# Overall Design

This is a very lightweight solution or framework. The frontend is an AngularJS SPA application which calls REST services from the backend to perform all the transactions.

Authentication is through a Java servlet filter (AuthFilter.java). All the configurations have been simplified using Guice (GuiceConfig.java).

The features are designed to be modular and exposed as reusable services.

# Technologies

|  |  |
| --- | --- |
| Frontend | AngularJS, Bootstrap, Angular UI, Font Awesome, LESS |
| Backend | Guice, GAE, Guava, Jackson, Lombok, Objectify, JavaX-WS-RS, Jersey |

# Backend

Once the users logged in to the site, all the services are exposed through REST.

Refer to the package “com.crossover.action”

|  |  |
| --- | --- |
| Class Name | Description |
| AccountService.java | REST services to retrieve accounts and transaction details. |
| AdminService.java | REST services to retrieve all accounts and transaction details. |

User permission is checked to prevent unauthorized access. A service code will be returned to indicate if the operation is successful, unauthorized or failed (Refer to ServiceCode.java)

# Frontend

The frontend is developed as an AngularJS application.

Refer to the “src” folder for the source code. Under this folder you can the pages for “admin”, “myspace” and “home”

|  |  |
| --- | --- |
| File Extension | Description |
| .js | Contains the processing logic. |
| .tpl.html | HTML template |
| .spec.js | Karma unit testing |
| .less | Style sheet |

Under “src\common\resources” folder you can find the AngularJS modules which are used to invoke the backend REST services.

|  |  |
| --- | --- |
| Module Name | Description |
| account.js | Module to invoke the account REST services. |
| admin.js | Module to invoke the admin REST services. |