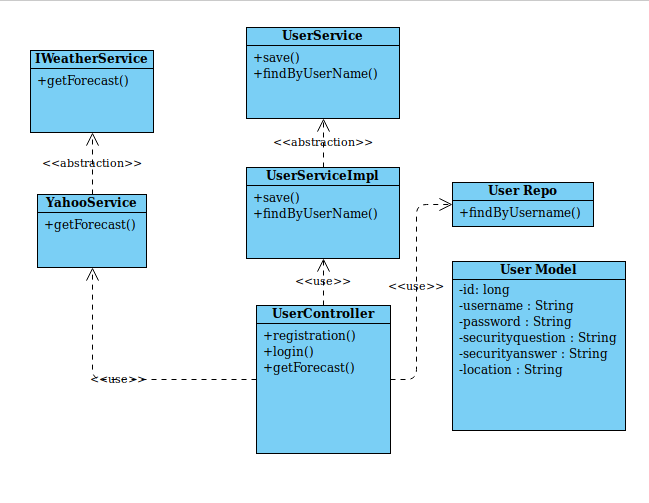
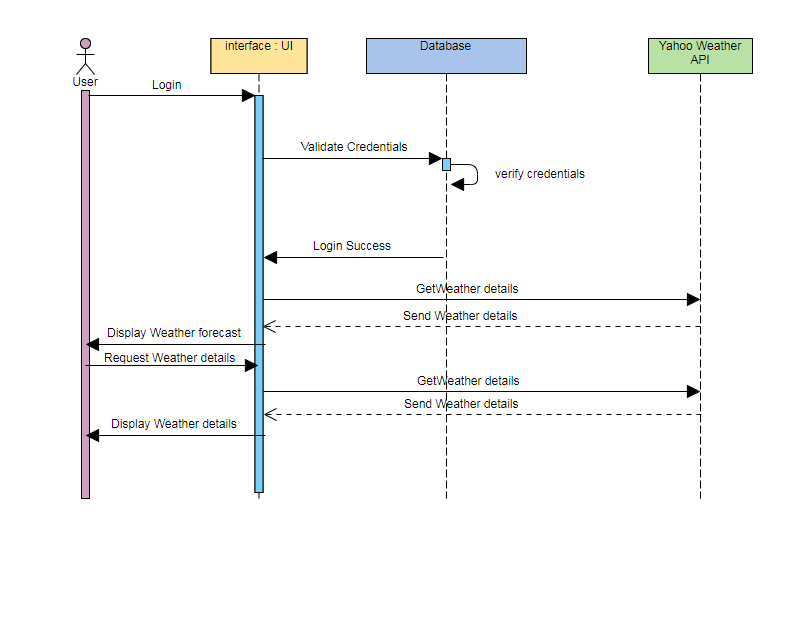
Weather Forecast

Class Diagram



Sequence Diagram



This is a simple application to give Weather forecast details for Places from around the world.

* Initially user has to login to use the application.
* On the landing page the user needs to login. Incase he does not have an account he needs to register.
* Registering a user takes from user username ,password ,location , and a security question and its answer.
* After registering, the user is redirected to welcome page.
* It by default displays forecast for the location specified by user during registration.
* The user will be able to check weather forecast for locations around the world.
* He needs to select if the temperature needs to be displayed in degree or Fahrenheit.
* Also, the forecast for coming # days needs to be specified by user.
* Weather forecast details include
  + max/min temp in Celsius by default(user can change it to Fahrenheit )
  + One word to describe weather(clear,cloudy,rain..etc)
* An external API(Yahoo Weather Java API ) has been integrated to get weather forecast details for various places.
* A logout button to logout from the application at any time after which the user will not be able to check weather forecast for other locations.

**Development steps.**

Ensure that java is installed on the machine. Followed by maven. The MySql database is used to store details of user and for login. Tomcat is used to deploy the war file. Alternatively the project can be imported in Spring Tool Suite as an existing Maven project and can be run in its embedded Tomcat.

Jenkins has been configured for continuous integration and deployment(to a tomcat container). SonarQube has been used for static code analysis.

Prerequisites:

* Java
* Maven
* Spring Tool Suite
* MySql
* Tomcat
* Jenkins
* SonarQube

Apache Maven has been chosen as the build and dependency management tool . The following dependencies have been used.

To start a web service project, developers just need to create a Maven project with the “Create a simple project (skip archetype selection)” option. Then, edit the *pom.xml* to insert the following xml snippet under the root (project) node,

spring-boot-starter-parent -> For creating a spring boot project this dependency is required in parent tag

spring-boot-starter-web -> To create a web project

spring-boot-starter-security -> To enable authority based login

spring-boot-starter-data-jpa -> For integrating any database

mysql-connector-java -> To connect to mySql

spring-boot-starter-tomcat -> For deploying the app on tomcat server

tomcat-embed-jasper -> Embed that in the application

jstl -> Template to display jsp pages

spring-boot-starter-test -> For junit testing

yahoo-weather-java-api -> External API for weather forecast details

Deployment

The war file can be deployed on tomcat . To generate the war file

Do a *mvn install* in the project location

And place this file in webapps folder to deploy it on default port

Alternatively the project can be imported in Spring Tool Suite as an existing Maven project and can be run in its embedded Tomcat (by running it as a spring boot app).

*mvn spring-boot:run*

The application will be available on

* http://localhost:{ default tomcat port}/

the user can register or signin with his credentials