**Project 3: Enhanced Mortgage Calculator (EMC)**

**Description:** EMC is a calculator to help user to calculator loans. Each user needs to log in the system and input the necessary information. Then EMC will output the monthly payment with the payment schedule. The user can also add additional principals for each month and EMC will calculate how much the user can save for interest.

**Reference Link:** <https://www8.bankofamerica.com/home-loans/mortgage-purchase/closing-cost-calculator.go>

**Responsibilities:**

* Register: A new user can register for a new account. Validation should be applied and a welcome email should be sent automatically.
* Login: The user can log in the system with valid username and password.
* GUI: Create an interface similar to the above link. The user can input necessary information.
* Mortgage Calculator: Provide a web service to calculate the mortgage with monthly payment and payment schedule. It also gives a summary of the total interest paid. The interest rate for each term should be retrieved from database.
* Enhanced feature: Add two input fields. One is the additional principal for each month and the other is the number of months to pay the additional principal. Based on the additional inputs, the mortgage calculator will output the updated monthly payment and payment scheduled, along with the amount of interest that the user can save.
* Display results: Use table to display the payment schedule and use charting to display the interest distribution through the whole term (google chart).
* Log off. The user can log off the system and his/her session will be expired.
* Admin account: There is an admin account for the administrator of the system. The admin can login to a different page. As an admin, he/she can change the annual interest rate for each type of loans.

**Implementation (The following technologies should be applied):**

* Core Java
* Spring Security, Spring MVC or Struts 2 MVC, Spring IoC, Spring Dao Support, Spring Cache
* Hibernate
* REST or SOAP web service
* HTML5, CSS3, JavaScript, jQuery, AJAX, AngularJS, Bootstrap
* Responsive Design
* Validation(front-end, jQuery/AngularJS)
* Oracle 10g
* GitHub
* Trello (<https://trello.com/>) project management tool -> JIRA
* JRebel (reloads changes to Java classes)
* MockRunner

**Issue**

* Precision
* Cache -> memory overflow
* Evict cache after admin updated the rate (so, once the user recalculate the same plan, it will use the latest rate)
* Verify user’s email (sign up) by using web service

**UI Design**

Main page (sign in, sign up; sign out, forget password)

Tags: Calculator, Result, Rate&Loan, Chart, \*Admin

Sign up page

Update Account page