Coupon System 27/07/2017

Part 2- Client side

**822.96**

**Handed in by:**

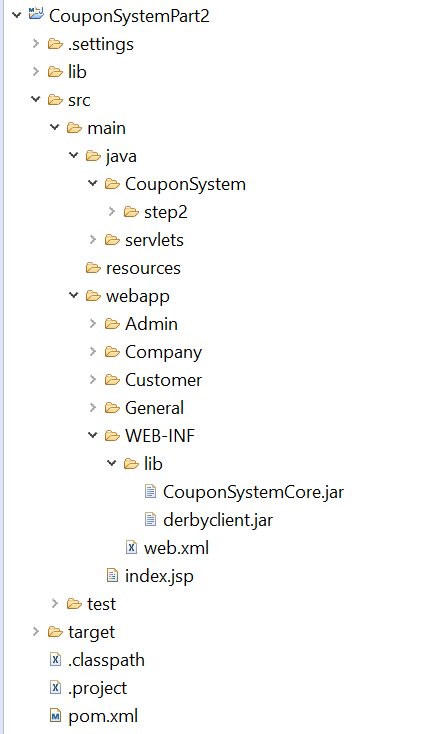
Alon Ben David 036997997

Batel Kanarek 200779635

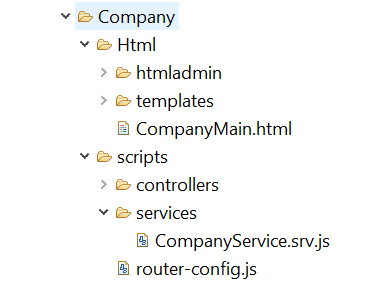
Sagi Mauda 066370628

**Project Overview:**

**This project represents a Coupon System in which companies offer coupons for their products or services. Customers can then purchase Coupons issued by different companies.**



Each user type has its own folder containing the following directories:



**Setup instructions:**

* The database folder should be located in ‘C:\’
* The log files can be found at “\eclipse-jee-mars-2-win32-x86\_64\eclipse”
* If it important to make sure that the BuiltPath to the Archive files is set up correctly.

**Web application Components:**

|  |  |  |
| --- | --- | --- |
| **Java Resources** | **WebApp Resources** | **Archives** |
| 3 Web APIs | html | CouponSystemCore.Jar |
| 2 Servlets | Angular controllers | servlet-api.jar |
| Java logger | Angular services |  |
|  | Javascript pages |  |
|  | bootstrap |  |
|  | CSS files |  |
|  | Ui router |  |
|  | images |  |

The project consists of the following components:

**Java Resources:**

* web services

There are 3 webservices – one for each user Façade:

1. AdminService
2. CustomerService
3. CompanyService

The web API receives requests and returns a response according to the response from the coupon system core.

*getFacadeFromSession*- an internal method which checks if the user has a session with a façade attribute set to it and returns it. If there is no session the method will return a response wrapped in an exception with status 401(“unauthorized”).Every method In the webAPI will call this method before performing any action, this will ensure that only logged in users can perform actions.

* CSLogger class-Java Logger with a fileHandler and a simpleFormatter in a singleton design pattern. This logger is embedded into the web API and logs every entry , exit and exception in every method.
* Servlets:

1. Login Servlet- receives user credentials from login.html form. Performs login Performs login, sets Facade on session and redirects to corresponding user page. If client check the “remember me” box, cookies containing login details will be planted in the response
2. Logout Servlet- invalidates the user session, removes the login cookies and redirects to login page.

**WebApp Resources:**

The web resources include html pages, Angular Controllers and Services, javaScript functions and design using CSS and Bootstrap.

1. Html pages:

* login.Html- the first page in the website which will then route users who successfully logged in to one of the following user pages:
* AdminMain.html
* CustomerMain.html
* Company Main.html
* Html Templates- the inner pages which are inserted into the “UI-VIEW” section of the main page. There are different templates per user and per method:

|  |  |  |  |
| --- | --- | --- | --- |
| **Html Templates** | | | |
| **Admin** | | **Customer** | **Company** |
| createCompany.html | createCustomer.html | PurchaseCoupon.html | createCoupon.html |
| removeCompany.html | removeCustomer.html | GetAllPurchasedCoupons.html | removeCoupon.html |
| updateCompany.html | updateCustomer.html |  | updateCoupon.html |
| getCompany.html | getCustomer.html |  | getCompany.html |
| getAllCompanies.html | getAllCustomers.html |  | GetCoupons.html |
| CreateCompany.html | createCustomer.html |  |  |

1. Angular controllers- most controllers contain one method, some contain several methods which refer to the same resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Angular Controllers** | | | |
| **Admin** | | **Customer** | **Company** |
| createCompany.ctrl.js | createCustomer.ctrl.js | PurchaseCoupon .ctrl.js | createCoupon.ctrl.js |
| removeCompany.ctrl.js | removeCustomer.ctrl.js | GetAllPurchasedCoupons.ctrl.js | removeCoupon.ctrl.js |
| updateCompany.ctrl.js | updateCustomer.ctrl.js |  | updateCoupon.ctrl.js |
| getCompany.ctrl.js | getCustomer.ctrl.js |  | getCompany.ctrl.js |
| getAllCompanies.ctrl.js | getAllCustomers.ctrl.js |  | GetCoupons.ctrl.js |
| CreateCompany.ctrl.js | createCustomer.ctrl.js |  |  |

1. Angular Services:

* AdminService.srv.js
* CustomerService.srv.js
* CompanyService.srv.js
* NgConfirmAlerts.js- general file pertaining to all user types. contains the ng-confirm pop up for the unauthorized users which redirects users to the login back. This service is injected into each controller.

1. Javascript pages

* Router-config.js- a file which defines the url state and associated html template and controller. There are three such files- one for each user type.
* LoginRouter.js- a javascript file with ‘checkForCookies’ function which runs as soon as the login page is loaded (event- onload). Checks if user has cookies and if so- uses them to fill out the login form automatically. enables ***Auto Login.***

1. Bootstrap- all of the bootstrap information, files CSS,SCSS images Etc. are stored in a folder called “htmladmin”. There is such a folder for each user type and each user page has the bootstrap embedded.
2. CSS files:

There are 3 files under the ‘general’ folder-as they pertain to all users and control the design of all Tables forms and ng-alerts throughout the web application.

* tablestyle.css
* formStyle.css
* angular-confirm.min.css

Archives (lib directory):

* CouponSystemCore.Jar- the Java project which was already handed in.
* servlet-api.jar

**Additional Features for extra credit: (beyond basic requirements):**

1. Cookies- auto Login
2. Response
3. Bootstrap
4. Carousel Plugin
5. Java Logger
6. Use of W3 Schools tags
7. HTML validation on all forms
8. Use of CSS3