Se452 Group 2

Rent All

-Benjamin Conrad

-Xin Guo

-Steve Nickerson

**GITHUB LINK:** [**https://github.com/GaryGuo007/SE452Group2**](https://github.com/GaryGuo007/SE452Group2)

Overview:

-Rent all is an application meant to mimic something similar to Angie’s List such that users can search and find information on contractor/maintenance workers.

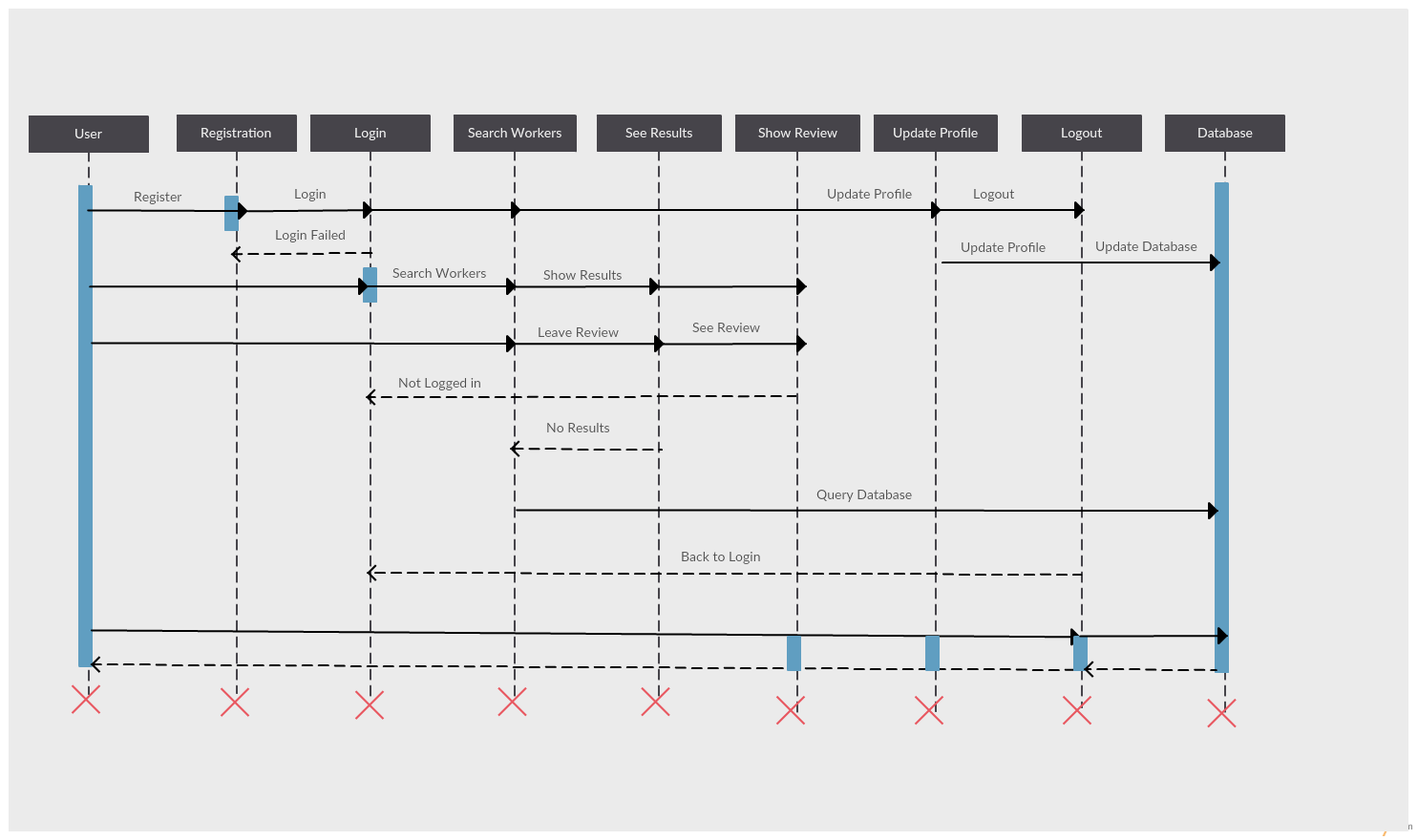
-The site should allow the user to create a profile and login. In addition, the site should allow a contractor to create a profile which is in turn added to the database.

Frameworks/Programs

* Spring MVC:
  + Used to take form data and using a controller, handle the data either to display on the frontend JSP pages, or add to the database via the application layer.
  + Uses request mapping to parse get requests and redirects from the frontend pages.
  + Create dynamic pages based on search parameters and session variables.
* MongoDB:
  + NoSQL database which is used to store data.
  + Running locally for the current project but persistence can be updated to add a cloud based option.
  + For this project, transactional integrity isn’t as important as storing lots of data which isn’t necessarily connected to each other.
* Hibernate:
  + Java framework that uses object relational mapping to provide a persistence layer to encapsulate the data.
  + Works in conjunction with Spring MVC to create a JPA layer which can host a multitude of different databases such as MongoDB or Neo4j.
* Maven:
  + All of this Is built as a maven project and dependencies must be added for each section in the POM.xml file.
* Bootstrap:
  + Front end web framework for easy templating and responsiveness.
  + Allows for basic form validation on most inputs.
  + Easy grid setup for responsive design on any device.

Functionality:

Sequence Diagram:



Usage:

No login:

-User starts search. Parameters are entered into form. Controller grabs form variables and a search object is created which is then queried from the database.

-User receives a list of results based on query. Each result is a clickable link which reloads the results page showing only the information of the reviewer. This is retrieved with Request. Get by passing an UID in the URL.

-Leaving a review will redirect back to the login screen using Springs redirect functionality.

Login:

-User registers and a database customer/contractor object is created which is then passed along to the database and customer is created.

-Login validates but query database for said customer. If validation succeeds a session variable will be creating using Springs Scope annotation. Information about the customer will be passed to the session variable.

-Search proceeds as highlighted before.

-If a review is created, session is validated for login, and the customer name is added to a review object. The review object then is added to the database with customer id and contractor id.

-The results page displays all reviews for selected contractor using Springs Expression Language.

-Update and Delete of customer information both contact the database via a contractor/worker service which provides a façade for the methods.

-Logout invalidates the session variable.

**Notes: Using Spring 4.3.1 and Apache Tomcat 8.0.3.**

**GITHUB LINK:** [**https://github.com/GaryGuo007/SE452Group2**](https://github.com/GaryGuo007/SE452Group2)