<Assignment A4>
Analysis and Design Document

Student:Budusan Ciprian
Group:30434

Table of Contents

1. Requirements Analysis 3
    1.1 Assignment Specification 3
    1.2 Functional Requirements 3
    1.3 Non-functional Requirements 3
2. Use-Case Model 3
3. System Architectural Design 3
4. UML Sequence Diagrams 5
5. Class Design 5
6. Data Model 6
7. System Testing 6
8. Bibliography 6

1. Requirements Analysis

1.1 Assignment Specification

Use appropriate Java/C# classes for Drawing and Drag & Dropping to design and implement an application similar with Paint. The application should implement the following actions:

(1) draw complex shapes using simple shapes and
(2) drag and drop simple shapes from a toolbar.

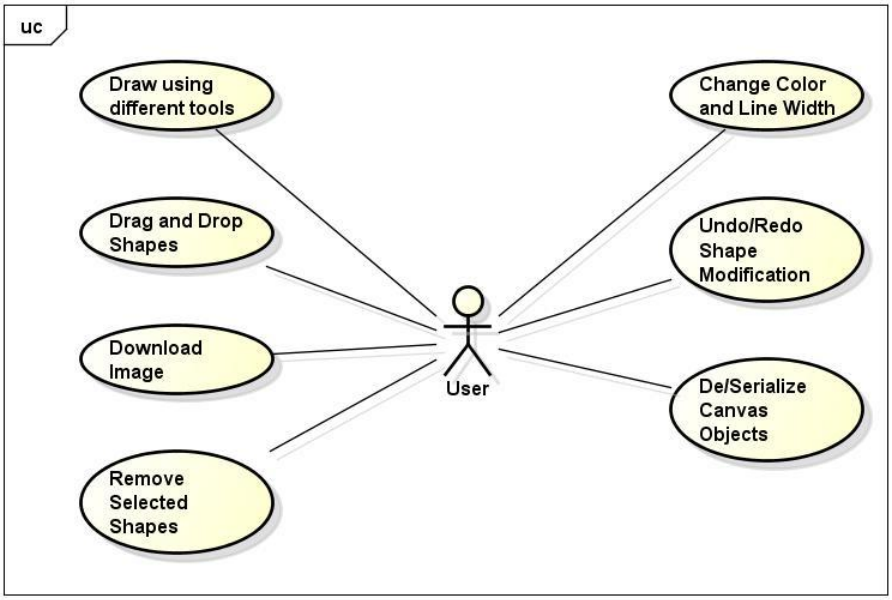
1.2 Functional Requirements

- draw complex shapes using simple shapes;
- drag and drop simple shapes from a toolbar;

1.3 Non-functional Requirements

- Performance: response time immediately;
- Availability: all the time;

2. Use-Case Model



Use case: Draw Shapes
Level: user-goal level
Primary actor: user
Main success scenario:

1. User clicks on Draw button and then clicks on a desired tool for drawing.
2. The user presses the mouse button and releases it on the canvas in order to

draw the shape just as in MS Paint.
3. The shape is drawn and the further drawings can be performed.
4. If he wants to save the image the user must click on Download Image.
5. Also if he wants to close the app and continue work later he can save the canvas which contains the object properties such as position, color.

Extensions: 5a. The browser does not support HTML5.
-5a1. He will not see the canvas and cannot draw.

3. System Architectural Design

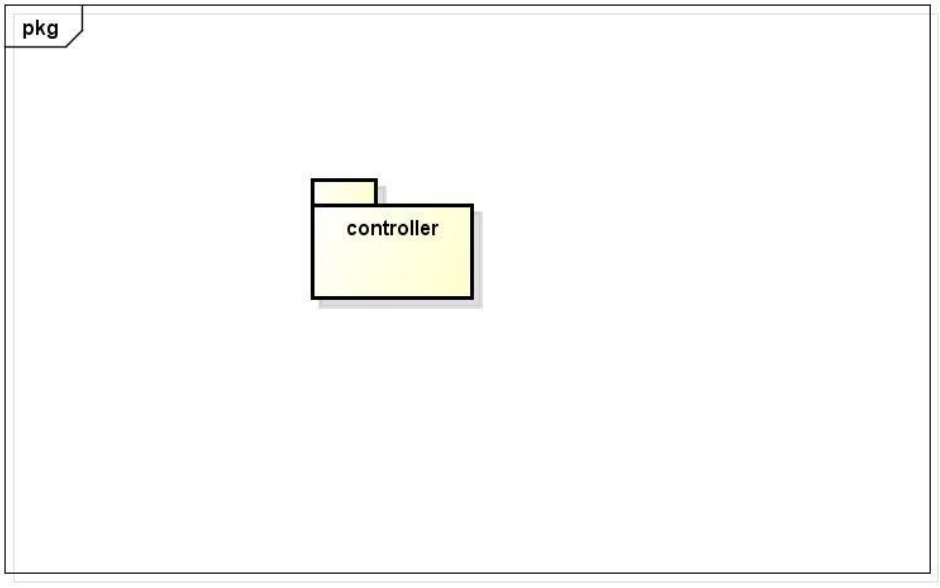
3.1 Architectural Pattern Description

Spring MVC is used to ensure the Model-View-Controller pattern, but the project contains only one controller class for displaying a single page, which contains buttons and a <canvas> element. The rest of the logic is done in a javascript file.

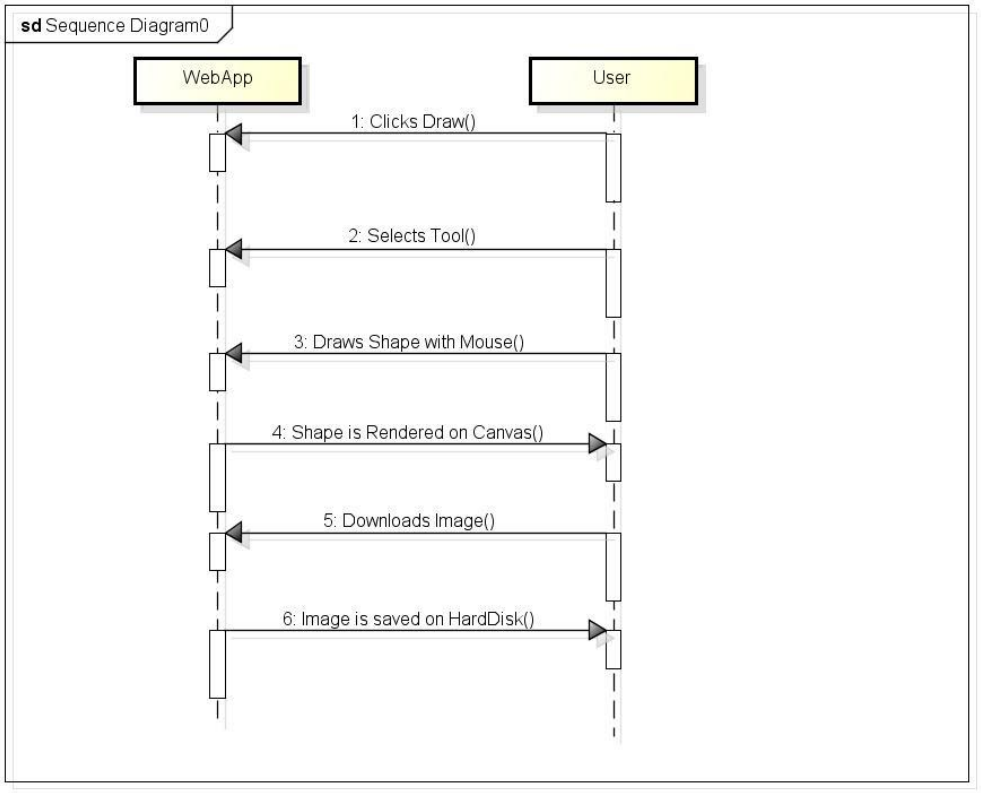
For the front-end part, view level, JSP files are used with javascript for canvas drawing and Bootstrap theme. Fabric JS library is also used to have <canvas> objects serialized in JSON format.

3.2 Diagrams

Package Diagram:



4. UML Sequence Diagrams



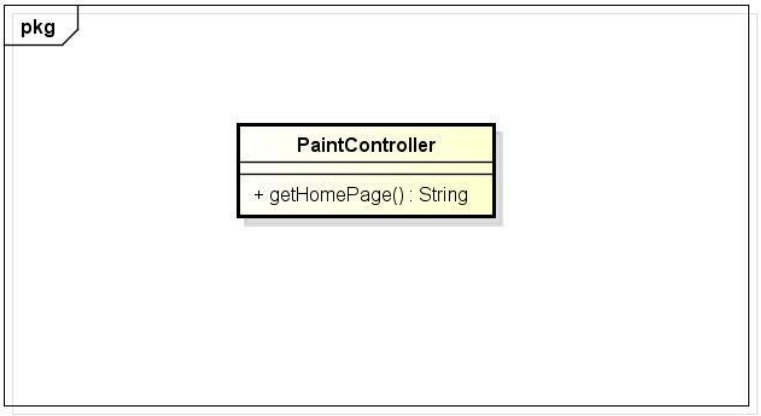
5. Class Design

5.1 Design Patterns Description

Spring MVC uses the following patterns:

- Dependency injection/ or IoC (inversion of control) – Is the main principle behind decoupling process that Spring does;
- Factory – Spring uses factory pattern to create objects of beans using Application Context reference;
- Proxy – used heavily in AOP, and remoting;
- Singleton – by default, beans defined in spring config file (xml) are only created once. No matter how many calls were made using getBean() method, it will always have only one bean. This is because, by default all beans in spring are singletons.

5.2 UML Class Diagram



6. Data Model

7. System Testing

The Spring MVC Test framework provides first class JUnit support for testing client and server-side Spring MVC code through a fluent API. Typically it loads the actual Spring configuration through the TestContext framework and always uses the DispatcherServlet to process requests thus approximating full integration tests without requiring a running Servlet container.

Client-side tests are RestTemplate-based and allow tests for code that relies on the RestTemplate without requiring a running server to respond to the requests.

I have also used console.log(message) to debugg the application by pressing F12 in Chrome Browser and viewing the message displayed from the javascript.

8. Bibliography

http://fabricjs.com/fabric-intro-part-1/
http://www.w3schools.com/bootstrap/
http://stackoverflow.com/questions/11829786/delete-multiple-objects-at-once-on-a-fabric-js-canvas-in-html5