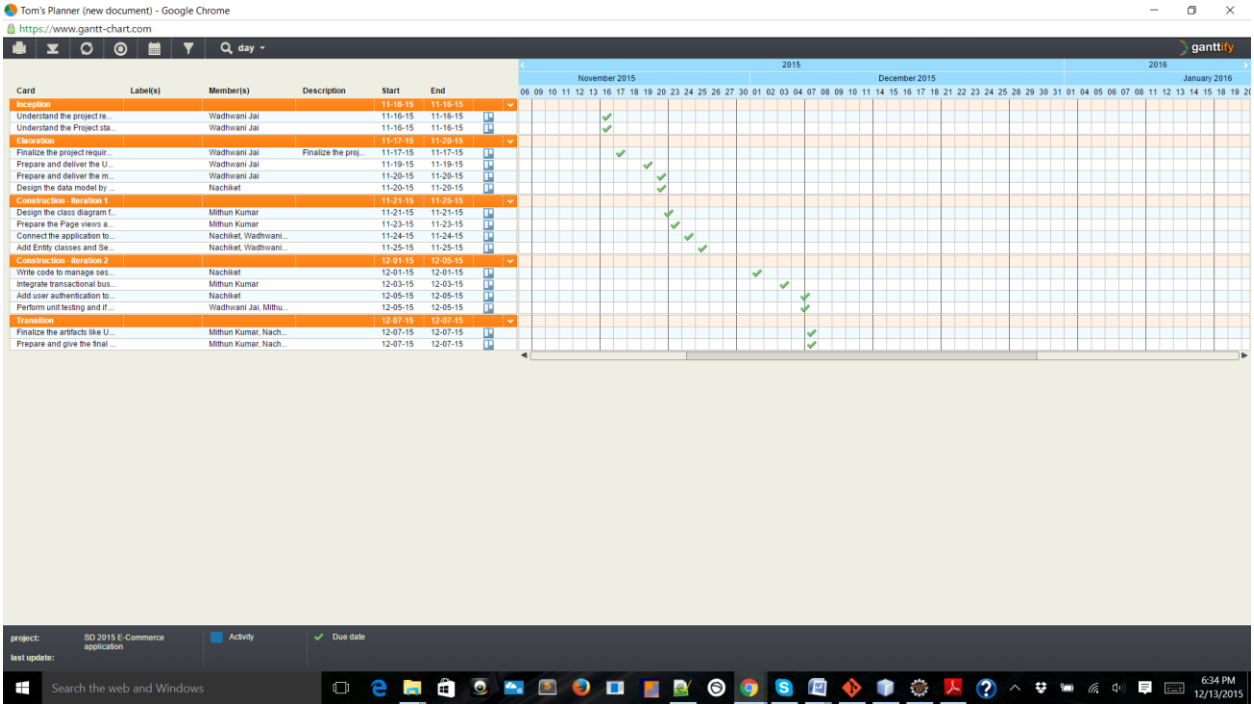
- Write code to manage sessions
- Integrate transactional business logic to the application
- Add user authentication to the system
- Perform unit testing and if possible write automated test cases

Transition (12/05/2015 - 12/07/2015)

- Finalize the artifacts like UML diagrams, source code, instructions to install the project, etc. to be submitted for the project
- Prepare and give the final demo on Dec 7 2015

Also, please find the Gantt chart of this schedule created using Trello and gantt-chart.com below

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu



Risk Management

The two primary concerns that need to be addressed while considering risk management are :

- Preventing unauthorized users from gaining access to protected content.
- Preventing protected content from being read while it is being transmitted.

The first concern, *access control*, is typically a two-step process that involves (1) determining whether a user is who he or she claims to be (i.e., *authentication*), and then (2) either granting or denying the user access to the requested resource (i.e.,*authorization*). A simple and common way to implement access control for web applications is with a login form that enables the server to compare user credentials with a pre-existing list of authenticated users.

The second concern, protecting data while it is in transit, typically involves using Transport Layer Security (TLS), or its predecessor, Secure Sockets Layer (SSL), in order to encrypt any data communicated between the client and server.

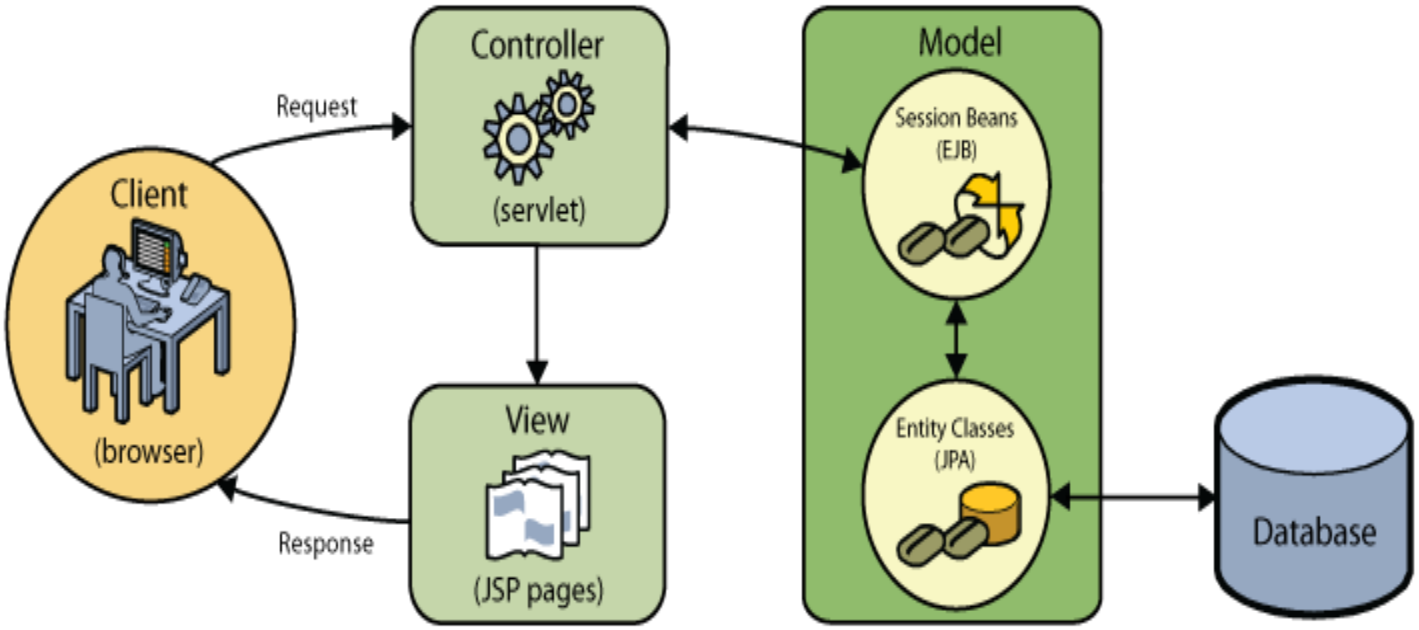
We are planning to secure the application in the following ways:

- Set up a login form for the administration console that enables staff members access to the console's services, and blocks unauthorized users.
- Configure secure data transport for both the customer checkout process, and for any data transmitted to and from the administration console.

In order to implement the above, we'll take advantage of NetBeans' visual editor for the web.xml deployment descriptor.

NetBeans visual editor enables you to view all customers and orders contained in the database. When you click either of the links in the left panel, the page will update to display a table listing customers or orders, depending on your choice

High level system architecture design:



The application uses the MVC (model-view-controller) pattern as its high level architecture design. This pattern divides the application into three interoperable components:

- **Model:** Represents the business data and any business logic that govern access to and modification of the data. The model notifies views when it changes and lets the view query the model about its state. It also lets the controller access application functionality encapsulated by the model.

- **View:** The view renders the contents of a model. It gets data from the model and specifies how that data should be presented. It updates data presentation when the model changes. A view also forwards user input to a controller.

- **Controller:** The controller defines application behavior. It dispatches user requests and selects views for presentation. It interprets user inputs and maps them into actions to be performed by the model. In a web application, user inputs are HTTP GET and POST requests. A controller selects the next view to display based on the user interactions and the outcome of the model operations.

Adhering to the MVC design pattern provides you with numerous benefits:

- **Separation of design concerns:** Because of the decoupling of presentation, control, and data persistence and behavior, the application becomes more flexible; modifications to one component have minimal impact on other components. You can, for example, create new views without needing to rewrite the model.
- **More easily maintainable and extensible:** Good structure can reduce code complexity. As such, code duplication is minimized.
- **Promotes division of labor:** Developers with different skill sets are able to focus on their core skills and collaborate through clearly defined interfaces.

Technologies Used:

Web technologies: HTML, CSS, JavaScript, JSP, EJB, JPA, JSTL, Glassfish server

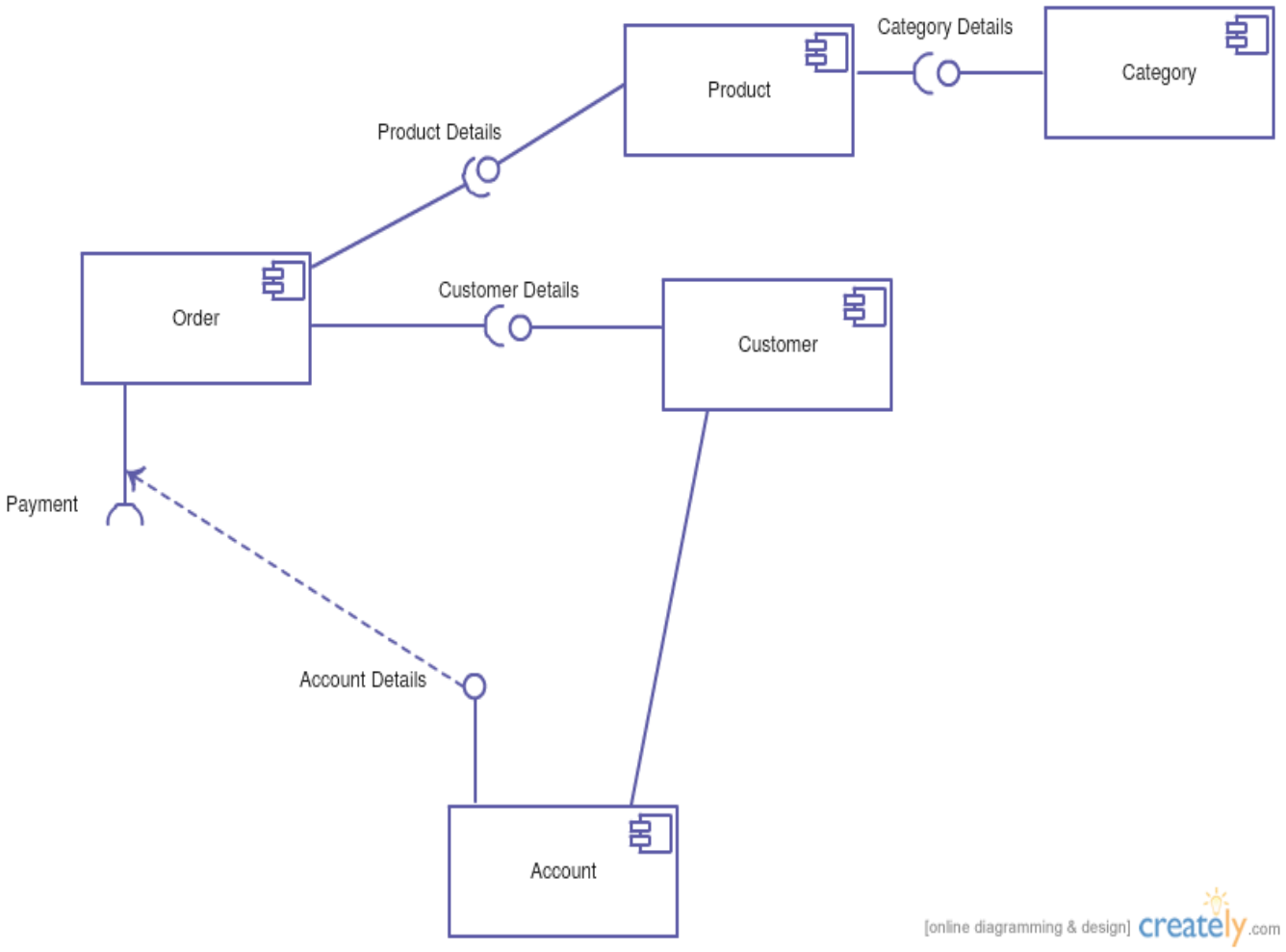
Database: MySQL 5.1, MySQL Workbench

Programming: Java 8, J2EE

IDE: NetBeans

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Component Diagram:



The component diagram above shows the various physical aspects, the organization and relationships among components of the system. It identifies five different components and their relationships:

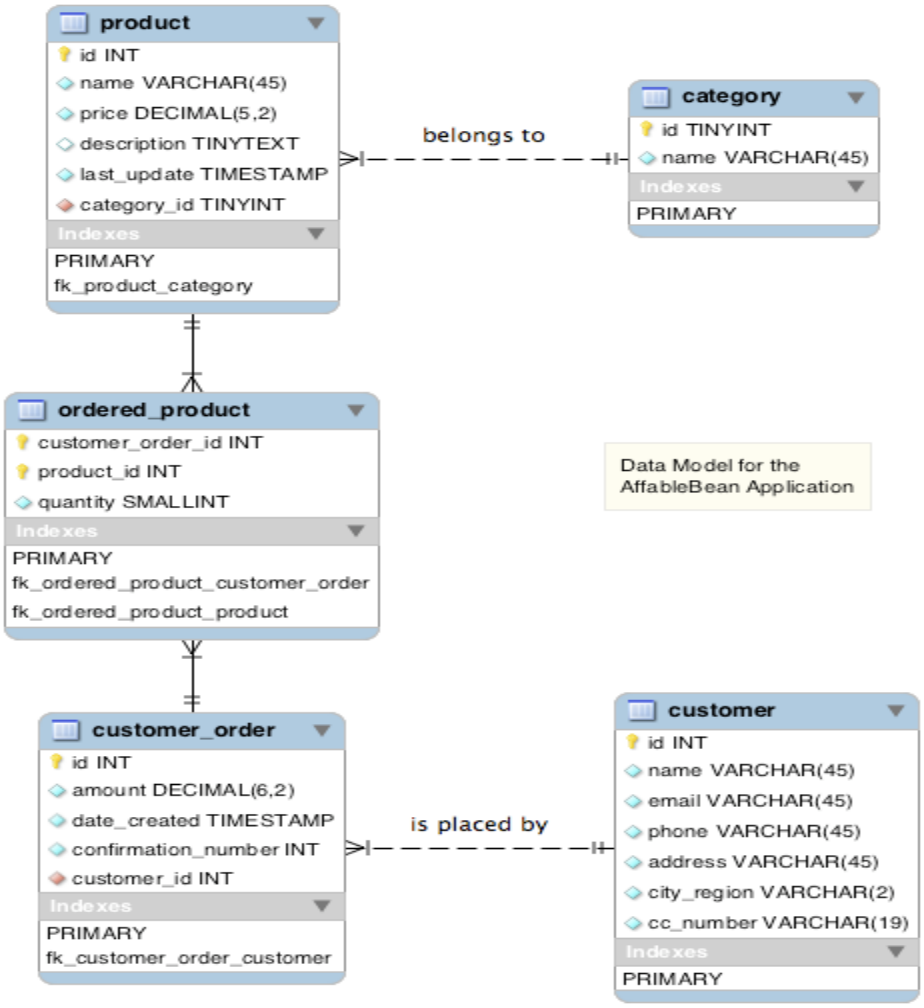
- Customer: This components stores all the information related to the customer such as customer_id, name, email, phone, address, etc.

- Account: This component contains information related to a customer_order such as date_created, confirmation_number, amount, etc.
- Order: The Order component is responsible to handle the different order details of a customer such as customer_order_id, quantity of products ordered, product_id, etc.

- Product: This component is required to carry out operations related to a particular product to be sold on the website. It has the name of the product, its description, which category it belongs to, etc.

- Category: This component groups similar kind of products into a particular category and holds all relevant information related to that category.

Data Model Diagram:



The Data Model diagram consists of four different entities:

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Category

Column	Datatype	PK	NN	UN	AI
id	TINYINT	✓	✓	✓	✓
name	VARCHAR(45)		✓		

customer_order

Column	Datatype	PK	NN	UN	AI	Default
id	INT	✓	✓	✓	✓	
amount	DECIMAL(6,2)		✓			
date_created	TIMESTAMP		✓			CURRENT_TIMESTAMP
confirmation_number	INT		✓	✓		

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Product

Column	Datatype	PK	NN	UN	AI	Default
id	INT	✓	✓	✓	✓	
name	VARCHAR(45)		✓			
price	DECIMAL(5,2)		✓			
description	TINYTEXT					
last_update	TIMESTAMP		✓			CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

Based on the Data Model diagram it can be deduced that the application has the following one-to-many relationships:

- A category must contain one or more products
- A customer must have placed one or more orders

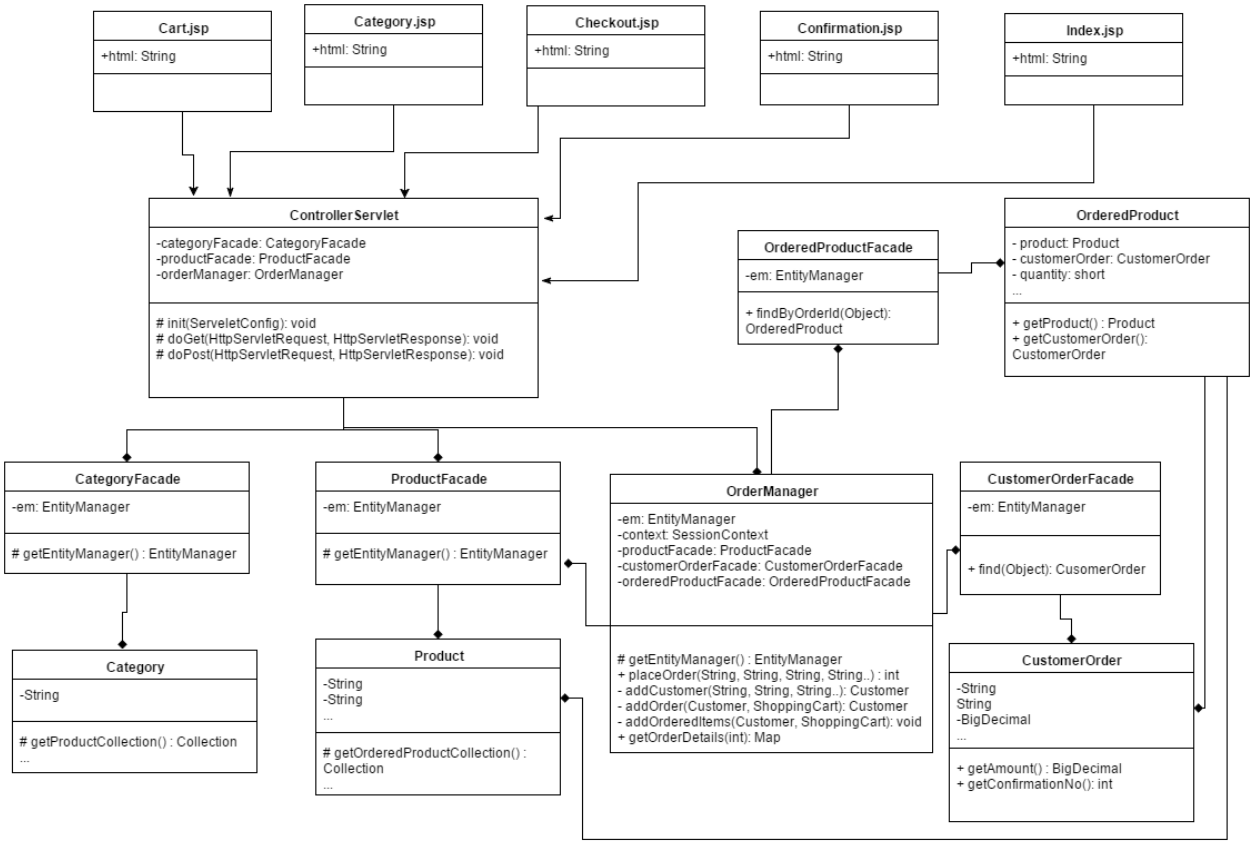
And the following many-to-many relationship:

- A category contains multiple products, and a product can belong to multiple categories

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Class Diagram:

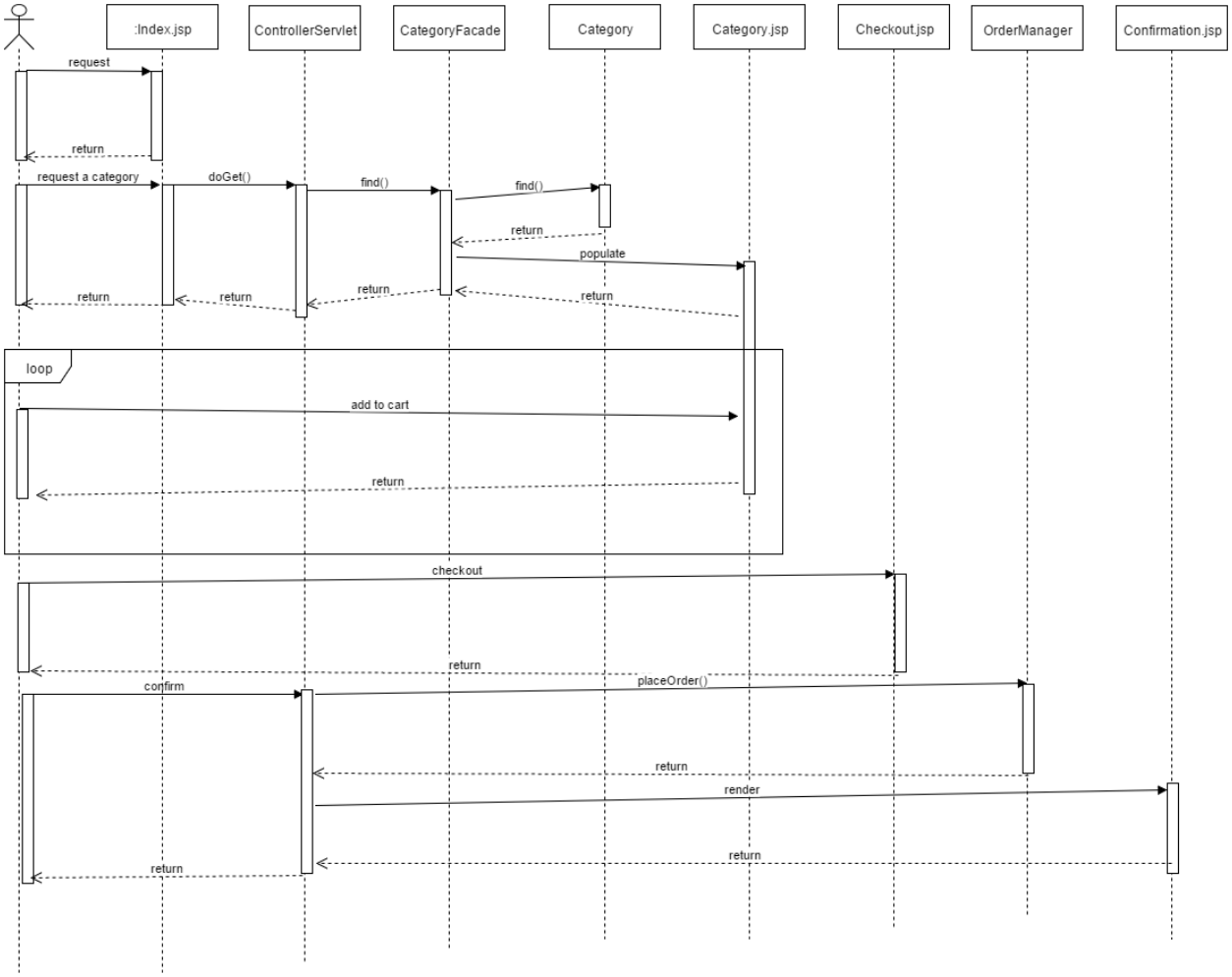
Our application uses MVC design pattern and the below class diagram tries to represent various classes in our project and the associations between them.



Sequence Diagram:

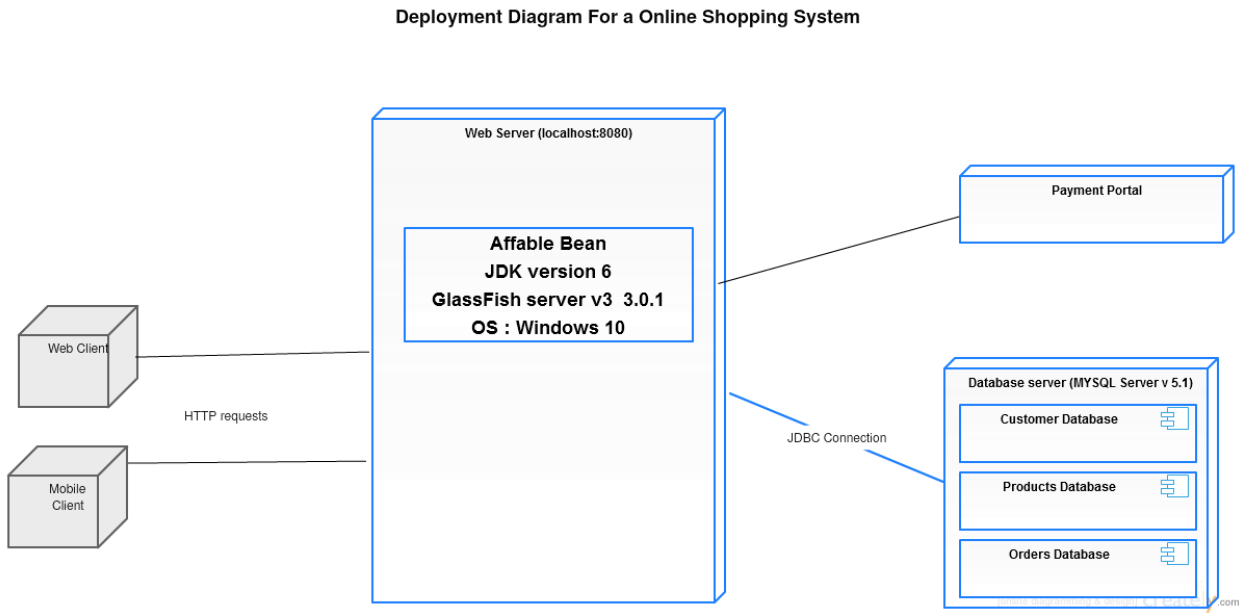
The below diagram covers the sequence of interactions in a use case, where user loads the category page, adds products to cart and checks out the cart in the end.

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu



Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Deployment Diagram:



The deployment diagram shows how our application will be deployed on the server's ready-to-use for clients. The client and server both are deployed on the same machine. The Affablebean application runs on JDK version 6, Glassfish server v3.0.1 and Operating System Windows 10. The server is connected to the MySQL server v5.1 through JDBC.

Quality Assurance:

Before delivering any application, it is important to ensure that it functions properly, and that it can perform acceptably for the demands expected of it. Web applications, especially e-commerce applications, by their very nature provide concurrent access to shared resources. In other words, the servers on which they are hosted must be able to respond to multiple users requesting the same resources over the same period of time. The developer should be mindful of this fact when during development your application appears to behave correctly as you click through web pages in your browser. The following points should be taken in consideration to build a good quality product:

1.) How will the application perform when handling multiple users simultaneously?
2.) Are there memory leaks that will degrade the server's performance after the application has been running for long periods of time?

3.) What steps can you take to ensure that your production server best handles traffic to your application?

The tool that we will be using for testing purposes is **JMeter**. It is a plugin which can be used in the Netbeans environment. Some of the important features that JMeter provides are :

- Allows the user to create test plan
- Different types of testing methods can be performed such as Load testing, stress testing
- Monitoring the performance of the application using Netbeans profiler and check for potential bugs
- Allows the user to check for any memory leaks, CPU usage, evaluate and analyze the Heap contents
- Analyze and view the performance of the application and which bug is degrading the performance

To create a test plan in JMeter the following use-case was created which mocks a list of user-initiated requests to the server:

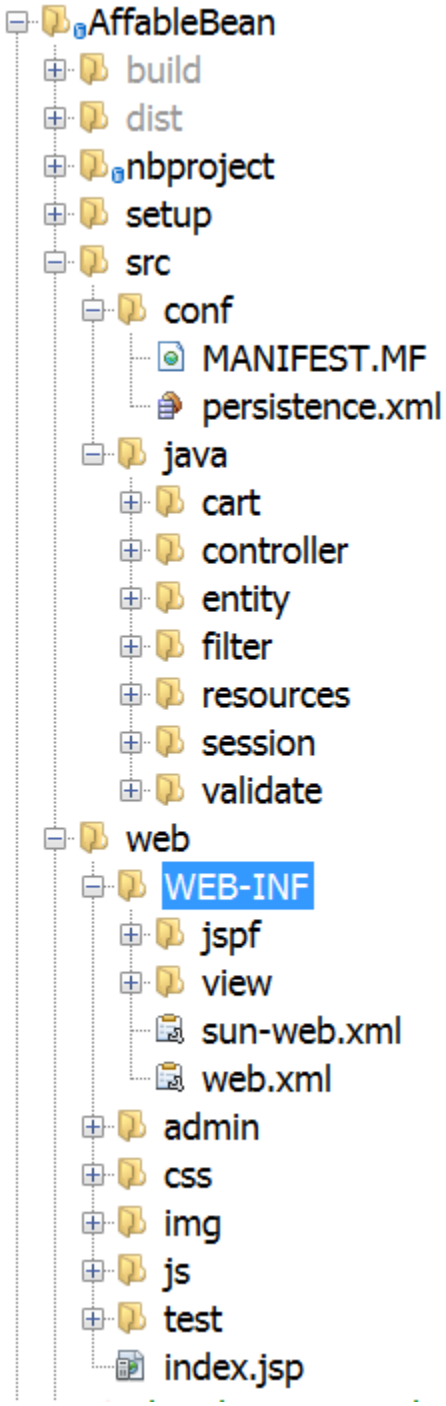
Use-Case	Server Request
Customer visits the welcome page...	/AffableBean/
...and selects a product category.	/AffableBean/category
Customer browses products within the selected category page, then adds a product to his or her shopping cart.	/AffableBean/addToCart
Customer continues shopping and selects a different category.	/AffableBean/category
Customer adds several products from this category to shopping cart.	/AffableBean/addToCart /AffableBean/addToCart

Customer selects 'view cart' option...	/AffableBean/viewCart
...and updates quantities for cart products in the cart page.	/AffableBean/updateCart
Customer verifies shopping cart contents and proceeds to checkout.	/AffableBean/checkout
In the checkout page, customer views the cost of the order and other information, fills in personal data, then submits his or her details.	/AffableBean/purchase
The order is processed and customer is taken to a confirmation page. The confirmation page provides a unique reference number for tracking the customer order, as well as a summary of the order.	(n/a)

Software Configuration Management:

The final product of this project is a 3-tier e-commerce application built for a US client Affable bean. The product is built using Java 1.8 EE and utilizes Glassfish v4 and MySQL Database v3.1. The product is built in NetBeans v8.1 and hence follows a general web application folder structure containing folders like WEB-INF for jsp files, src for Java related files, etc. Please find a snapshot of the folder structure for further clarity.

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu



The application connects to a MySQL database which should have a schema affablebean at port number 3306. The sql queries needed to import the schema has been provided under the setup folder, as part of this project for easier deployments across machines and platforms. For more information about setting up the project, please refer to README.txt in "Setup" folder in the project.

Software Development - Group 6
Nachiket Barve
Jai Wadhwani
Mithun Nallu

Source Code and Project Documentation:

Source code and project documentation can be accessed from the team's github repository at: https://github.com/jai-knows-how-to-code/Affablebean_Group6