|  |
| --- |
| CAT Plug-in Project Documentation |
| SWE 574 - Fall 2016 Instructor: Dr. Suzan Üsküdarlı |
| CAT Plug-in Team: Arda Ofluoğlu, Okan Menevşeoğlu, Şenol Kanca, Şeyda Karadağ, Taner Yiğit, Tolga Caner |

# Table of Contents

[1 Table of Contents 2](#_Toc470500498)

[2 Introduction 5](#_Toc470500499)

[3 Requirements 5](#_Toc470500500)

[3.1 Authentication 5](#_Toc470500501)

[3.1.1 Sign in using credentials 5](#_Toc470500502)

[3.1.2 Sign up with e-mail, username, and password 5](#_Toc470500503)

[3.1.3 Maintain email uniqueness 5](#_Toc470500504)

[3.1.4 Have member and admin user types 5](#_Toc470500505)

[3.1.5 Change Password 6](#_Toc470500506)

[3.2 Authentication 6](#_Toc470500507)

[3.2.1 Make private and public annotations 6](#_Toc470500508)

[3.2.2 Annotate HTML pages and texts 6](#_Toc470500509)

[3.2.3 Don't include HTML tag in annotations 6](#_Toc470500510)

[3.2.4 Annotate images 6](#_Toc470500511)

[3.2.5 Only annotate the source of img tags 6](#_Toc470500512)

[3.2.6 Highlight annotated texts 6](#_Toc470500513)

[3.2.7 Manage annotations 6](#_Toc470500514)

[3.2.8 Search and delete annotations as admin 6](#_Toc470500515)

[3.2.9 Delete other members' annotations 7](#_Toc470500516)

[3.2.10 View annotations of a page 7](#_Toc470500517)

[3.2.11 View annotations of a target 7](#_Toc470500518)

[3.2.12 Filter annotations 7](#_Toc470500519)

[3.2.13 Handle content changes 7](#_Toc470500520)

[3.2.14 Won't Have Requirements 7](#_Toc470500521)

[4 Mock-ups 8](#_Toc470500522)

[4.1 View for an Anonymous User 8](#_Toc470500523)

[4.2 Sign Up Screen 9](#_Toc470500524)

[4.3 Sign In Screen 9](#_Toc470500525)

[4.4 After Sign In 10](#_Toc470500526)

[4.5 Text Annotation 10](#_Toc470500527)

[4.6 Image Annotation 11](#_Toc470500528)

[4.7 Change Password Screen 11](#_Toc470500529)

[4.8 Manage Annotations 12](#_Toc470500530)

[5 Project Plan 12](#_Toc470500531)

[6 UML Design Diagrams 13](#_Toc470500532)

[6.1 Use Case Diagram 13](#_Toc470500533)

[6.1.1 Use Cases for Admin, User, and Non-User 13](#_Toc470500534)

[6.2 Activity Diagrams 14](#_Toc470500535)

[6.2.1 Create Annotation Activity Diagram 15](#_Toc470500536)

[6.2.2 View Annotation Activity Diagram 15](#_Toc470500537)

[6.3 Sequence Diagrams 16](#_Toc470500538)

[6.3.1 Create Annotation Sequence Diagram 16](#_Toc470500539)

[6.3.2 View Annotation Sequence Diagram 17](#_Toc470500540)

[6.4 Class Diagram 17](#_Toc470500541)

[7 High-Level Architecture 19](#_Toc470500542)

[8 Features and User Guide 20](#_Toc470500543)

[9 Server Configuration 22](#_Toc470500544)

[9.1 Java 22](#_Toc470500545)

[9.2 IntelliJ or Spring Tool Suite 23](#_Toc470500546)

[9.3 Tomcat 23](#_Toc470500547)

[9.4 MySQL 23](#_Toc470500548)

[9.5 Security 23](#_Toc470500549)

[10 Test Cases 24](#_Toc470500550)

# Introduction

This document provides information about our web annotation plug-in, CAT, for Mozilla Firefox web browser, as a SWE574 course project in Boğaziçi University. It provides details about the requirements, design diagrams, and project plan. It also includes a list of features and a user guide.

Using our plug-in, users shall be able to add annotations to web resources, delete, or modify them, and view a complete list of annotations about a web resource.

The plug-in adheres to the standards described by The World-Wide Consortium (W3C); adopting their proposed protocol, data model, and vocabulary.

# Requirements

This part describes the requirements of CAT Plug-in. The requirements are categorized in terms of functionality, and prioritized by MoSCoW prioritization standards. (MH = Must Have, SH = Should Have)

## Authentication

### Sign in using credentials

An authorized user must be able to sign in to the system with his/her username and password. (MH)

### Sign up with e-mail, username, and password

A non-member must be able to sign up by entering the following fields: e-mail, username, and password. (MH)

### Maintain email uniqueness

A non-member must not be able to sign up with the same e-mail address which is recorded in the system. (MH)

### Have member and admin user types

There should be two types of member: a regular member and an administrator. (SH)

### Change Password

An authorized user should be able to edit her/his password. (SH)

## Authentication

### Make private and public annotations

An authorized user must be able to make their annotations private or public on creation and can update later. (MH)

### Annotate HTML pages and texts

An authorized user must be able to annotate any HTML web page or a specific text in the web page. (MH)

### Don't include HTML tag in annotations

The html tag itself cannot be annotated. (MH)

### Annotate images

An authorized user shall be able to annotate an image with coordinates of the origin, width and height, in a rectangular manner. (MH)

### Only annotate the source of img tags

Only, the source of the "img" tags will be annotated. /(Ex.: <img src="image link" />)/ (MH)

### Highlight annotated texts

An authorized user should be able to highlight an annotated text. (MH)

### Manage annotations

An authorized user shall be able to create, edit or delete an annotation. The user can only delete and edit his/her own annotations. (MH)

### Search and delete annotations as admin

An administrator should be able to search and delete any annotation, whenever they find the annotation is inappropriate. (MH)

### Delete other members' annotations

In addition to all functions that the normal member do, the admin can also delete the other members' annotations. (MH)

### View annotations of a page

Any user shall be able to view number of all annotations of a specific page on the right menu of the plugin. (MH)

### View annotations of a target

Any user shall be able to view number of all annotations of a specific target on the right menu of the plugin. (MH)

### Filter annotations

Any user shall be able to filter annotation by uniqueness. The bodies that are added to the target annotation will be shown when the annotation is selected. (SH)

### Handle content changes

The annotated web pages' content may be changed, so the tool should take into account the current version of the web page. If a content change occurs, the tool should return a message that says "source might have changed". (SH)

### Won't Have Requirements

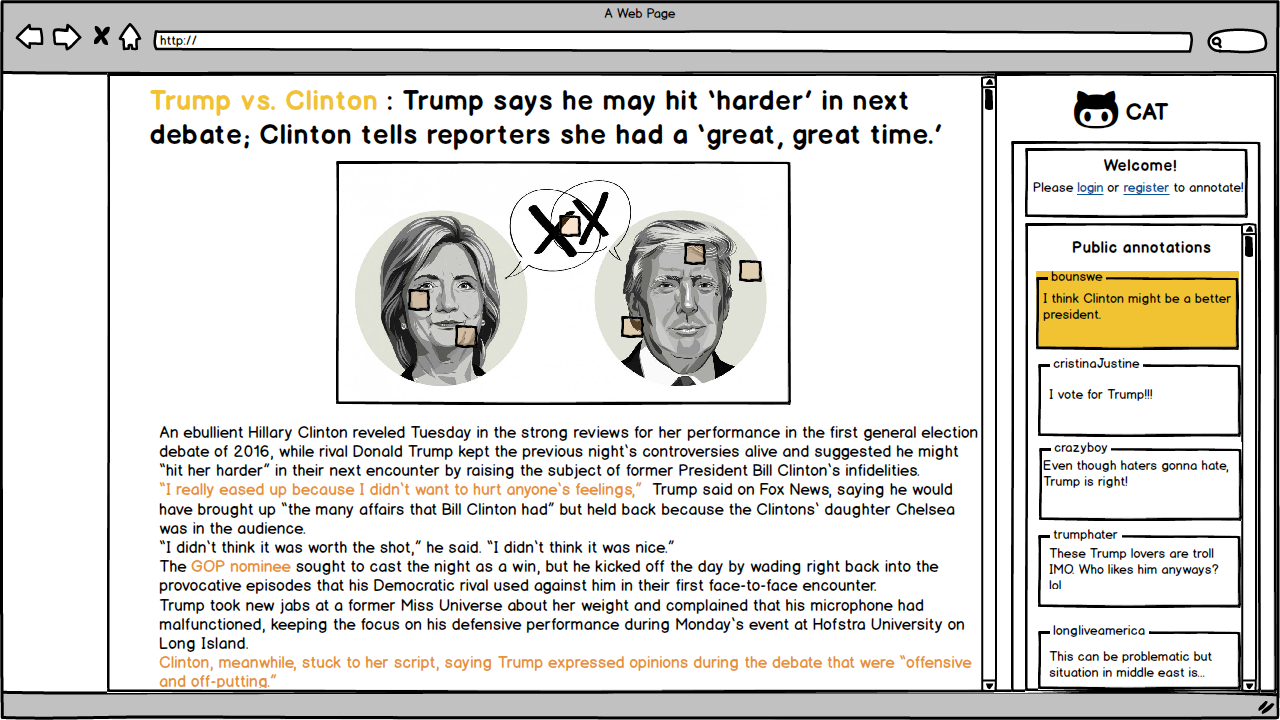
* An authorized user won't be able to annotate pdf, video and audio files.
* The tool won't offer the ability to make a correction if the content of the annotation is wrong. An annotation can only be updated by its author.
* There won't be a rating system for the annotations.
* There won't be a report system for annotations.
* An authorized user won't be able to use his/her Facebook/Twitter/Google ID to sign-in.

# Mock-ups

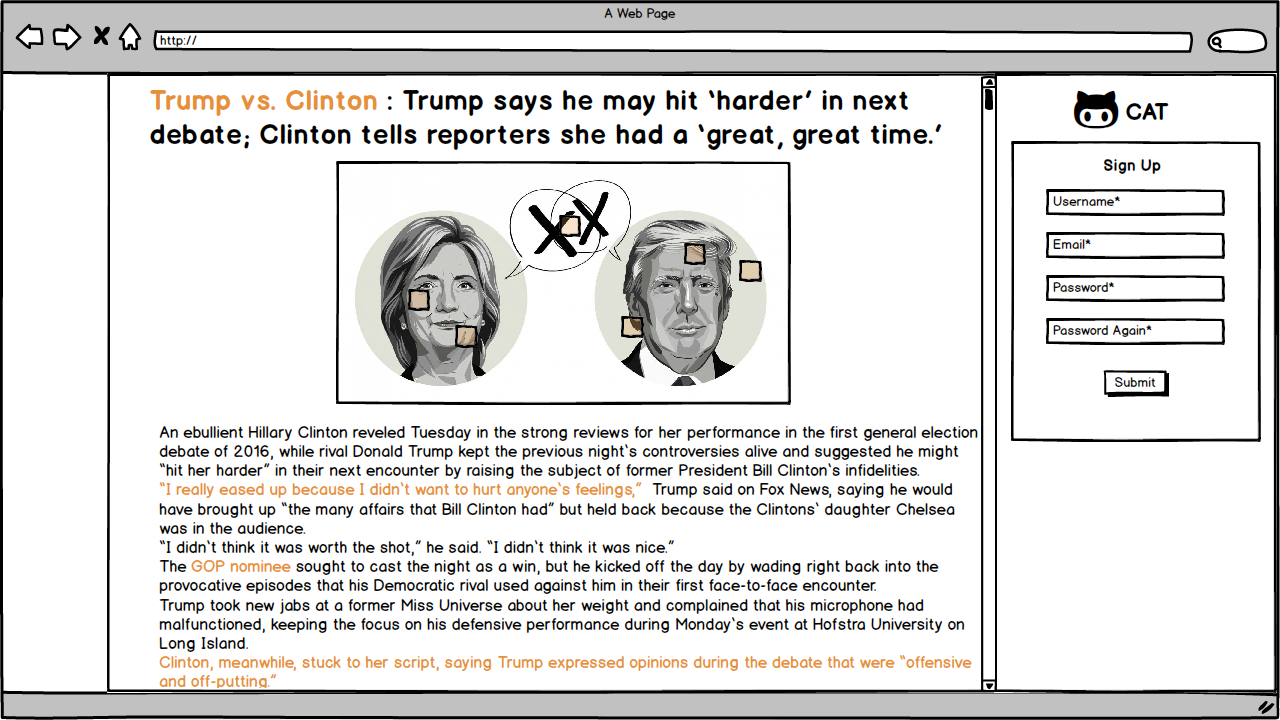
In manufacturing and design, a mockup, or mock-up, is a scale or full-size model of a design or device, used for teaching, demonstration, design evaluation, promotion, and other purposes. A mockup is a prototype if it provides at least part of the functionality of a system and enables testing of a design. (For more information, visit: <https://en.wikipedia.org/wiki/Mockup>)

This are more of a guideline than exact design models. This means that the actual design might include differences.

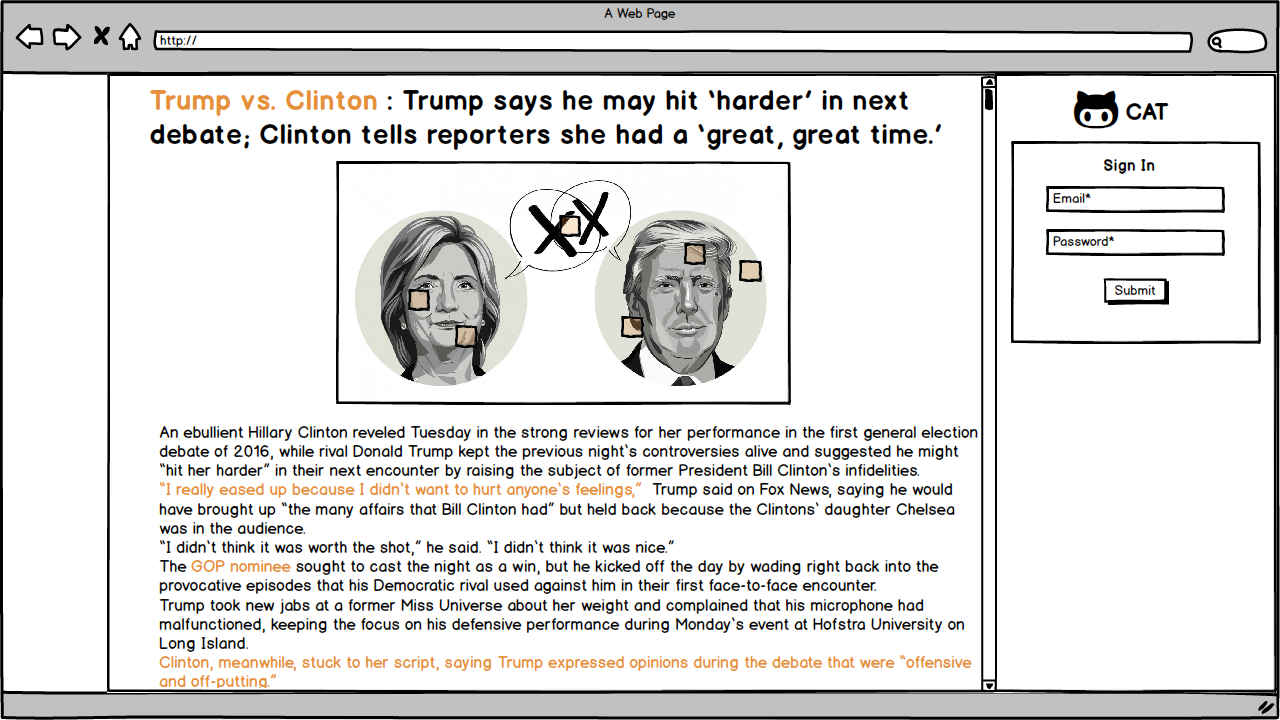
## View for an Anonymous User



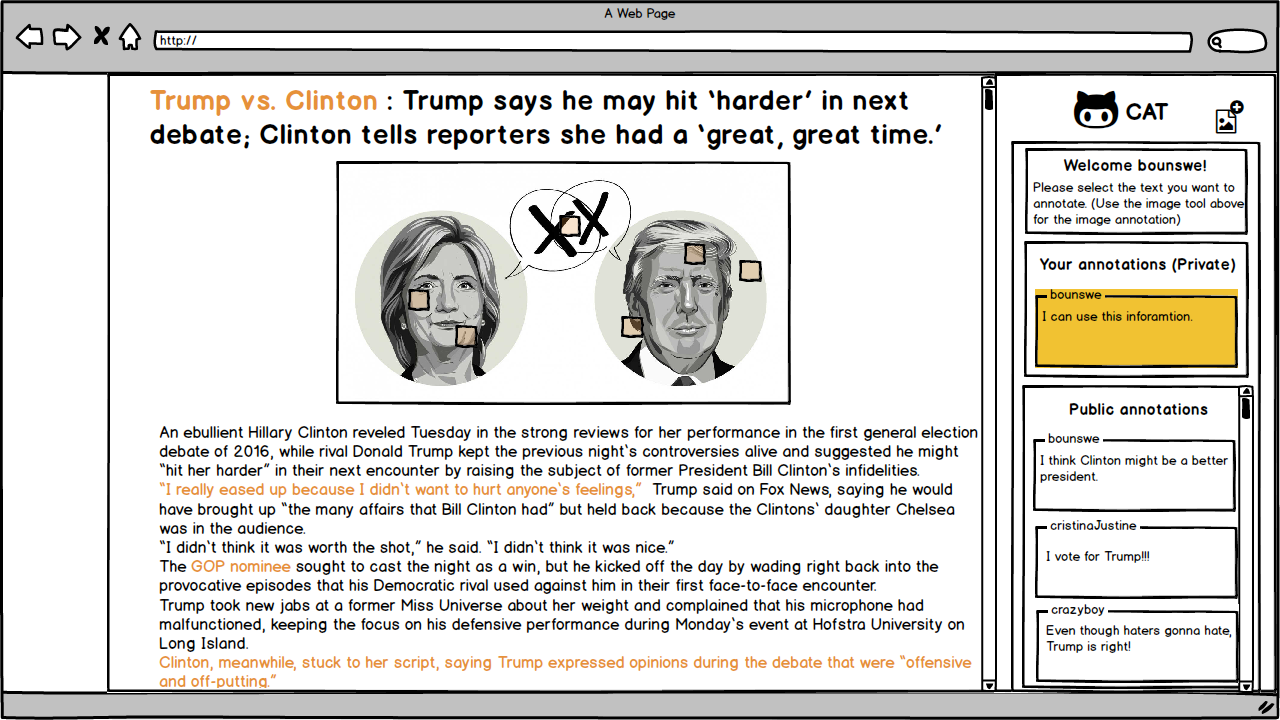
## Sign Up Screen



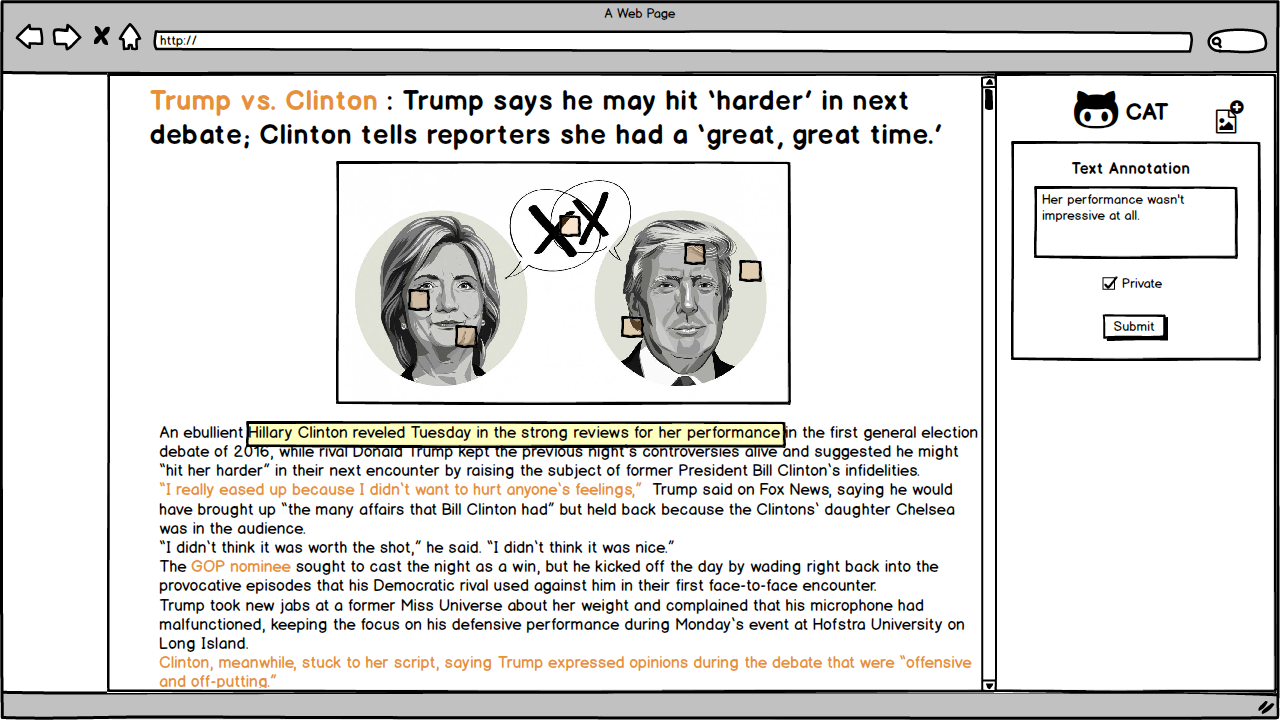
## Sign In Screen



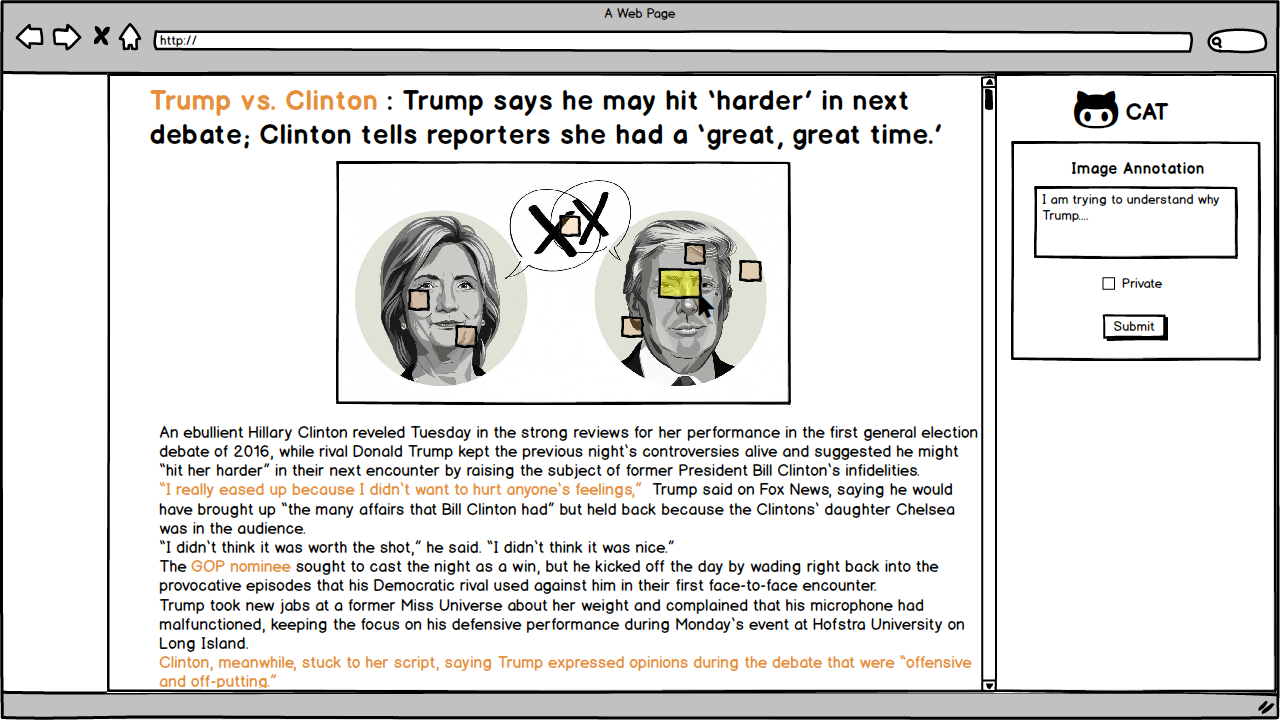
## After Sign In



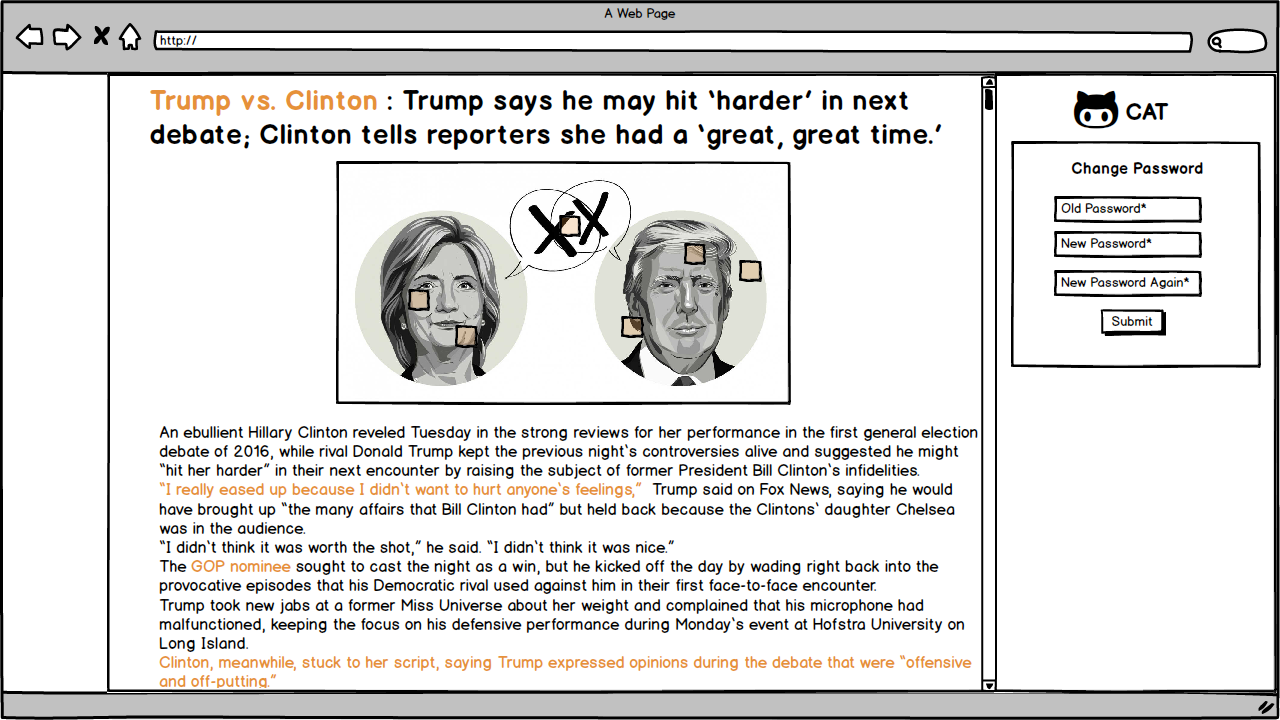
## Text Annotation



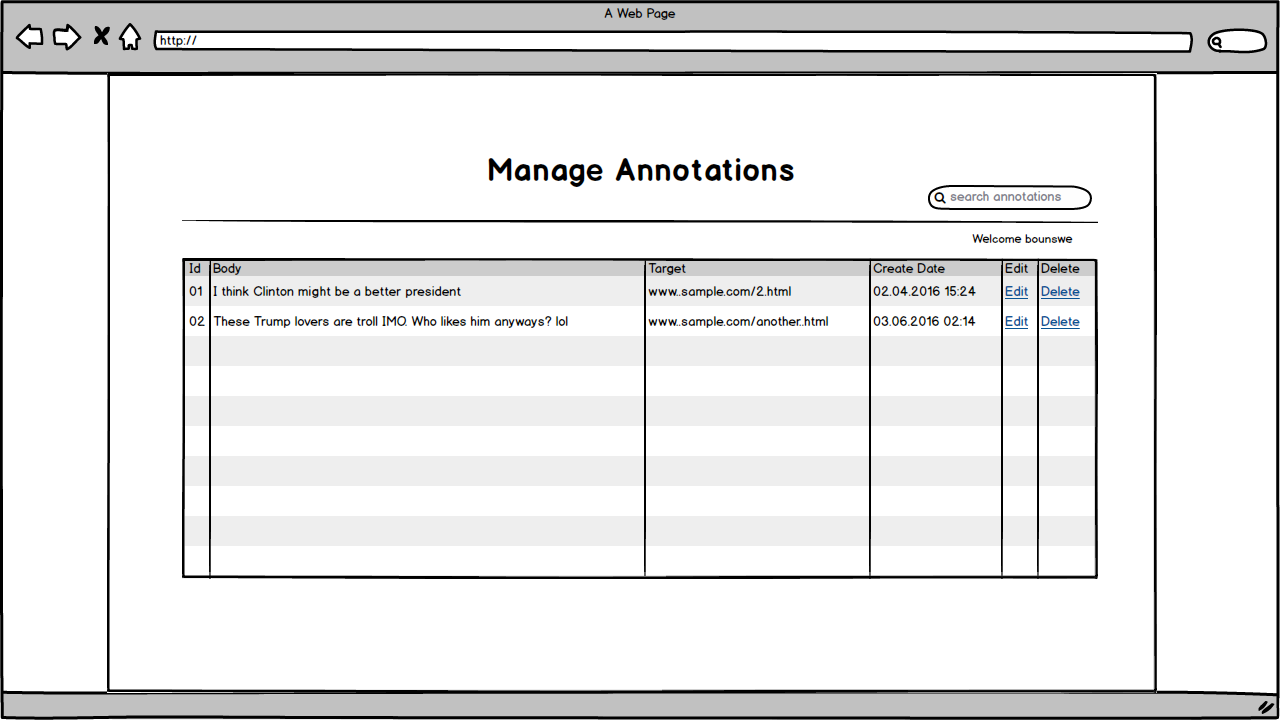
## Image Annotation



## Change Password Screen

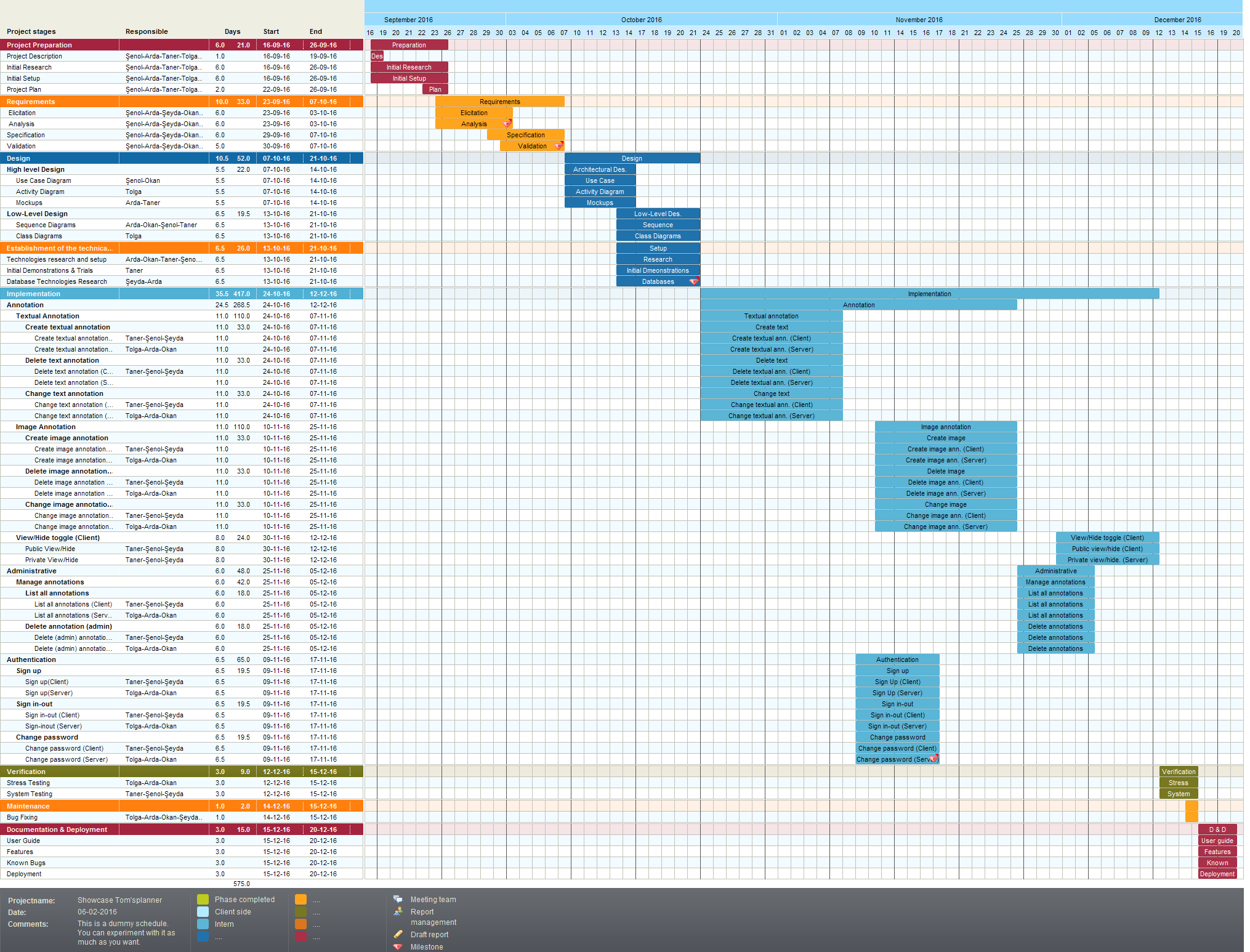


## Manage Annotations



# Project Plan

A project plan, according to the Project Management Body of Knowledge (PMBOK), is: "...a formal, approved document used to guide both project execution and project control. The primary uses of the project plan are to document planning assumptions and decisions, facilitate communication among project stakeholders, and document approved scope, cost, and schedule baselines. A project plan may be summarized or detailed." (For more information, visit: <https://en.wikipedia.org/wiki/Project_plan>)



# UML Design Diagrams

## Use Case Diagram

Use case diagrams are consisting of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a functionality of a system. So, to model the entire system, numbers of use case diagrams are used. (For more information, visit: <http://www.tutorialspoint.com/uml/uml_use_case_diagram.htm>)

### Use Cases for Admin, User, and Non-User

The following use case diagram shows the actions available to users, admins, and non-users.

use-case-diagram.png

## Activity Diagrams

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc. (For more information, visit: <http://www.tutorialspoint.com/uml/uml_activity_diagram.htm>)

### Create Annotation Activity Diagram

The following diagram shows the logic of creating an annotation.

create-annotation.png

### View Annotation Activity Diagram

The following diagram shows the logic of viewing an annotation.

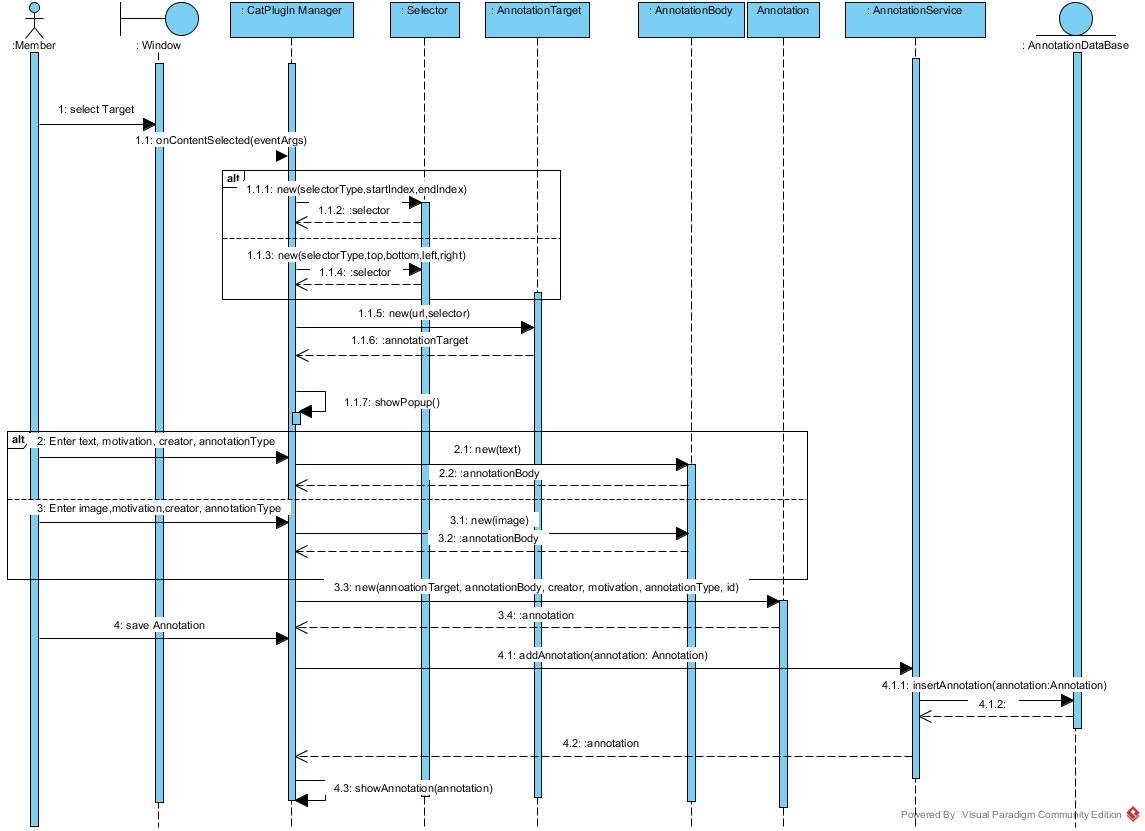
activity_diagram_view_annotations.png

## Sequence Diagrams

The sequence diagram is part of the interaction diagram. Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages. (For more information, visit: <https://www.tutorialspoint.com/uml/uml_interaction_diagram.htm>)

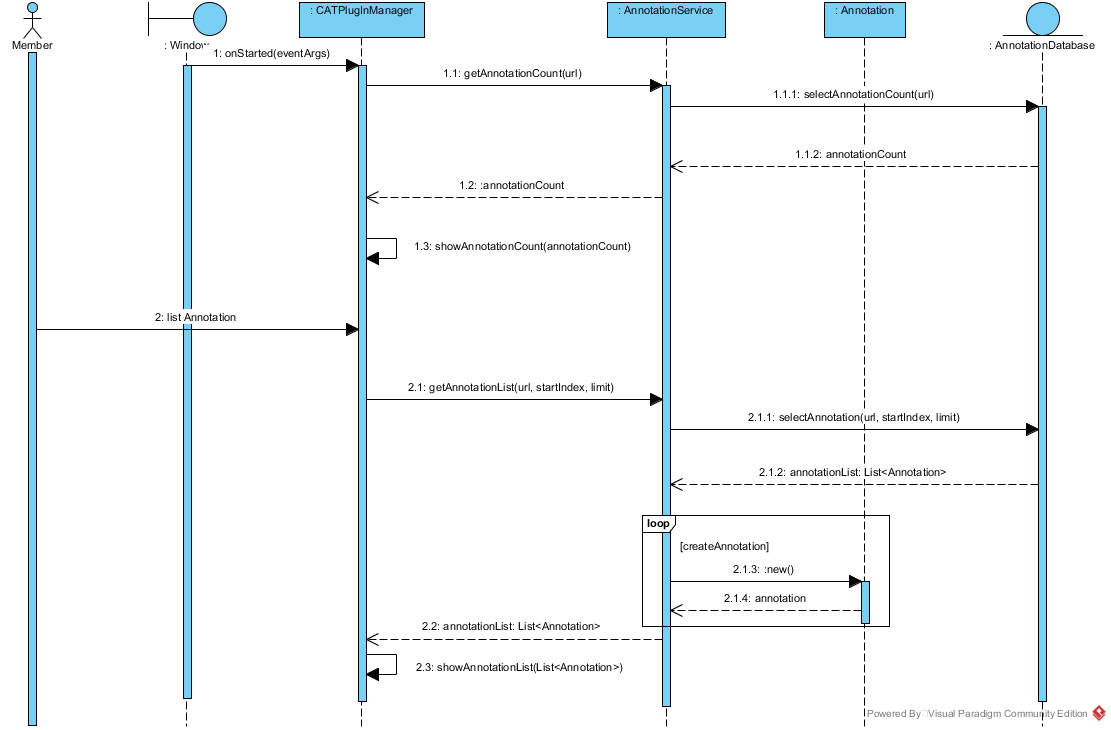
### Create Annotation Sequence Diagram

The following sequence diagram shows the creation of an annotation.



### View Annotation Sequence Diagram

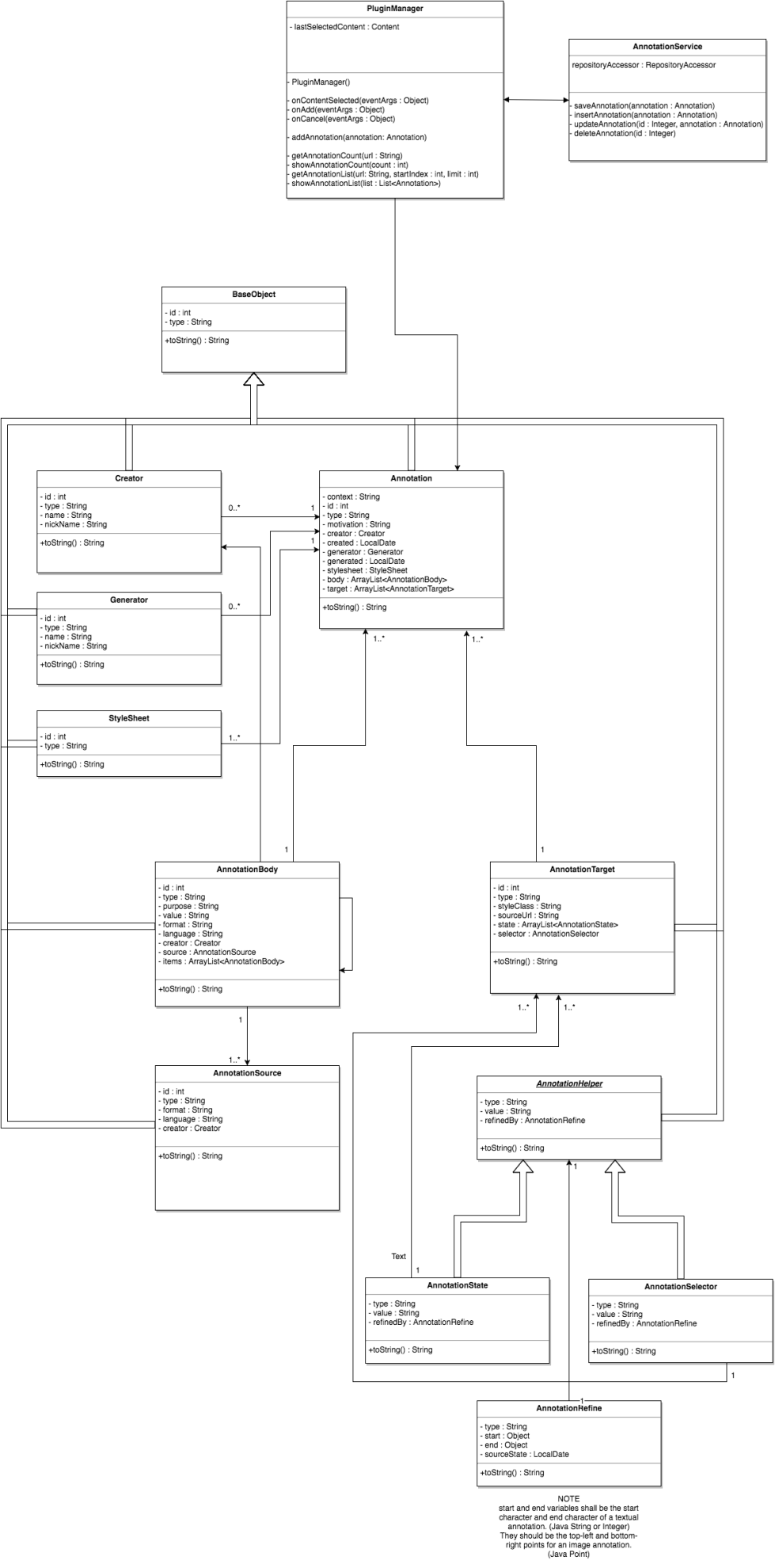
The following sequence diagram shows the displaying of an annotation.



## Class Diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram. (For more information, visit: <https://www.tutorialspoint.com/uml/uml_class_diagram.htm>)

The following class diagram shows the models used in the project.



# High-Level Architecture

The main architecture used in the project is the REST Architecture.

C:\Users\okanm\Downloads\Untitled Diagram (1).png

The main communication of the system can be seen from the figure 7-1. All of the requests and responses via HTTP are based JSON content type.

Also, a very detailed MVC architecture is used in the project. There are 5 layers for this architecture.

First layer is the model layer where the communication between database is done via data access objects.

Second layer is the repository layer which is implemented via CrudRepository, PagingAndSortingRepository or JPARepository in Spring Framework. These are the data transfer objects that handles create, read, update and delete operations easily.

Third layer is the service interface layer. This layer is created for abstraction and provides the signatures of the methods that are used in the service implementation.

Fourth layer is the actual service implementation layer in which the logic of the system is handled, exceptions are solved.

Fifth and final layer is the controller layer which handles the requests that come from the client via the front controller named as dispatcher servlet of the Spring Framework. After the request is handled, a response is sent back to the client.



This layered architecture provides a very strong internal structure and abstraction. All of the layers are separated from each other. Thus, changing any layer is not effecting a lower or a higher layer.

# Features and User Guide

To launch Cat Annotation Tool, the user should download cat annotation plug-in from Firefox library. After set up, a cat icon will appear on the browser and by clicking on this cat icon the plug-in shall launch.

When the Cat Annotation tool is launched, the login screen appears. If you do not have an account yet, you can create a new account by clicking on "Sign Up" button.

"Sign Up" page includes the required fields to register to the tool. the required fields are, "username", "email", and "password".

In sign up page, there is a control algorithm that checks the length of the password, whether it is correct, and if all the required fields are filled. For each case, there is an error message that appears below the related field.

When the user fills the form correctly and clicks on the register button, the register page redirects the user to the login page.

In the login page, username and password fields will appear.

The user has to enter his/her username and password. If the user doesn't enter their username or password, an error message appears again. If the user enters the username and password correctly and clicks the login button, they will be redirected to the manage page.

Manage page was created to manage annotations that are created by other users and the logged in user. There are two tabs in this page; "Basic Search" and "Advanced Search".

In the basic search, the users can search the annotations by word and a given time interval. The user enters to the text box the word that he wants to search and determines a start date and end date. When the user clicks to search button, the system displays all annotations that include the word given by the user and created between start date and end date.

In the advanced search, the user can search the annotations by type of annotation, by the motivation of the annotation, by creator and by target. Besides, the user can prefer to see his own annotations.

To list annotations that created are by him, the user should check "show my annotations only" button. Moreover, the manage page asks the user how many annotations to display. There is an annotation count selection box and the user can choose 1, 5, 10, 25, 50, or 100. These numbers refer to the number of annotations that will be listed on the page. If the user wants to search the annotations by type, he should select "image" or "text" options from the option box named "selector" located at the top of the page. If the user wants to search by motivation type of the annotation, he should select an option from "motivation selection box".

This selection box includes "assessing", "bookmarking", "classifying", "commenting", "describing", "editing", "highlighting", "identifying", "linking", "moderating", "questioning", "replying" and "tagging" options.

If the user prefers to search annotations by target, he should enter the target of the annotation to the text box named target.

In addition to target, the user can search annotations by the creator and in this case, he should write the creator to the creator text box.

After the user made all the selections, he clicks to "search&show" button and all the annotations will be listed in pursuance of his selections.

There is no need for a page to annotate an image or a text. When the user wants to annotate an image or a text, he should select the area or he should select the text and then a small annotate button appears on the selected area. When the user clicks on this annotate message, the annotation page will be appeared on the right side of the browser.

The user should enter the motivation of the annotation that explained in the search section (assessing, bookmarking, classifying, commenting, describing, editing, highlighting, identifying, linking, moderating, questioning, replying and tagging) and click to annotate button.

The annotated image or text will appear on the right side of the browser and when the user clicks on the annotate button, the system keeps his annotation and displays it on the top of the page.

# Server Configuration

The project consists of 2 main parts. The first part of the project is the “server side”. The main programming language that is used in the server side is “Java (SDK 8)”. In addition to that, “Spring Framework (Spring Boot 1.4.1)” is used as the main framework. For the web server and a servlet container. Apache Tomcat (Version 8) is used as a web server and servlet container.

Finally, MySQL is used as the database of the project. XAMPP Server is used for the development phase but any MySQL based database that is running on 3306 port is OK to be able to run the project.

## Java

As it is stated before, to be able to compile and run the application at least Java SDK 8 is required. To get more information about how to install Java, please refer to this site: <http://docs.oracle.com/javase/8/docs/technotes/guides/install/install_overview.html>

## IntelliJ or Spring Tool Suite

After the installation of Java, an IDE is required. Even though you can use any Java IDE, it is advised to install IntelliJ or STS. You can follow up the following site: <https://www.jetbrains.com/idea/> and for the IntelliJ and <https://spring.io/tools> for the STS.

## Tomcat

If you install one of the popular IDE’s, Tomcat can be downloaded quite easily from them. However, if you want to install separately and run as a war file, you can look up to this site: <https://tomcat.apache.org/download-80.cgi>

## MySQL

Finally, MySQL database is needed to be able to create our database. Even though it can be installed separately, it is suggested that to install the Wamp Server. It can easily start the database via phpMyAdmin from the 3306 port. It can be installed from site: <http://www.wampserver.com/en/>

## Security

Finally, there is also the security settings. HTTP Basic Authentication is used for the security part and the configuration this can be found in the WebSecurityConfig class. Annotation based page restrictions are used with @PreAuthorize annotation. To change the access levels, you can change this annotation and give or set permissions to the pages easily.

# Test Cases

Finally, we have created various test cases about authentication and adding an annotation.

















