|  |
| --- |
| CAT Plug-in Project Documentation |
| SWE 574 - Fall 2016 |
| CAT Plug-in Team |

Table of Contents

[2 Introduction 4](#_Toc470355090)

[3 Requirements 4](#_Toc470355091)

[3.1 Authentication 4](#_Toc470355092)

[3.1.1 Sign in using credentials 4](#_Toc470355093)

[3.1.2 Sign up with e-mail, username, and password 4](#_Toc470355094)

[3.1.3 Maintain email uniqueness 4](#_Toc470355095)

[3.1.4 Have member and admin user types 4](#_Toc470355096)

[3.1.5 Change Password 4](#_Toc470355097)

[3.1.6 Make private and public annotations 4](#_Toc470355098)

[3.1.7 Annotate HTML pages and texts 5](#_Toc470355099)

[3.1.8 Don't include HTML tag in annotations 5](#_Toc470355100)

[3.1.9 Annotate images 5](#_Toc470355101)

[3.1.10 Only annotate the source of img tags 5](#_Toc470355102)

[3.1.11 Highlight annotated texts 5](#_Toc470355103)

[3.1.12 Manage annotations 5](#_Toc470355104)

[3.1.13 Search and delete annotations as admin 5](#_Toc470355105)

[3.1.14 Delete other members' annotations 5](#_Toc470355106)

[3.1.15 View annotations of a page 5](#_Toc470355107)

[3.1.16 View annotations of a target 5](#_Toc470355108)

[3.1.17 Filter annotations 5](#_Toc470355109)

[3.1.18 Handle content changes 6](#_Toc470355110)

[3.1.19 Won't Have Requirements 6](#_Toc470355111)

[4 Mock-ups 6](#_Toc470355112)

[4.1 View for an Anonymous User 6](#_Toc470355113)

[4.2 Sign Up Screen 7](#_Toc470355114)

[4.3 Sign In Screen 7](#_Toc470355115)

[4.4 After Sign In 8](#_Toc470355116)

[4.5 Text Annotation 8](#_Toc470355117)

[4.6 Image Annotation 8](#_Toc470355118)

[4.7 Change Password Screen 9](#_Toc470355119)

[4.8 Manage Annotations 10](#_Toc470355120)

[5 Project Plan 10](#_Toc470355121)

[6 UML Design Diagrams 11](#_Toc470355122)

[6.1 Use Case Diagram 11](#_Toc470355123)

[6.1.1 Use Cases for Admin, User, and Non-User 11](#_Toc470355124)

[6.2 Activity Diagrams 12](#_Toc470355125)

[6.2.1 Create Annotation Activity Diagram 12](#_Toc470355126)

[6.2.2 View Annotation Activity Diagram 13](#_Toc470355127)

[6.3 Sequence Diagrams 14](#_Toc470355128)

[6.3.1 Create Annotation Sequence Diagram 14](#_Toc470355129)

[6.3.2 View Annotation Sequence Diagram 15](#_Toc470355130)

[6.4 Class Diagram 16](#_Toc470355131)

# 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 particular 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.

## 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)

### 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)

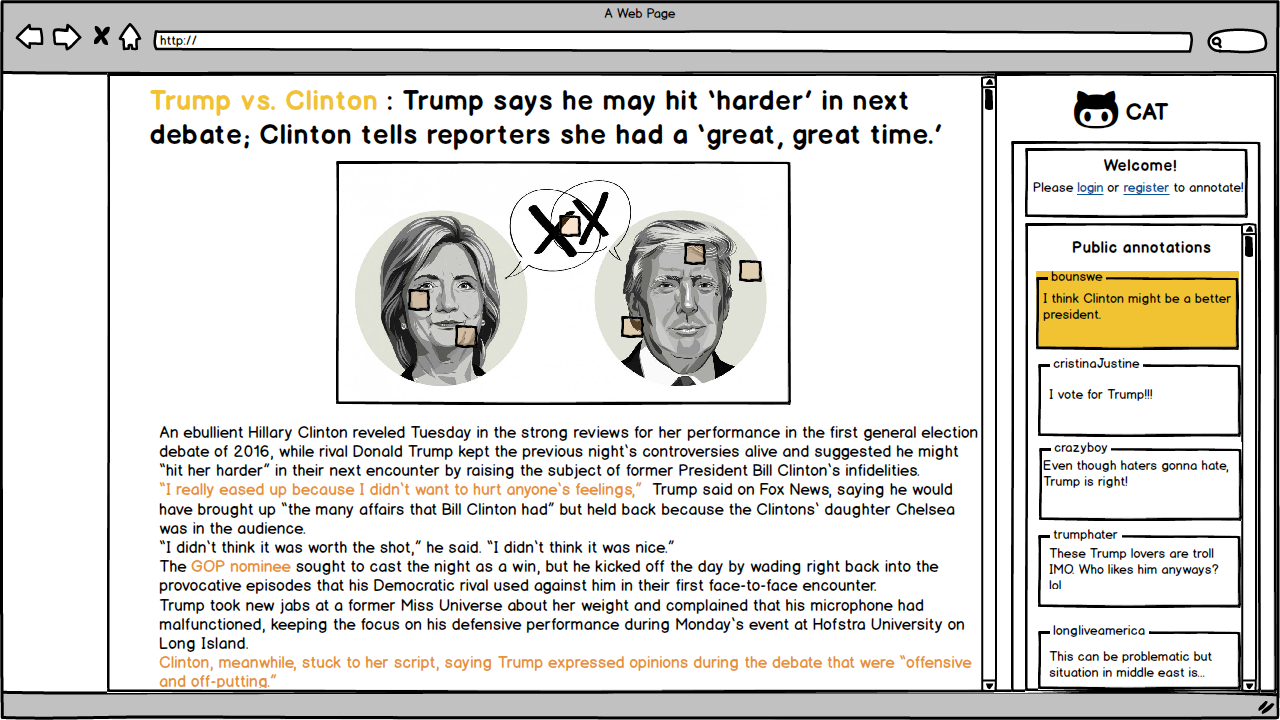
### Won't Have Requirements

* 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.

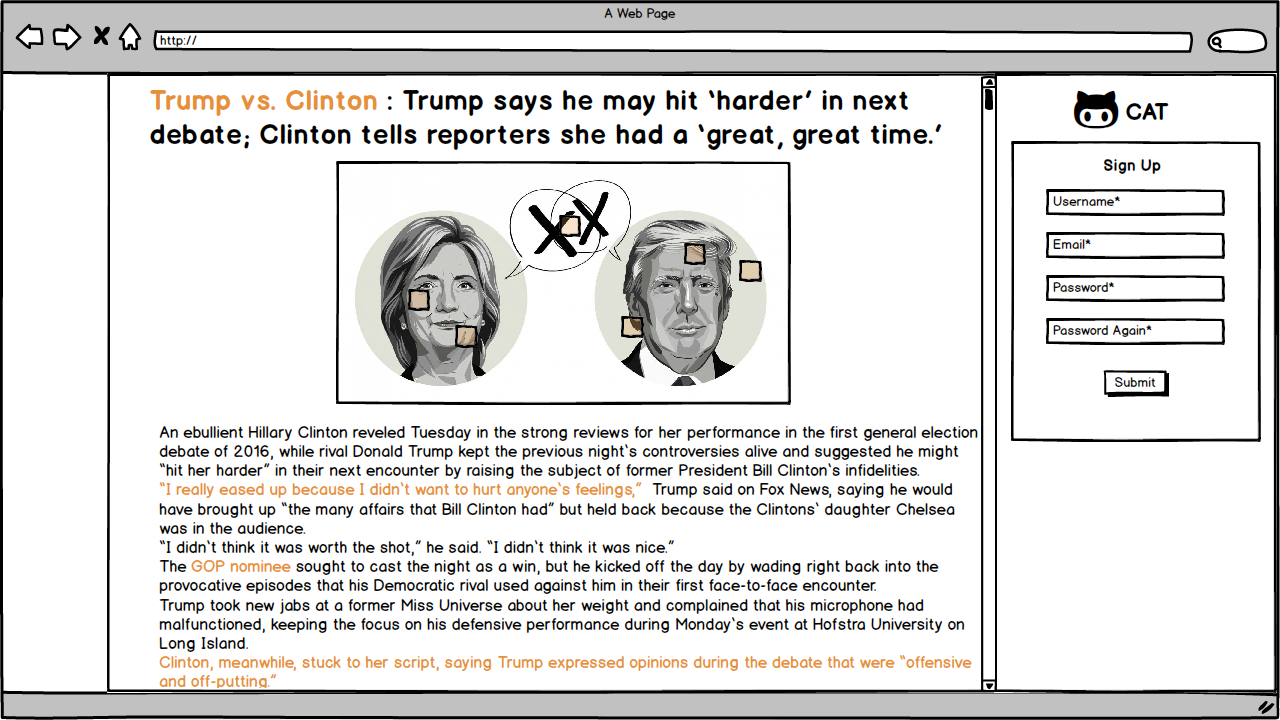
# Mock-ups

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: [Mockup](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.

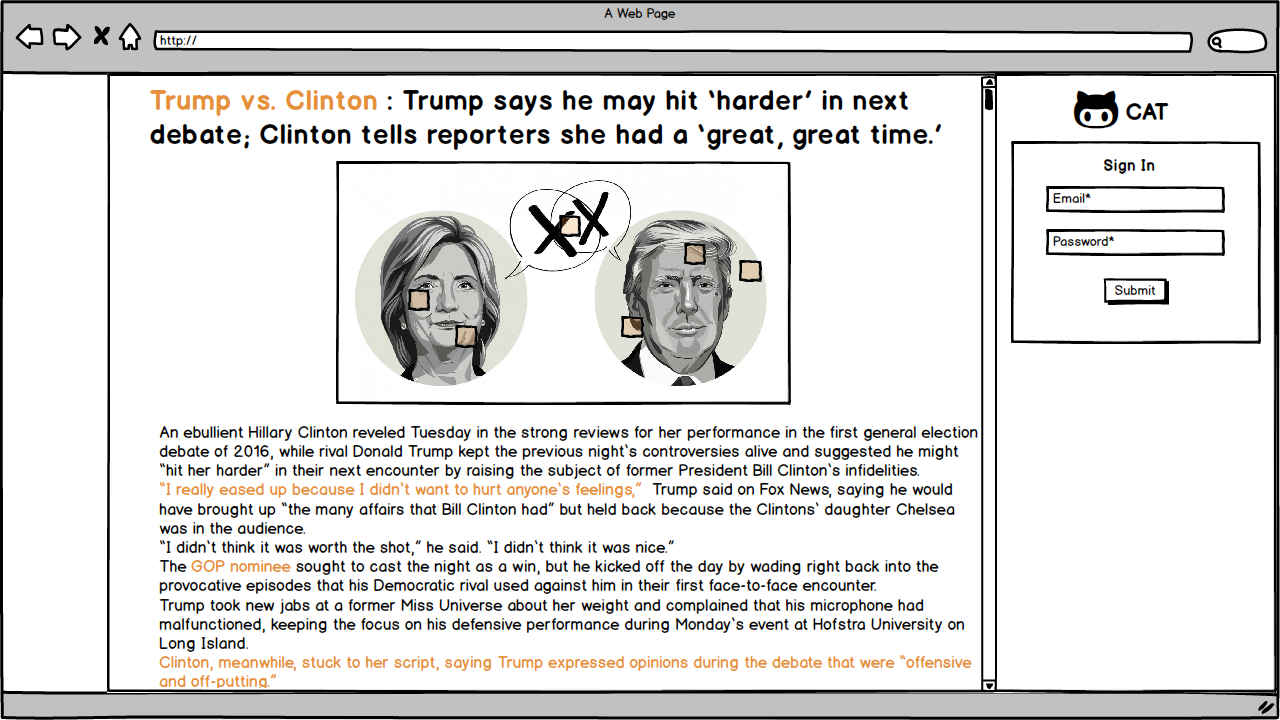
## View for an Anonymous User



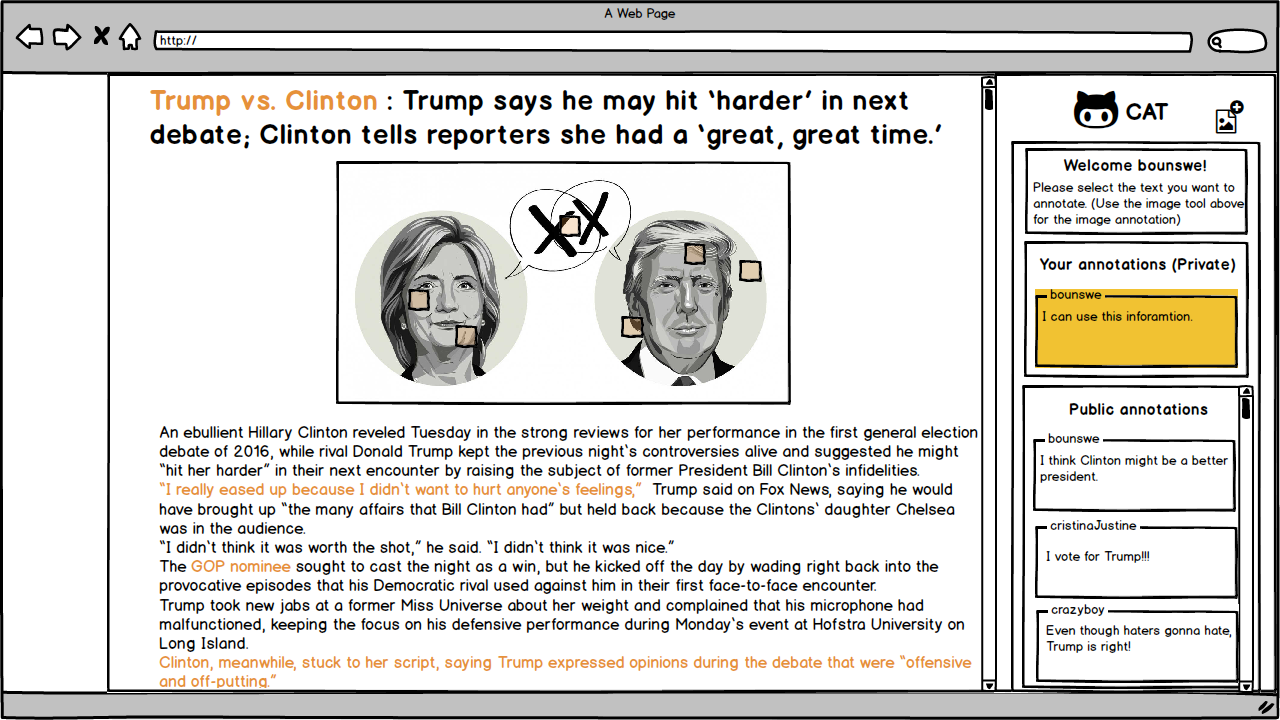
## Sign Up Screen



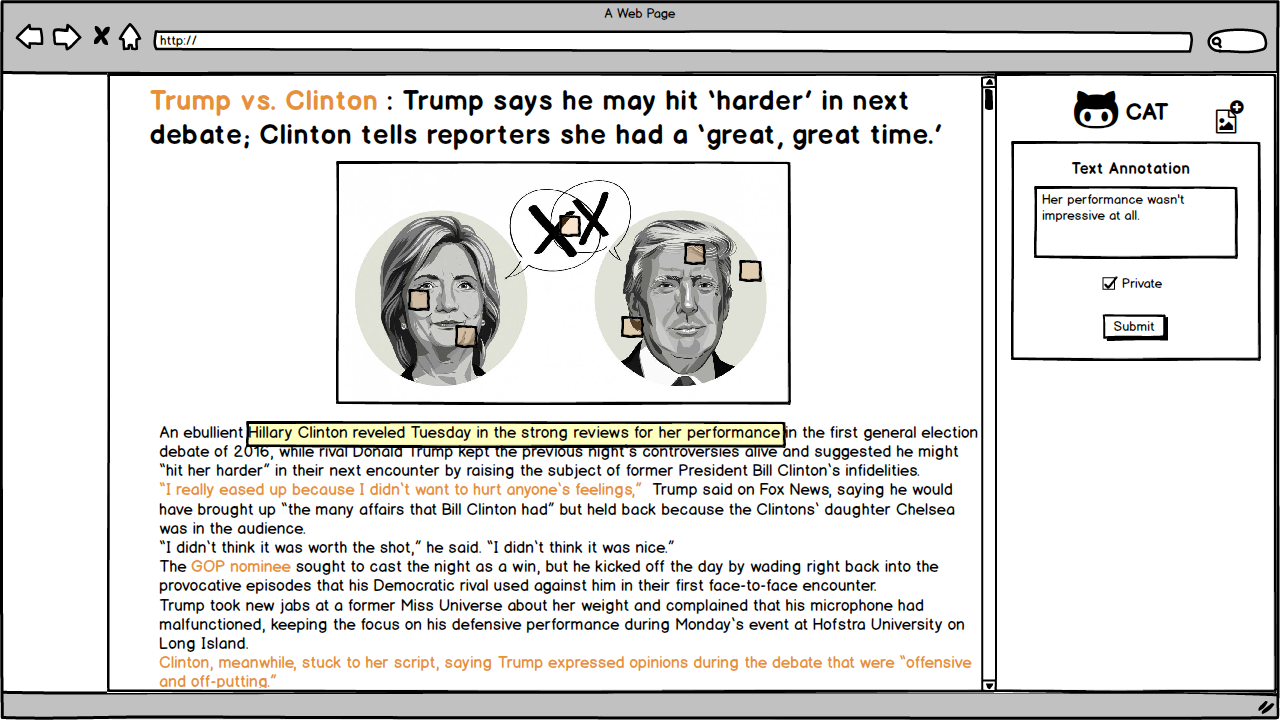
## Sign In Screen



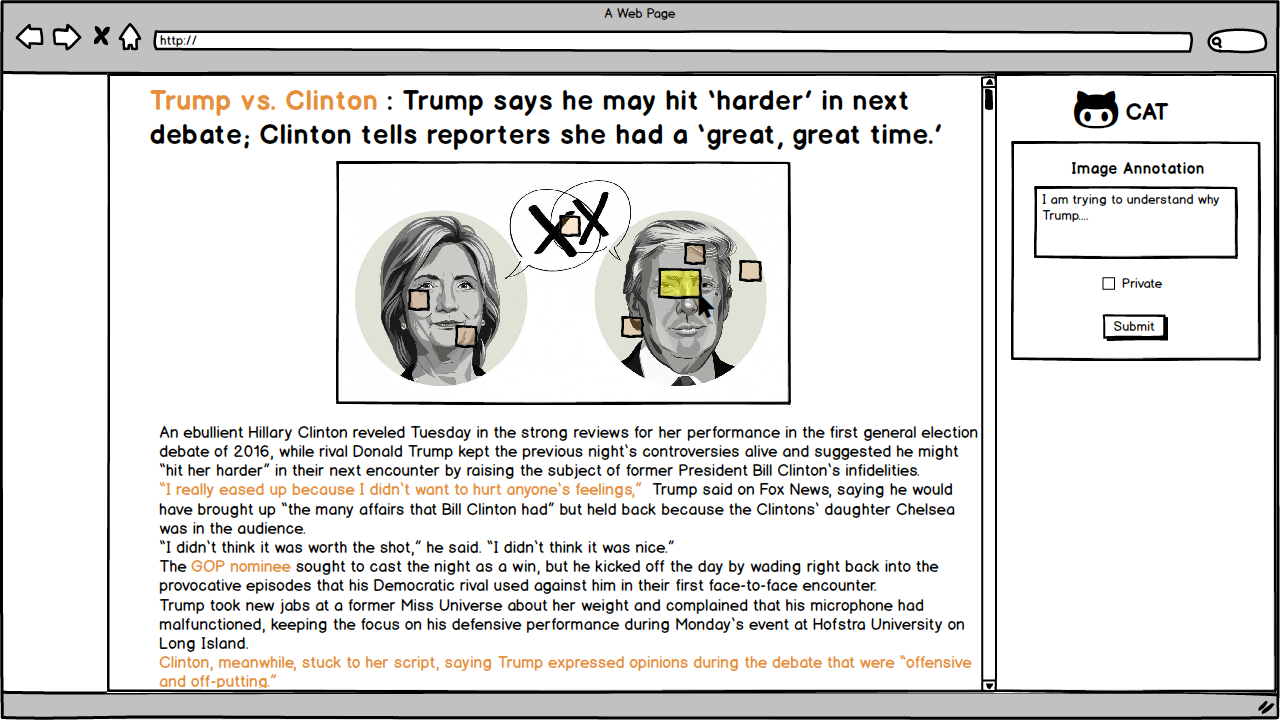
## After Sign In



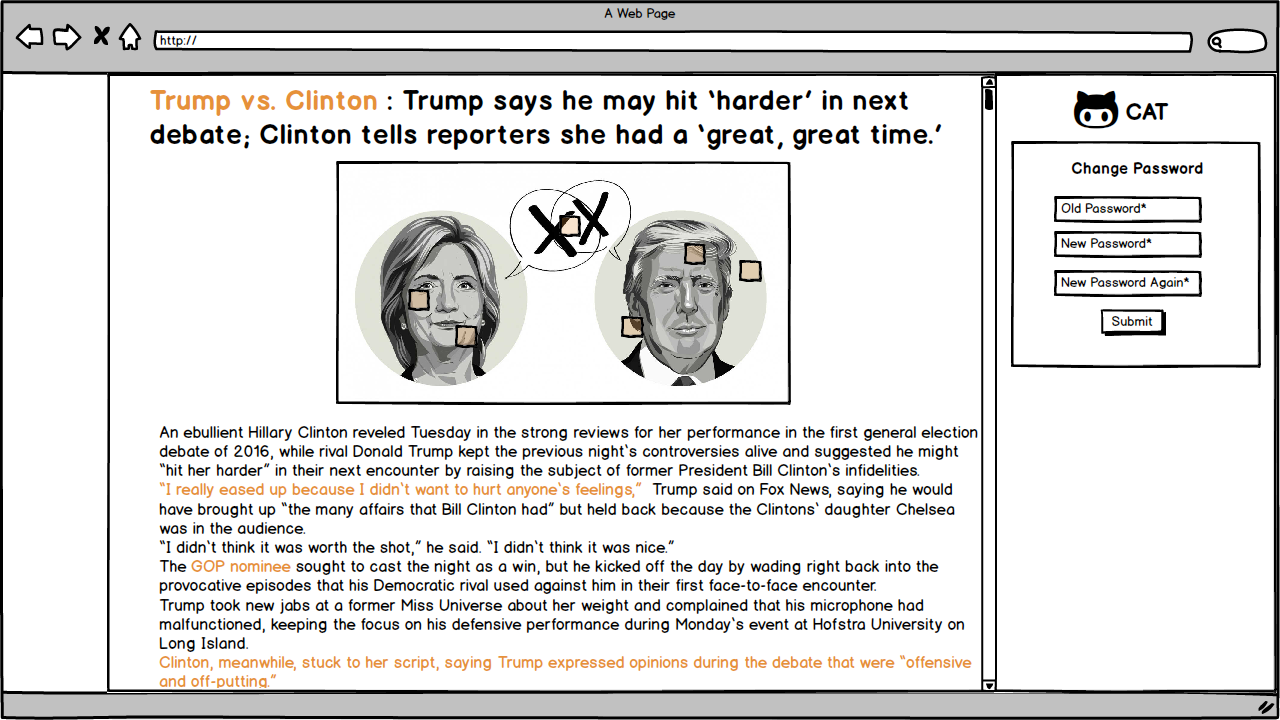
## Text Annotation



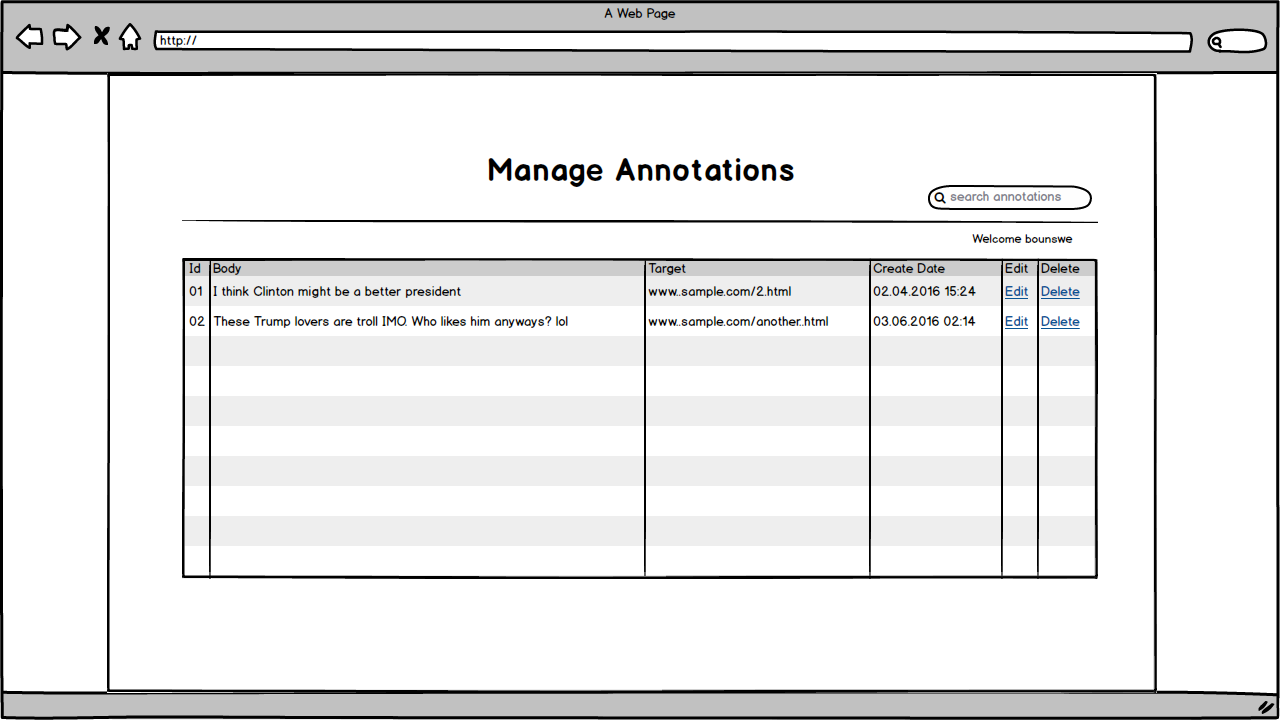
## Image Annotation



## Change Password Screen

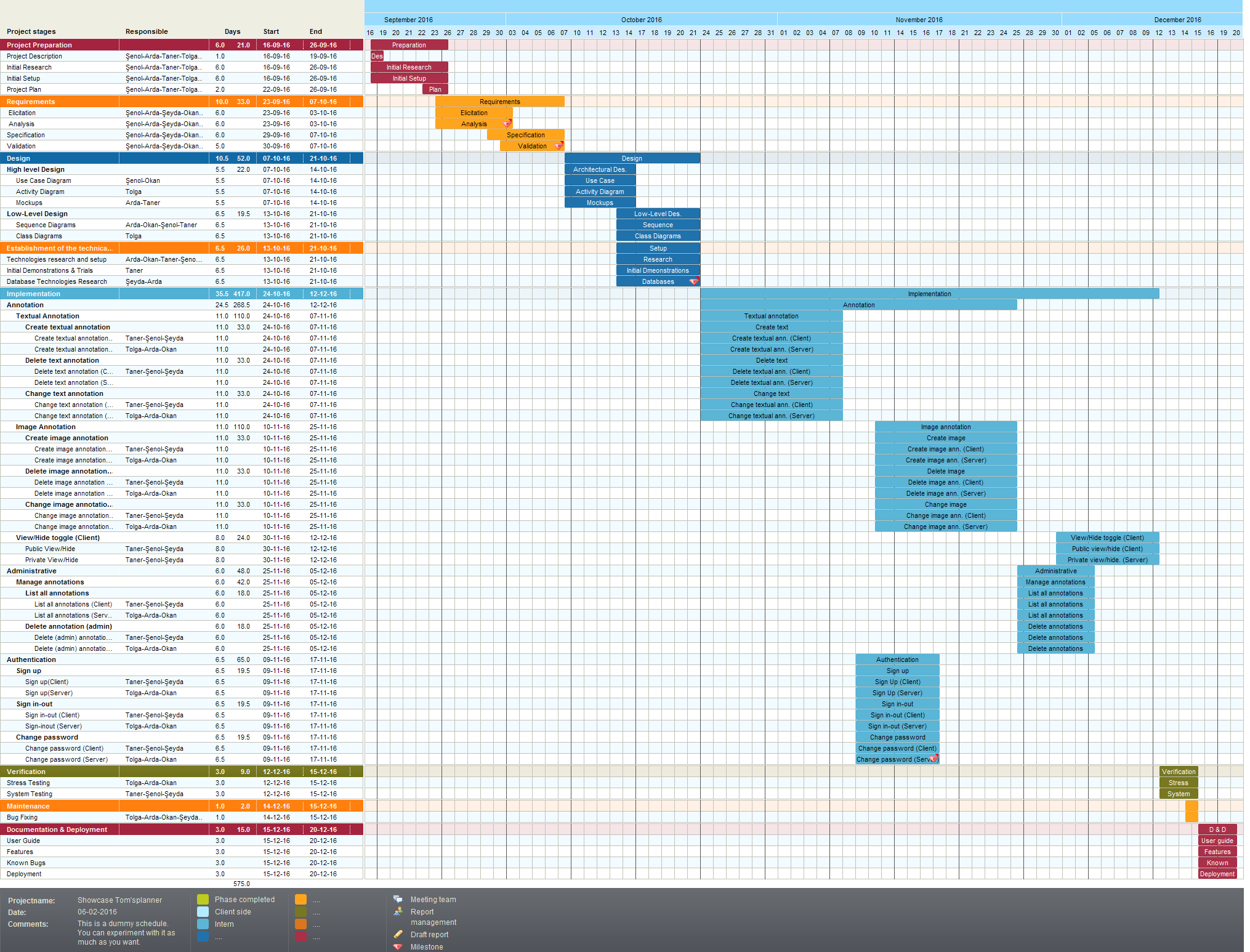


## Manage Annotations



# 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: [Project plan](https://en.wikipedia.org/wiki/Project_plan))



# UML Design Diagrams

## Use Case Diagram

Use case diagrams are consists 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 particular functionality of a system.  
So, to model the entire system, numbers of use case diagrams are used. (For more information, visit: [UML - Use Case Diagrams](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: [Tutorials Point (UML - Activity Diagram)](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.

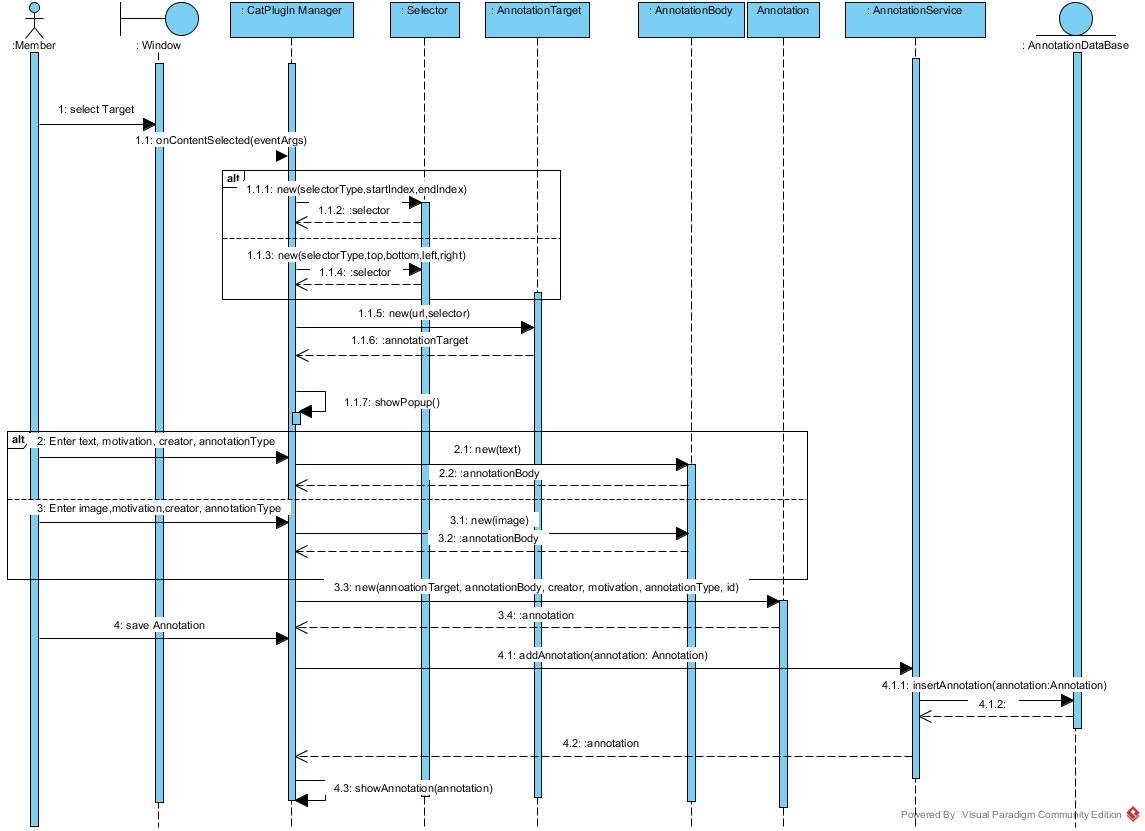
activity_diagram_view_annotations.png

## 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: [UML - Sequence Diagrams](https://www.tutorialspoint.com/uml/uml_interaction_diagram.htm))

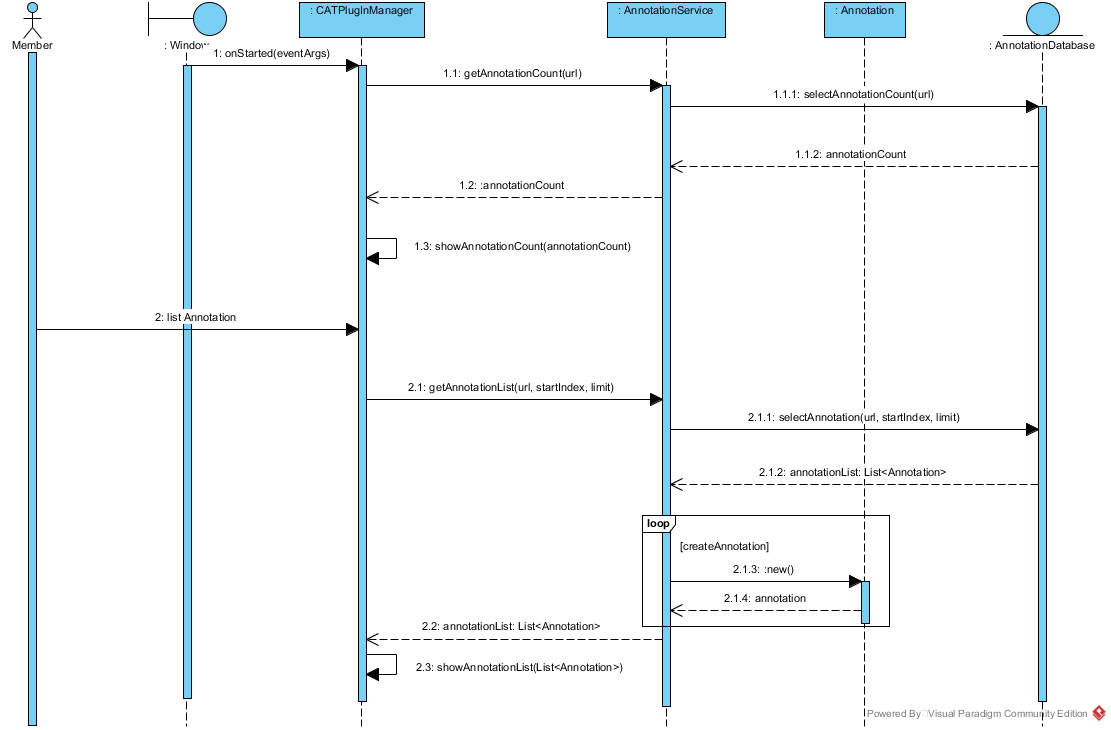
### Create Annotation Sequence Diagram

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: [UML - Class Diagram](https://www.tutorialspoint.com/uml/uml_class_diagram.htm))

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

