SAN JOSÉ STATE
UNIVERSITY

FINAL PROJECT REPORT RESUBMISSION

Project: Virtual Grocery Multitenant Application

SPRING 2014

CMPE281

Instructor

Prof: Paul Nguyen

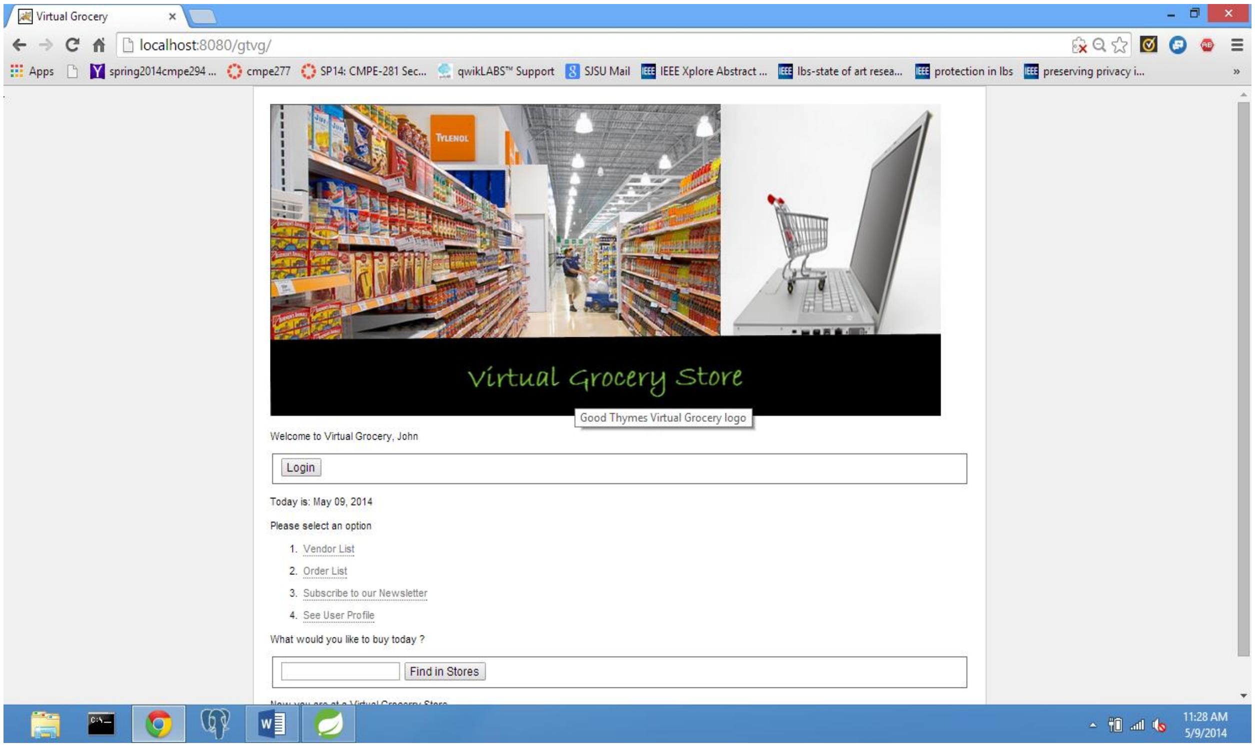
Submitted By

Geetha Anne(008741057)-Team 10

PROJECT REQUIREMENT PART 1: UI, Data Modelling and Database on Salesfroce.com

Intent of our project is to develop a multitenant online groceries store (Virtual Grocery Store) which accommodates five brands and can accommodate more number of stores in future to be added themselves as tenants. The objective was to implement a single instance multi-tenant solution which includes multiple stores and enables users to shop from different brands using a single interface. There are five grocery stores (tenants) accommodated in the Virtual Grocery SaaS application which are Sprouts, Safeway, Trader Joes, Target and Whole foods.

User Interface of the application is designed using HTML, CSS and HTML thyme leaf extension. The initial basic skeleton of the UI of the application is shown in the figure1. Further improvements are added to each individual tenant's user interface the existing UI to bring a rich looking interface for the application users. My part of the project was to design the UI for the Sprouts tenant, design its database and data model on salesforce.com which are shown in the following figures. "Buy Now" button is associated with a REST PayPal API integration to allow users to purchase the items instantly online.



Welcome to Virtual Grocery, John

Login

Today is: May 09, 2014

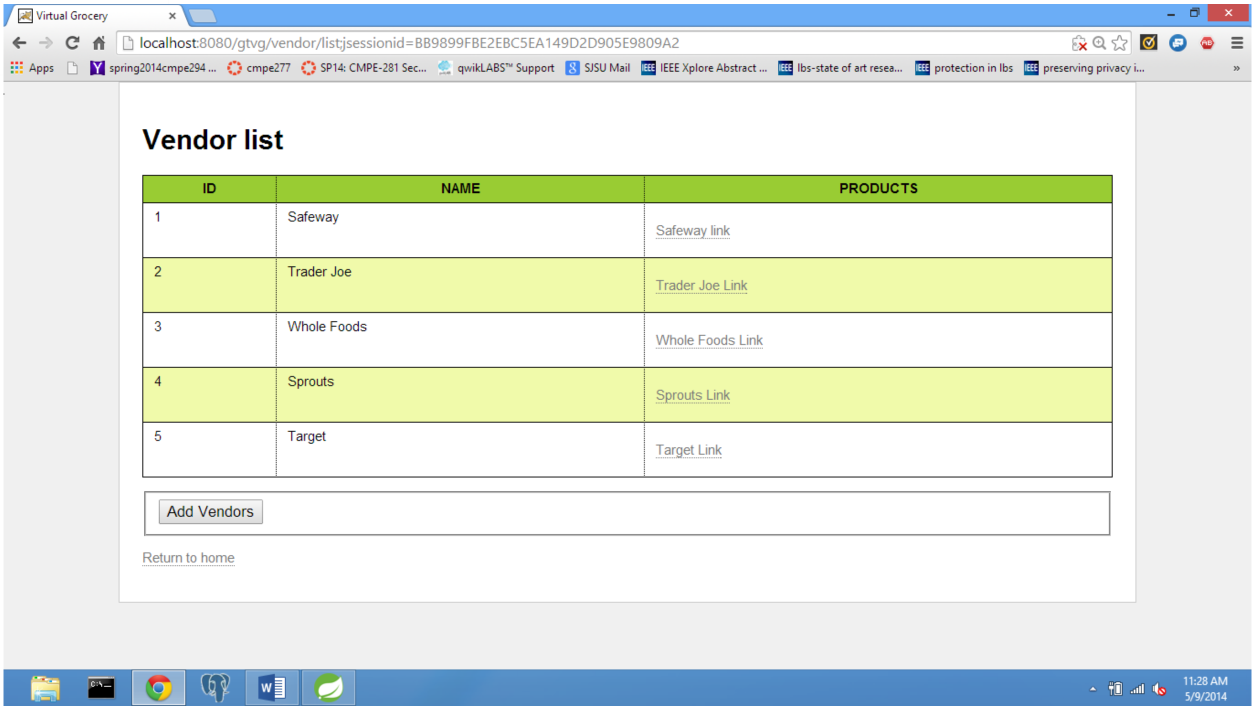
Please select an option

1. Vendor List
2. Order List
3. Subscribe to our Newsletter
4. See User Profile

What would you like to buy today ?

Find in Stores

Figure 1: Intial UI developed for the multi tenant application-Home Page.



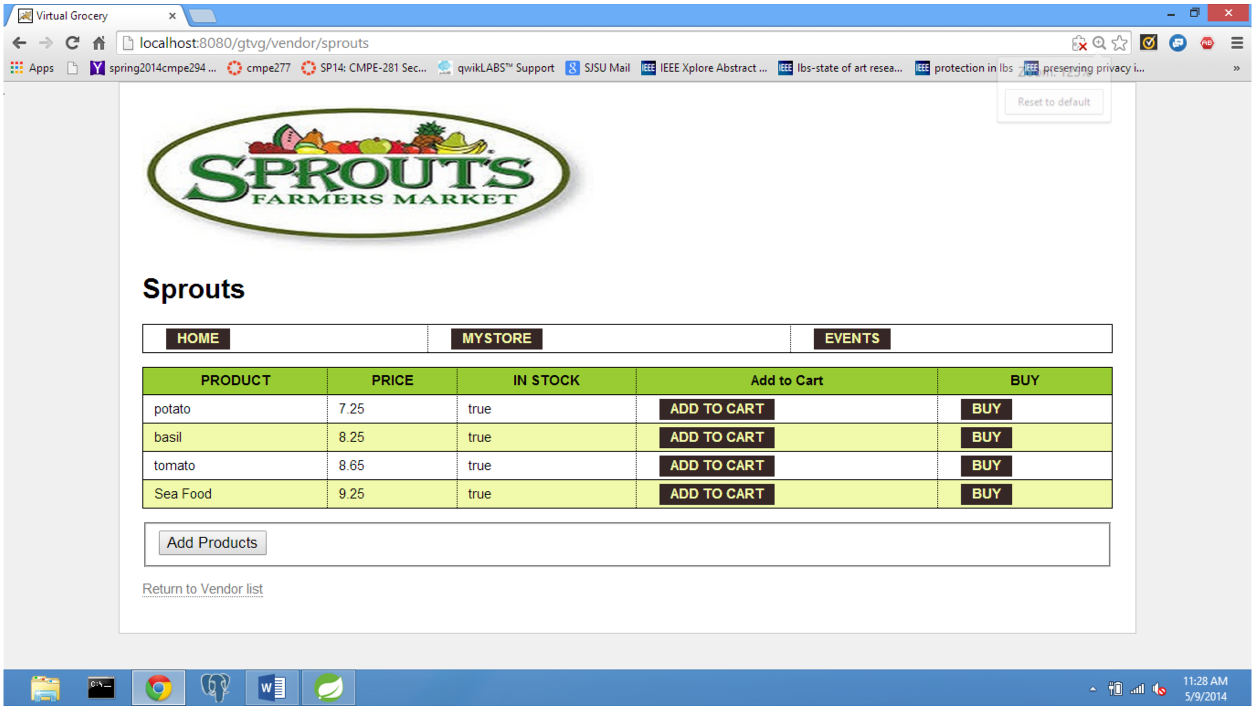
Vendor list

ID	NAME	PRODUCTS
1	Safeway	Safeway link
2	Trader Joe	Trader Joe Link
3	Whole Foods	Whole Foods Link
4	Sprouts	Sprouts Link
5	Target	Target Link

Add Vendors

Return to home

Figure 2: UI developed using HTML extension called thyme leaf for all the vendors.



Sprouts

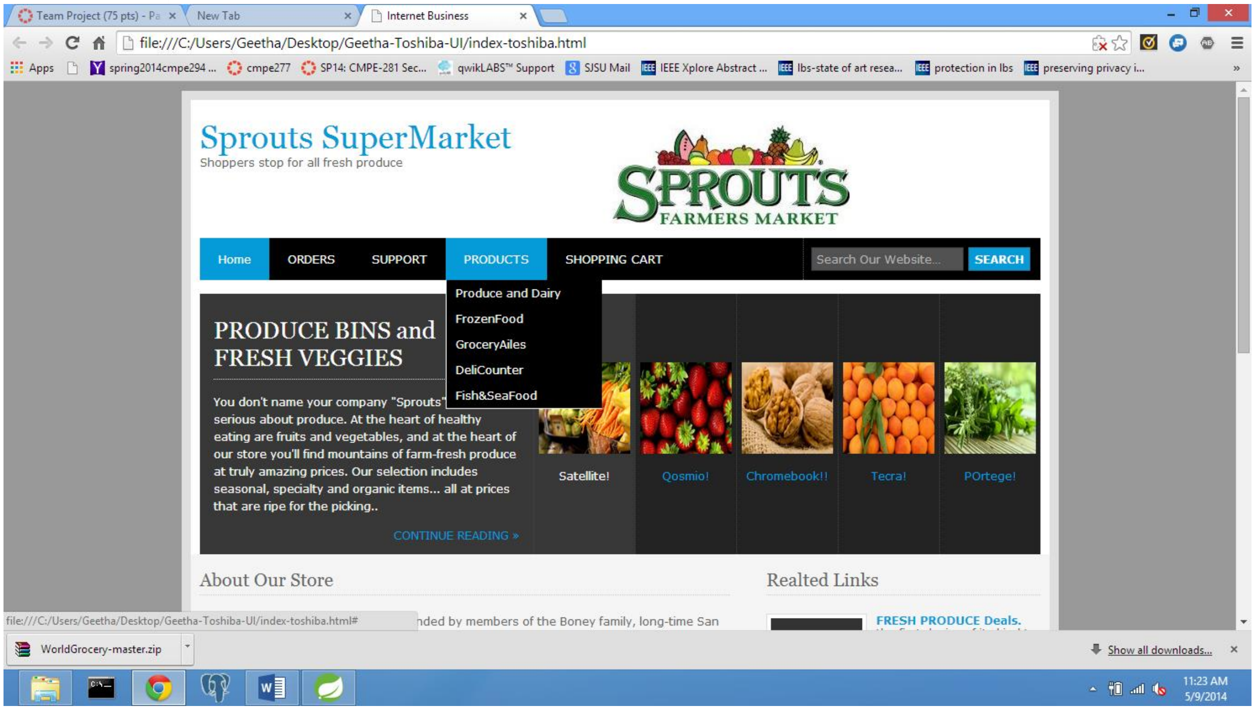
HOME	MYSTORE	EVENTS

PRODUCT	PRICE	IN STOCK	Add to Cart	BUY
potato	7.25	true	ADD TO CART	BUY
basil	8.25	true	ADD TO CART	BUY
tomato	8.65	true	ADD TO CART	BUY
Sea Food	9.25	true	ADD TO CART	BUY

Add Products

Return to Vendor list

Figure 3: Dedicated Intital UI designed for the sprouts client in HTML thyme leaf.



file:///C:/Users/Geetha/Desktop/Geetha-Toshiba-UI/index-toshiba.html

Sprouts SuperMarket

Shoppers stop for all fresh produce

PRODUCE BINS and FRESH VEGGIES

You don't name your company "Sprouts" serious about produce. At the heart of healthy eating are fruits and vegetables, and at the heart of our store you'll find mountains of farm-fresh produce at truly amazing prices. Our selection includes seasonal, specialty and organic items... all at prices that are ripe for the picking..

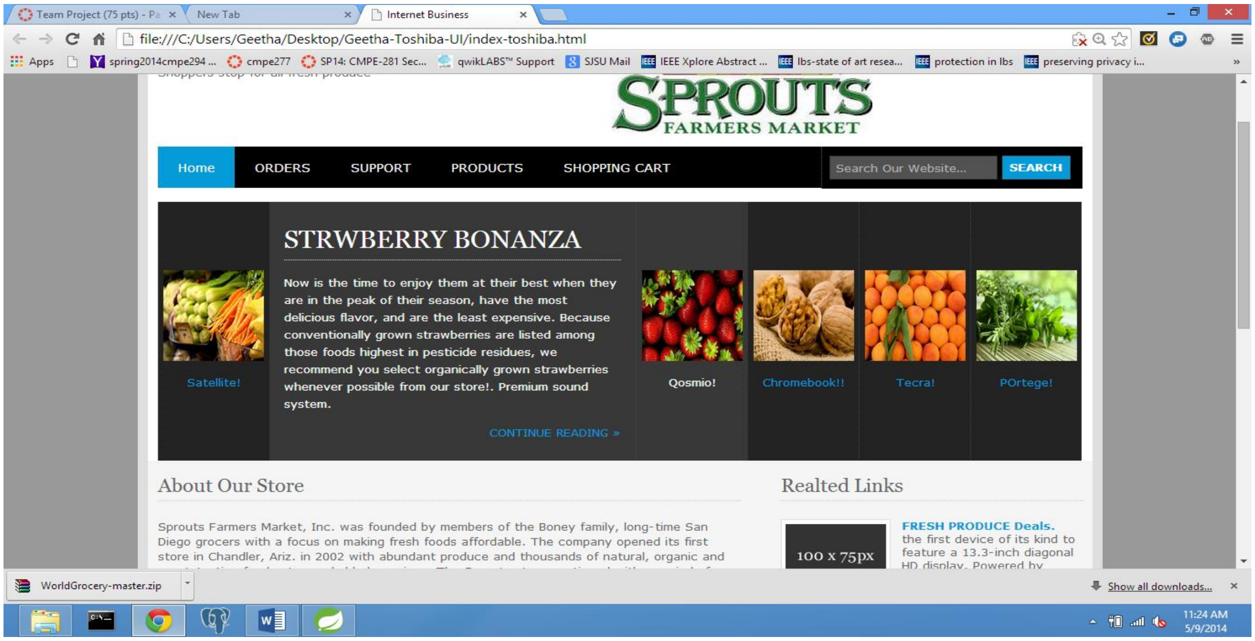
About Our Store

Realted Links

nded by members of the Boney family, long-time San

FRESH PRODUCE Deals.

Figure 4: Final UI of the Sprouts vendor after all the developments like pictures, fonts



file:///C:/Users/Geetha/Desktop/Geetha-Toshiba-UI/index-toshiba.html

STRWBERRY BONANZA

Now is the time to enjoy them at their best when they are in the peak of their season, have the most delicious flavor, and are the least expensive. Because conventionally grown strawberries are listed among those foods highest in pesticide residues, we recommend you select organically grown strawberries whenever possible from our store!. Premium sound system.

About Our Store

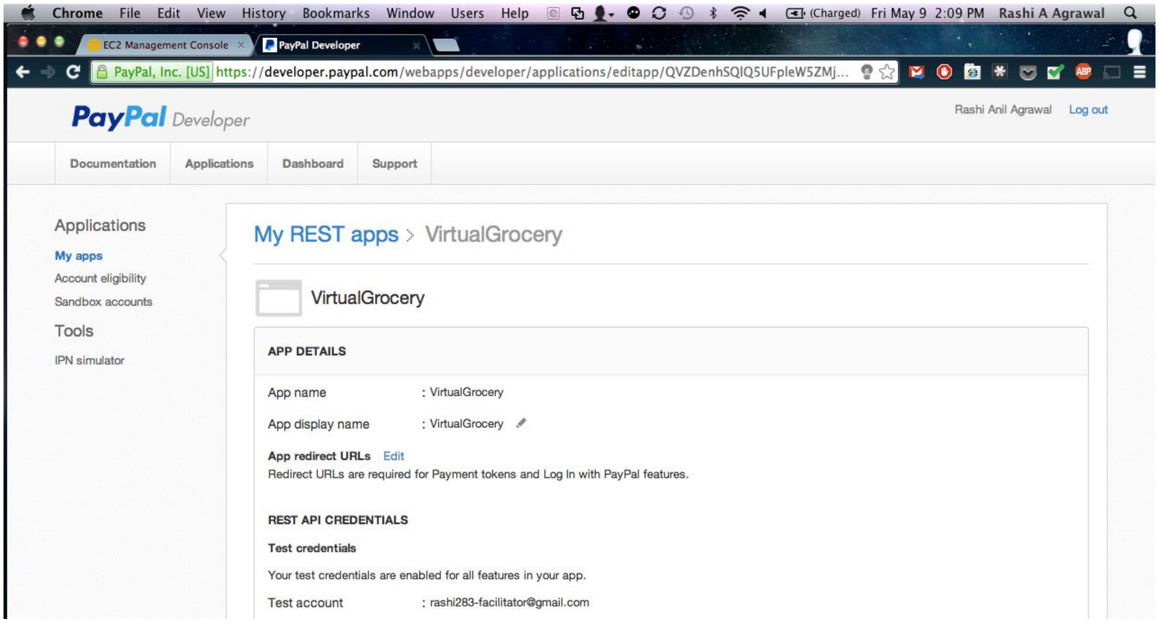
Sprouts Farmers Market, Inc. was founded by members of the Boney family, long-time San Diego grocers with a focus on making fresh foods affordable. The company opened its first store in Chandler, Ariz. in 2002 with abundant produce and thousands of natural, organic and

Realted Links

FRESH PRODUCE Deals.

the first device of its kind to feature a 13.3-inch diagonal HD display. Powered by

Figure 5: Final UI of the Sprouts vendor after all the developments like pictures, fonts.



https://developer.paypal.com/webapps/developer/applications/editapp/QVZDenhSQlQ5UFpleW5ZMj...

VirtualGrocery

APP DETAILS

App name : VirtualGrocery

App display name : VirtualGrocery

App redirect URLs Edit

Redirect URLs are required for Payment tokens and Log In with PayPal features.

REST API CREDENTIALS

Test credentials

Your test credentials are enabled for all features in your app.

Test account : rashi283-facilitator@gmail.com

Figure 6: Paypal integrated into the application using REST for the buynow button.

Object Dynamic models:

Virtual Grocery application's classes are represented in the form of dynamic objects in the following class diagrams. Relationships and multiplicities are shown between each class. The below mentioned tasks are achieved as part of the project requirement 3.

1. Object Dynamic Models are drawn for the entire Virtual Grocery application
2. Object Dynamic Model (class Diagram) id drawn for the individual tenant (Sprouts) as part of individual contribution showing all the classes that this tenant specific.
3. User or Customer profile Database is created for the individual tenant i.e. Sprouts in force.com
4. Data Modelling is done using the Schema Builder in force.com

Class diagram for the sprouts vendor:

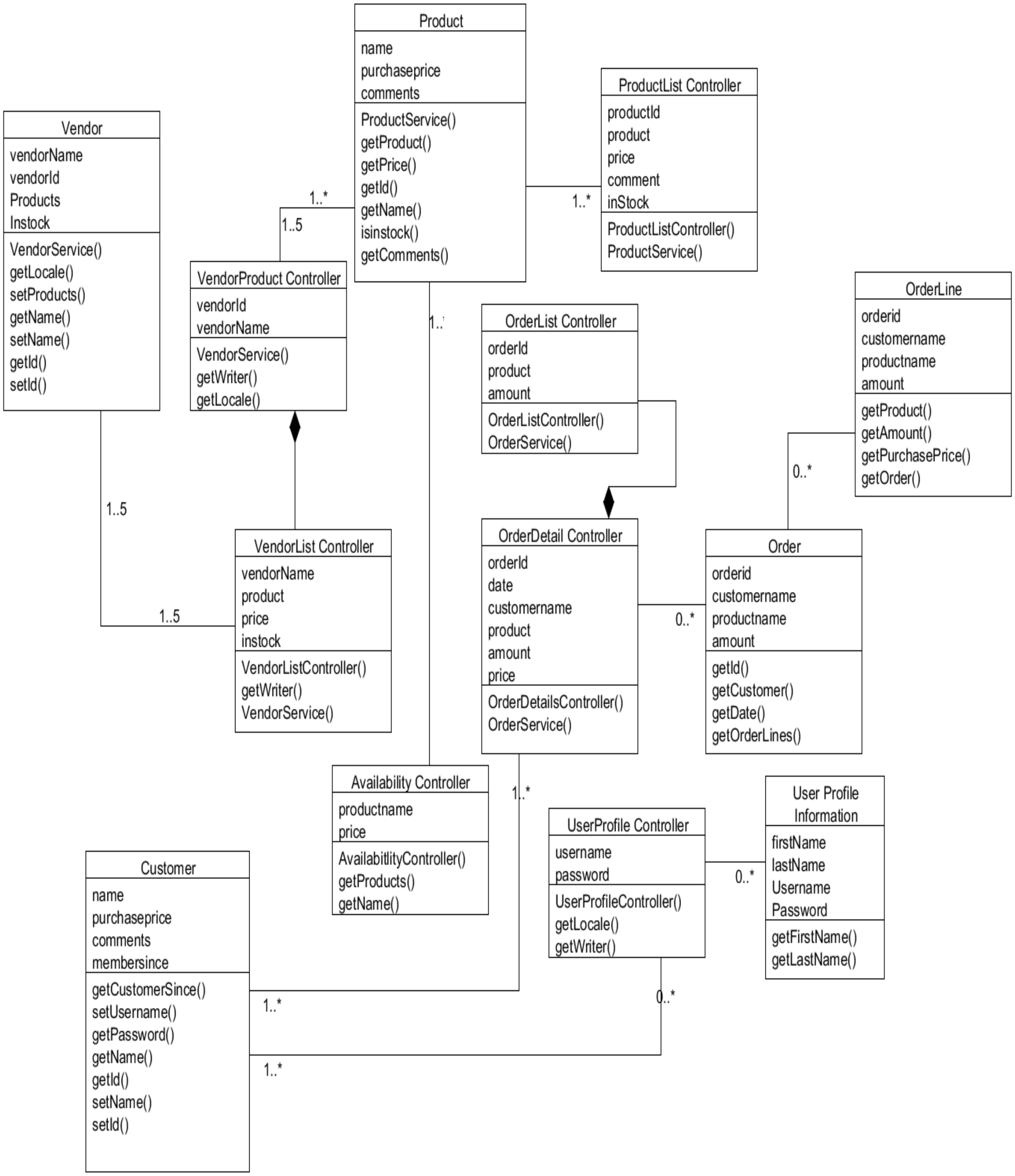


Figure 7: Object Dynamic Model for the entire Virtual Grocery application

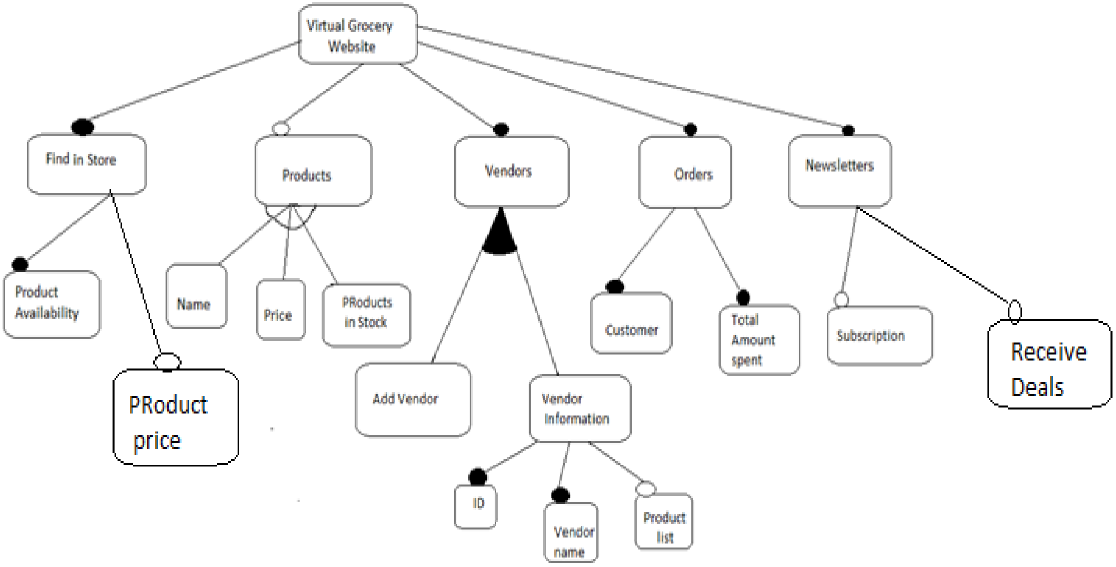


Figure 8: Feature model

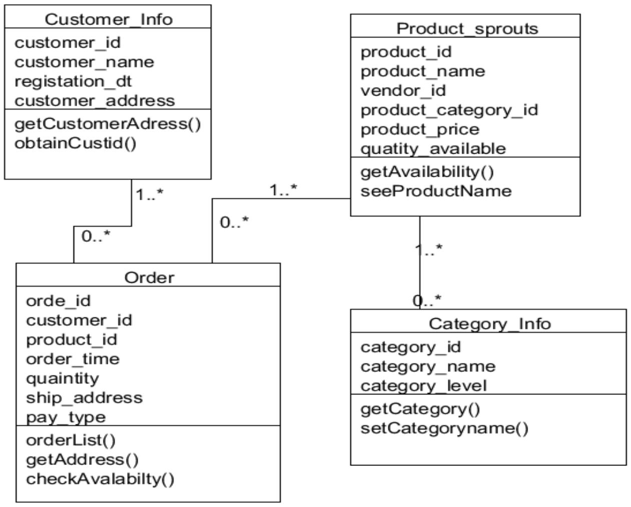
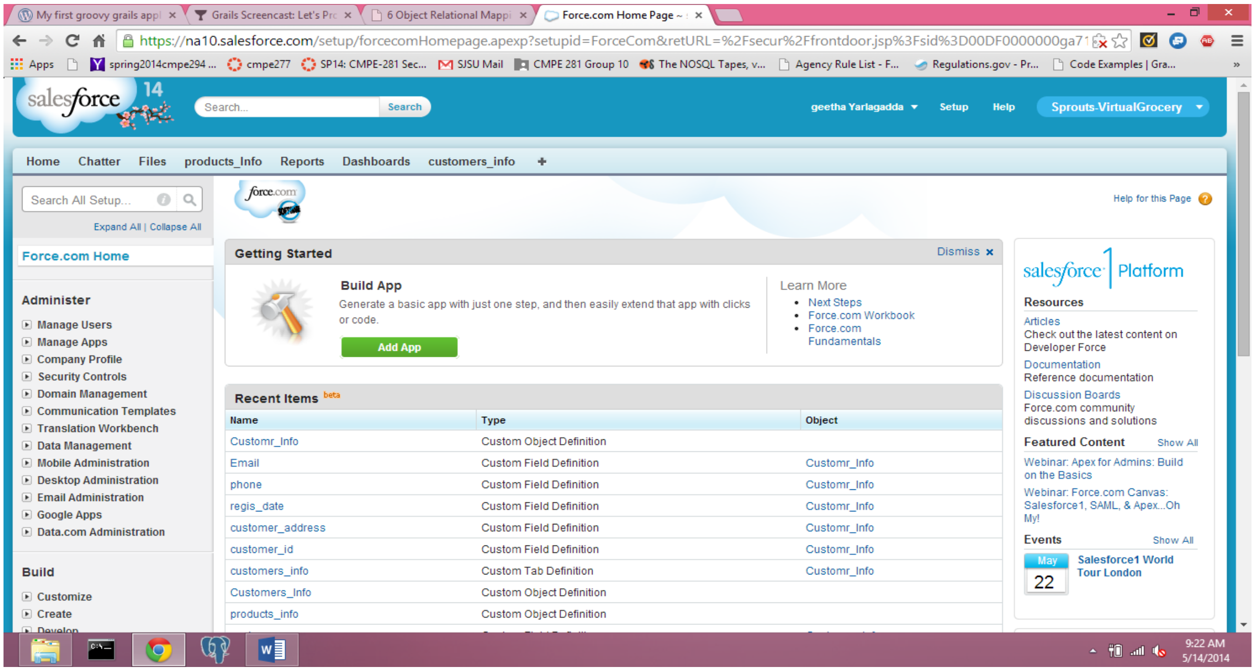


Figure 9: Object Model for Sprouts tenant:

SCREEN CAPTURES OF SALESFORCE- SPROUTS VIRTUALGROCERY APP:

User and Product Databases on SALESFORCE.com:



Getting Started

Build App

Generate a basic app with just one step, and then easily extend that app with clicks or code.

Learn More
- Next Steps
- Force.com Workbook
- Force.com Fundamentals

Recent Items beta

Name	Type	Object
Customr_Info	Custom Object Definition	
Email	Custom Field Definition	Customr_Info
phone	Custom Field Definition	Customr_Info
regis_date	Custom Field Definition	Customr_Info
customer_address	Custom Field Definition	Customr_Info
customer_id	Custom Field Definition	Customr_Info
customers_info	Custom Tab Definition	Customr_Info
Customers_Info	Custom Object Definition	
products_info	Custom Object Definition	

salesforce Platform

Resources

Articles
Check out the latest content on Developer Force

Documentation
Reference documentation

Discussion Boards
Force.com community discussions and solutions

Featured Content Show All

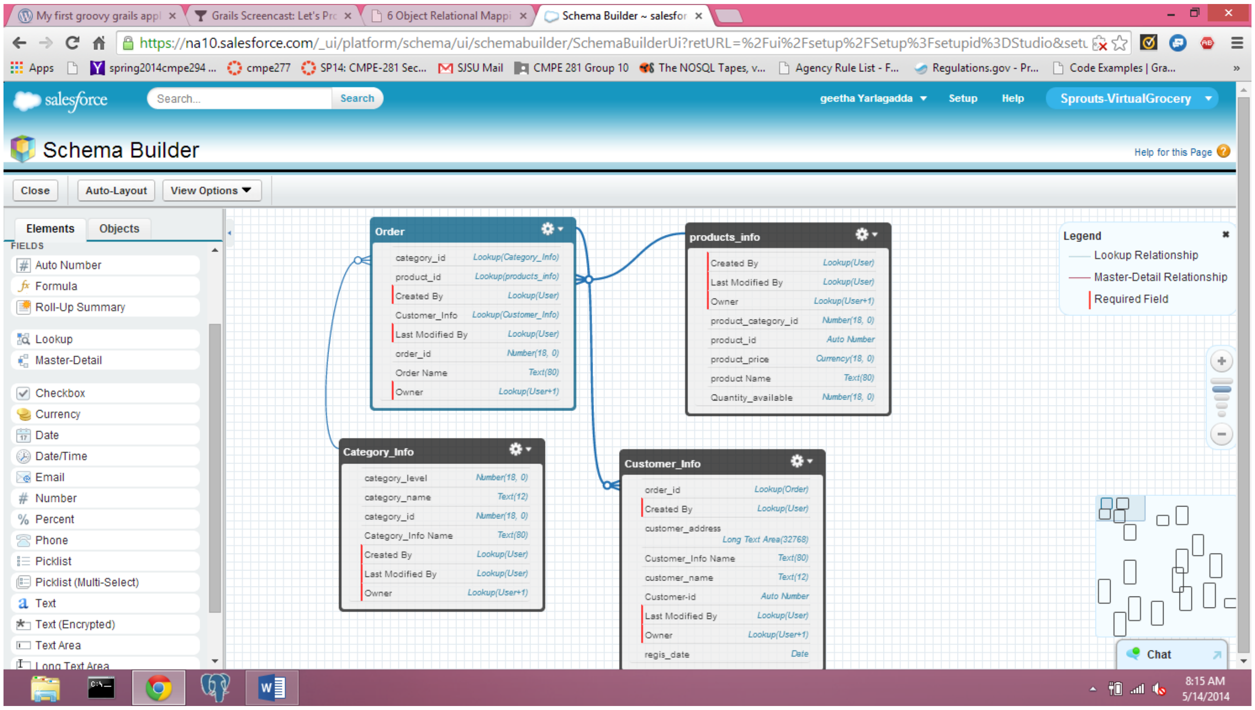
Webinar: Apex for Admins: Build on the Basics

Webinar: Force.com Canvas: Salesforce1, SAML, & Apex...Oh My!

Events Show All

May 22 Salesforce1 World Tour London

Figure 10: Final UI of the Sprouts vendor after all the developments like pictures, fonts.



Order

category_id	Lookup(Category_Info)
product_id	Lookup(products_info)
Created By	Lookup(User)
Customer_Info	Lookup(Customer_Info)
Last Modified By	Lookup(User)
order_id	Number(18, 0)
Order Name	Text(80)
Owner	Lookup(User+1)

products_info

Created By	Lookup(User)
Last Modified By	Lookup(User)
Owner	Lookup(User+1)
product_category_id	Number(18, 0)
product_id	Auto Number
product_price	Currency(18, 0)
product Name	Text(80)
Quantity_available	Number(18, 0)

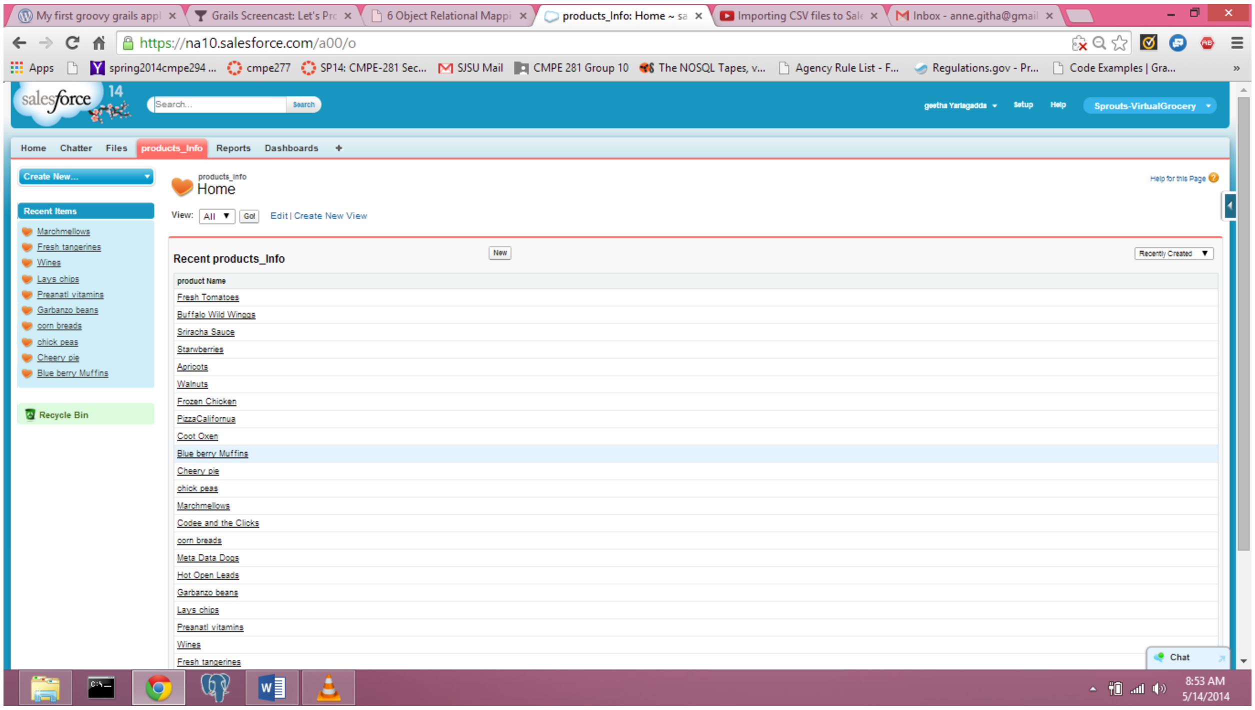
Category_Info

category_level	Number(18, 0)
category_name	Text(12)
category_id	Number(18, 0)
Category_Info Name	Text(80)
Created By	Lookup(User)
Last Modified By	Lookup(User)
Owner	Lookup(User+1)

Customer_Info

order_id	Lookup(Order)
Created By	Lookup(User)
customer_address	Long Text Area(32768)
Customer_Info Name	Text(80)
customer_name	Text(12)
Customer-id	Auto Number
Last Modified By	Lookup(User)
Owner	Lookup(User+1)
regis_date	Date

Figure 11: Data Model in the Schema Builder view:



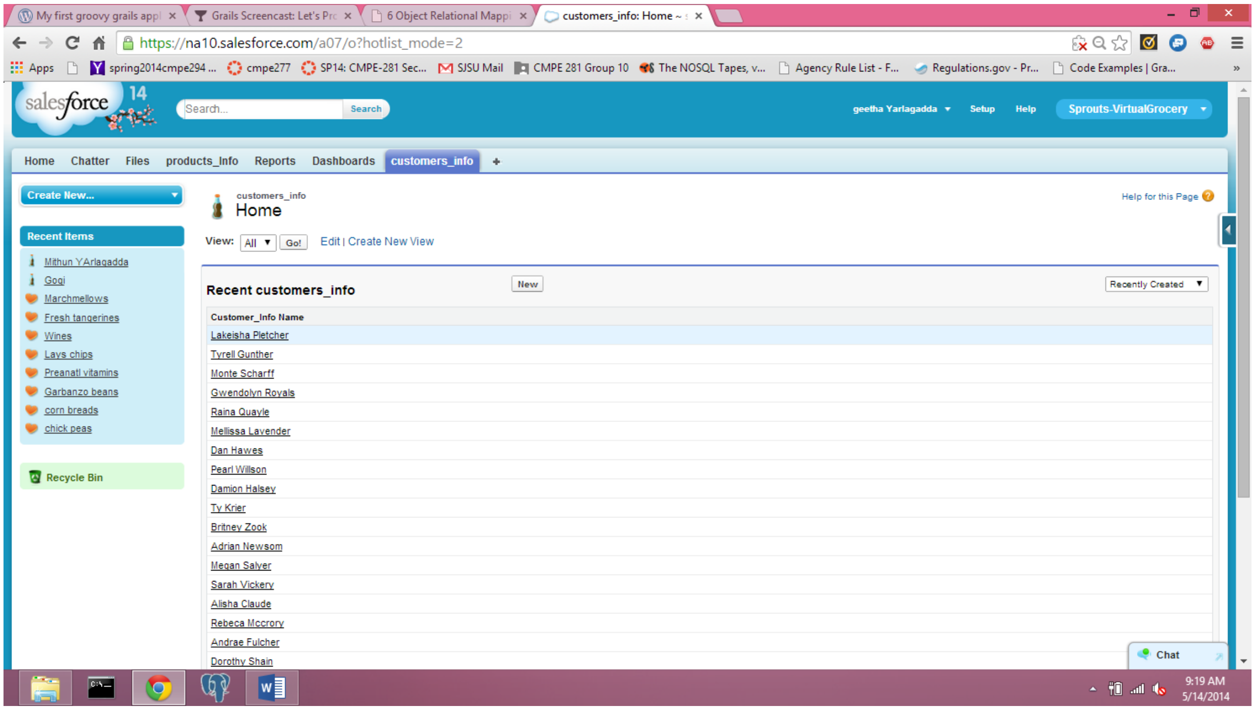
products_Info
Home

Recent Items
- Marchmellows
- Fresh tangerines
- Wines
- Lays chips
- Preanatl vitamins
- Garbanzo beans
- corn breads
- chick peas
- Cheery pie
- Blue berry Muffins

Recent products_Info

product Name
Fresh Tomatoes
Buffalo Wild Winggs
Sriracha Sauce
Starwberries
Apricots
Walnuts
Frozen Chicken
PizzaCalifornua
Coot Oxen
Blue berry Muffins
Cheery pie
chick peas
Marchmellows
Codee and the Clicks
corn breads
Meta Data Dogs
Hot Open Leads
Garbanzo beans
Lays chips
Preanatl vitamins
Wines
Fresh tangerines

Figure 12: Product Information Database on Salesforce:



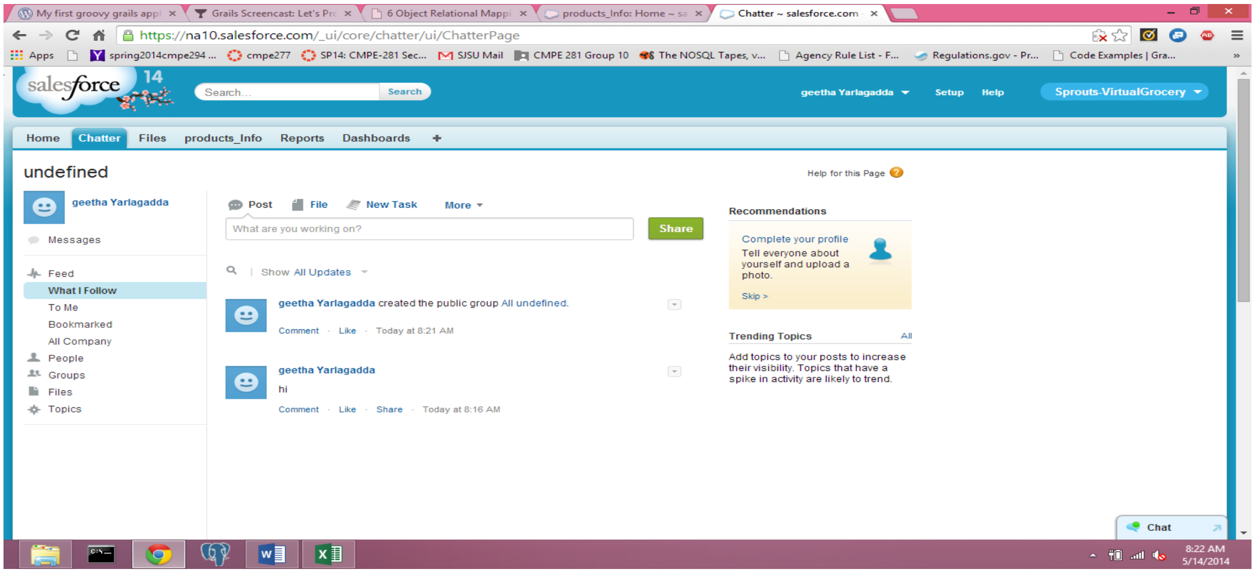
customers_info
Home

Recent Items
- Mithun YArlagadda
- Gogi
- Marchmellows
- Fresh tangerines
- Wines
- Lays chips
- Preanatl vitamins
- Garbanzo beans
- corn breads
- chick peas

Recent customers_info

Customer_Info Name
Lakeisha Pletcher
Tyrell Gunther
Monte Scharff
Gwendolyn Royals
Raina Quayle
Mellissa Lavender
Dan Hawes
Pearl Willson
Damion Halsey
Ty Krier
Britney Zook
Adrian Newsom
Megan Salyer
Sarah Vickery
Alisha Claude
Rebeca Mccrory
Andrae Fulcher
Dorothy Shain

Figure 13: Customer Data base on Salesforce.com:



undefined

geetha Yarlagadda created the public group All undefined.

Comment · Like · Today at 8:21 AM

geetha Yarlagadda

hi

Comment · Like · Share · Today at 8:16 AM

Recommendations

Complete your profile
Tell everyone about yourself and upload a photo.
Skip >

Trending Topics All

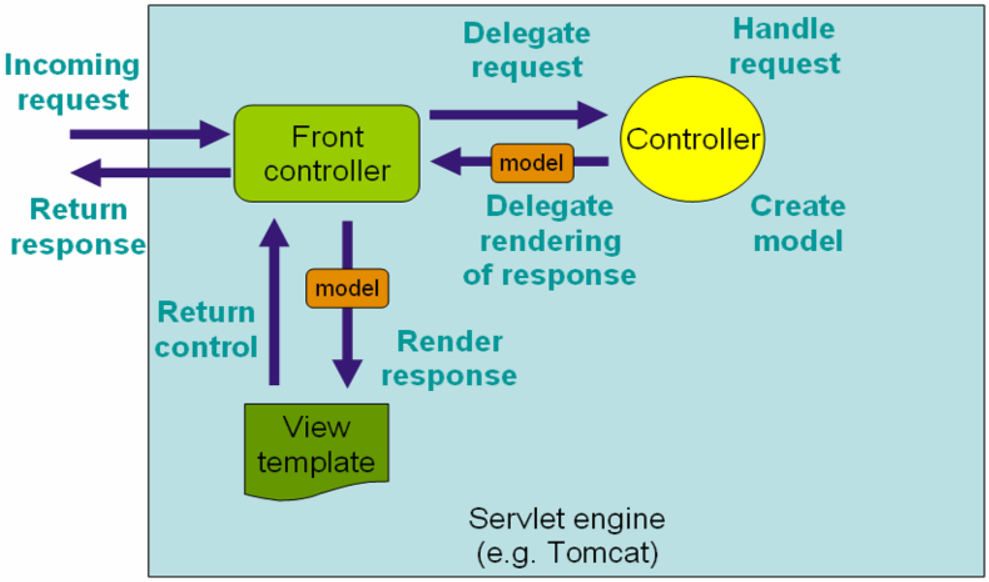
Add topics to your posts to increase their visibility. Topics that have a spike in activity are likely to trend.

Figure 14: Chatter view of the Virtual Grocery Application on desktop:

PROJECT REQUIREMET 2: (IDE tools, Prototypes, Code samples from Spring Tool Suite)

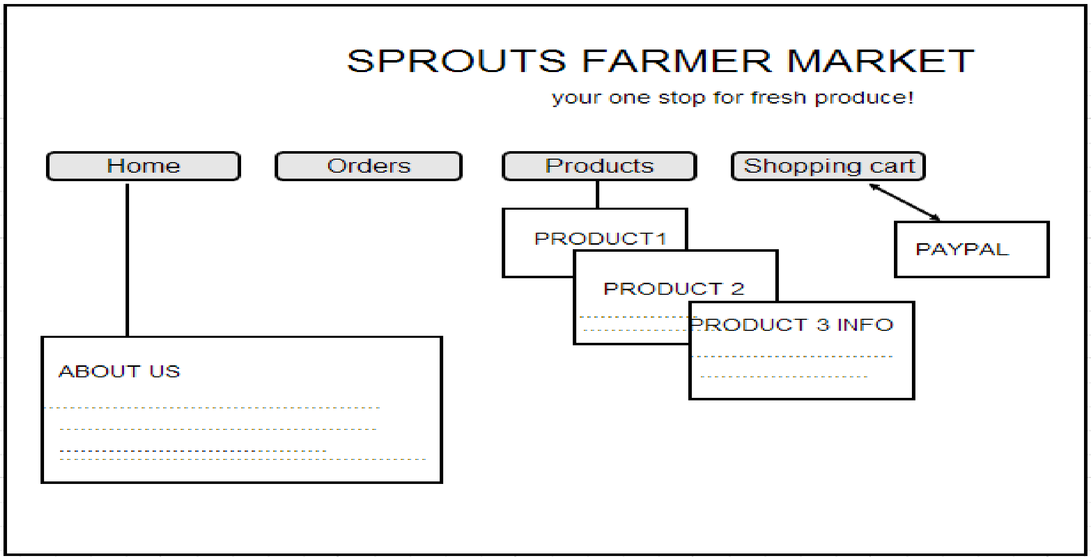
As part of the individual effort to design a UI, data model and development of sprouts vendor, the IDE used is the Spring Tool Suite (STS). Spring MVC architecture defines Model, views and controllers. The below shown are the screenshots of the code snippets for the sprouts vendor individually.

Individual Research Work on the IDE Spring Tool Suite's MVC architecture:

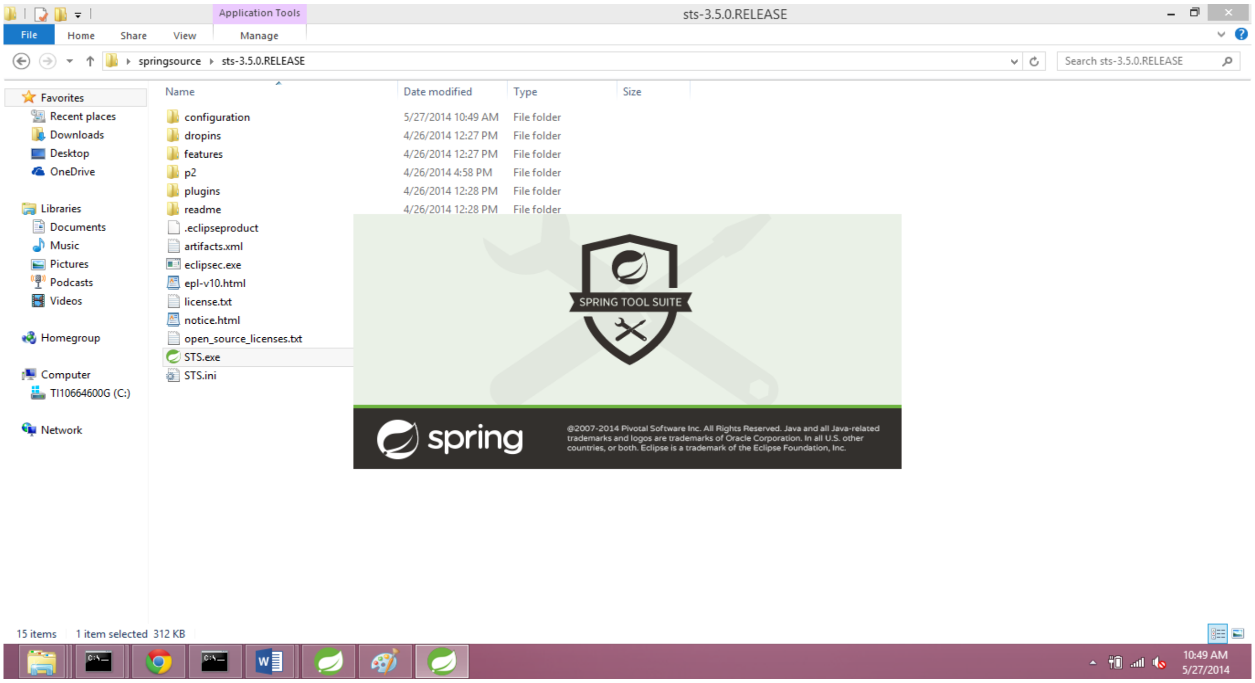


The above diagram shows the classic Model View controller architecture of the Spring framework. This type of MVC architecture ensures clear separation between several objects like model object, controllers, Dispatcher Servlets, handler mappings, view resolvers etc. Reusable business code is an important feature of this MVC model without duplication.

Prototypes for the Sprouts vendor: (Mockup)



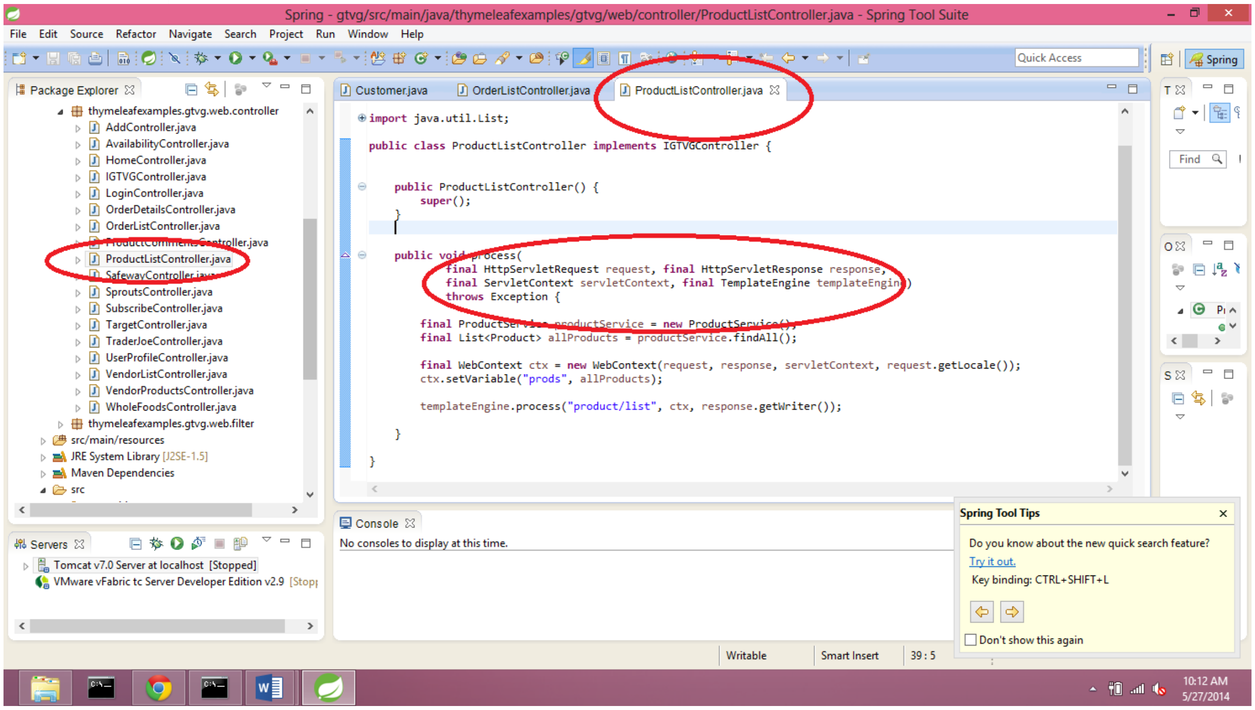
SCREEN CAPTURES OF IDE TOOL (STS), CODE SNIPPETS:



Name	Date modified	Type	Size
configuration	5/27/2014 10:49 AM	File folder	
dropins	4/26/2014 12:27 PM	File folder	
features	4/26/2014 12:27 PM	File folder	
p2	4/26/2014 4:58 PM	File folder	
plugins	4/26/2014 12:28 PM	File folder	
readme	4/26/2014 12:28 PM	File folder	
.eclipseproduct			
artifacts.xml			
eclipsec.exe			
epl-v10.html			
license.txt			
notice.html			
open_source_licenses.txt			
STS.exe			
STS.ini			

@2007-2014 Pivotal Software Inc. All Rights Reserved. Java and all Java-related trademarks and logos are trademarks of Oracle Corporation. In all U.S. other countries, or both. Eclipse is a trademark of the Eclipse Foundation, Inc.

Figure 15: Installation of Spring STS tool suite



Spring - gtvg/src/main/java/thymeleafexamples/gtvg/web/controller/ProductListController.java - Spring Tool Suite

```java
import java.util.List;

public class ProductListController implements IGTVGController {


    public ProductListController() {
        super();
    }

    public void process(
            final HttpServletRequest request, final HttpServletResponse response,
            final ServletContext servletContext, final TemplateEngine templateEngine)
            throws Exception {

        final ProductService productService = new ProductService();
        final List<Product> allProducts = productService.findAll();

        final WebContext ctx = new WebContext(request, response, servletContext, request.getLocale());
        ctx.setVariable("prods", allProducts);

        templateEngine.process("product/list", ctx, response.getWriter());

    }

}
```

No consoles to display at this time.

Spring Tool Tips

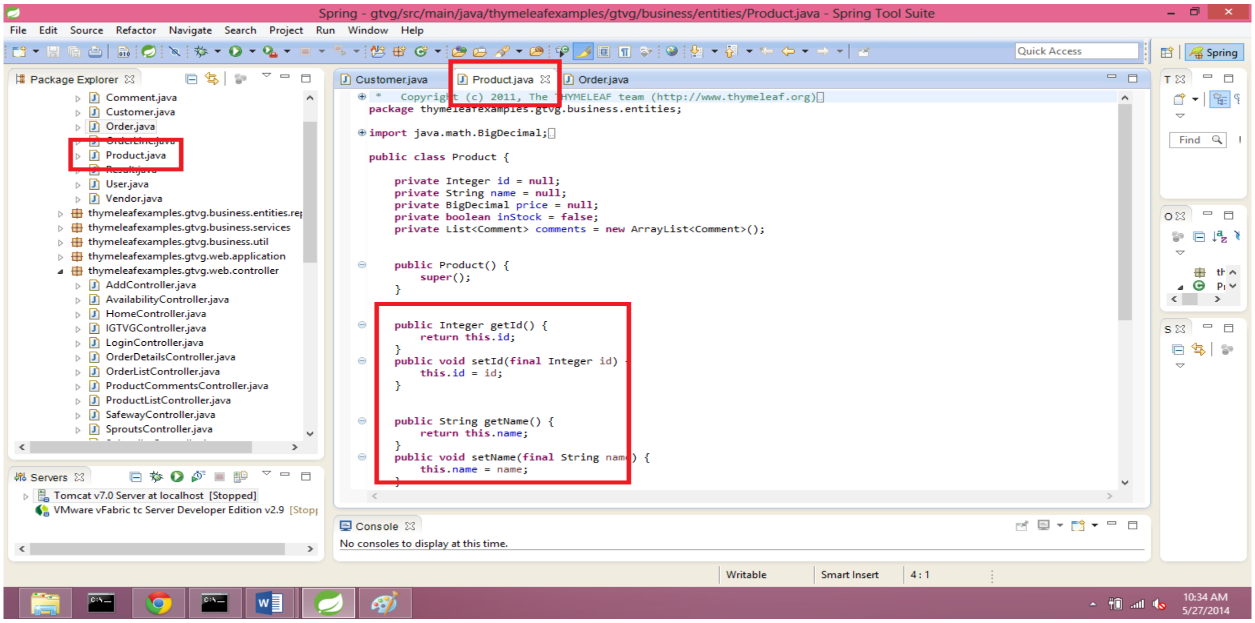
Do you know about the new quick search feature?

Try it out.

Key binding: CTRL+SHIFT+L

□ Don't show this again

Figure 16: Controller written in java for the product list of sprouts vendor

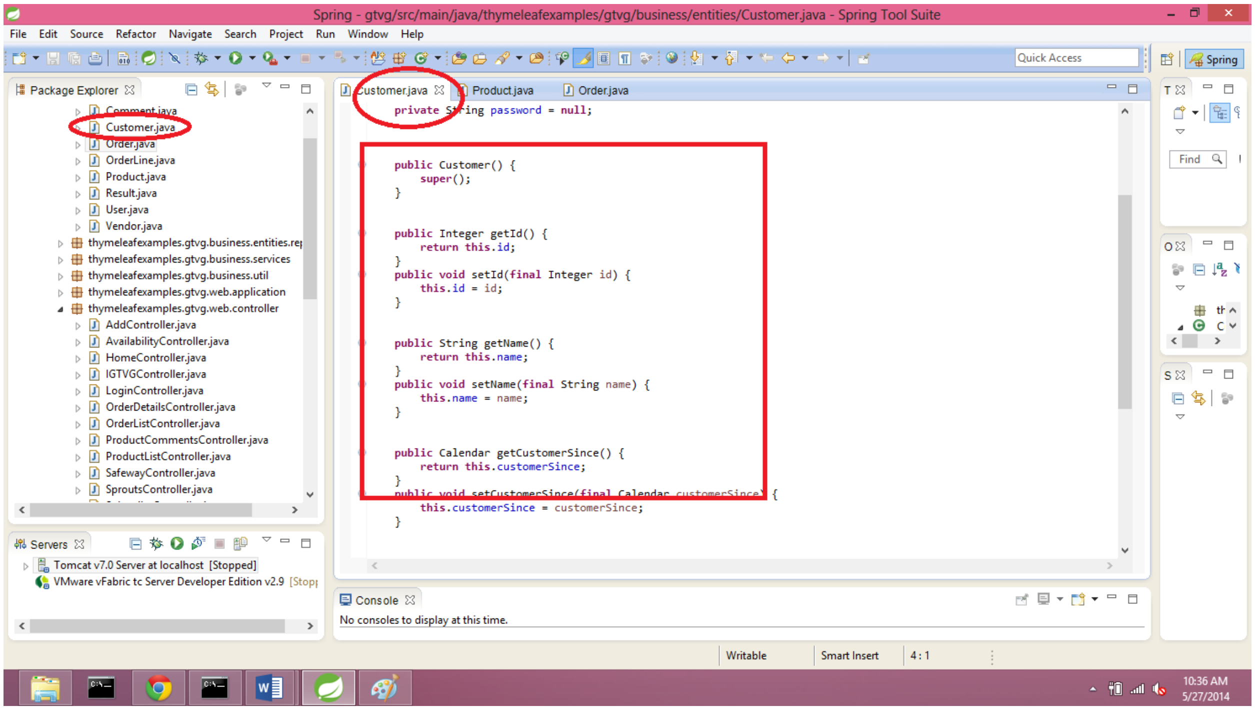


Spring - gtvg/src/main/java/thymeleafexamples/gtvg/business/entities/Product.java - Spring Tool Suite

```java
 *   Copyright (c) 2011, The THYMELEAF team (http://www.thymeleaf.org)
package thymeleafexamples.gtvg.business.entities;

import java.math.BigDecimal;

public class Product {

    private Integer id = null;
    private String name = null;
    private BigDecimal price = null;
    private boolean inStock = false;
    private List<Comment> comments = new ArrayList<Comment>();


    public Product() {
        super();
    }


    public Integer getId() {
        return this.id;
    }
    public void setId(final Integer id) {
        this.id = id;
    }


    public String getName() {
        return this.name;
    }
    public void setName(final String name) {
        this.name = name;
    }
```

No consoles to display at this time.

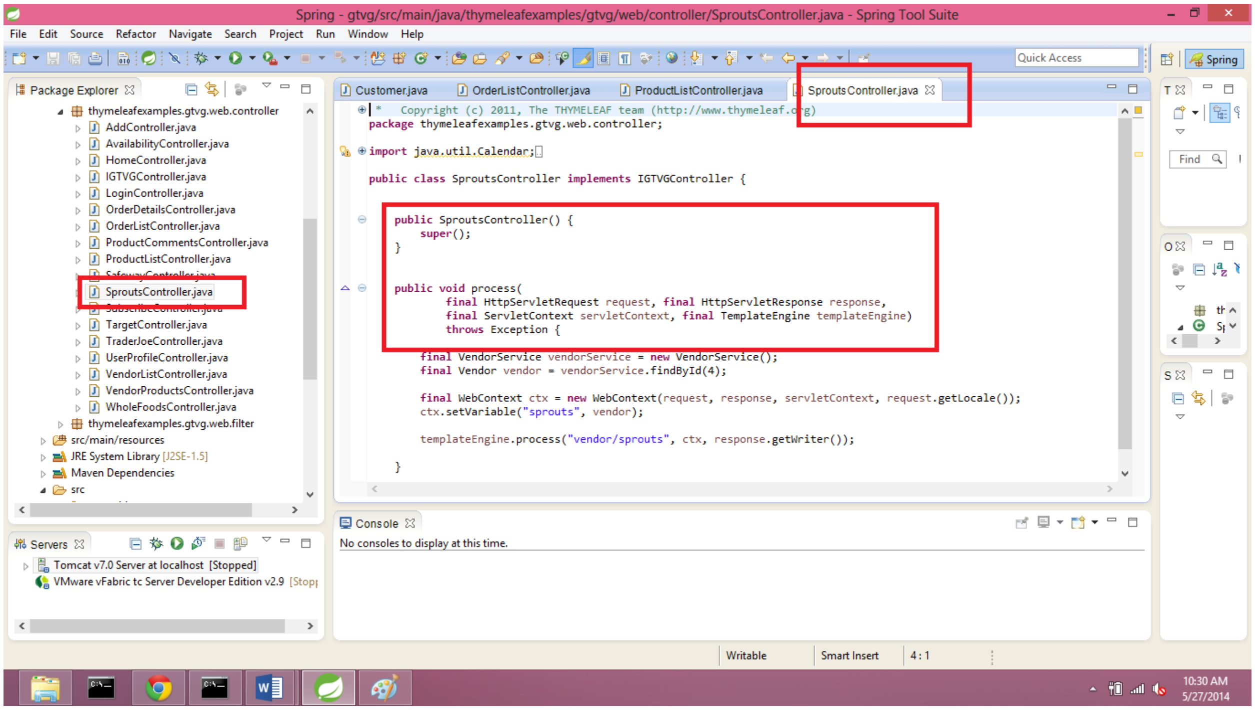
Figure 17: Model part of SpringMVC, Product.java code snippet having methods like getName(), getPrice() etc..


Spring - gtvg/src/main/java/thymeleafexamples/gtvg/business/entities/Customer.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Spring
Package Explorer
Comment.java
Customer.java
Order.java
OrderLine.java
Product.java
Result.java
User.java
Vendor.java
thymeleafexamples.gtvg.business.entities.rep
thymeleafexamples.gtvg.business.services
thymeleafexamples.gtvg.business.util
thymeleafexamples.gtvg.web.application
thymeleafexamples.gtvg.web.controller
AddController.java
AvailabilityController.java
HomeController.java
IGTVGController.java
LoginController.java
OrderDetailsController.java
OrderListController.java
ProductCommentsController.java
ProductListController.java
SafewayController.java
SproutsController.java
Customer.java
Product.java
Order.java
Servers
Tomcat v7.0 Server at localhost [Stopped]
VMware vFabric tc Server Developer Edition v2.9 [Stopp
Console
Writable
Smart Insert
4 : 1
10:36 AM
5/27/2014

```java
    private String password = null;


    public Customer() {
        super();
    }


    public Integer getId() {
        return this.id;
    }
    public void setId(final Integer id) {
        this.id = id;
    }


    public String getName() {
        return this.name;
    }
    public void setName(final String name) {
        this.name = name;
    }


    public Calendar getCustomerSince() {
        return this.customerSince;
    }
    public void setCustomerSince(final Calendar customerSince) {
        this.customerSince = customerSince;
    }
```

No consoles to display at this time.

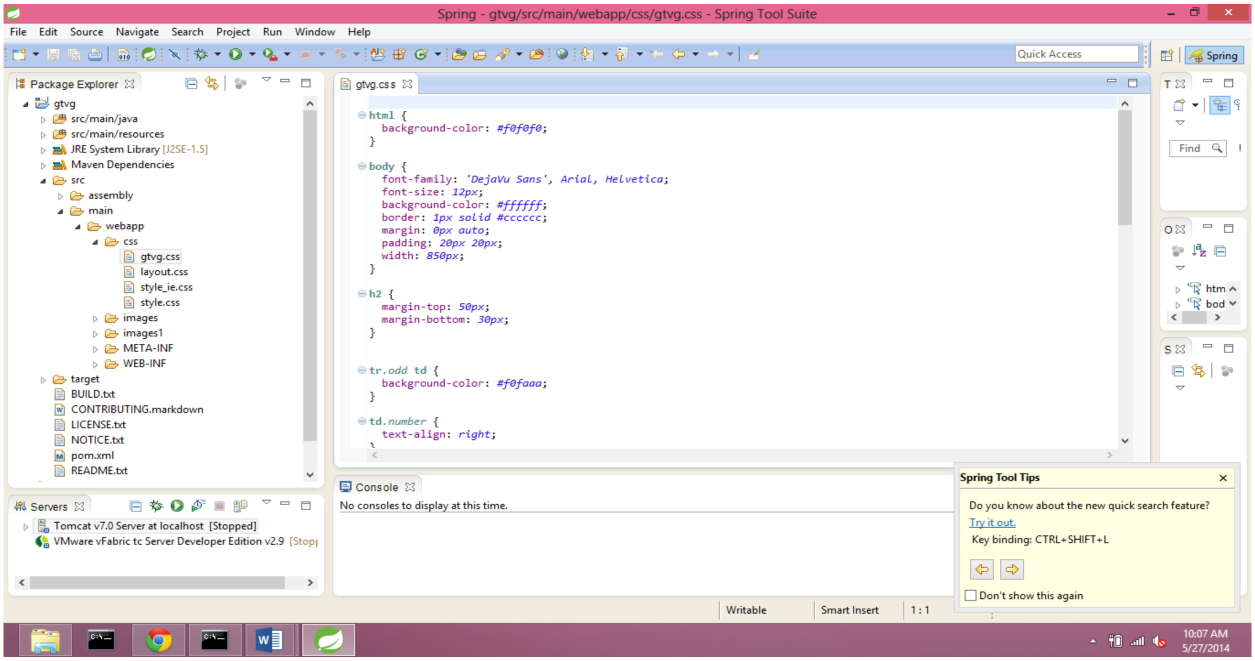
Figure 18: Customer.java containing getName(), getCustomerSince() methods etc..


Spring - gtvg/src/main/java/thymeleafexamples/gtvg/web/controller/SproutsController.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Spring
Package Explorer
thymeleafexamples.gtvg.web.controller
AddController.java
AvailabilityController.java
HomeController.java
IGTVGController.java
LoginController.java
OrderDetailsController.java
OrderListController.java
ProductCommentsController.java
ProductListController.java
SafewayController.java
SproutsController.java
SubscribeController.java
TargetController.java
TraderJoeController.java
UserProfileController.java
VendorListController.java
VendorProductsController.java
WholeFoodsController.java
thymeleafexamples.gtvg.web.filter
src/main/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
Customer.java
OrderListController.java
ProductListController.java
SproutsController.java
Servers
Tomcat v7.0 Server at localhost [Stopped]
VMware vFabric tc Server Developer Edition v2.9 [Stopp
Console
Writable
Smart Insert
4 : 1
10:30 AM
5/27/2014

```java
 *   Copyright (c) 2011, The THYMELEAF team (http://www.thymeleaf.org)
package thymeleafexamples.gtvg.web.controller;

import java.util.Calendar;

public class SproutsController implements IGTVGController {


    public SproutsController() {
        super();
    }


    public void process(
            final HttpServletRequest request, final HttpServletResponse response,
            final ServletContext servletContext, final TemplateEngine templateEngine)
            throws Exception {

        final VendorService vendorService = new VendorService();
        final Vendor vendor = vendorService.findById(4);

        final WebContext ctx = new WebContext(request, response, servletContext, request.getLocale());
        ctx.setVariable("sprouts", vendor);

        templateEngine.process("vendor/sprouts", ctx, response.getWriter());

    }
```

No consoles to display at this time.

Figure 19: One of the sprouts vendor controller



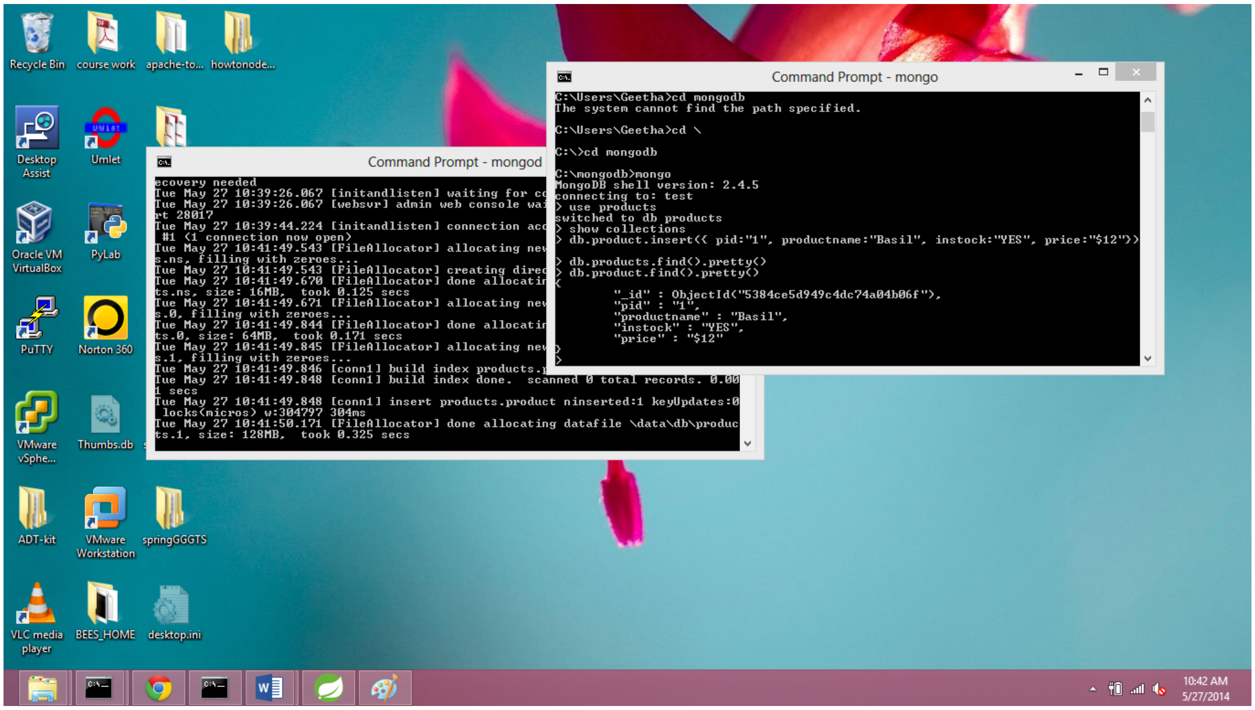
```
html {
  background-color: #f0f0f0;
}

body {
  font-family: 'DejaVu Sans', Arial, Helvetica;
  font-size: 12px;
  background-color: #ffffff;
  border: 1px solid #cccccc;
  margin: 0px auto;
  padding: 20px 20px;
  width: 850px;
}

h2 {
  margin-top: 50px;
  margin-bottom: 30px;
}


tr.odd td {
  background-color: #f0faaa;
}

td.number {
  text-align: right;
```

No consoles to display at this time.

Do you know about the new quick search feature?
Try it out.
Key binding: CTRL+SHIFT+L

□ Don't show this again

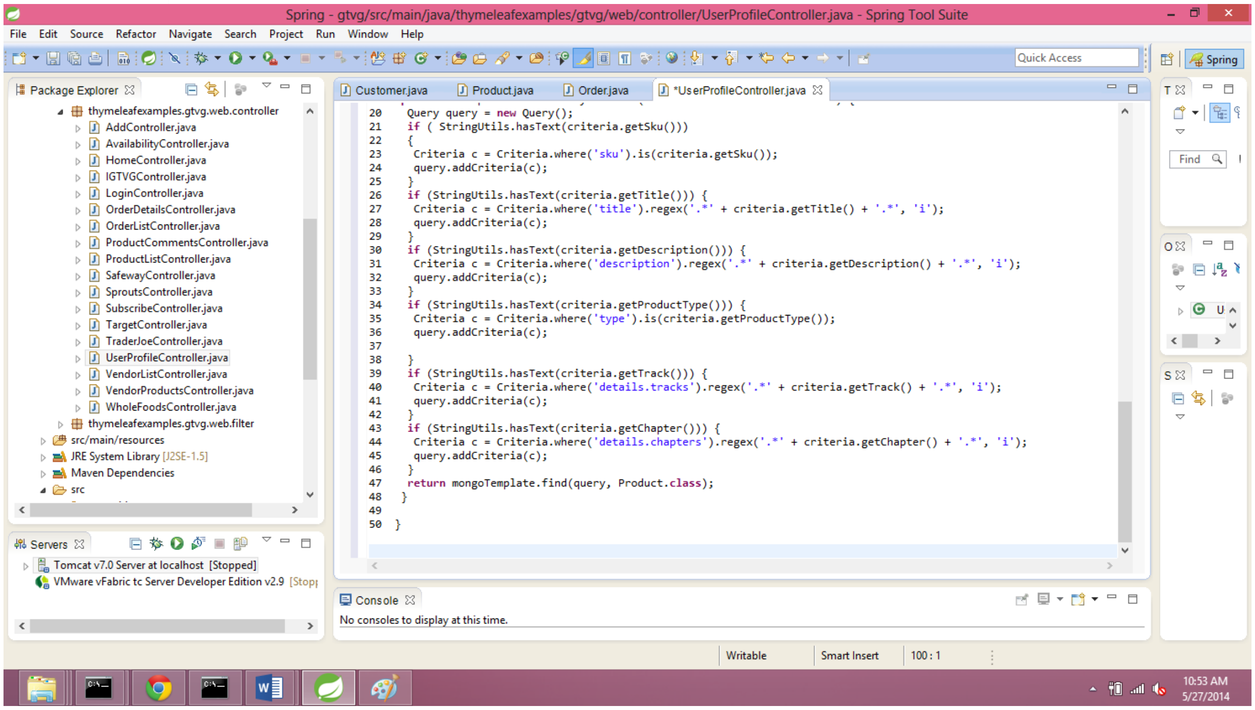
Figure 20 : View part of Spring MVC -CSS for the Sprouts vendor shown from Spring Tool Suite (STS)



```
ecovery needed
Tue May 27 10:39:26.067 [initandlisten] waiting for co
Tue May 27 10:39:26.067 [websvr] admin web console wai
rt 28017
Tue May 27 10:39:44.224 [initandlisten] connection acc
 #1 (1 connection now open)
Tue May 27 10:41:49.543 [FileAllocator] allocating new
s.ns, filling with zeroes...
Tue May 27 10:41:49.543 [FileAllocator] creating direc
Tue May 27 10:41:49.670 [FileAllocator] done allocatin
ts.ns, size: 16MB,  took 0.125 secs
Tue May 27 10:41:49.671 [FileAllocator] allocating new
s.0, filling with zeroes...
Tue May 27 10:41:49.844 [FileAllocator] done allocatin
ts.0, size: 64MB,  took 0.171 secs
Tue May 27 10:41:49.845 [FileAllocator] allocating new
s.1, filling with zeroes...
Tue May 27 10:41:49.846 [conn1] build index products.p
Tue May 27 10:41:49.848 [conn1] build index done.  scanned 0 total records. 0.00
1 secs
Tue May 27 10:41:49.848 [conn1] insert products.product ninserted:1 keyUpdates:0
 locks(micros) w:304797 304ms
Tue May 27 10:41:50.171 [FileAllocator] done allocating datafile \data\db\produc
ts.1, size: 128MB,  took 0.325 secs
```

```
C:\Users\Geetha>cd mongodb
The system cannot find the path specified.

C:\Users\Geetha>cd \

C:\>cd mongodb

C:\mongodb>mongo
MongoDB shell version: 2.4.5
connecting to: test
> use products
switched to db products
> show collections
> db.product.insert({ pid:"1", productname:"Basil", instock:"YES", price:"$12"})

> db.products.find().pretty()
> db.product.find().pretty()
{
        "_id" : ObjectId("5384ce5d949c4dc74a04b06f"),
        "pid" : "1",
        "productname" : "Basil",
        "instock" : "YES",
        "price" : "$12"
}
>
```

Figure 21: Installation ad usage of Individual Mongo Instance of the sprouts vendor containing products data.


Spring - gtvg/src/main/java/thymeleafexamples/gtvg/web/controller/UserProfileController.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Spring
Package Explorer
thymeleafexamples.gtvg.web.controller
AddController.java
AvailabilityController.java
HomeController.java
IGTVGController.java
LoginController.java
OrderDetailsController.java
OrderListController.java
ProductCommentsController.java
ProductListController.java
SafewayController.java
SproutsController.java
SubscribeController.java
TargetController.java
TraderJoeController.java
UserProfileController.java
VendorListController.java
VendorProductsController.java
WholeFoodsController.java
thymeleafexamples.gtvg.web.filter
src/main/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
Servers
Tomcat v7.0 Server at localhost [Stopped]
VMware vFabric tc Server Developer Edition v2.9 [Stop
Customer.java
Product.java
Order.java
*UserProfileController.java
Console
Writable
Smart Insert
100 : 1
10:53 AM
5/27/2014

```
  Query query = new Query();
  if ( StringUtils.hasText(criteria.getSku()))
  {
   Criteria c = Criteria.where('sku').is(criteria.getSku());
   query.addCriteria(c);
  }
  if (StringUtils.hasText(criteria.getTitle())) {
   Criteria c = Criteria.where('title').regex('.*' + criteria.getTitle() + '.*', 'i');
   query.addCriteria(c);
  }
  if (StringUtils.hasText(criteria.getDescription())) {
   Criteria c = Criteria.where('description').regex('.*' + criteria.getDescription() + '.*', 'i');
   query.addCriteria(c);
  }
  if (StringUtils.hasText(criteria.getProductType())) {
   Criteria c = Criteria.where('type').is(criteria.getProductType());
   query.addCriteria(c);

  }
  if (StringUtils.hasText(criteria.getTrack())) {
   Criteria c = Criteria.where('details.tracks').regex('.*' + criteria.getTrack() + '.*', 'i');
   query.addCriteria(c);
  }
  if (StringUtils.hasText(criteria.getChapter())) {
   Criteria c = Criteria.where('details.chapters').regex('.*' + criteria.getChapter() + '.*', 'i');
   query.addCriteria(c);
  }
  return mongoTemplate.find(query, Product.class);
 }

}
```

No consoles to display at this time.

Figure 22: Code snippet of mongo integration in STS.

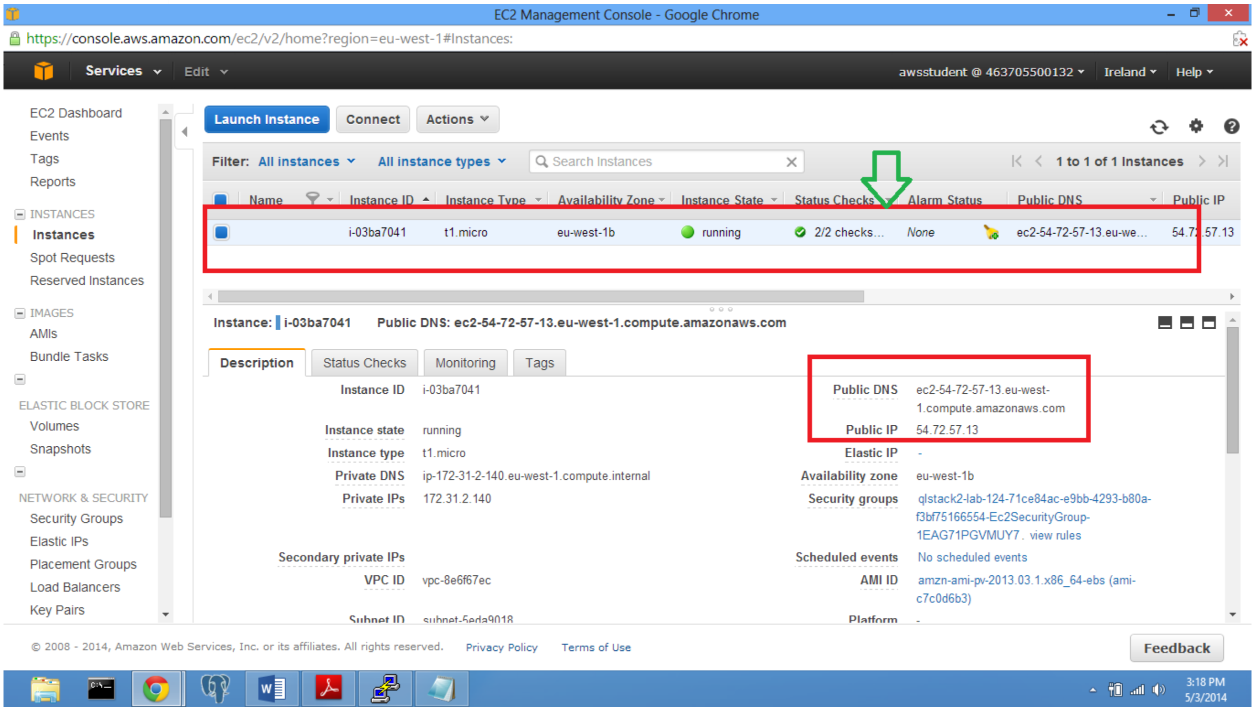
PROJECT REQUIREMENT 3: (IaaS and Paas CloutPlatform Information)

This part of the project was to deploy the application's any one component on one of the available Infrastructure as Service platforms and Platform As Service applications online. Virtual Grocery multitenant application has following technology components:

1. Data Base for individual and Shared : MONGODB
2. Iaas: Amazon AWS EC2 having the Mongo DB instance running on it. So it is acting as the Iaas.
3. Paas: Cloudbees Paas Platform.
4. Development: Spring MVC framework, STS, HTML, CSS(front end)

So after the development of the application is done using the model, view, and controller framework i.e. Spring MVC, the virtual grocery application's embedded .war file is created and uploaded on to the cloudbees PaaS application. This completes the deployment of the app on one of the Paas platforms. For data base purposes, project saved the documents on the Mongo DB which is hosted on AWS EC2 (IaaS).

SCREEN CAPTURES OF REQUIREMNET 3:



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
	i-03ba7041	t1.micro	eu-west-1b	running	2/2 checks...	None	ec2-54-72-57-13.eu-we...	54.72.57.13

Instance: i-03ba7041   Public DNS: ec2-54-72-57-13.eu-west-1.compute.amazonaws.com

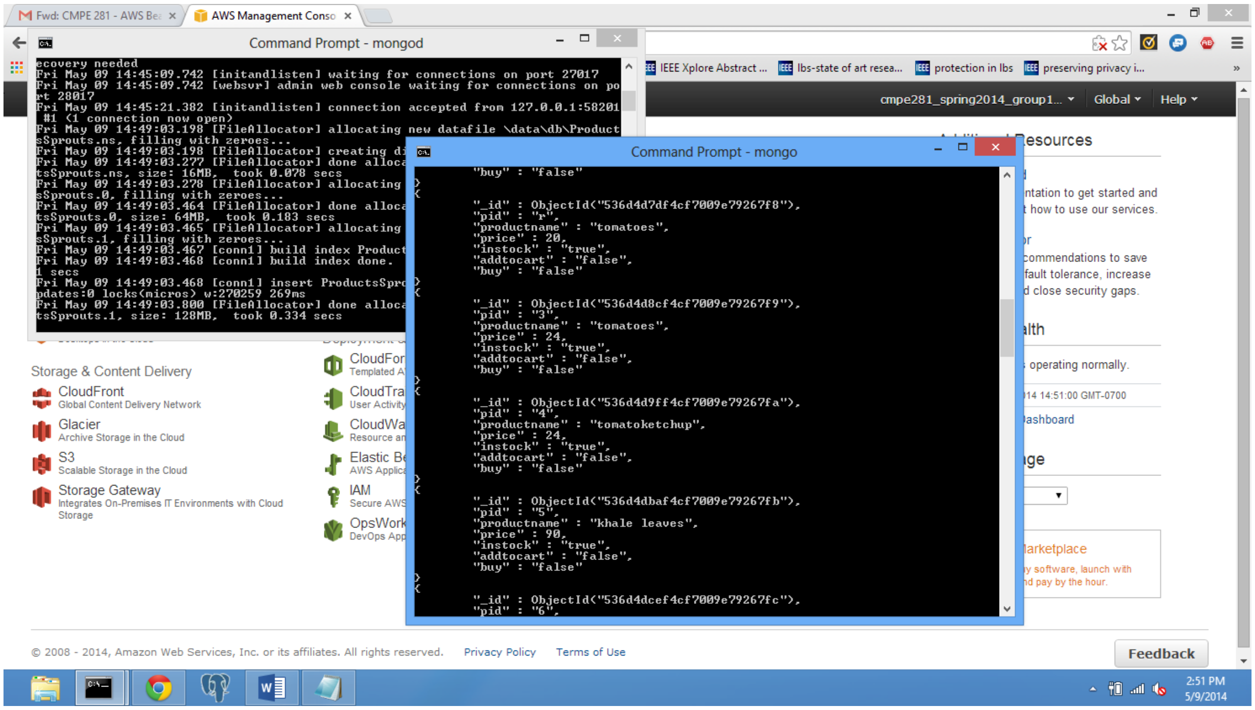
Description | Status Checks | Monitoring | Tags

Instance ID	i-03ba7041	Public DNS	ec2-54-72-57-13.eu-west-1.compute.amazonaws.com
Instance state	running	Public IP	54.72.57.13
Instance type	t1.micro	Elastic IP	-
Private DNS	ip-172-31-2-140.eu-west-1.compute.internal	Availability zone	eu-west-1b
Private IPs	172.31.2.140	Security groups	qlstack2-lab-124-71ce84ac-e9bb-4293-b80a-f3bf75166554-Ec2SecurityGroup-1EAG71PGVMUY7. view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-8e6f67ec	AMI ID	amzn-ami-pv-2013.03.1.x86_64-ebs (ami-c7c0d6b3)
Subnet ID	subnet-5eda9018	Platform	-

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

3:18 PM
5/3/2014

Figure 23: Figure showing a running instance having MYSQL on AWS EC2 on an Ubuntu VM



Command Prompt - mongod

```
ecovery needed
Fri May 09 14:45:09.742 [initandlisten] waiting for connections on port 27017
Fri May 09 14:45:09.742 [websvr] admin web console waiting for connections on po
rt 28017
Fri May 09 14:45:21.382 [initandlisten] connection accepted from 127.0.0.1:58201
 #1 (1 connection now open)
Fri May 09 14:49:03.198 [FileAllocator] allocating new datafile \data\db\Product
sSprouts.ns, filling with zeroes...
Fri May 09 14:49:03.198 [FileAllocator] creating di
Fri May 09 14:49:03.277 [FileAllocator] done alloca
tsSprouts.ns, size: 16MB,  took 0.078 secs
Fri May 09 14:49:03.278 [FileAllocator] allocating
sSprouts.0, filling with zeroes...
Fri May 09 14:49:03.464 [FileAllocator] done alloca
tsSprouts.0, size: 64MB,  took 0.183 secs
Fri May 09 14:49:03.465 [FileAllocator] allocating
sSprouts.1, filling with zeroes...
Fri May 09 14:49:03.467 [conn1] build index Product
Fri May 09 14:49:03.468 [conn1] build index done.
1 secs
Fri May 09 14:49:03.468 [conn1] insert ProductsSpro
pdates:0 locks(micros) w:270259 269ms
Fri May 09 14:49:03.800 [FileAllocator] done alloca
tsSprouts.1, size: 128MB,  took 0.334 secs
```

Command Prompt - mongo

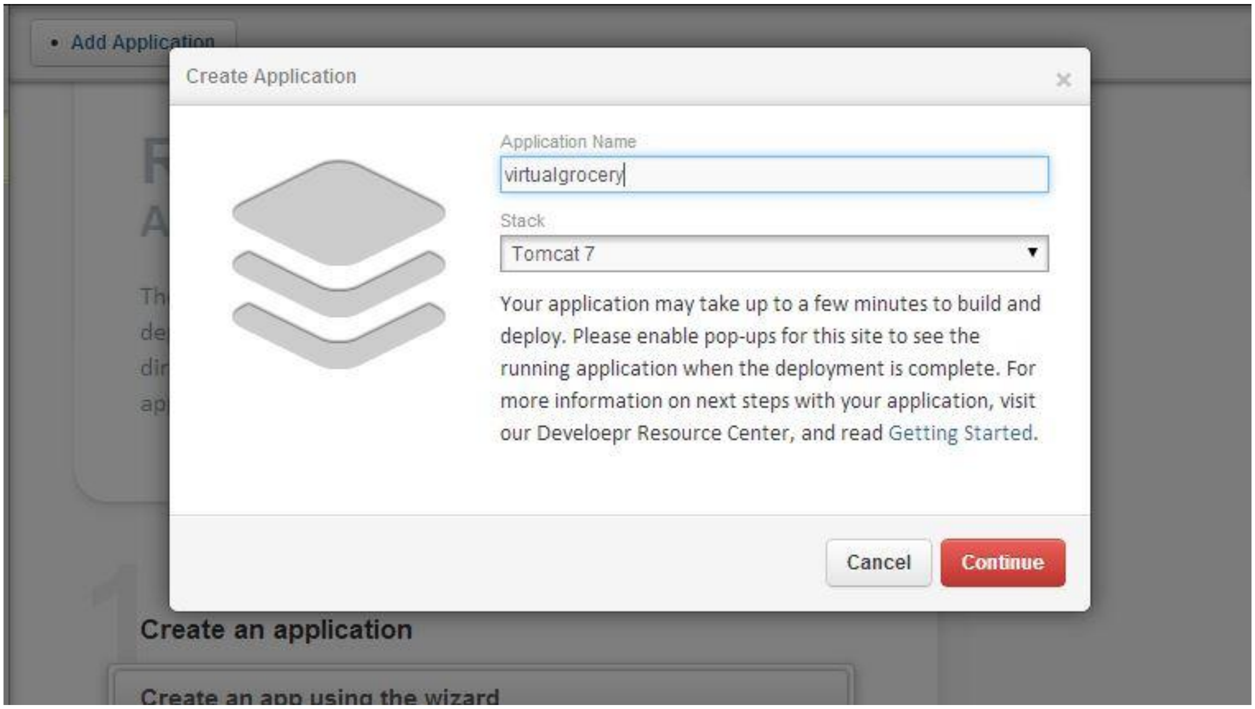
```
        "buy" : "false"
}
{
        "_id" : ObjectId("536d4d7df4cf7009e79267f8"),
        "pid" : "r",
        "productname" : "tomatoes",
        "price" : 20,
        "instock" : "true",
        "addtocart" : "false",
        "buy" : "false"
}
{
        "_id" : ObjectId("536d4d8cf4cf7009e79267f9"),
        "pid" : "3",
        "productname" : "tomatoes",
        "price" : 24,
        "instock" : "true",
        "addtocart" : "false",
        "buy" : "false"
}
{
        "_id" : ObjectId("536d4d9ff4cf7009e79267fa"),
        "pid" : "4",
        "productname" : "tomatoketchup",
        "price" : 24,
        "instock" : "true",
        "addtocart" : "false",
        "buy" : "false"
}
{
        "_id" : ObjectId("536d4dbaf4cf7009e79267fb"),
        "pid" : "5",
        "productname" : "khale leaves",
        "price" : 90,
        "instock" : "true",
        "addtocart" : "false",
        "buy" : "false"
}
{
        "_id" : ObjectId("536d4dcef4cf7009e79267fc"),
        "pid" : "6",
```

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

2:51 PM
5/9/2014

Figure 24: Figure showing a running instance having MYSQL on AWS EC2 on a Ubuntu VM

Platform as Service : Cloud bees



Create Application

Application Name

virtualgrocery

Stack

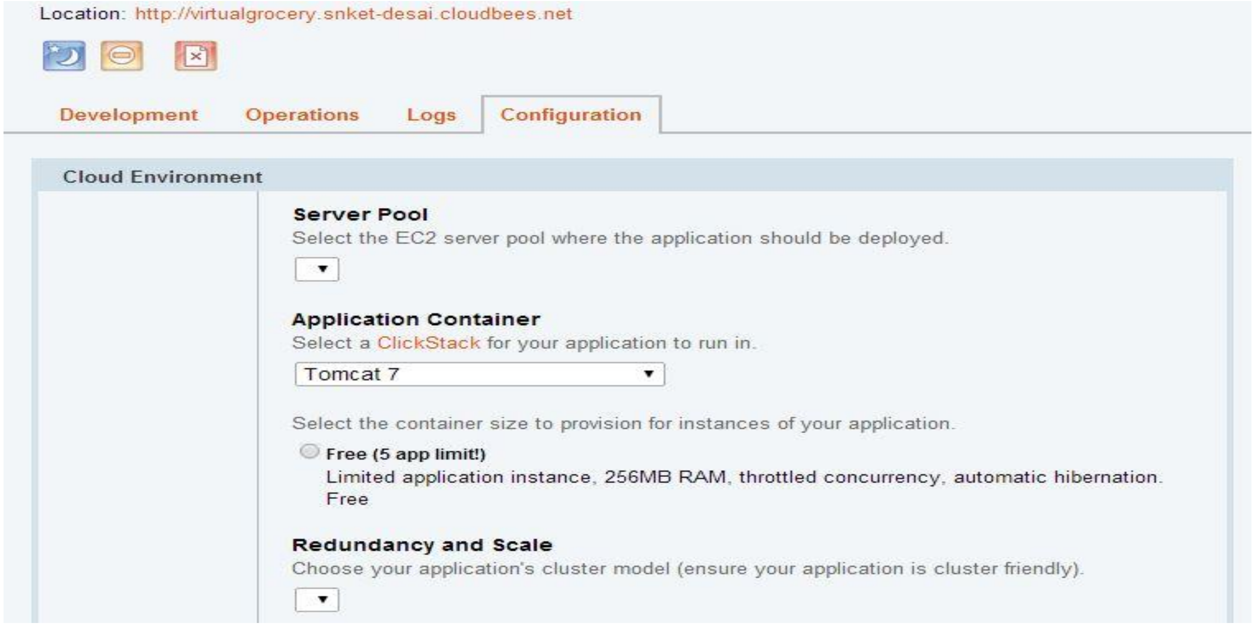
Tomcat 7

Your application may take up to a few minutes to build and deploy. Please enable pop-ups for this site to see the running application when the deployment is complete. For more information on next steps with your application, visit our Develoepr Resource Center, and read Getting Started.

Create an application

Create an app using the wizard

Figure 25: Paas Application being deployed on Cloud bees



Location: http://virtualgrocery.snket-desai.cloudbees.net

Cloud Environment

Server Pool
Select the EC2 server pool where the application should be deployed.

Application Container
Select a ClickStack for your application to run in.

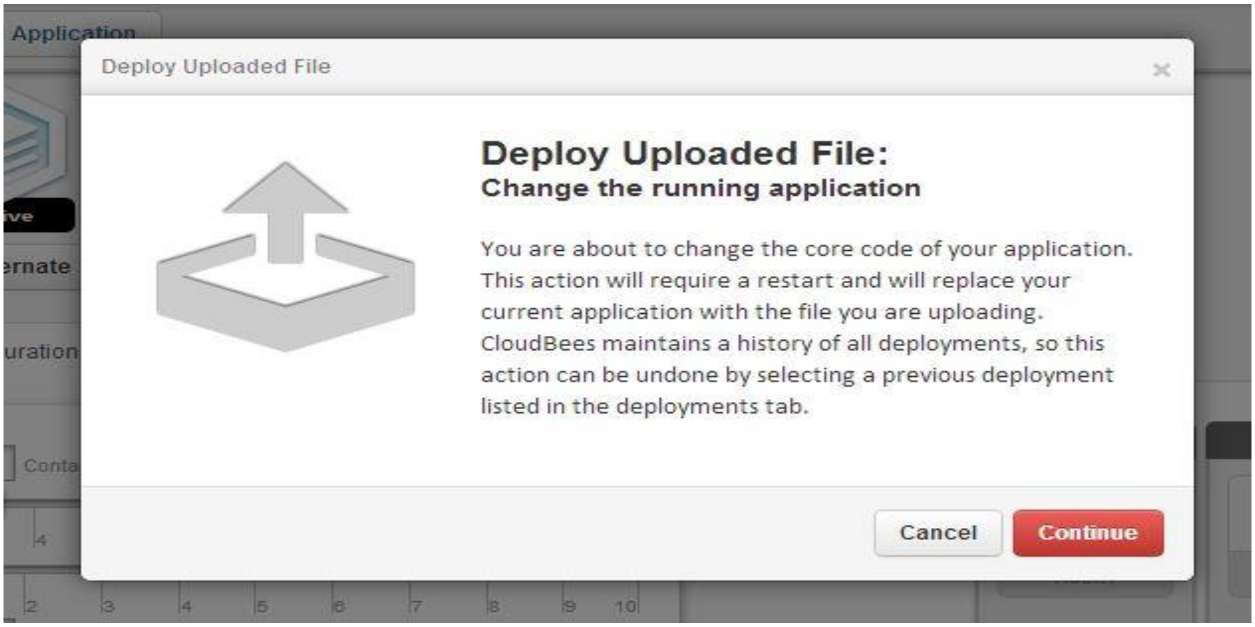
Tomcat 7

Select the container size to provision for instances of your application.

○ **Free (5 app limit!)**
Limited application instance, 256MB RAM, throttled concurrency, automatic hibernation.
Free

Redundancy and Scale
Choose your application's cluster model (ensure your application is cluster friendly).

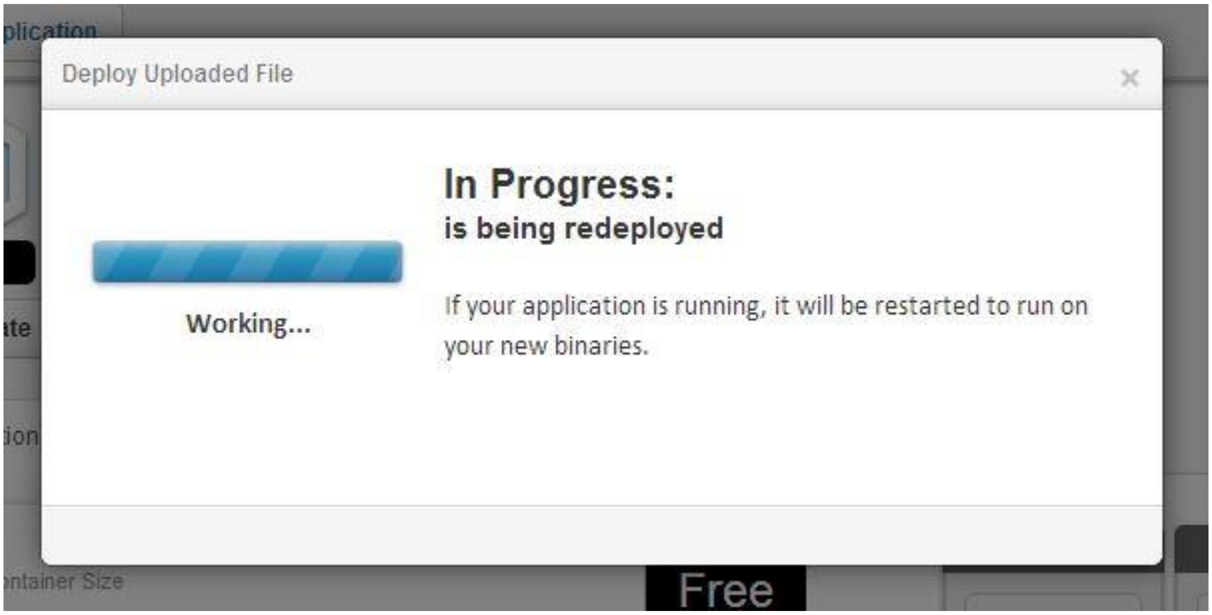
Figure 26: Paas: App on Cloudbees- Server selection



Deploy Uploaded File:
Change the running application

You are about to change the core code of your application. This action will require a restart and will replace your current application with the file you are uploading. CloudBees maintains a history of all deployments, so this action can be undone by selecting a previous deployment listed in the deployments tab.

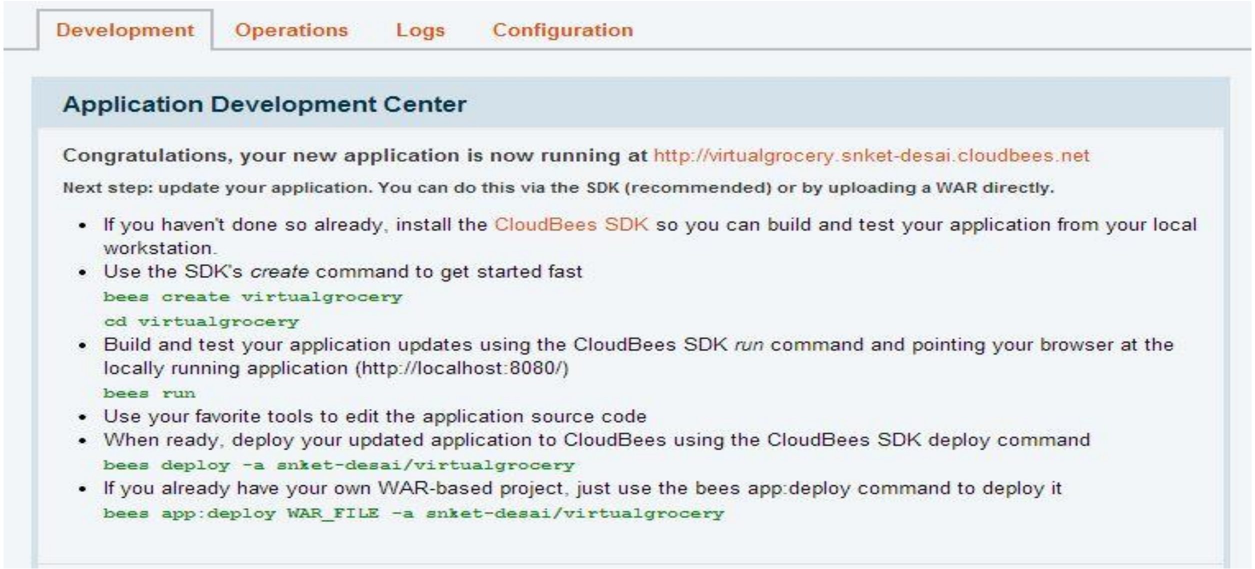
Figure 27: Paas: App on Cloudbees- Uploaded



In Progress:
is being redeployed

If your application is running, it will be restarted to run on your new binaries.

Figure 28: Deployment in progress on cloudbees



Application Development Center

Congratulations, your new application is now running at http://virtualgrocery.snket-desai.cloudbees.net

Next step: update your application. You can do this via the SDK (recommended) or by uploading a WAR directly.

- If you haven't done so already, install the CloudBees SDK so you can build and test your application from your local workstation.
- Use the SDK's *create* command to get started fast

```
bees create virtualgrocery
cd virtualgrocery
```

- Build and test your application updates using the CloudBees SDK *run* command and pointing your browser at the locally running application (http://localhost:8080/)

```
bees run
```

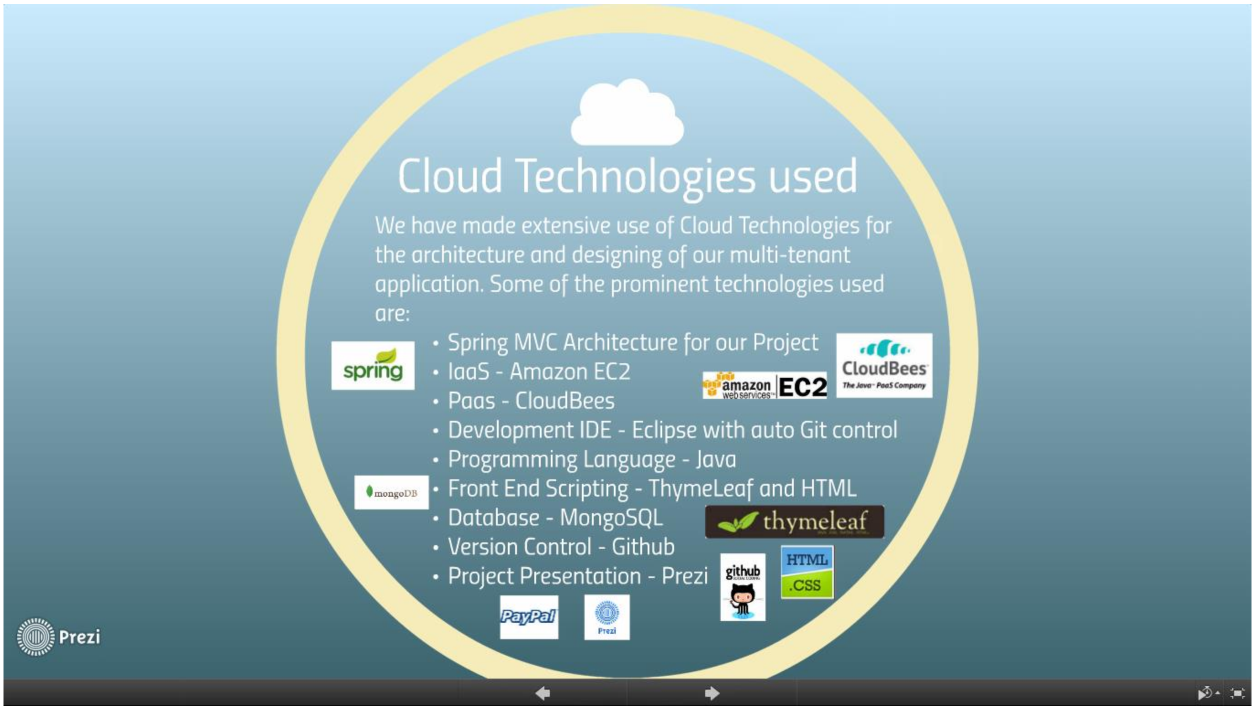
- Use your favorite tools to edit the application source code
- When ready, deploy your updated application to CloudBees using the CloudBees SDK deploy command

```
bees deploy -a snket-desai/virtualgrocery
```

- If you already have your own WAR-based project, just use the bees app:deploy command to deploy it

```
bees app:deploy WAR_FILE -a snket-desai/virtualgrocery
```

Figure 29: A running instance of the application deployed on Cloud bees (PaaS)



Cloud Technologies used

We have made extensive use of Cloud Technologies for the architecture and designing of our multi-tenant application. Some of the prominent technologies used are:

- Spring MVC Architecture for our Project
- IaaS - Amazon EC2
- Paas - CloudBees
- Development IDE - Eclipse with auto Git control
- Programming Language - Java
- Front End Scripting - ThymeLeaf and HTML
- Database - MongoSQL
- Version Control - Github
- Project Presentation - Prezi