DRAWIUM

SOFTWARE REQUIREMENTS SPECIFICATIONS

Ozan Çelik
Ozan Tahiroğlu
Özgür Saygın Bican
Seçkin Can Şahin

Table of Contents

Figure List

1. INTRODUCTION

This document is a software requirement specification for Drawium project. To introduce you the document, we will first give the purpose and scope of this document, then follow with an overall description of the system. After completing a general introduction to the system, we will address specific requirements for it. In the third part of this document, we will address functional requirements as well as nonfunctional requirements.

In this document, we used “users” and “web-developers” interchangeably to refer to the users of our systems(who are presumably web-developers). We refer to the users of the web-developer’s websites as “users” but this should not cause ambiguity as it is easy to determine which users we are referring to from context.

1.1 Problem Definition

When a new user lands on a new website that he has not visited before, there is a steep learning curve. The fact that it is –at least kind of- hard to learn how to use the website costs the website owner a lot in terms of the number of users that they lose – users that just close their page or press on the back button.

In fact, this problem is so important that big corporations like Facebook and Google. They dedicate teams to solve these problems by developing custom solutions. For instance when a new user registers for a gmail account, gmail clearly indicates the button that they should use to compose an email by a box that explains it.

So what is the problem we are trying to solve if Google and Facebook already solved this aforementioned problem? The problem is that there is no toolkit for this purpose in the market for general use. The solutions those big corporations have developed over time are not open sourced and smaller websites are either forced to develop their solution in-house by devoting considerable resources or just continue losing those frustrated customers.

We aim to solve this problem by developing a very easy to integrate Javascript library and an accompanying website to customize that library.

1.2 Purpose

The purpose of this document is to give a complete description of the behavior of the Drawium project to be developed. This document is intended to establish the basis for agreement between customers and the suppliers on what the software product is to do, decrease the effort needed for development, provide a basis for validation and verification. What we are going to address will basically constitute a basis for functionality, external interfaces, performance, attributes and the design constraints of the system. This document is better suited for the customers, users, and developers .

1.3 Scope

The software we are going to introduce is named Drawium. It is a tool for web developers to help users understand their web site, and to share their designs with other developers.

1.4 User and Literature Survey

There are already some javascript libraries to help people with drawing. But, none of them has a purpose of helping web developers, and all of them are static libraries. Drawium is a dynamic library, it fits to the user's needs. Below there are two examples of other libraries:

Raphael: This is also a javascript library. Similar to ours, this also helps people with drawing. The differences between raphael and drawium are; Drawium's goal is to help web developers. Raphael is only a drawing tool (like an “opengl” for javascript) Raphael is a static library; it is the same library for all users. Drawium creates different drawing libraries for all its users.

JsDraw2D: This library is used to add drawing functionalities to web applications or web sites. Its only similarity with drawium is that both are going to be used by web developers.

1.5 Definitions and Abbreviations

Drawium: It is a drawing --and not just the output of the drawing process but the process itself-- with additional properties. Additional properties include but are not limited by the timing of the drawing(when it starts being drawn and stops) and whether it is animated or a still image.

Drawia: The plural form of Drawium.

JS Library: Javascript Library

SRS: Software Requirements Specification

1.6 References

1. Canvas Element. Retrieved 22.11.2011 from http://en.wikipedia.org/wiki/Canvas_element

2. Copyright. Retrieved 24.11.2011 from http://en.wikipedia.org/wiki/Copyright

3. HTML5. Retrieved 22.11.2011 from http://en.wikipedia.org/wiki/Html5

4. IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

5. JavaScript. Retrieved 23.11.2011 from http://en.wikipedia.org/wiki/Javascript

1.7 Overview

Following section of this document will focus on describing the system in terms of product perspective, product functions, user characteristics, assumptions and dependencies. In the third section, we will address specific requirements of the system, which will enclose external interface requirements, functional requirements of the system, performance requirements , and other requirements.

2. OVERALL DESCRIPTION

In this part, background information about specific requirements of the system will be provided briefly. General issues that affect the product and outline of the functional requirements will be mentioned, too.

2.1 Product Perspective

The product will be a Javascript library and an accompanying web site for the use of web developers. The product's main purpose is to provide a JS library to the users. The JS library will include the functions to help create Drawia. These functions shall take the ID of an HTML element along with other possible configuration parameters (like the time to start drawing etc.) as a parameter in order to attach a Drawium to the HTML element. For instance, there will be a function called draw_circle(id) which takes the id of the html element to cover with a circle and animates the drawing of a circle to grab the attention of the users. This is an example image of what we are envisioning:

Figure 1. An example image of what we are envisioning

Moreover, the system shall provide a drawing environment to the web-designers which includes basic drawing and text box creation tools. As an example use of the text-box that aims to teach something to the visitors, have a look at this screenshot from Google Docs web interface:

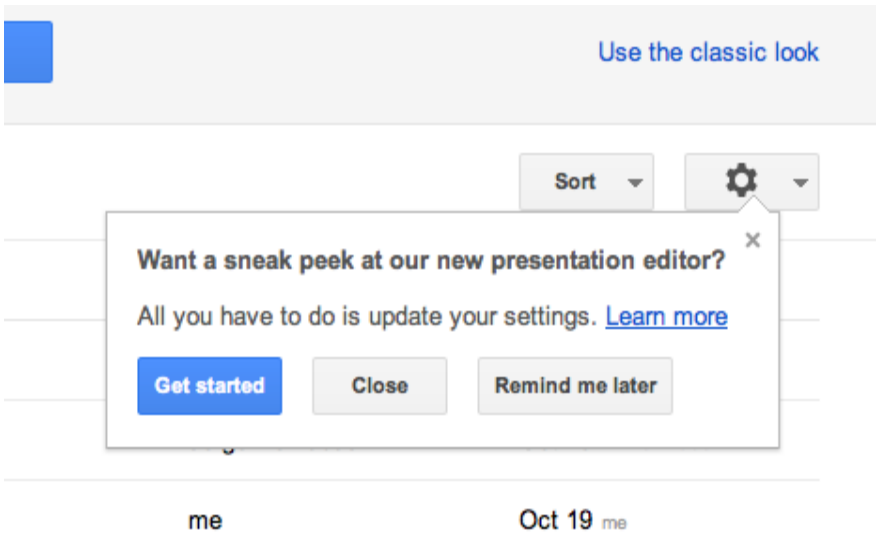

Figure 2. screenshot from Google Docs web interface

The system will be a website with a database. It will be independent and totally self contained. Users reach to the functions that they can include in their custom JS library (the Drawia Library) via the web-interface which shall store and serve the drawia in and from the database. After each operation as input, the changes shall reflect the database in a way that will insert, remove and change the related fields. The system shall deal with the database issues, i.e searching and retrieving data from the database.

Please check the overall diagram of the system below.

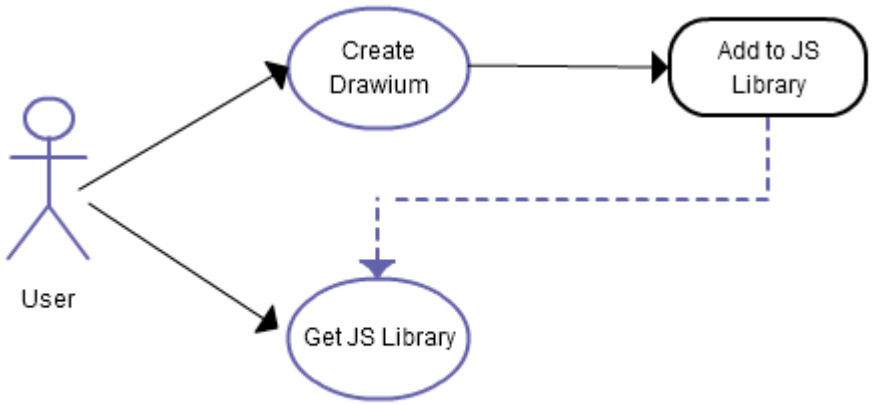

Figure 3. Overall Diagram of the system

2.2 Product Functions

The user shall provide specific sequence of inputs to the system to make it operate properly. The system shall produce the corresponding outputs via a set of functions for the given inputs. We can categorize the functions into three:

1. Create JS library or drawium

This category includes the functions that fulfil the main aim of the system. These functions enables the user to produce new drawia. By using their own or other users' drawia they shall create their JS library.

2.Communicative functions

The functions in this category enable users to share drawia and make connection with other users. The users will be able to display and follow other users' profile. The drawia will be able to sharable and usable by other users. In addition users will be notified about the newly created drawia via news feed.

3.Administrator functions

The functions in this category are usable by only admins. The admins shall have privileged functions so that they can delete users and drawia.

2.3 Constraints, Assumptions and Dependencies

The target community of the product is web developers. Since the system's main aim is to provide a JS library to the users, the user should have enough information about using JS library.

2.3.1 Intellectual Property Protection Rights

Like all software developers, we will need some protection for our software. This will be done with a patent, trademarks and copyright.
Here is a small explanation about what they are:

Patent: A set of exclusive rights for a public disclosure of an invention.

Trademark: This is a symbol that represents the product belongs to a person or company.

Copyright: A legal concept giving the owner exclusive rights to the product.

2.3.2 Code Obfuscation

Since anything on the web can not be completely protected, some smart people will try to copy our source code, modify it and claim that it is theirs. To prevent this, we will obfuscate our source code. This way, these smart people can not understand our code, thus can not modify it.

3. SPECIFIC REQUIREMENTS

3.1 Interface Requirements

3.1.1 User Interfaces

Profile: Here the user will see his/her profile. The profile will include all drawium done by him/her, as small images. The user can also follow or unfollow other users from this page, using small buttons.

News Feed: Here the user will see both his/her and all his followee's drawium, again as small images.

Display Drawium Module: Here the user will be able to display a drawium, as a large image. The user can animate the drawing process of the drawium (from scratch) by clicking a button here. There will also be some information about this drawium; such as how many people are using this drawium. The user can also use this drawium with clicking a button here.

Create New Drawium: The user can create a new drawium here. This page will contain some basic drawing tools, to help the user create the drawium. It will include a save button, which saves the drawium.

3.1.2 Admin Interfaces

In addition to the user interfaces;

Profile: The admin can also delete users with clicking a button on that user's profile. Of course, another button will pop up and ask the admin if he/she is sure.

Admin Panel: The admin can list all the users registered here, with clicking a button. The list will contain links to the users' profiles.

3.1.3 Software Interfaces

HTML5: This is the fifth version of the HTML software. It is used for structuring and presenting content for web. It is a popular choice among web developers, a report released on September 2011 says that 34 of the worlds top 100 web sites use this software.

JavaScript: This one is a prototype based scripting language. It is multi paradigm, in other words it supports functional, imperative and object oriented programming. It is mainly used for web development, but it can be used for anything else too.

Canvas: It is a tool used for drawing using scripting, mainly used with javascript. It can be used to draw images, graphs, make photograph compositions and make animations.

3.2 Functional and Non Functional Requirements

3.2.1 Profile

Diagram

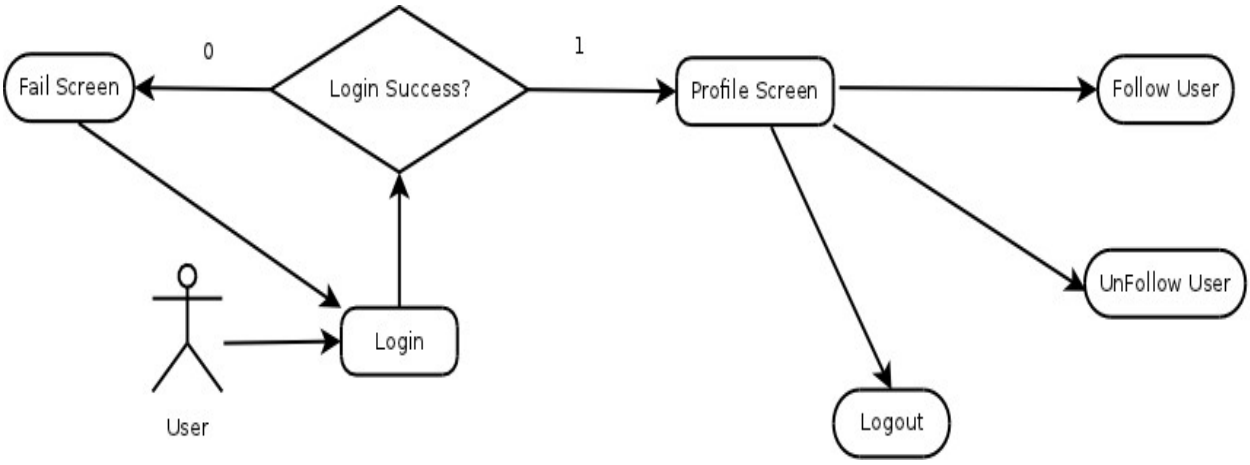

Figure 4. Diagram of Profile Module

Description

Our users are going to have a profile where they will show the “Drawia” they have created and also the Drawia they saw on the website and wanted to share with their followers. It is also a gateway to all the information they have provided to the website over the course of their being user. See specific functionalities section for more details.

- **Primary Actor** **User**
- **Goal in Context** **To view their information and drawia**
- **Preconditions** **User should be logged in**
- **Trigger** **User wants to view his or her information and drawia on the Drawium website.**

3.2.1.1 Specific Functionalities

See Followees and Followers

Diagram

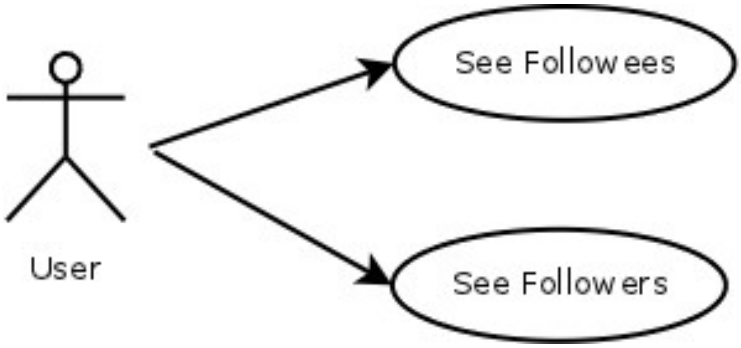

Figure 5. Diagram of See followers and followees

Description

This section is designed to show who the user follows and which users follow him or her. Number of followers and followees will also be shown here.

- **Primary Actor** **User**
- **Goal in Context** **For user to view the followers and followees he or another user has**
- **Preconditions** **User should be logged in**
- **Trigger** **User clicks on a profile link on the website and looks at the Followers or Followees sections of the page.**

Normal Flow of Events

1- User logs in to the system with correct credentials
2- User is authenticated and forwarded to the news feed page.
3- User clicks on the profile link shown on the page.
4- User lands on the profile page.
5- User looks at the Followers and Followees sections of the page.

Alternative Flow of Events

The steps 1 and 2 are optional and user can go through the 3,4,5 phases by going to her profile and seeing a drawium he has developed on another person's profile page.

Functional Requirements

This section should be visible even if the profile-owner user does not have any followers and does not follow anybody.

Non-Functional Requirements

Having a lot of followees or followers should not decrease the user experience quality or the performance of the website substantially.

Follow

Diagram

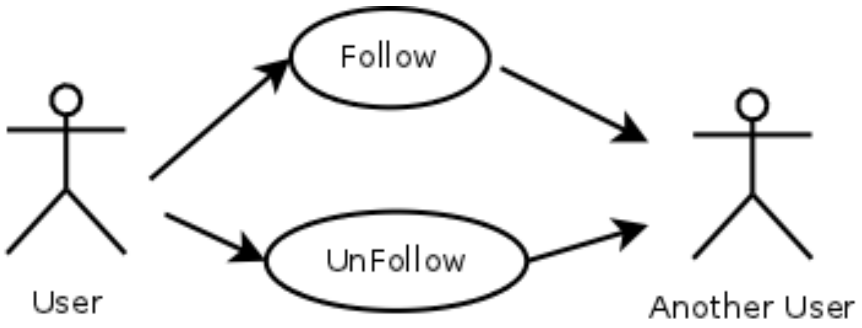

Figure 6. Diagram of Follow

Description

This button lets users follow each other. Following means that the follower is going to see the activities of the followee in his news feed. This helps users to be kept informed about the activities of the followee.

- **Primary Actor** — **User**
- **Goal in Context** — **To be kept informed about a user's activities.**
- **Preconditions** — **User should be logged in. The profile should belong to someone else and the viewing user should not already be following the profile owner.**
- **Trigger** — **User views the profile of another user and wants to follow him/her.**

Normal Flow of Events

1- User logs in to the system with correct credentials.
2- User is authenticated and forwarded to the news feed page.
3- User clicks on the profile link of another user from the news feed.
4- User lands on the profile page of that user.

5- User clicks the Follow button.
6- User starts following the other user.

Unfollow

Diagram

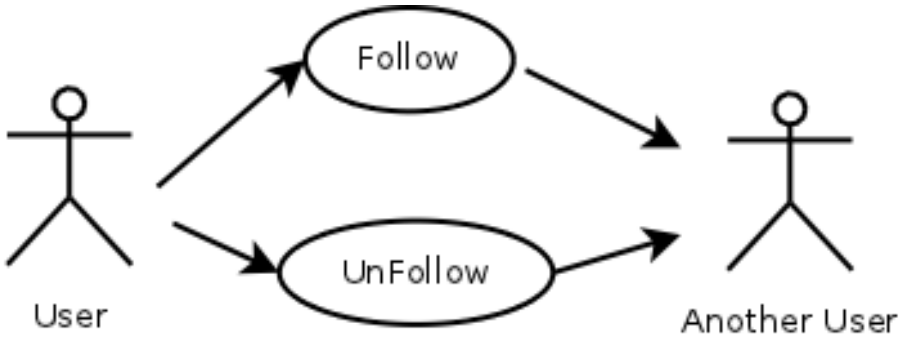

Figure 7. Diagram of Profile Unfollow

Description

This button lets users unfollow each other. This helps users to be unsubscribed from users who are no longer interesting for him/her.

- **Primary Actor** — **User**
- **Goal in Context** — **To stop getting updates about the activities of a user.**
- **Preconditions** — **User should be logged in. The profile should belong to someone else and the viewing user should already be following the profile owner.**
- **Trigger** — **User views the profile of another user she is already following and wants to unfollow him/her.**

Normal Flow of Events
1- User logs in to the system with correct credentials.
2- User is authenticated and forwarded to the news feed page.
3- User clicks on the profile link of another user that he or she is already following from the news feed.
4- User lands on the profile page of that user.

5- User clicks the Unfollow button.
6- User stops getting updates about that user.

Alternative Flow of Events
1- User logs in to the system with correct credentials.
2- User is authenticated and forwarded to the news feed page.
3- User goes to her own profile.
4- User clicks on the profile link of another user that he or she is already following from the Following section of the profile page.
5- User lands on the profile page of that user.
6- User clicks the Unfollow button.
7- User stops getting updates about that user.

3.2.2 Display Drawium Module

Diagram

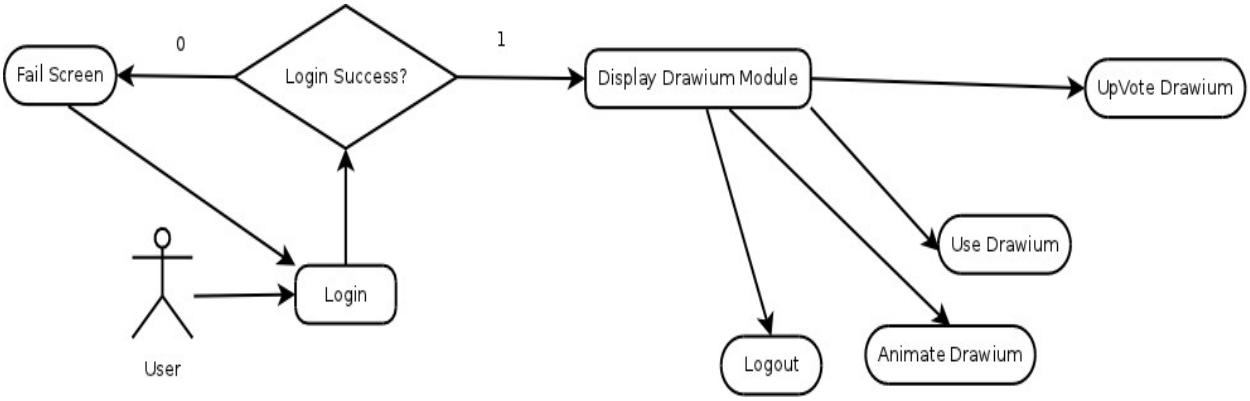

Figure 8. Diagram of Display Drawium Module

Description

A drawium will be able to be displayed at user profiles and news feeds. This module will include

- the creator
- number of votes
- number of users using that drawium

The animation of a drawium will be watchable. Besides users will be able to upvote or click on the “use this” button of the drawium. The “use this” button makes the drawium usable for the user who presses the “use this” button.

- **Primary Actor** **User**
- **Goal in Context** **To make the drawia watched, voted and used by users**
- **Preconditions** **User should be logged in**
- **Trigger** **User wants to view drawia at his/her profile or news feed**

3.2.2.1 Specific Functionalities

Show Animation of Drawium

Diagram

Figure 9. Diagram of Show animation of a drawium

Description

This button lets users to watch the animation of a drawium.

- **Primary Actor** **User**
- **Goal in Context** **To let the users to watch animations of drawia**
- **Preconditions** **User should be logged in**
- **Trigger** **User wants to watch the animation of a drawium**

Normal Flow of Events

1- User logs in to the system with correct credentials.
2- User is authenticated and forwarded to the news feed page.
3- User clicks on the the icon to watch the animation of a drawium
4- The animation of the drawium will be showed to user

Alternative Flow of Events

1- User logs in to the system with correct credentials.
2- User is authenticated and forwarded to the news feed page.
3- User views another user's profile.
4- User clicks on the the icon to watch the animation of a drawium
5- The animation of the drawium will be showed to user

Functional Requirements

The system shall check if the drawium that's being animated is marked as private in which case it is not going to allow the user to see the drawium or the animation of it.

If the browser of the user is incompatible, this module should 'catch' the error and shall not create a visible error.

Non-Functional Requirements

The drawing should not create too much load on a user's computer while it is being rendered in the browser.

Upvote

Diagram

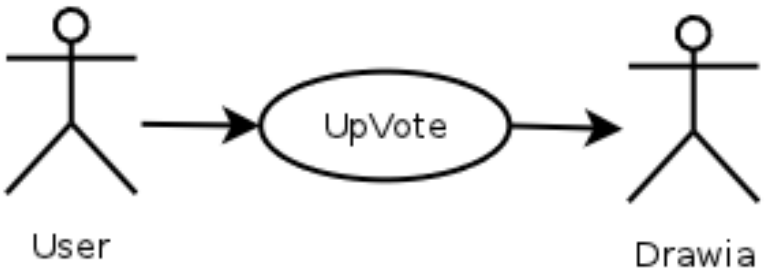

Figure 10. Diagram of Upvote

Description

The users should be able to upvote other users' drawia. When a user upvotes a drawium, that drawium will be displayed at his/her profile and number of votes of that drawium will be increased by one. The upvoted drawium will be available to use for the user. Each user will be able to upvote a drawium only one time.

- **Primary Actor** **User**
- **Goal in Context** **To make drawia voted by other users and displayed the drawia in users' profile**
- **Preconditions** **User should be logged in**
- **Trigger** **User wants to upvote a drawium**

Normal Flow of Events

1- User logs in to the system with correct credentials.
2- User is authenticated and forwarded to the news feed page.
3- User clicks upvote button of drawium that she/he wants to upvote
4- The drawium displayed at the user's profile
5- Number of votes increases by one of that drawium
6- The upvote button of that drawium becomes invisible for the user

Alternative Flow of Events

1- User logs in to the system with correct credentials.
2- User is authenticated and forwarded to the news feed page.
3- User views another user's profile.
4- User clicks upvote button of drawium that she/he wants to upvote
5- The drawium displayed at the user's profile
6- Number of votes increases by one of that drawium
7- The upvote button of that drawium becomes invisible for the user

Functional Requirements

The system should not allow users to upvote the same content more than once.

Non-Functional Requirements

When a user upvotes a content, he or she should be notified that the upvote is being sent to the backend server to be saved in the database (something like a loading sign should be used)

Use This

Diagram

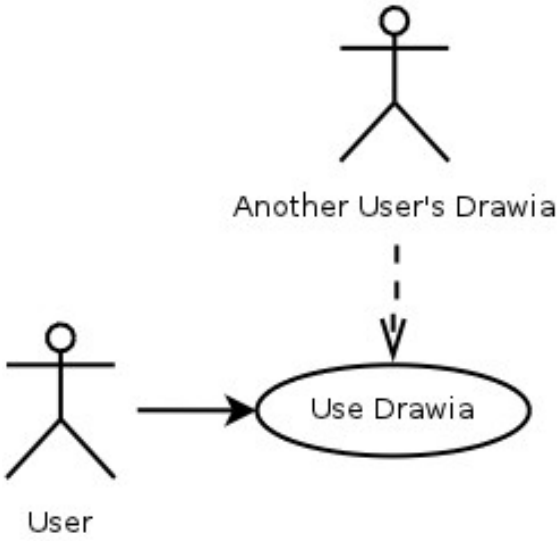

Figure 11. Diagram of Use this

Description

The users should be able to use other users' drawia. When a user clicks the "use this" button of a drawium, that drawium will be available to use for the user and displayed at his/her profile.

- **Primary Actor** **User**
- **Goal in Context** **To make the drawia available to use for other users and displayed the drawia in users' profile**
- **Preconditions** **User should be logged in**
- **Trigger** **User wants to use a drawium**

Normal Flow of Events

1- User logs in to the system with correct credentials.
2- User is authenticated and forwarded to the news feed page.
3- User clicks "use this" button of drawium that she/he wants to use.
4- The drawium displayed at the user's profile.
5- Number of users uses that drawium increases by one.
6- The "use this" button of that drawium becomes invisible for the user.

Alternative Flow of Events

1- User logs in to the system with correct credentials.
2- User is authenticated and forwarded to the news feed page.
3- User views another user's profile.
4- User clicks "use this" button of drawium that she/he wants to use.

5- The drawium displayed at the user's profile.
6- Number of users uses that drawium increases by one.
7- The "use this" button of that drawium becomes invisible for the user.

Functional Requirements

The system shall check to see if this button has been used before, and it shall make sure that it does not add the same drawium more than once to the list of drawia that the user can use.

Non-Functional Requirements

The user should be able to click on at least a hundred different "Use This" buttons without having serious performance issues while processing the "Use This" request and also while loading his profile page (where all the Drawia for which he clicked on Use This is going to appear in).

3.2.3 News Feed

Diagram

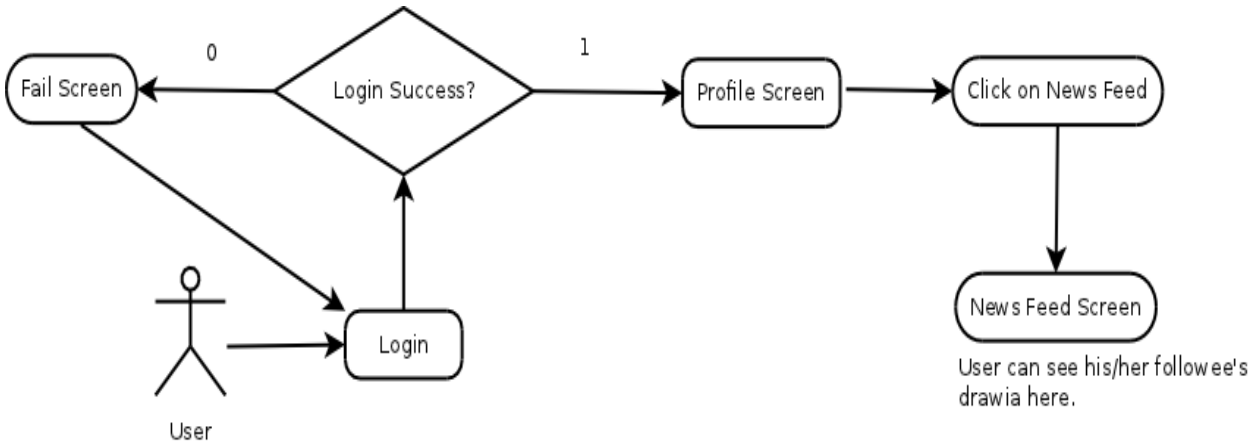

Figure 12. Diagram of News Feed Module

Description

The users will be forwarded to the news feed when they log in. News feed let the users to see the drawia of their followees.

- **Primary Actor** **User**
- **Goal in Context** **To notify the user about the user's followees' drawia**

- **Preconditions** **User should be logged in**
- **Trigger** **User is logged in or wants to display his/her news feed**

Normal Flow of Events

1- User logs in to the system with correct credentials.
2- User is authenticated and forwarded to the news feed page.
3- Display the drawia of the user's followees

Alternative Flow of Events

1- User clicks to New Feed button from anywhere in the web site.
2- Display the drawia of the user's followees

Functional Requirements

The system should make sure that the private drawia is not visible for other people in their news feeds.

Non-Functional Requirements

This should only show a subset of all the available content to make sure that news feed page loads are not taking too much time. (i.e. pagination)

3.2.4 Create JS Library Module

Diagram

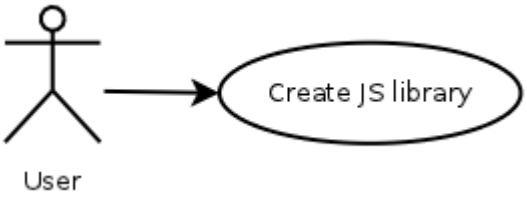

Figure 13. Diagram of Create JS Library Module

Description

The main product we are designing and building is a customizable Javascript Library that allows web designers to incorporate drawing elements into their websites. They will be able to use this product to, for instance, draw a circle around the Sign Up button that they want to get their users' attention to. Any drawia which the user shared in his/her profile and drawia he/she created will be available to be selected to be put in the library. The user will be able select the drawia that he/she wants to use in the JS

library out of these. Moreover, how to call the function of each drawium will be shown to the user. Finally the JS library will be created and presented to the user.

- **Primary Actor** **User**
- **Goal in Context** **To create a custom JS library to use for their websites.**
- **Preconditions** **User should be logged in. The profile should belong to the user who views this profile page.**
- **Trigger** **User wants to make use of our JS library solution.**

Normal Flow of Events

1- User logs in to the system with correct credentials
2- User is authenticated and forwarded to the news feed page.
3- User clicks on the profile link shown on the page.
4- User lands on the profile page.
5- User clicks on the Create a JS Library button.
6- User is forwarded to the Create a JS Library page.
7- User selects drawia that he/she wants to use
8- How to call the function of each drawium will be showed to user
9- The JS library will be created and presented to the user

Alternative Flow of Events

1- User logs in to the system with correct credentials
2- User is authenticated and forwarded to the news feed page.
3- User clicks on the profile link shown on the page.
4- User lands on the profile page.
5- User clicks on the Create a JS Library button.
6- User is forwarded to the Create a JS Library page.
7- User can click cancel at any step.
8- User is forwarded to the profile without saving changes.

Functional Requirements

The functions in the JS library shall take ID of an HTML element as parameter in order to attach the drawium to the HTML element.

Non-Functional Requirements

This page should not make the user click on so many buttons/radio buttons/check boxes and the user should not enter too much text to create a simple library. The interface that helps users customize the library should also be intuitive and should not be cumbersome.

3.2.5 Create new Drawium

Diagram

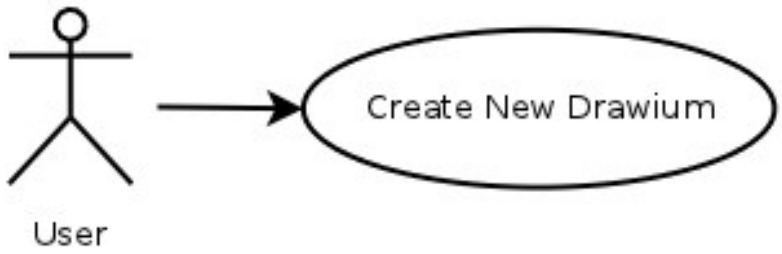

Figure 14. Diagram of Create new Module

Description

This module presents users a Drawium creation tool. User will be able to draw with a drawing tool. The user will be able to save his/her work. The new drawium of the user will be displayed in his/her profile. Moreover, the new drawium of the user will be displayed in the news feed of the user's followers. The tool that will be provided to the user will include basic text and drawing tools.

- **Primary Actor** **User**
- **Goal in Context** **To let the user to create a new drawium**
- **Preconditions** **User should be logged in. The profile should belong to the user who views this profile page.**
- **Trigger** **User wants to create a new Drawium.**

Normal Flow of Events

1- User logs in to the system with correct credentials
2- User is authenticated and forwarded to the news feed page.
3- User clicks on the profile link shown on the page.
4- User lands on the profile page.
5- User clicks the Create a New Drawium Button.
6- User is redirected to the the Drawium design page.
7- User draws a new drawium
8- User saves his/her working
9- The new drawium of the user is displayed in his/her profile
10- The drawium is displayed in the news feed of the user's followers

Alternative Flow of Events

1- User logs in to the system with correct credentials
2- User is authenticated and forwarded to the news feed page.
3- User clicks on the profile link shown on the page.
4- User lands on the profile page.
5- User clicks the Create a New Drawium Button.
6- User is redirected to the the Drawium design page.
7- User can click cancel at any step.
8- User is forwarded to the profile without saving changes.

Functional Requirements

The system shall attach a unique id to a drawium once it has been successfully created and all other systems shall use that id to refer to the particular drawia created including but not limited to the upvote, profile and the newsfeed systems.

Non-Functional Requirements

This tool should not be too CPU-intensive and should not lock the browser.

3.2.6 Administrator Access

Diagram

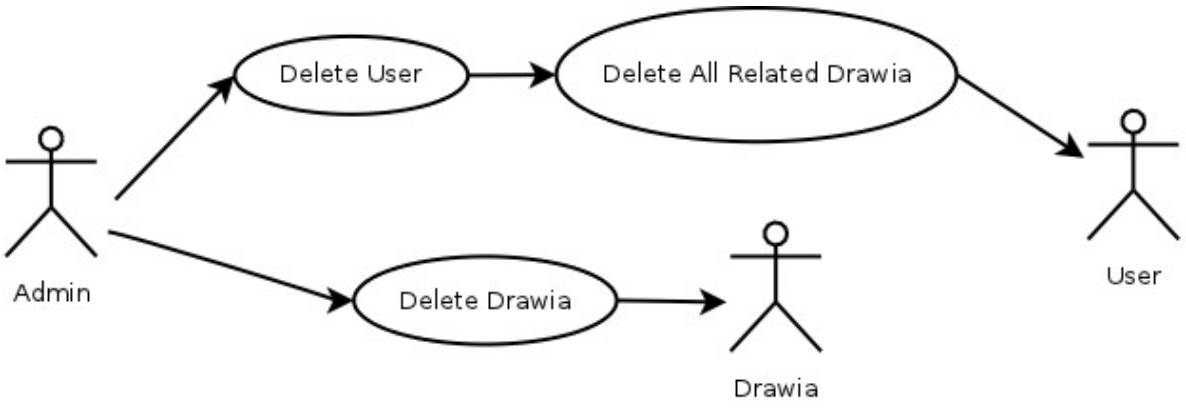

Figure 15. Diagram of Administrator Access Module

Description

Administrators are special users. They all features normal users have. The additional features of admins are:

- They can list all users.
- They can delete any user either using the news feed section or the user's profile.
- They can delete any drawium of any user.

3.2.6.1 Specific Functionalities

List all users

Diagram

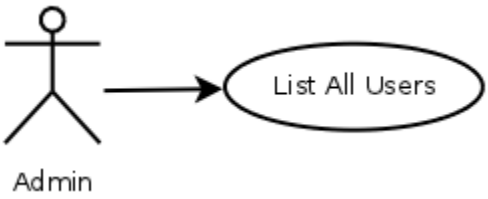

Figure 16. Diagram of List All Users

Description

This option lets the admins list all of the users so that admins can display the profiles of the users. Via this list the admins will be able access any user's profile page.

- **Primary Actor** **Administrator**
- **Goal in Context** **To let the admins to list all the users**
- **Preconditions** **Admin should be logged in.**
- **Trigger** **Admin wants to list all the users.**

Normal Flow of Events

1- Administrator logs in to the system with correct credentials
2- Administrator is authenticated and forwarded to the news feed page.
3- Administrator requests list of all users
4- The system displays the list of all users
5- Administrator may access any user's profile via the list

Functional Requirements

If users give any private data to drawium.com for any reason, administrators should not be able to see them without appropriate authorization by the website owners.

Non-Functional Requirements

The system work in such a way that when an administrator removes a content, it should be unavailable even for the creator of the content.

Delete user

Diagram

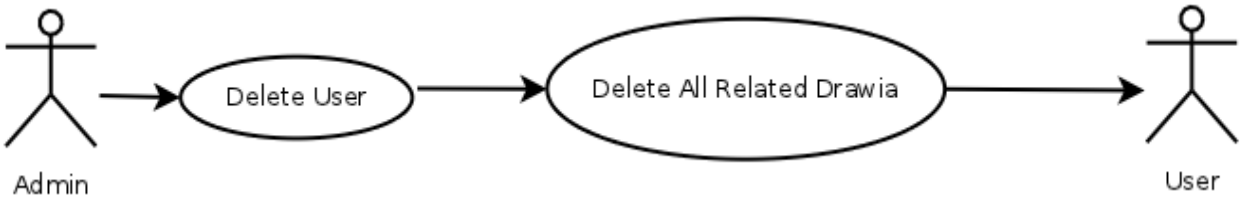

Figure 17. Diagram of Delete User

Description

Admins will be able to delete users who dont obey the rules of the website. When a user is deleted, the drawia of that user should be deleted automatically. In addition, the followers and followees of other user should be adjusted automatically as well.

- **Primary Actor** **Administrator**
- **Goal in Context** **To let the admins to delete any user**
- **Preconditions** **Admin should be logged in.**
- **Trigger** **Admin wants to delete a user.**

Normal Flow of Events

1- Administrator logs in to the system with correct credentials
2- Administrator is authenticated and forwarded to the news feed page.
3- Administrator requests list of all users
4- The system displays the list of all users
5- Administrator may access any user's profile via the list
6- Administrator deletes the user
7- Drawia of the deleted user is deleted by the system
8- The followers and followees of other users is adjusted

Alternative Flow of Events

1- Administrator logs in to the system with correct credentials
2- Administrator is authenticated and forwarded to the news feed page.
3- Administrator accesses any user's profile via his/her news feed
4- Administrator deletes the user
5- Drawia of the deleted user is deleted by the system
6- The followers and followees of other users is adjusted

Functional Requirements

Administrators should be able to delete any user. The information of who removed whom shall be kept in the database for future reference. When a user is removed, their data shall not be removed but be invisible as if they don't exist at all. This is for easily bringing back the users who were removed by a mistake.

Non-Functional Requirements

If the administrator receives a "User successfully removed" notification after trying to remove the user, that information shall always be correct, i.e, the system shall be 99.9% reliable for this function.

Delete drawium

Diagram

Figure 18. Diagram of Delete Drawium

Description

Admins will be able to delete drawia which are inappropriate such as drawia whose contents are insulting. When a drawium is deleted, it should also be removed automatically from the profiles and news feeds that are displaying that drawium.

- **Primary Actor** **Administrator**
- **Goal in Context** **To let the admins to delete any drawium**
- **Preconditions** **Admin should be logged in.**

- **Trigger** **Admin wants to delete a drawium.**

Normal Flow of Events

1- Administrator logs in to the system with correct credentials
2- Administrator is authenticated and forwarded to the news feed page.
3- Administrator accesses any user's profile
4- Administrator deletes the drawium
5- Drawia will be removed from the profiles and news feeds that are displaying that drawium.

Alternative Flow of Events

1- Administrator logs in to the system with correct credentials
2- Administrator is authenticated and forwarded to the news feed page.
3- Administrator deletes the drawium via the news feed page
4- Drawia will be removed from the profiles and news feeds that are displaying that drawium.

Functional Requirements

The drawium, once removed, shall not be available to the creator of it also. The drawium object shall not be removed from the database. It shall be marked as invisible and treated as invisible by the system from that point on.

4. DATA MODEL and DESCRIPTION

In this section the information of data models of software is mentioned.

4.1 Data Description

Basically, 4 types of data objects will be manipulated by the software. These data objects are namely user object, drawium object, js library object.

4.1.1 Data Objects

- **User Object: This object contains the user's information, namely the general information of user's (ID etc).**

- **Drawium Object: This object contains the drawium's informations, namely the ID and creation time of the drawium, the owner information of the drawium, the information of how many users upvote the drawium and the information of how many users use the drawium.**

- **Java Script Library Object: This object contains the JS Library's information, namely the ID and owner information of the JS and the information of the drawia inside it.**

4.1.2 Relationships

"User" object triggers the "JS library" and drawium object.

"JS Library" object waits data from drawium object.

4.1.3 Data Object Diagrams

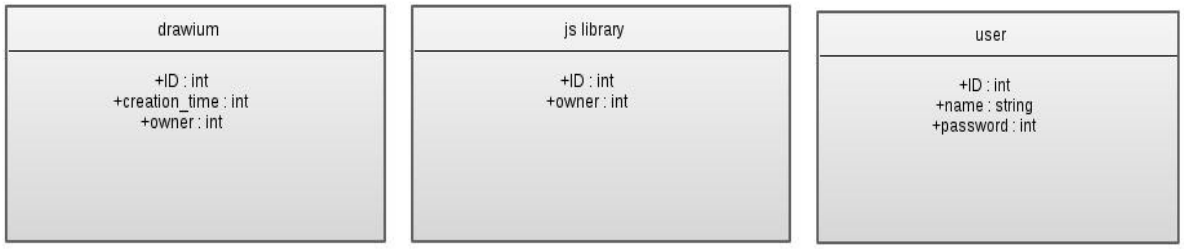

Figure 19. Data Objects

4.1.4 Complete Data Model

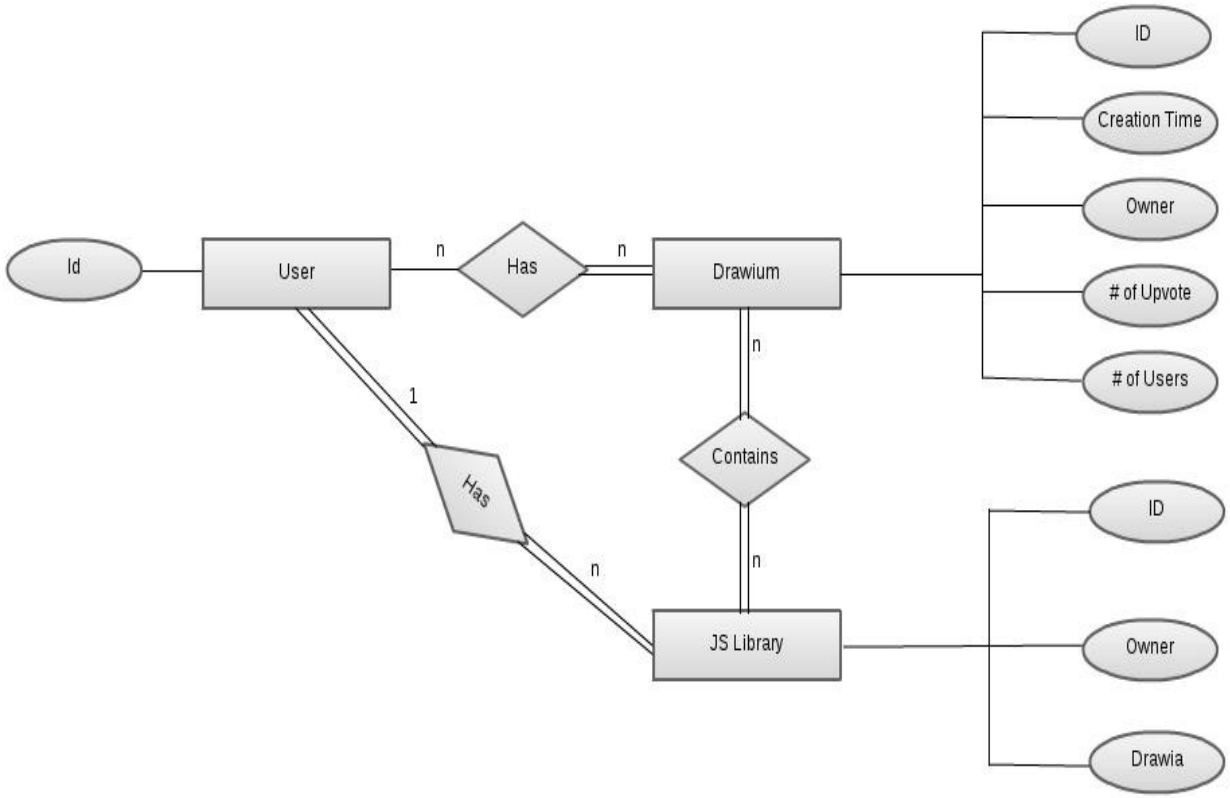

Figure 20. Entity Relationship Diagram

4.1.5 Data Dictionary

User

Field	***Type***	***Key?***
ID	int	Yes
name	string	No
password	int	No

Drawium

Field	***Type***	***Key?***
ID	int	Yes
creation time	int	No
owner	int	No

JS Library

Field	***Type***	***Key?***
ID	int	Yes
owner	int	No

5. BEHAVIORAL MODEL and DESCRIPTION

All behavioural models do is to describe how system works. It depicts the dynamic behavior of the system as a function of specific events and these specific events are triggered by the actors who use the system.

There are two types of actors in this system depending on their access levels.

1. Users
2. Admins

How they play the role in this system will be explained in the following section.

5.1 Description for Software Behavior

5.1.1 Users

After logging into drawium, the news feed page will be shown to the users . In this page, users can see the drawia which their followees have created. To use it later, users can display the drawia of their followees and they can choose “use this” option or just upvote the drawium if they like it. The users can go their followees profile from the news feed and they can see the followers and followees of their followees. They can

also see the all drawia of their followees and again they can choose “upvote” or “use this” or they can go the followers or followees of their followees and can also see that user’s profile. Another thing the users can do in news feed page is that they can go their profile page. User can create JS library, create a new drawium, display his/her drawia or other drawia posted in his/her profile, see their followees and followers in their profile page. After seeing their followees and followers page, users can follow if they do not follow that user or users can unfollow if they follow that user already.

5.1.2 Admins

After loging in to drawium, the news feed page will be shown to the admins, like the users. In this page, admins can do the same things the user can do. In addition to them, admins can display a drawium of their followees from their news feed and delete it. Admins can also list all users and delete the user they choose. As the admin deletes a user, all drawia of that user will be deleted automatically.

5.2 State Transition Diagrams

5.2.1 User

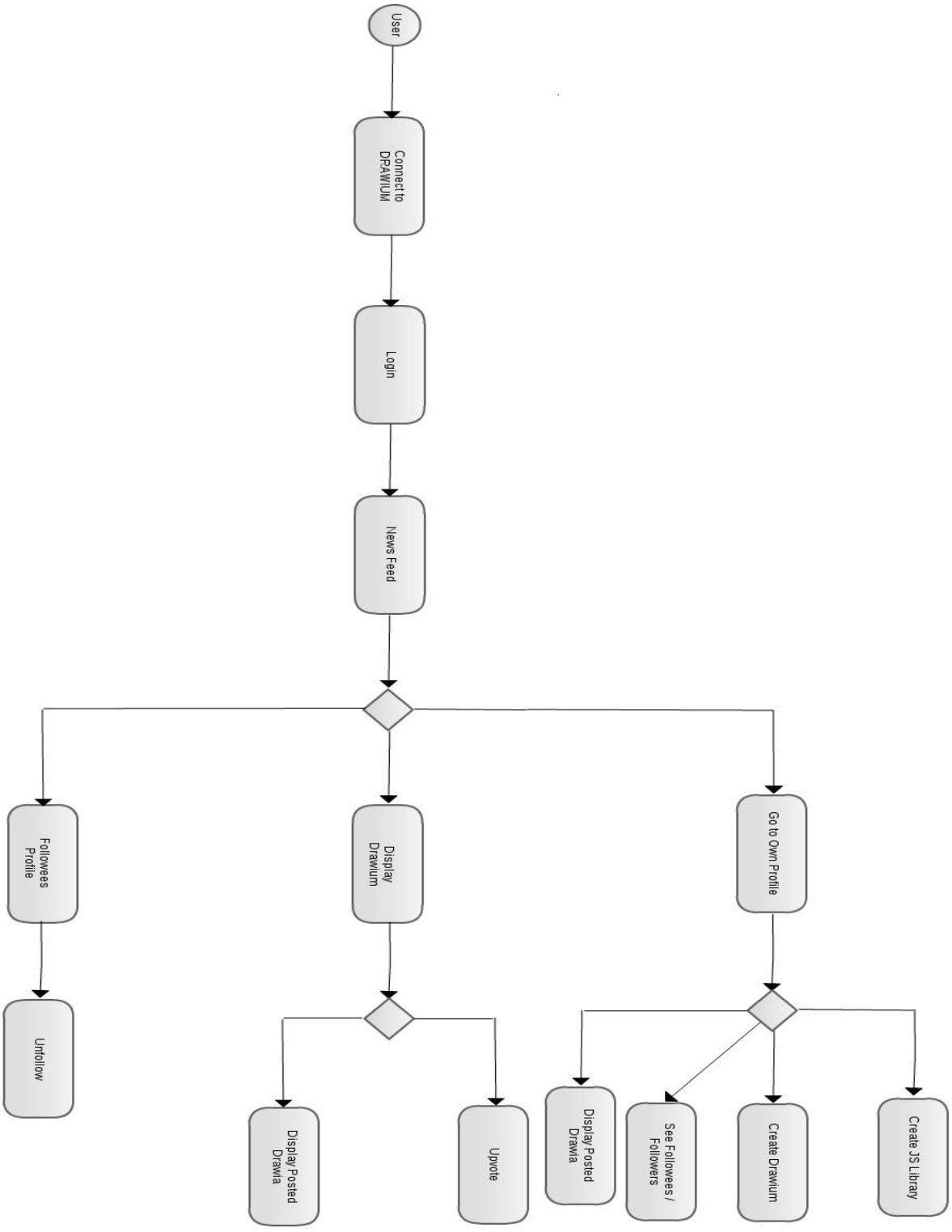

Figure 21. State Transition diagram of User

5.2.2 Admin

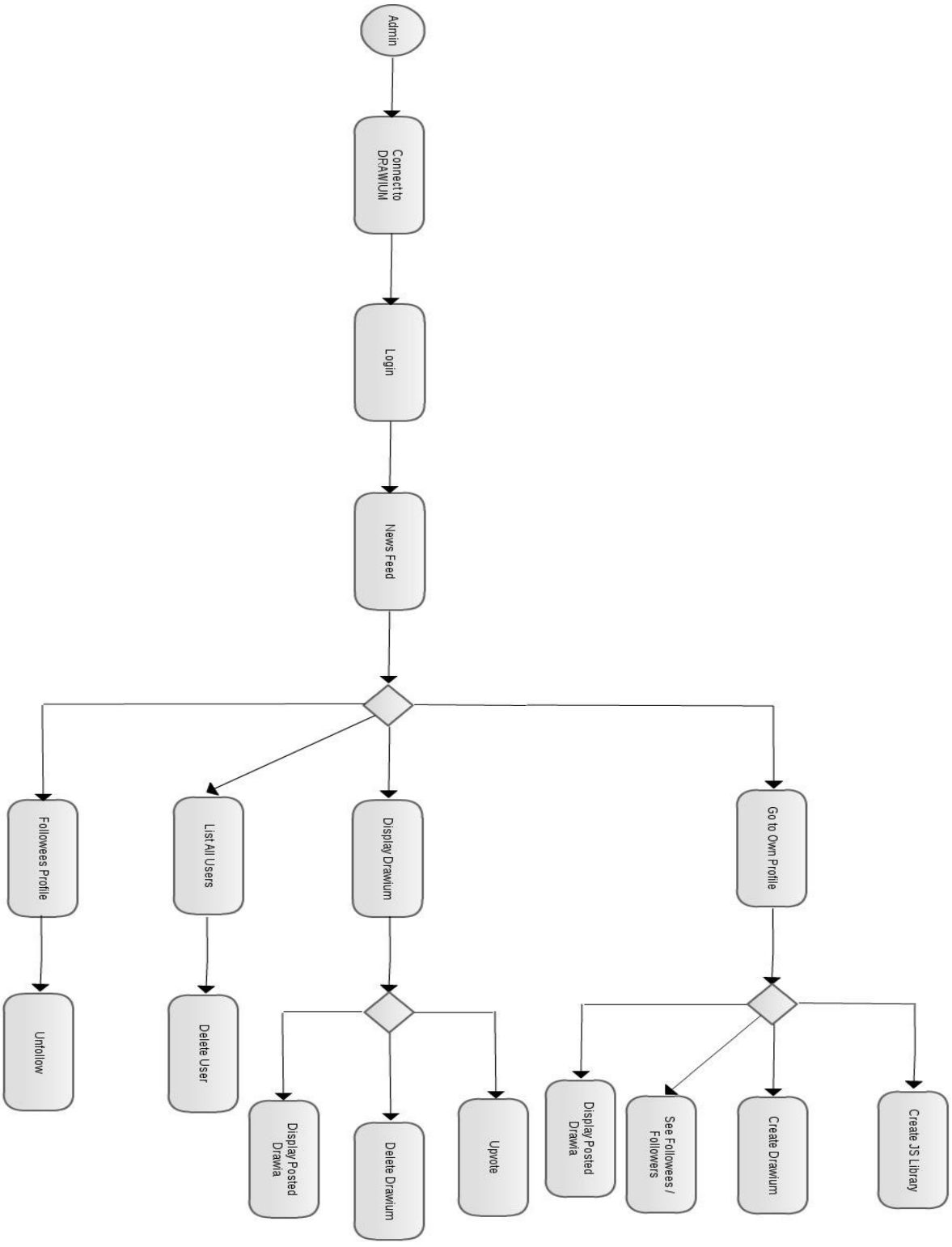

Figure 22. State Transition diagram of Admin

6. PLANNING

6.1 Team Structure

Seçkin Can Şahin : Technology Researcher, Software Engineering, Drawium Module Developer, Project Leader

Özgür Saygın Bican : Technology Researcher, Software Engineering, Drawium Module Developer, Public Relations, Task Manager

Mustafa Ozan Çelik : User Interface Designer, Profile Module Developer, Software Engineering, Reporter

Ozan Tahiroğlu : Profile Module Developer, Creating JS Module Developer, Software Engineering, Reporter

6.2 Basic Schedule

Please check the end of the document for gant chart.

6.3 Process Model

Although, waterfall process model is not the best model for our project, taking into consideration the course schedule waterfall model becomes a better choice. Hence our process model is the waterfall model. Waterfall process, is a sequential process which includes requirements analysis, software design, implementing, testing and maintenance. However, through the end of the project we would like to improve our product iteratively.

7.CONCLUSION

As we have mentioned, we are trying to make lives of the web developers a lot easier by solving a problem their users face when they first arrive to their websites. We are going to help the developers teach their websites in very short time which will ultimately result in less frustrated users. We believe that our solution is also going to yield a higher conversion rate as the users will better know what that random website that they landed on is actually trying to offer them.

We are eager to improve this proposed solution with feedback from you and the developers who use our tools and our website.

	1	2	3	4	5	6	7	8	9	10	10+
Simple drawing by a single person	X										
Intellectual Property(IP) Research	X	X	X								
Saving the drawings and serving them		X									
Saving drawing process and serving them.		X	X	X							
Developing the Drawia creation tool				X	X	X	X	X			
Developing the Drawia Library creation page					X	X	X	X	X		
Implementing IP protection methods			X								
First Drawia Library(the JS library) prototype			X	X	X	X					
Developing the prototype of Drawia Library							X	X	X	X	
Profiles for users				X	X	X					
Following mechanism for users					X						
Activity feed (newsfeed) of followed users							X	X	X		
Upvoting mechanism for drawia on profiles and the feed								X	X		
Iteration on the product										X	X