

# The *ComicVerse*<sup>TM</sup>

## Software Requirements Specification

# *The ComicVerse*<sup>TM</sup>

**Authors:** Daeun Park, Himchan Yun, Seokhoon Kim, Thomas Vetere  
Team White<sup>TM</sup>

**Based on IEEE std 830<sup>TM</sup>-1998 (R2009) document format**

Copyright © 2019 Team White

*No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher*

# **1 Introduction**

Traditionally, comics have been enjoyed through printed media such as newspapers and graphic novels. In the digital age, where a large amount of content is created and shared online, it seems natural for comics to migrate onto the web. Comics are an artistic medium through which people can express ideas, be humorous, or tell a story. They are meant to be read and enjoyed by people other than the author himself. Therefore, the act of publishing a comic is essentially a social interaction, one that can be hosted on a web application platform.

While it is true that people can currently share comics over social media and sites that host comics already exist, both amateur and professional comic writers are in need of a platform that can centralize all of the aspects of developing a comic: creating it, publishing it, and receiving commentary on it. Unfortunately, printed media limits the amount of contact that a comic writer has with his or her audience. If a comic writer wants to submit to a local newspaper, they must first create the comic using traditional artistic tools and then send it in. When it is finally published, newspaper readers cannot directly comment on the comic nor can they revisit the comic on demand should they throw away their newspaper. The entire process is both time consuming and tedious.

Our site seeks to streamline this entire process and even provides a platform for users to create their own versions of blank comics uploaded by other users. Our site seeks to extend the capabilities of current comic sites. This site does not aim to be merely a hosting service for comics, but also a collaborative and creative platform for users to develop and share their work.

***The ComicVerse*** will introduce a new paradigm into comic web applications by centralizing the entire process of creating and publishing a comic. The site will be functional, user-friendly, and will follow common web design patterns.

## **1.1 Purpose**

This document serves as a blueprint for ***The ComicVerse*** application. It describes in detail the functionality and the appearance of the site. The target audience for this document is the software development team as well as management personnel. All of the content presented in this document was constructed by the authors and the site's design was the result of a consensus between all parties involved. This document should give the reader a good sense of the structure of the application, how the application will function, and the needs that the application is intended to meet. It is important to note that the designs in this document serve as a guide for the development team and should not necessarily be taken verbatim.

## 1.2 Scope

The goal of this application is to provide a centralized web service for creating, publishing, and commenting on comics. The site should be accessible to both amateur and professional comic writers. It will follow good design practices for building web user interfaces and will stand out from competing sites by allowing users to edit and make their own versions of incomplete comics that are submitted by other users.

## 1.3 Definitions, acronyms, and abbreviations

**IEEE** – Institute of Electrical and Electronics Engineers, the “world’s largest professional association for the advancement of technology”.

**Java** -Java is a general purpose, high-level programming language. The source code files (files with a *.java* extension) are compiled into a format called *bytecode* (files with a *.class* extension), which can then be executed by a Java interpreter.

**UML** – Unified Modeling Language, a standard set of document formats for designing software graphically.

**Use Case Diagram** – A UML document format that specifies how a user will interact with a system. Note that these diagrams do not include technical details. Instead, they are fed as input into the design stage (stage after this one) where the appropriate software designs are constructed based in part on the Use Cases specified in the SRS.

## 1.4 References

**IEEE Std 830™-1998 (R2009)** - IEEE Recommended Practice for Software Requirements Specification

## 1.5 Overview

This document will describe in detail how *The ComicVerse* application should operate and look. The scope of this document is relatively high level and does not describe how the software should be implemented using certain technologies. For example, it does not present detailed UML class diagrams. Section 2 of this document provides an overarching description of the application. Section 3 describes specific software requirements in more detail. Section 4 contains a table of contents, an index, and addendices.

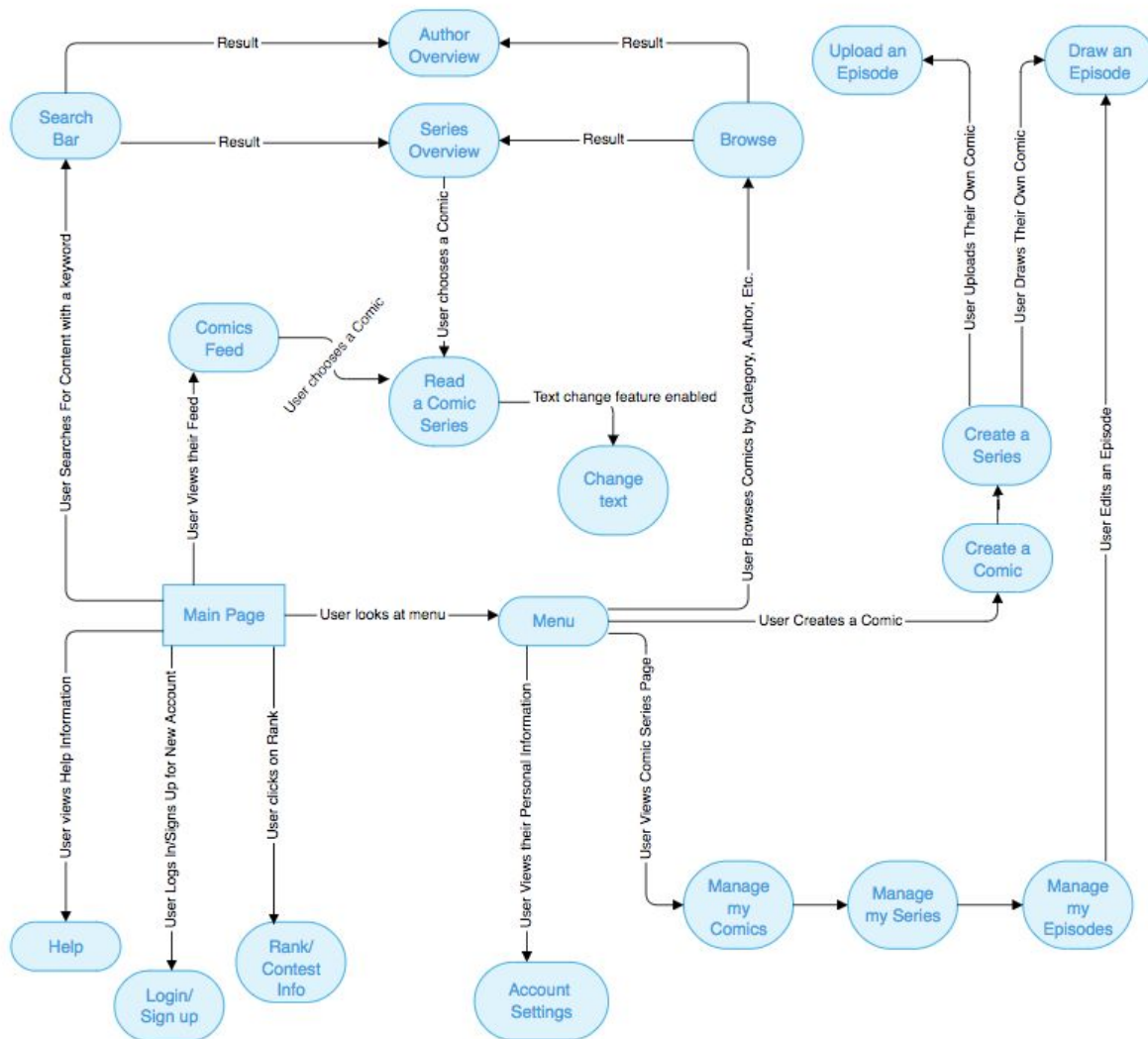
## **2. Overall description**

Since a large amount of content is created and shared online today, it makes sense to provide a platform for both amateur and professional comic writers to create and share their work. Printed media limits the audience that a comic writer can reach and also makes communication between readers and the author difficult. A web application to create, publish, and share comics would greatly increase the reach that comic authors have with respect to their audience and would serve as a platform where readers can comment on comics or quickly contact the author. Current comic hosting sites do exist already but what separates our site from the competition is that we also provide a means to create comics on our site and we foster creativity by including a feature where users can upload unfinished comics and have other users create their own versions of them by filling them in.

### **2.1 Product perspective**

Our *ComicVerse* application will centralize the entire process of creating, publishing, and commenting on comics. It is unique in that it allows users to edit unfinished comics published by other users. In order to access the services provided by our site, users must have a JavaScript enabled web browser and an Internet connection.

**Figure 2.1 Page Flow Diagrams**



### 2.1.1 System Interface

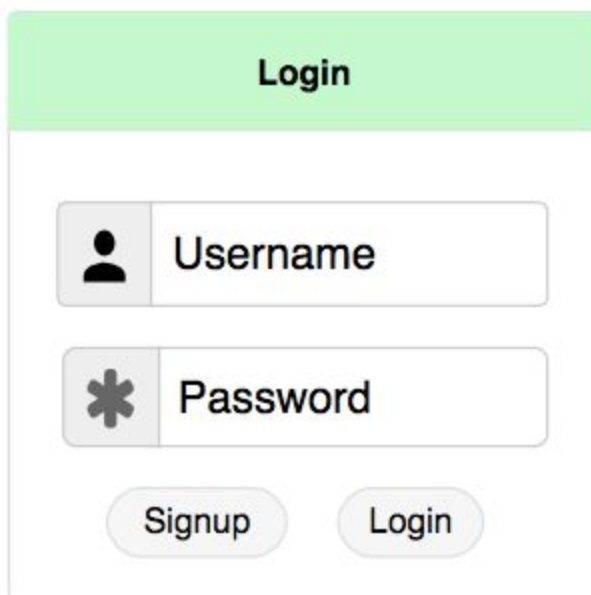
The *ComicVerse* website is strongly depending on many toolbars that are responsible for navigating pages and user-interacting actions. Figure 2.2 is Top Toolbar which user can use to navigate many different pages in the website. Since *ComicVerse* is the website that is consist of many pages, Top Toolbar will be really helpful for user to easily navigate between many pages. Figure 2.3 is Login Toolbar with which user can login or create account with. Mostly, login or sign up is what user will do first, so it is located in Home page and very visible. Figure 2.4 is Search Toolbar which will help users to easily find the comics that they want. Some users will

have specific comics or authors that they want to find. This toolbar will be really helpful for those to find desired comics or authors. Figure 2.5 is Drawing Toolbar which assist user to draw the comics. This website is centralized from making and reading comics, so sophisticated drawing toolbar is really important feature.

**Figure 2.2 Top Toolbar**



**Figure 2.3 Login Toolbar**

A login form with a green header labeled 'Login'. It contains two input fields: 'Username' with a person icon and 'Password' with an asterisk icon. Below the fields are two buttons: 'Signup' and 'Login'. The form is enclosed in a light gray border.

**Figure 2.3 Search Toolbar**

A search bar with the 'ComicVerse' logo on the left. The search input field has a dropdown menu labeled 'Title' and a 'Go!' button on the right. The search bar is a single line with a light gray border.

**Figure 2.4 Drawing Toolbar**



### **2.1.2 User Interfaces**

#### **Actor Description**

##### **All App User**

All App user is anyone who visits our website, including first time visitors, and logged in users.

All App User can be a reader, an author or both at the same time.

##### **Author**

Author is a logged in user who wants to create a comic series or has created some series. Author has exclusive access to management of comics.

##### **Reader**

Reader is an all app user who reads a comic series. A reader has limited access to some features such as commenting or likes when reader is not logged in.

**Figure 2.2** below summarizes all the user cases with which user can interact with our website, Comic Verse. All the UI interfaces are specifically introduced in section 3.1. Detailed explanation about all the user interfaces will be illustrated in Use Case UML Diagram in 2.1.2. Here is overview of use cases about our website.

Use Case #	UI Region / Screen	Use Case
2.1	Login Toolbar / Sign Up Page	<b>Application Sign Up</b>
2.2	Home / Login Toolbar	<b>Application Login</b>
2.3	Home / Login Toolbar	<b>Sign Out</b>
2.4	Account Setting / Top Tool Bar	<b>View Profile</b>
2.5	Account Setting	<b>Change Password</b>
2.6	Account Setting	<b>Delete Account</b>
2.7	Home / Suggested List Bar	<b>Read Comic Series From Suggested List</b>
2.8	Home / Search Bar	<b>Search Comic By Title</b>
2.9	Home / Search Bar	<b>Search Comic By Author</b>
2.10	Home / Search Bar	<b>Search Comic By All</b>
2.11	Upload Page	<b>Upload Episode</b>
2.12	Draw Page	<b>Upload Drawn Episode</b>
2.13	Create Page	<b>Create New Series</b>
2.14	Drawing Page	<b>Choose Pen Type</b>
2.15	Drawing Page	<b>Add Text Box</b>
2.16	Drawing Page	<b>Fill Paint</b>
2.17	Drawing Page	<b>Add Shape</b>
2.18	Drawing Page	<b>Add Image to the Canvas</b>
2.19	Drawing Page	<b>Choose Pen Color</b>
2.20	Drawing Page	<b>Zoom in Canvas</b>
2.21	Drawing Page	<b>Undo</b>



2.22	Episode Page	<b>Like Episode</b>
2.23	Episode Page	<b>Dislike Episode</b>
2.24	Episode Page	<b>Comment On Episode</b>
2.25	Episode Page	<b>Navigate to the previous episode</b>
2.26	Episode Page	<b>Navigate to the next episode</b>
2.27	Browse Page	<b>Browse by Categories</b>
2.28	Browse Page	<b>Browse by Author name(alphabetical)</b>
2.29	Browse Page	<b>Browse by Series name(alphabetical)</b>
2.30	Browse Page	<b>Read Comic Series In Browse Page</b>
2.31	Comic Series Page	<b>Subscribe a Comic Series</b>
2.32	Comic Series Page	<b>Unsubscribe a Comic Series</b>
2.33	Comic Series Page	<b>View Episode</b>
2.34	Managing Series Page	<b>Edit a Series</b>
2.35	Managing Series Page	<b>Delete a Series</b>
2.36	Managing Episodes Page / Edit Episode Page	<b>Edit an Episode</b>
2.37	Managing Episodes Page / Edit Episode Page	<b>Delete an Episode</b>
2.38	Contest Information Page	<b>View Contest</b>
2.42	Episode Page	<b>Fill in the bubbles</b>

### 2.1.3 Product functions

#### Use Case 2.1

Use Case:	Application Login
Primary Actor:	All App User
Goal in Context:	Have the user successfully login to the web application

Preconditions:	The user already has an account with the web application and is currently on the web app site.
Trigger:	The app user clicks on the “Login” button.
Scenario:	<ol style="list-style-type: none"> <li>1. App users navigates to the web application site and find login toolbar.</li> <li>2. App users fill in correct username and password in appropriate blank.</li> <li>3. App users clicks the login button.</li> <li>4. App user is successfully logged in if their credentials are correct, is re-prompted with error message if their credentials are invalid.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. If the user enters invalid login information, they will be notified and then re-prompted for their credentials</li> </ol>
Priority:	Essential, must be implemented
When Available:	First Benchmark
Frequency of Use:	Every time user login
Open Issues:	<ol style="list-style-type: none"> <li>1. Should we impose password requirements on the user?</li> <li>2. How can we implement Facebook/Google connect?</li> <li>3. How to implement password change?</li> </ol>

## Use Case 2.2

Use Case:	Application Signup
Primary Actor:	All App User
Goal in Context:	Have the user successfully sign up to web browser
Preconditions:	The user doesn't have any account registered to this website.
Trigger:	The app user clicks on the “Sign Up” button.
Scenario:	<ol style="list-style-type: none"> <li>1. App users navigates to the web application site and find login toolbar.</li> <li>2. App users clicks ‘sign up’ button under the toolbar.</li> <li>3. App users will be redirected to the sign in page.</li> <li>4. App users fill all the informations including username, password, and email that are needed for sign up.</li> <li>5. If same username or same email doesn't exist, user will be again redirected to homepage successfully logged in.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. If user put duplicate email or username, they will not be signed up and prompted a appropriate message.</li> </ol>
Priority:	Essential, must be implemented
When Available:	First Benchmark
Frequency of Use:	Every time new user comes in.

Open Issues:	<ol style="list-style-type: none"> <li>1. Should the username and password checked case-sensitively?</li> <li>2. Should there be any restrictions on password for security issue?</li> <li>3. How to implement synchronizing accounts with facebook or google.</li> </ol>
--------------	---

### Use Case 2.3

Use Case:	Application Sign Out
Primary Actor:	All App User
Goal in Context:	Have a user successfully signed out of web page
Preconditions:	The user is successfully logged in.
Trigger:	The App User clicks sign out button in the Login Toolbar
Scenario:	<ol style="list-style-type: none"> <li>1. The app user navigates to the home page.</li> <li>2. The app user clicks sign out button in the Login Toolbar.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. If the app user is not logged in to the website, they won't see sign out button in the Login Toolbar.</li> </ol>
Priority:	Essential, must be implemented
When Available:	First Benchmark
Frequency of Use:	Every time user wants to log out.
Open Issues:	N/A

### Use Case 2.4

Use Case:	View Profile
Primary Actor:	All App User
Goal in Context:	Have a user successfully view their personal informations
Preconditions:	The user is successfully logged in.
Trigger:	The App user clicks 'Account Setting' button on the Top Toolbar
Scenario:	<ol style="list-style-type: none"> <li>1. The App user navigates to the homepage of website</li> <li>2. The App user clicks 'Account Setting' button on the Top Toolbar</li> <li>3. The App user will be redirected to Account Setting page.</li> <li>4. The App will see all the information about the account there.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. If the app user is not logged in when they click 'Account Setting' button on the Top Toolbar, they will be prompted with pop up window with appropriate message.</li> </ol>
Priority:	Essential, must be implemented

When Available:	First Benchmark
Frequency of Use:	Not frequently.
Open Issues:	N/A

### Use Case 2.5

Use Case:	Change Password
Primary Actor:	All App User
Goal in Context:	Have a user to successfully change account password
Preconditions:	The user is successfully logged in.
Trigger:	The App user clicks change password button under the Account setting page.
Scenario:	<ol style="list-style-type: none"> <li>1. The App user navigates to the homepage of website.</li> <li>2. The App user clicks 'Account Setting' button on the Top Toolbar.</li> <li>3. The App user will be redirected to Account Setting page.</li> <li>4. The App user will click change password button located bottom of page.</li> <li>5. The App user will be prompted with pop-up window asking new password.</li> <li>6. The App user will write new password in blank and lock change button to successfully change password.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. If the app user is not logged in when they click 'Account Setting' button on the Top Toolbar, they will be prompted with pop up window with appropriate message.</li> </ol>
Priority:	Not Essential, Extra functionality
When Available:	First Benchmark
Frequency of Use:	Not frequently.
Open Issues:	<ol style="list-style-type: none"> <li>1. Should users be authenticated before they can change password?</li> </ol>

### Use Case 2.6

Use Case:	Delete Account
Primary Actor:	All App User
Goal in Context:	Have a user to successfully change account password.
Preconditions:	The user is successfully logged in.
Trigger:	The App user clicks delete account button under the Account setting page.
Scenario:	<ol style="list-style-type: none"> <li>1. The App user navigates to the homepage of website</li> <li>2. The App user clicks 'Account Setting' button on the Top Toolbar</li> </ol>

	<ol style="list-style-type: none"> <li>The App user will be redirected to Account Setting page.</li> <li>The App user will click delete account button located bottom of page.</li> <li>The App user will be prompted with pop up window asking if user really wants to delete the account</li> <li>If user clicks ok, an account is successfully deleted</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>If the app user is not logged in when they click 'Account Setting' button on the Top Toolbar, they will be prompted with pop up window with appropriate message.</li> </ol>
Priority:	Not Essential. Extra functionality
When Available:	First Benchmark
Frequency of Use:	Not frequently.
Open Issues:	<ol style="list-style-type: none"> <li>Should users be authenticated before they can delete account.</li> </ol>

### Use Case 2.7

Use Case:	Read Comic Series From Suggested List
Primary Actor:	Reader
Goal in Context:	Access to a comic series from personalized suggested list of comic series.
Preconditions:	The user has successfully signed in and is on Home page.
Trigger:	Clicks on one of comics series from suggested list.
Scenario:	<ol style="list-style-type: none"> <li>Reader navigates to Home page</li> <li>Clicks on a comic series from suggested list</li> <li>Navigated to the selected series page.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>When the user is not signed in, prompt to sing in.</li> </ol>
Priority:	Not essential
When Available:	Third Benchmark
Frequency of Use:	Once per visit
Open Issues:	N/A

### Use Case 2.8

Use Case:	Search Comic By title
Primary Actor:	Reader
Goal in Context:	Search a specific comic series when the reader knows the exact or partial keyword of title of the series to search.

Preconditions:	The user is in main page.
Trigger:	Clicks on the search button on the search bar.
Scenario:	<ol style="list-style-type: none"> <li>1. The user navigates to main page.</li> <li>2. The user selects the search option as Title and enters the keyword.</li> <li>3. Clicks on the search button.</li> <li>4. Returns the result as a list of comic series matching the keyword.</li> </ol>
Exceptions:	<ul style="list-style-type: none"> <li>• If there is no matching result, show the empty list.</li> <li>• If the select option is not specified, search by all.</li> </ul>
Priority:	Essential, must be implemented
When Available:	Third Benchmark
Frequency of Use:	Often, once per visit
Open Issues:	N/A

### Use Case 2.9

Use Case:	Search a comic series By Author
Primary Actor:	Reader
Goal in Context:	Search a specific comic series when the reader knows the exact or partial keyword of author name of the series to search.
Preconditions:	The user is in main page.
Trigger:	Clicks on the search button on the search bar.
Scenario:	<ol style="list-style-type: none"> <li>1. The user navigates to main page.</li> <li>2. The user selects the search option as Author and enters the keyword.</li> <li>3. Clicks on the search button.</li> <li>4. Returns the result as a list of comic series matching the keyword.</li> </ol>
Exceptions:	<ul style="list-style-type: none"> <li>• If there is no matching result, show the empty list.</li> <li>• If the select option is not specified, search by all.</li> </ul>
Priority:	Essential, must be implemented
When Available:	Third Benchmark
Frequency of Use:	Often, once per visit
Open Issues:	N/A

### Use Case 2.10

Use Case:	Search a comic series By All
-----------	------------------------------

Primary Actor:	Reader
Goal in Context:	Search a specific comic series when the reader knows the exact or partial keyword of the series to search.
Preconditions:	The user is in main page.
Trigger:	Clicks on the search button on the search bar.
Scenario:	<ol style="list-style-type: none"> <li>1. The user navigates to main page.</li> <li>2. The user selects the search option as All and enters the keyword.</li> <li>3. Clicks on the search button.</li> <li>4. Returns the result as a list of comic series matching the keyword.</li> </ol>
Exceptions:	<ul style="list-style-type: none"> <li>• If there is no matching result, show the empty list.</li> <li>• If the select option is not specified, search by all.</li> </ul>
Priority:	Essential, must be implemented
When Available:	Third Benchmark
Frequency of Use:	Often, once per visit
Open Issues:	N/A

## Use Case 2.11

Use Case:	Upload Episode
Primary Actor:	Author
Goal in Context:	Upload Episode drawn offline
Preconditions:	The user has successfully signed in. The user has created a comic series and is on Upload page. The user has selected a comic series that the episode to be uploaded.
Trigger:	Clicks on Upload Button from a drop-down at menu bar
Scenario:	<ol style="list-style-type: none"> <li>1. Navigates to Upload page</li> <li>2. The user select a comic series that the episode to be uploaded</li> <li>3. Clicks on Upload button.</li> </ol>
Exceptions:	<p>If the user is not signed in, prompt to sign in.</p> <p>If the user has not selected a comic series, prompt to select.</p> <p>The image file should not be empty.</p>
Priority:	Essential
When Available:	Second benchmark
Frequency of Use:	Once per visit
Open Issues:	N/A

### Use Case 2.12

Use Case:	Upload Drawn Episode
Primary Actor:	Author
Goal in Context:	Draw and upload an Episode
Preconditions:	The user has successfully signed in. The user has created a comic series and is on Draw page. The user has drawn a series and selected a comic series that the episode to be uploaded.
Trigger:	Clicks on upload button
Scenario:	<ol style="list-style-type: none"><li>1. Navigates to Upload page</li><li>2. The user select a comic series that the episode to be uploaded.</li><li>3. The use has completed drawing.</li><li>4. Clicks on Upload button.</li></ol>
Exceptions:	If the user is not signed in, prompt to sign in. If the user has not selected a comic series, prompt to select. The drawing image should not be empty.
Priority:	Essential
When Available:	Second benchmark
Frequency of Use:	Once per visit
Open Issues:	N/A

### Use Case 2.13

Use Case:	Create New Series
Primary Actor:	Author
Goal in Context:	Create a new comic series.
Preconditions:	The user is successfully logged in and is on Create page. The user has selected category and entered the name of the series.
Trigger:	Clicks on Create button
Scenario:	<ol style="list-style-type: none"><li>1. The user navigates to Create page.</li><li>2. The user selects category</li><li>3. The user enters the name of comic series</li><li>4. The user clicks on Create button.</li><li>5. The new comic series is created.</li></ol>



Exceptions:	If the user is not signed in, prompt to sign in. If the user has not selected a category, prompt to select. If the user has not checked the name of comic series, prompt to check. If the name of comic series is duplicate, prompt that the name already exists.
Priority:	Essential
When Available:	Second benchmark
Frequency of Use:	Not often, once per several visits
Open Issues:	N/A

### Use Case 2.14

Use Case:	Choose Pen Type
Primary Actor:	Author
Goal in Context:	Have users to successfully change pen type when drawing on canvas.
Preconditions:	The user is successfully logged in and now on drawing page.
Trigger:	The app user click first button from top in Drawing Toolbar.
Scenario:	<ol style="list-style-type: none"> <li>1. App user navigates to drawing page.</li> <li>2. User clicks first button from the top in Drawing Toolbar.</li> <li>3. Many pen types that are available will be shown right by choose pen button.</li> <li>4. User clicks desired pen type.</li> </ol>
Exceptions:	N/A
Priority:	Essential. Must be implemented.
When Available:	Second Benchmark
Frequency of Use:	Everytime author wants to draw cartoons to upload.
Open Issues:	N/A

### Use Case 2.15

Use Case:	Add Textbox
Primary Actor:	Author
Goal in Context:	Have users to successfully add textbox to the cartoons when they are drawing.
Preconditions:	The user is successfully logged in and now on drawing page.
Trigger:	The app user click second button from top in Drawing Toolbar.

Scenario:	<ol style="list-style-type: none"> <li>1. App user navigates to drawing page.</li> <li>2. User clicks second button from the top in Drawing Toolbar.</li> <li>3. Cursor will change its shape to white square.</li> <li>4. User clicks and drags to the size which user desire while pressing.</li> <li>5. Textbox is added to the drawing canvas at desired location.</li> </ol>
Exceptions:	N/A
Priority:	Essential. Must be implemented.
When Available:	Second Benchmark
Frequency of Use:	Everytime author wants to draw cartoons to upload.
Open Issues:	N/A

### Use Case 2.16

Use Case:	Fill Paint
Primary Actor:	Author
Goal in Context:	Have users to successfully use paint to fill the region enclosed by pen.
Preconditions:	The user is successfully logged in and now on drawing page.
Trigger:	The app user click third button from top in Drawing Toolbar.
Scenario:	<ol style="list-style-type: none"> <li>1. App user navigates to drawing page.</li> <li>2. User clicks third button from the top in Drawing Toolbar.</li> <li>3. Cursor will change to paint shape.</li> <li>4. User clicks area which user wants to pour paint on.</li> <li>5. All the area that is enclosed by pen sketch will be painted to same color.</li> </ol>
Exceptions:	N/A
Priority:	Essential. Must be implemented.
When Available:	Second Benchmark
Frequency of Use:	Everytime author wants to draw cartoons to upload.
Open Issues:	N/A

### Use Case 2.17

Use Case:	Add Shape
Primary Actor:	Author
Goal in Context:	Have users to successfully add shape to the drawing canvas.

Preconditions:	The user is successfully logged in and now on drawing page.
Trigger:	The app user click fourth button from top in Drawing Toolbar.
Scenario:	<ol style="list-style-type: none"> <li>1. App user navigates to drawing page.</li> <li>2. User clicks fourth button from the top in Drawing Toolbar.</li> <li>3. List of shapes that are available will be shown right side of button.</li> <li>4. User clicks and drags to the size which user desire while pressing.</li> <li>5. New Shape is added to the drawing canvas at desired location.</li> </ol>
Exceptions:	N/A
Priority:	Essential. Must be implemented.
When Available:	Second Benchmark
Frequency of Use:	Everytime author wants to draw cartoons to upload.
Open Issues:	N/A

### Use Case 2.18

Use Case:	Add image to Canvas
Primary Actor:	Author
Goal in Context:	Have users to successfully add image to canvas.
Preconditions:	The user is successfully logged in and now on drawing page.
Trigger:	The app user click seventh button from top in Drawing Toolbar.
Scenario:	<ol style="list-style-type: none"> <li>1. App user navigates to drawing page.</li> <li>2. User clicks seventh button from the top in Drawing Toolbar.</li> <li>3. User will be prompted with choose file pop up window where user can choose image to add.</li> <li>4. User drag image to the desired location</li> </ol>
Exceptions:	N/A
Priority:	Essential. Must be implemented.
When Available:	Second Benchmark
Frequency of Use:	Everytime author wants to draw cartoons to upload.
Open Issues:	N/A

### Use Case 2.19

Use Case:	Choose Pen Color
-----------	------------------

Primary Actor:	Author
Goal in Context:	Have users to successfully change pen color to draw.
Preconditions:	The user is successfully logged in and now on drawing page.
Trigger:	The app user click fifth button from top in Drawing Toolbar.
Scenario:	<ol style="list-style-type: none"> <li>1. App user navigates to drawing page.</li> <li>2. User clicks fifth button from the top in Drawing Toolbar.</li> <li>3. List of colors that are available will be shown right side of button.</li> <li>4. User clicks desired color.</li> </ol>
Exceptions:	N/A
Priority:	Essential. Must be implemented.
When Available:	Second Benchmark
Frequency of Use:	Everytime author wants to draw cartoons to upload.
Open Issues:	N/A

## Use Case 2.20

Use Case:	Zoom in canvas
Primary Actor:	Author
Goal in Context:	Have users to successfully zoom in the canvas.
Preconditions:	The user is successfully logged in and now on drawing page.
Trigger:	The app user click sixth button from top in Drawing Toolbar.
Scenario:	<ol style="list-style-type: none"> <li>1. App user navigates to drawing page.</li> <li>2. User clicks sixth button from the top in Drawing Toolbar.</li> <li>3. The canvas will zoomed in by 10% every click.</li> <li>4. If canvas is zoomed in by any ratio, there will be zoom out button located right by zoom in button.</li> </ol>
Exceptions:	N/A
Priority:	Essential. Must be implemented.
When Available:	Second Benchmark
Frequency of Use:	Everytime author wants to draw cartoons to upload.
Open Issues:	N/A

## Use Case 2.21

Use Case:	Undo
Primary Actor:	Author
Goal in Context:	Have users to successfully undo last action made to canvas.
Preconditions:	The user is successfully logged in and now on drawing page.
Trigger:	The app user click last button from top in Drawing Toolbar.
Scenario:	<ol style="list-style-type: none"> <li>1. App user navigates to drawing page.</li> <li>2. User clicks last button from the top in Drawing Toolbar.</li> <li>3. The last action user made to canvas will be rolled back.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. If user made no change to the canvas yet, undo button will do nothing.</li> </ol>
Priority:	Essential. Must be implemented.
When Available:	Second Benchmark
Frequency of Use:	Everytime author wants to draw cartoons to upload.
Open Issues:	N/A

## Use Case 2.22

Use Case:	Like an episode
Primary Actor:	Reader
Goal in Context:	Like an episode to give feedback to author and to evaluate the quality of the episode.
Preconditions:	The user is successfully signed in. The user is in an episode page.
Trigger:	The user clicks on the like button.
Scenario:	<ol style="list-style-type: none"> <li>1. The user navigates to an episode page.</li> <li>2. Clicks on the like button.</li> <li>3. The episode is successfully liked with prompt.</li> </ol>
Exceptions:	If the user is not signed in, prompt the user to sign in. If the user already likes the episode, undo like with prompt.
Priority:	Essential, must be implemented
When Available:	3rd benchmark
Frequency of Use:	Often, once per read
Open Issues:	

### Use Case 2.23

Use Case:	Dislike an episode
Primary Actor:	Reader
Goal in Context:	Dislike an episode to give feedback to author and to evaluate the quality of the episode.
Preconditions:	The user is successfully signed in. The user is in an episode page.
Trigger:	The user clicks on the dislike button.
Scenario:	<ol style="list-style-type: none"><li>1. The user navigates to an episode page.</li><li>2. Clicks on the dislike button.</li><li>3. The episode is successfully disliked with prompt.</li></ol>
Exceptions:	If the user is not signed in, prompt the user to sign in. If the user already likes the episode, undo dislike with prompt
Priority:	Essential, must be implemented
When Available:	3rd benchmark
Frequency of Use:	Often, once per read
Open Issues:	

### Use Case 2.24

Use Case:	Comment on an episode
Primary Actor:	Reader
Goal in Context:	Comment on an episode
Preconditions:	The user is successfully signed in. The user is in an episode.
Trigger:	The user clicks on Comment button.
Scenario:	<ol style="list-style-type: none"><li>1. The user navigates to the episode page.</li><li>2. Enters the comment on textbox.</li><li>3. Clicks on the Comment.</li></ol>
Exceptions:	If the user is not signed in, prompt to sign in.
Priority:	Not essential
When Available:	Third benchmark
Frequency of Use:	Once per read
Open Issues:	N/A

### Use Case 2.25

Use Case:	Navigate to the previous episode
Primary Actor:	Reader
Goal in Context:	The user can navigate to the previous episode in an episode page.
Preconditions:	The user is in an episode page.
Trigger:	Clicks on the previous button.
Scenario:	<ol style="list-style-type: none"><li>1. The user navigates to an episode page.</li><li>2. The user clicks on the previous button.</li><li>3. The user successfully navigates to the previous episode page.</li></ol>
Exceptions:	<ol style="list-style-type: none"><li>1. If the user clicks on the previous button where the current episode is the first episode, prompt the user that this is the first episode.</li></ol>
Priority:	Not essential, helpful to reader
When Available:	Fourth benchmark
Frequency of Use:	Often, once per read
Open Issues:	N/A

### Use Case 2.26

Use Case:	Navigate to the next episode
Primary Actor:	Reader
Goal in Context:	The user can navigate to the next episode in an episode page.
Preconditions:	The user is in an episode page.
Trigger:	Clicks on the previous button.
Scenario:	<ol style="list-style-type: none"><li>1. The user navigates to an episode page.</li><li>2. The user clicks on the next button.</li><li>3. The user successfully navigates to the next episode page.</li></ol>
Exceptions:	<ol style="list-style-type: none"><li>1. If the user clicks on the next button where the current episode is the last episode, prompt the user that this is the last episode.</li></ol>
Priority:	Not essential, helpful to reader
When Available:	Fourth benchmark
Frequency of Use:	Often, once per read
Open Issues:	N/A

### Use Case 2.27

Use Case:	Browse a comic series by Category
Primary Actor:	Reader
Goal in Context:	Reader browse a list of comic series when reader does not know the exact or partial keyword.
Preconditions:	The reader is in browse page.
Trigger:	Clicks on a specific category to browse.
Scenario:	<ol style="list-style-type: none"><li>1. The user navigates to browse page.</li><li>2. The user clicks on the specific category.</li><li>3. The sub-category or the list of a comic series that belongs to the specific category is displayed.</li></ol>
Exceptions:	N/A
Priority:	Essential, must be implemented
When Available:	Third Benchmark
Frequency of Use:	Often, once per visit
Open Issues:	N/A

### Use Case 2.28

Use Case:	Browse a comic series by Author name
Primary Actor:	Reader
Goal in Context:	Reader browse a list of comic series when reader does not know the exact or partial keyword but only knows the first letter of the author.
Preconditions:	The reader is in browse page.
Trigger:	Clicks on a alphabet under Author tab to browse.
Scenario:	<ol style="list-style-type: none"><li>1. The user navigates to browse page.</li><li>2. The user clicks on the specific alphabet that is the first letter of the author..</li><li>3. The list of authors is displayed.</li><li>4. When the user clicks on the name of author, the list of a comic series of the author will be displayed</li></ol>
Exceptions:	N/A
Priority:	Not Essential



When Available:	Third benchmark
Frequency of Use:	Often, once per visit
Open Issues:	N/A

### Use Case 2.29

Use Case:	Browse a comic series by Series name
Primary Actor:	Reader
Goal in Context:	Reader browse a list of comic series when reader does not know the exact or partial keyword but only knows the first letter of the author.
Preconditions:	The reader is in browse page.
Trigger:	Clicks on a alphabet under Author tab to browse.
Scenario:	<ol style="list-style-type: none"> <li>1. The user navigates to browse page.</li> <li>2. The user clicks on the specific alphabet that is the first letter of the author..</li> <li>3. The list of series is displayed.</li> </ol>
Exceptions:	N/A
Priority:	Not Essential
When Available:	3rd benchmark
Frequency of Use:	Often, once per visit
Open Issues:	

### Use Case 2.30

Use Case:	Read Comic Series In Browse Page
Primary Actor:	Reader
Goal in Context:	Reader browse a list of comic series and read desired comic series
Preconditions:	The reader is in browse page.
Trigger:	Clicks on a comic series that are listed on the browse page
Scenario:	<ol style="list-style-type: none"> <li>1. User clicks thumbnail of comic series that user wants to read.</li> </ol>
Exceptions:	N/A
Priority:	Not Essential

When Available:	3rd benchmark
Frequency of Use:	Often, once per visit
Open Issues:	N/A

### Use Case 2.31

Use Case:	Subscribe a comic series
Primary Actor:	Reader
Goal in Context:	Subscribe a comic series to receive updates regularly.
Preconditions:	The user is successfully signed in and is on a comic series page.
Trigger:	Clicks on subscribe button.
Scenario:	<ol style="list-style-type: none"> <li>1. The user navigates to a comic series page.</li> <li>2. Clicks on subscribe button.</li> <li>3. The comic series is added to the list of subscription of the user.</li> </ol>
Exceptions:	If not signed in, prompt the user to sign in.
Priority:	Essential, must be implemented
When Available:	Second benchmark
Frequency of Use:	Not often, once per several visits
Open Issues:	

### Use Case 2.32

Use Case:	Unsubscribe to comic series
Primary Actor:	Reader
Goal in Context:	Unsubscribe a subscribed comic series.
Preconditions:	The user is successfully logged in and subscribed the comic series that the user want to unsubscribe. The user is either on a comic series page or personal page with a subscription list.
Trigger:	The user clicks on the unsubscribe button.
Scenario:	<ol style="list-style-type: none"> <li>1. The user navigates to either the comic series page or personal page with a subscription list.</li> <li>2. Clicks on the unsubscribe button.</li> <li>3. The comic series is removed from the list of subscription.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. If the user is not signed in, prompt to sign in.</li> </ol>

	<ol style="list-style-type: none"> <li>2. If the comic series to unsubscribe is not subscribed, <ol style="list-style-type: none"> <li>a. On comic series page, the unsubscribe button is disabled and there is a subscribed button instead.</li> <li>b. On subscription list in personal page, the comic series is not there in the list.</li> </ol> </li> </ol>
Priority:	Essential, must be implemented
When Available:	Second benchmark
Frequency of Use:	Not often, once per several visits
Open Issues:	N/A

### Use Case 2.33

Use Case:	View Episode
Primary Actor:	Reader
Goal in Context:	Display an episode of a comic series, along with option to rate, comment, like, and navigate to previous or next episode.
Preconditions:	The app user is in a comic series page.
Trigger:	The app user clicks on an episode on the list of episodes.
Scenario:	<ol style="list-style-type: none"> <li>1. The app user navigates to the page of a comic series.</li> <li>2. The app user clicks on an episode.</li> <li>3. The selected episode is displayed with available options.</li> </ol>
Exceptions:	N/A
Priority:	Essential, must be implemented
When Available:	Second Benchmark
Frequency of Use:	Very often, Many times per visit
Open Issues:	N/A

### Use Case 2.34

Use Case:	Edit a Series
Primary Actor:	Author
Goal in Context:	The user choose a series that the user wants to delete and successfully gets redirected to page that has all the list of episode in the series

Preconditions:	The app user is in a Managing series page and user is successfully logged i.
Trigger:	The app user clicks on an episode on the list of episodes.
Scenario:	<ol style="list-style-type: none"> <li>1. The user navigates to Managing Series page</li> <li>2. The user clicks the pen looking button.</li> <li>3. The user is redirected to Managing Episode page.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. If user has no comic series that was uploaded before, nothing will be shown in Edit a Series page</li> </ol>
Priority:	Essential, must be implemented
When Available:	Second Benchmark
Frequency of Use:	Very often, Many times per visit
Open Issues:	N/A

### Use Case 2.35

Use Case:	Delete a Series
Primary Actor:	Author
Goal in Context:	Delete a series with all episodes.
Preconditions:	The user is signed in and is on Managing episodes page. The use has created a comic series and an episode.
Trigger:	The user clicks on Delete button.
Scenario:	<ol style="list-style-type: none"> <li>1. The user navigates to Managing Series page.</li> <li>2. The user clicks on Delete button on the series to be deleted.</li> <li>3. Prompt to make sure the deletion pop up.</li> <li>4. The user agrees to delete.</li> <li>5. The selected series is deleted.</li> </ol>
Exceptions:	<p>If the user is not signed in, prompt to sign in.</p> <p>If the user selected cancel on the prompt, cancel the deletion.</p>
Priority:	Essential
When Available:	2nd benchmark
Frequency of Use:	Not often
Open Issues:	N/A

### Use Case 2.36

Use Case:	Edit an episode
Primary Actor:	Author
Goal in Context:	Have the user successfully edit desired episode.
Preconditions:	The user already has an account with the web application and is currently on the Managing Episode Page
Trigger:	The app user clicks on the pen-looking button on episode that user wants to delete.
Scenario:	<ol style="list-style-type: none"> <li>1. App users navigates to the Managing Episode Page.</li> <li>2. App users click on the pen-looking button on the episode that users want to delete.</li> <li>3. App users will be redirected to Edit Episode Page.</li> <li>4. App users will be redirected to drawing page with current episode on it.</li> <li>5. User can use same UI for drawing to edit each episode.</li> </ol>
Exceptions:	N/A
Priority:	Essential, must be implemented
When Available:	Second Benchmark
Frequency of Use:	Very rarely
Open Issues:	<ol style="list-style-type: none"> <li>1. Should author be authenticated before they edit the episode?</li> </ol>

### Use Case 2.37

Use Case:	Delete an episode
Primary Actor:	Author
Goal in Context:	Have the user successfully delete whole episode.
Preconditions:	The user already has an account with the web application and is currently on the Managing Episode Page
Trigger:	The app user clicks on the pen-looking button on episode that user wants to delete.
Scenario:	<ol style="list-style-type: none"> <li>1. App users navigates to the Managing Episode Page.</li> <li>2. App users click on the pen-looking button on the episode that users want to delete.</li> <li>3. App users will be redirected to Edit Episode Page.</li> <li>4. App users click Delete Episode button.</li> <li>5. App user will be prompted with confirming message.</li> <li>6. If user clicks OK, then user will be redirected to the Managing Episode Page and selected episode is now deleted.</li> </ol>
Exceptions:	N/A
Priority:	Essential, must be implemented
When Available:	Second Benchmark

Frequency of Use:	Very rarely
Open Issues:	2. Should author be authenticated before they delete the episode?

### Use Case 2.38

Use Case:	View Contest
Primary Actor:	Reader, Author
Goal in Context:	Display an ongoing or past contest with time remaining, reward details, and ranking.
Preconditions:	The user is on Home page. At least one contest has been posted.
Trigger:	The user clicks on the name of a contest.
Scenario:	<ol style="list-style-type: none"> <li>1. The user navigates to the Home Page</li> <li>2. Clicks any part of Rank UI that is located in left side of homepage.</li> <li>3. User is redirected to View Contest Page.</li> <li>4. Display contest information including time remaining, rewards, and ranking.</li> </ol>
Exceptions:	If there is no contest posted, nothing will be displayed on the View Contest Page.
Priority:	Not essential
When Available:	Fourth benchmark
Frequency of Use:	Not often, once per visit
Open Issues:	N/A

### Use Case 2.39

Use Case:	Fill in the bubbles
Primary Actor:	Reader
Goal in Context:	Give user an option to change any text bubbles in each comics to make a parody.
Preconditions:	The user is logged in, and the episode that user is reading has “Fill in the bubble” option
Trigger:	The User clicks change text buttons which looks like alphabet “T”
Scenario:	<ol style="list-style-type: none"> <li>1. The app user is in Reading Episode Page</li> <li>2. If the episode that user is reading is offering option to fill in the bubbles, ‘T’ button right by the episode name will be enabled.</li> <li>3. User clicks ‘T’ button and all the dialogs in the bubble will be ready to be changed.</li> <li>4. User clicks the dialog that user wants to change and click “Save” to save the change.</li> </ol>
Exceptions:	1. If the episode is not offering such option, nothing will happen when user clicks “T”

	button located right side of episode name.
Priority:	Not Essential. Additional Functionality
When Available:	Fourth Benchmark
Frequency of Use:	Very Rarely.
Open Issues:	1. Should we save changed version of dialog to the database?

## 2.3 User characteristics

The typical user of this application would be someone that is at least 15 years of age and either is an amateur or professional comic writer. Some users might use the site simply to view comics, but all users will have the ability to create and publish a comic. Due to the large age range and the dichotomy between amateur and professional comic writers, the user interface must be easy enough for all users and must also incorporate design tools that both amateur and professional comic writers would utilize when creating comics. The site also might involve some proprietary material created by users and therefore must be able to protect these property rights. Each unique comic creation should be attributed to a user in order to maintain artistic integrity.

## 2.4 Constraints

This project will use a 750x650 canvas resolution for reading and creating comics. The reason for this selected resolution is that this is a common resolution for images on most social medias.

## 2.5 Assumptions and dependencies

Note that at the moment we are only developing a web application but only with 10 comics. The application should be developed, however, such that additional comics can be added without having to change any source code.

# 3 Specific requirements

## 3.1 External interfaces

The following wireframe mockups provide a look at the types of controls and layout to be used for the User Interface.

**Figure 3.1 Home (Default)**

Home	Browse Comics	Create a Comic	Manage my Comics	Account Settings
------	---------------	----------------	------------------	------------------

ComicVerse

Title ▼ Search... Go!

Login

Username

Password

Signup

Login

Rank

Serie 4

Serie 1

Serie 2

Serie 6

Help

help.comicverse@gmail.com

Serie 1 - Suggestion +

Serie 2 - Suggestion +

Serie 3 - Suggestion +

Serie 4 - Suggestion +



Figure 3.2 Home (User Signed In)

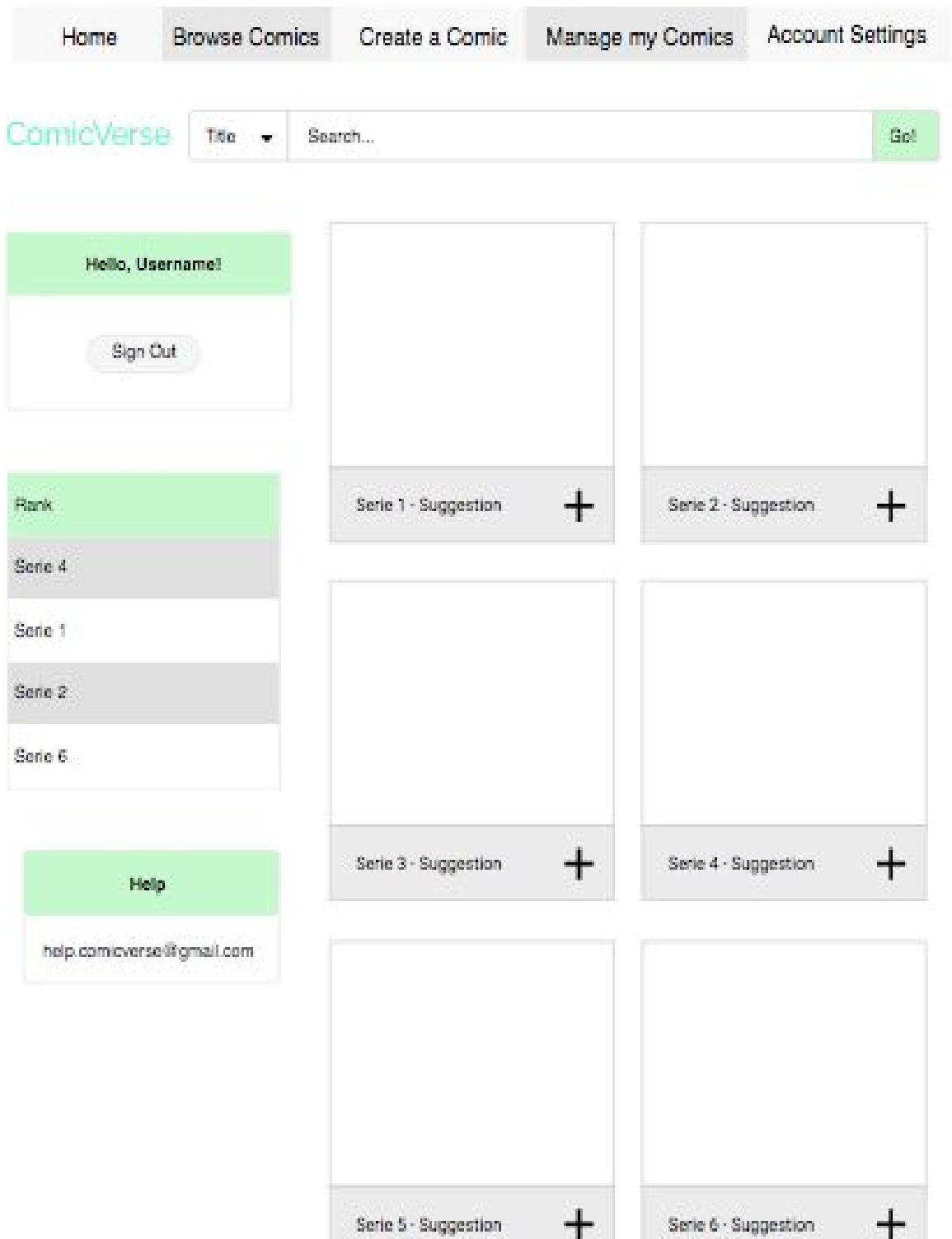


Figure 3.3 Home (With a drop down bar)

Home

Browse Comics

Create a Comic ▾

Create a Serie

Upload an Episode

Draw an Episode

Manage my Comics

Account Settings

ComicVerse

Title ▾

Go!

Login

Username

Password

Signup

Login

Rank

Serie 4

Serie 1

Serie 2

Serie 6

Help

help.comicverse@gmail.com

Serie 1 - Suggestion

+

Serie 2 - Suggestion

+

Serie 3 - Suggestion

+

Serie 4 - Suggestion

+

Serie 5 - Suggestion

+

Serie 6 - Suggestion

+

Figure 3.4 Browse

HomeBrowse ComicsCreate a ComicManage my ComicsAccount Settings

Browse

▼ Categories

☒ Cate 1

☒ Cate 2

☐ Cate 3

▼ Authors

☒ Author 1

☒ Author 2

☐ Author 3

▼ Ratings

☒ \*\*\*\*\*

☒ More than \*\*\*\*

☐ More than \*\*\*

Comic Serie 1+

Comic Serie 2✓

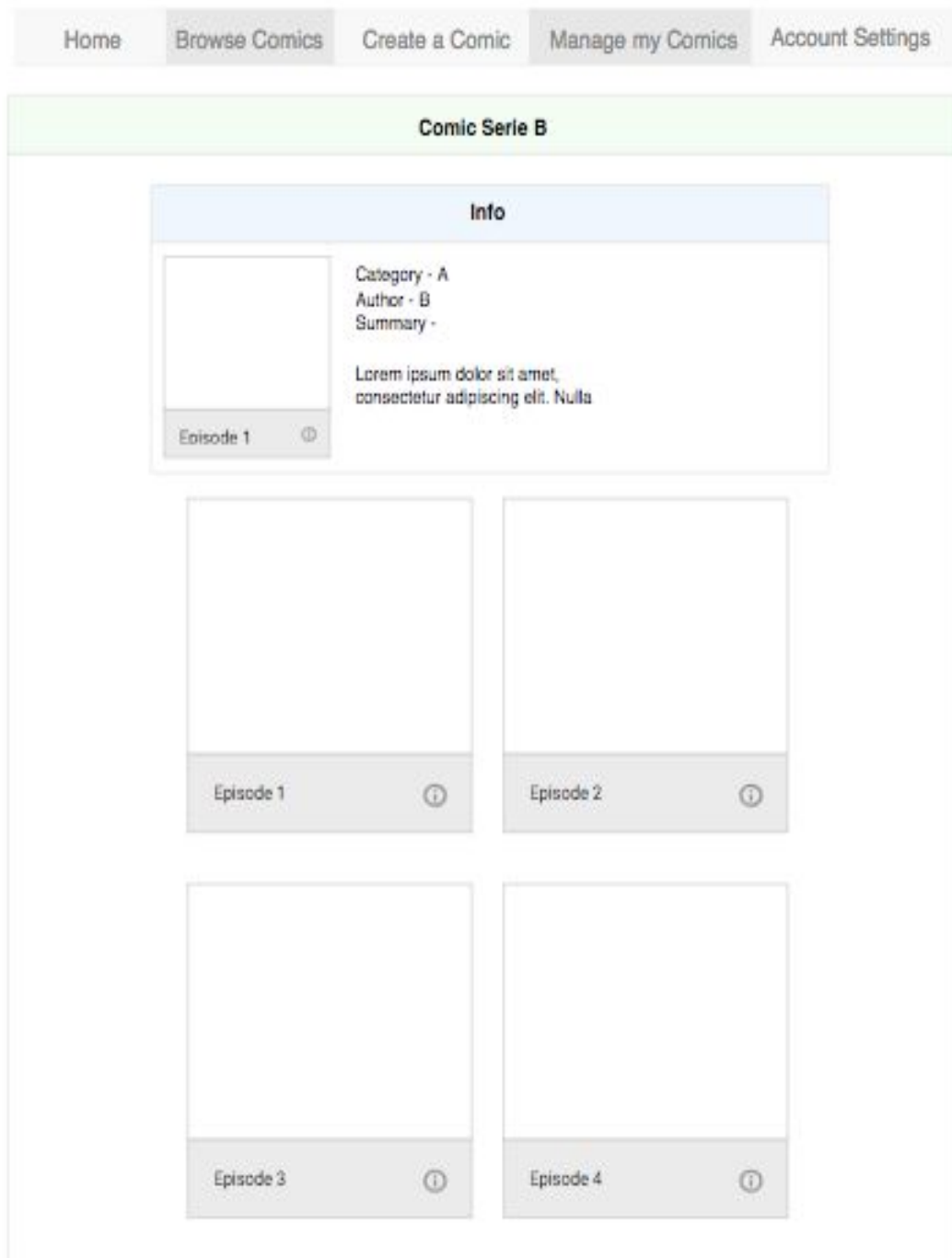
Comic Serie 2✓

Comic Serie 1+

Comic Serie 1+

Comic Serie 2✓

Figure 3.5 Read Comic Series



**Figure 3.6 Overview of an Author**



**Figure 3.7 Create a Comic Series**

[Home](#) [Browse Comics](#) [Create a Comic](#) [Manage my Comics](#) [Account Settings](#)

Create a Comic Serie

Comic Serie Name

Check

Category

▼

Create

**Figure 3.8 Upload an Episode**

[Home](#) [Browse Comics](#) [Create a Comic](#) [Manage my Comics](#) [Account Settings](#)

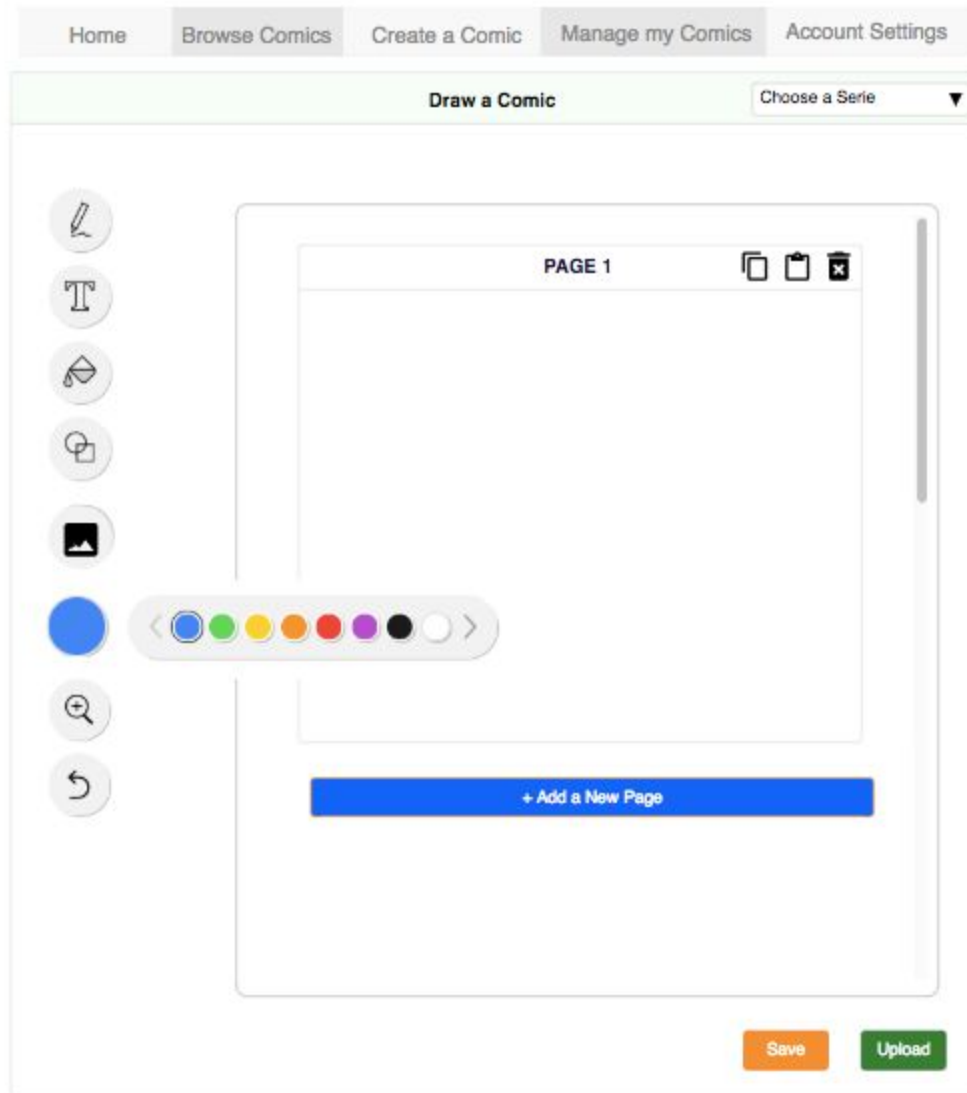
Upload an Episode

Choose a Serie

Choose a File

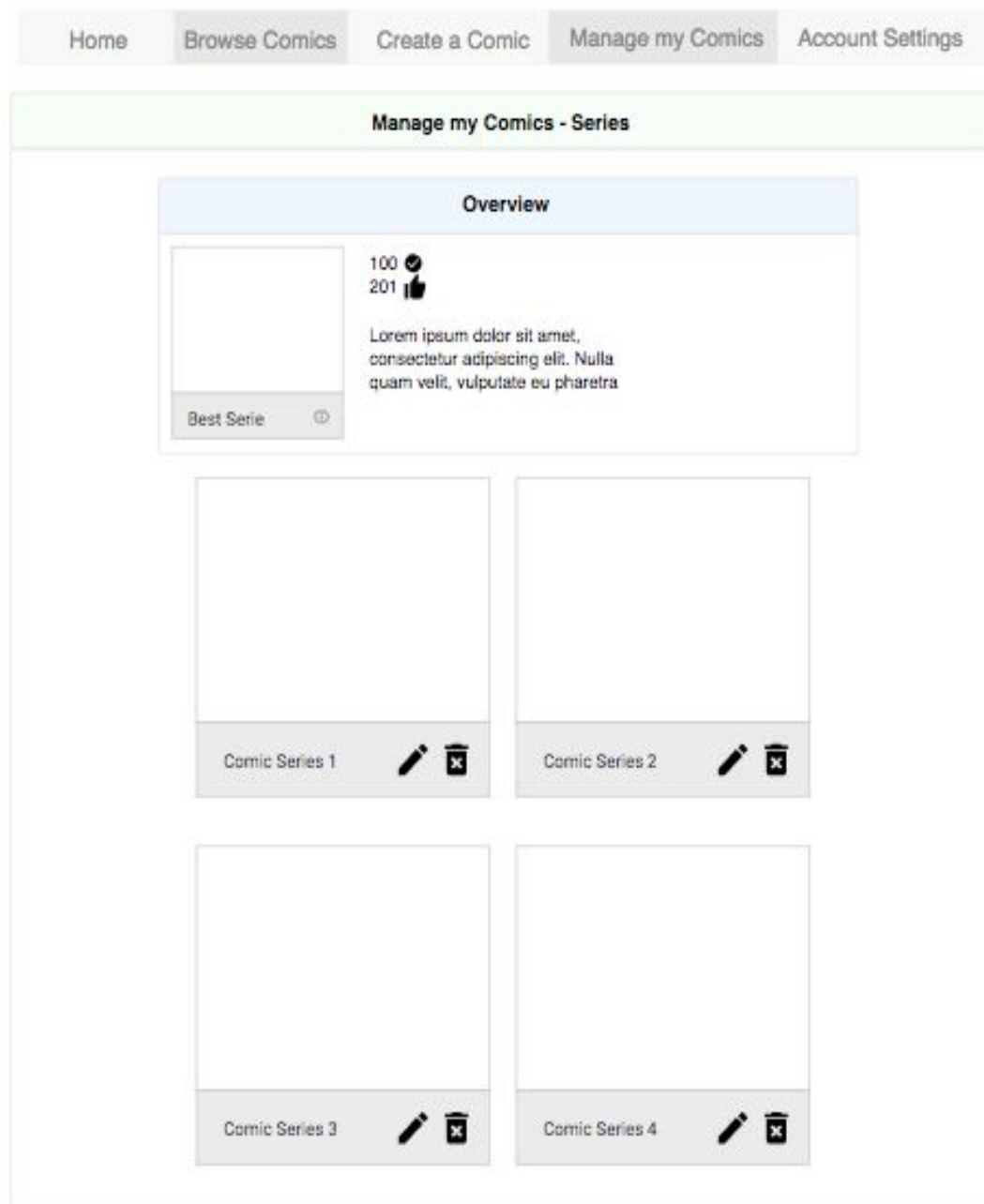
Upload

**Figure 3.9 Drawing Episode**





**Figure 3.10 Managing series**




**Figure 3.11 Manage Episodes**





**Figure 3.12 Account Settings**


[Home](#) [Browse Comics](#) [Create a Comic](#) [Manage my Comics](#) [Account Settings](#)

Personal Information

 Username

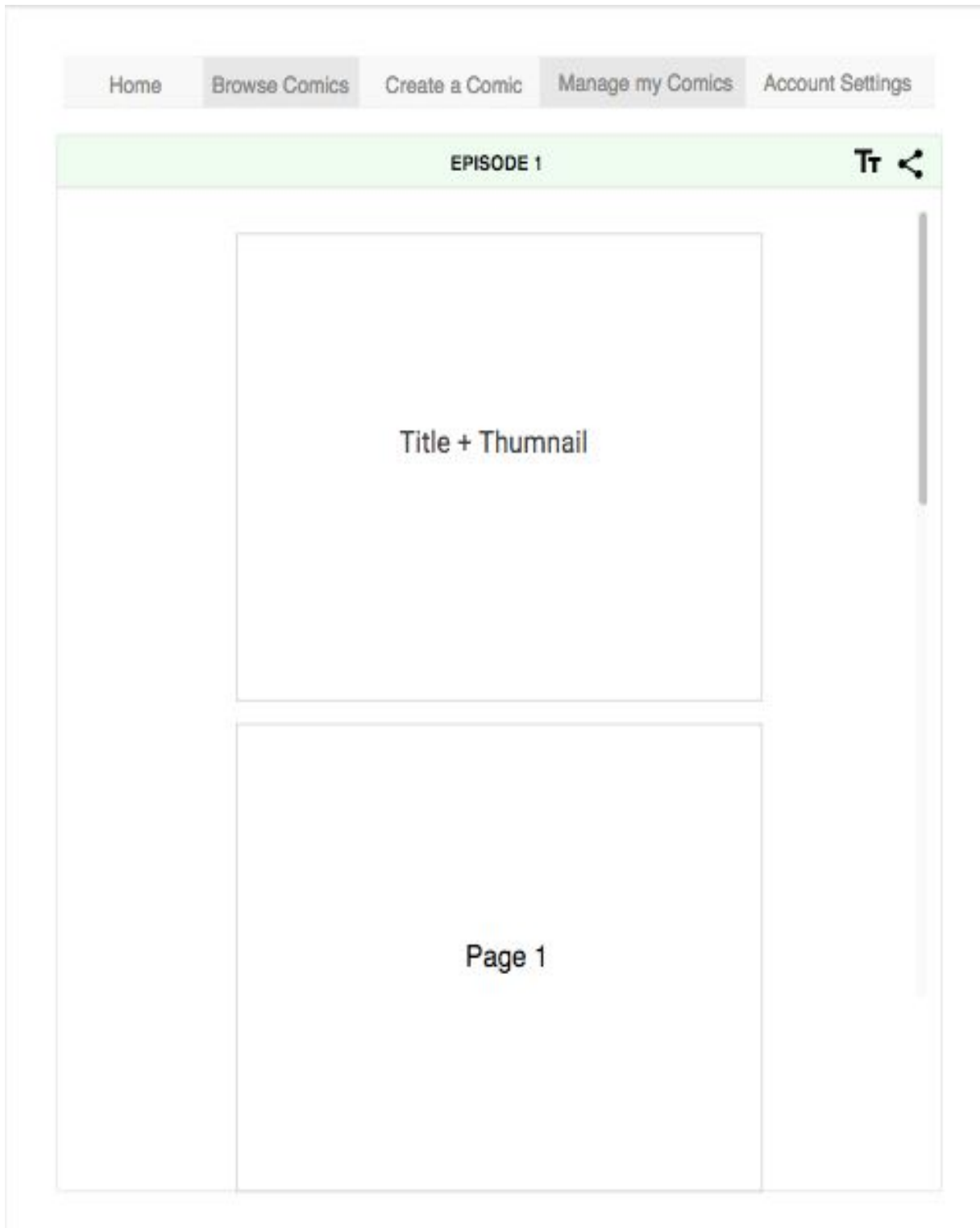
 Email

 Password [Change Password](#)

 Rewards

Delete Account

**Figure 3.13 Read an Episode - Top of the page**



**Figure 3.14 Read an Episode - Bottom of the page (Default - Text change feature disabled)**

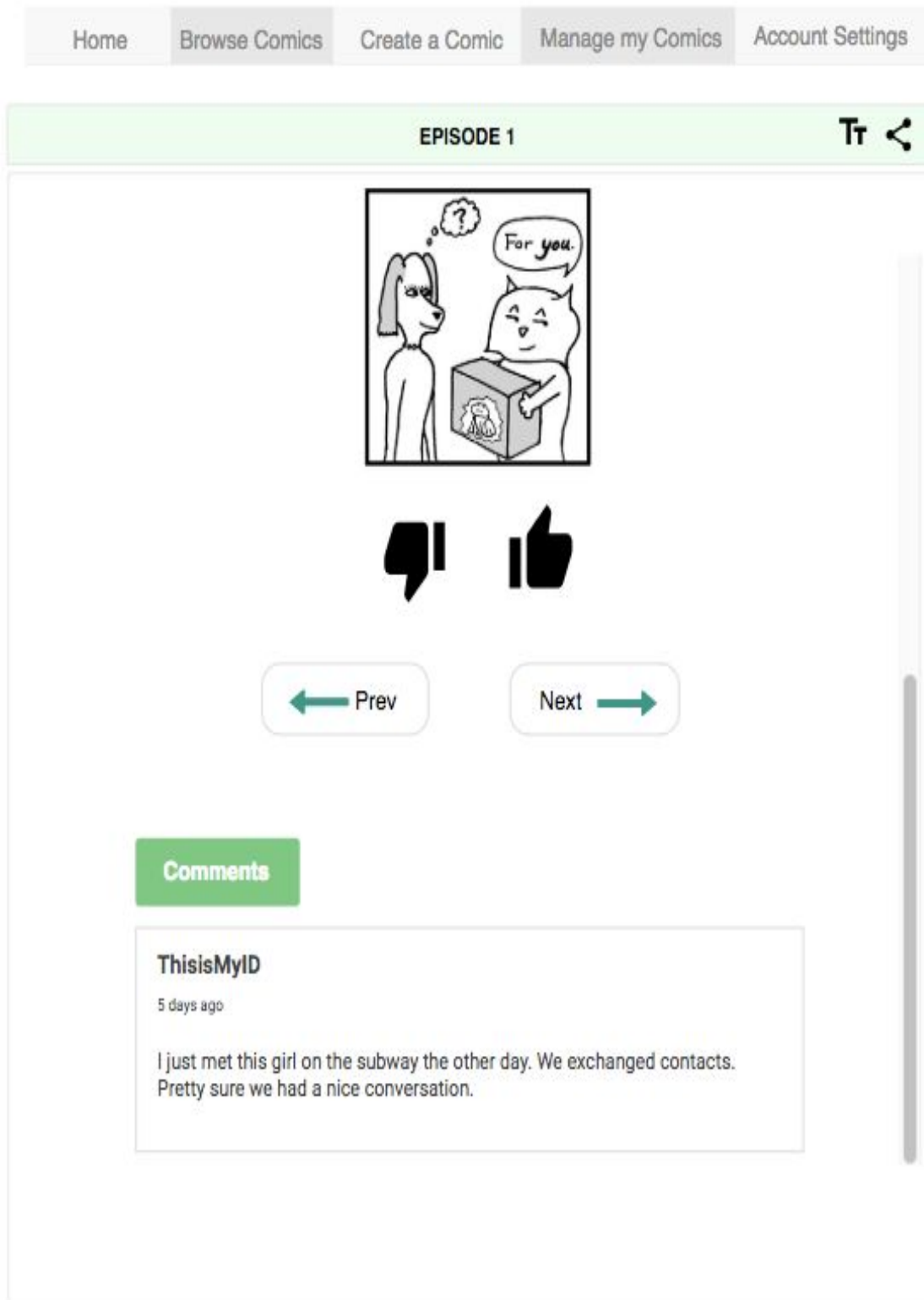


Figure 3.15 Read an Episode - Bottom of the page (Default - Text change feature enabled)

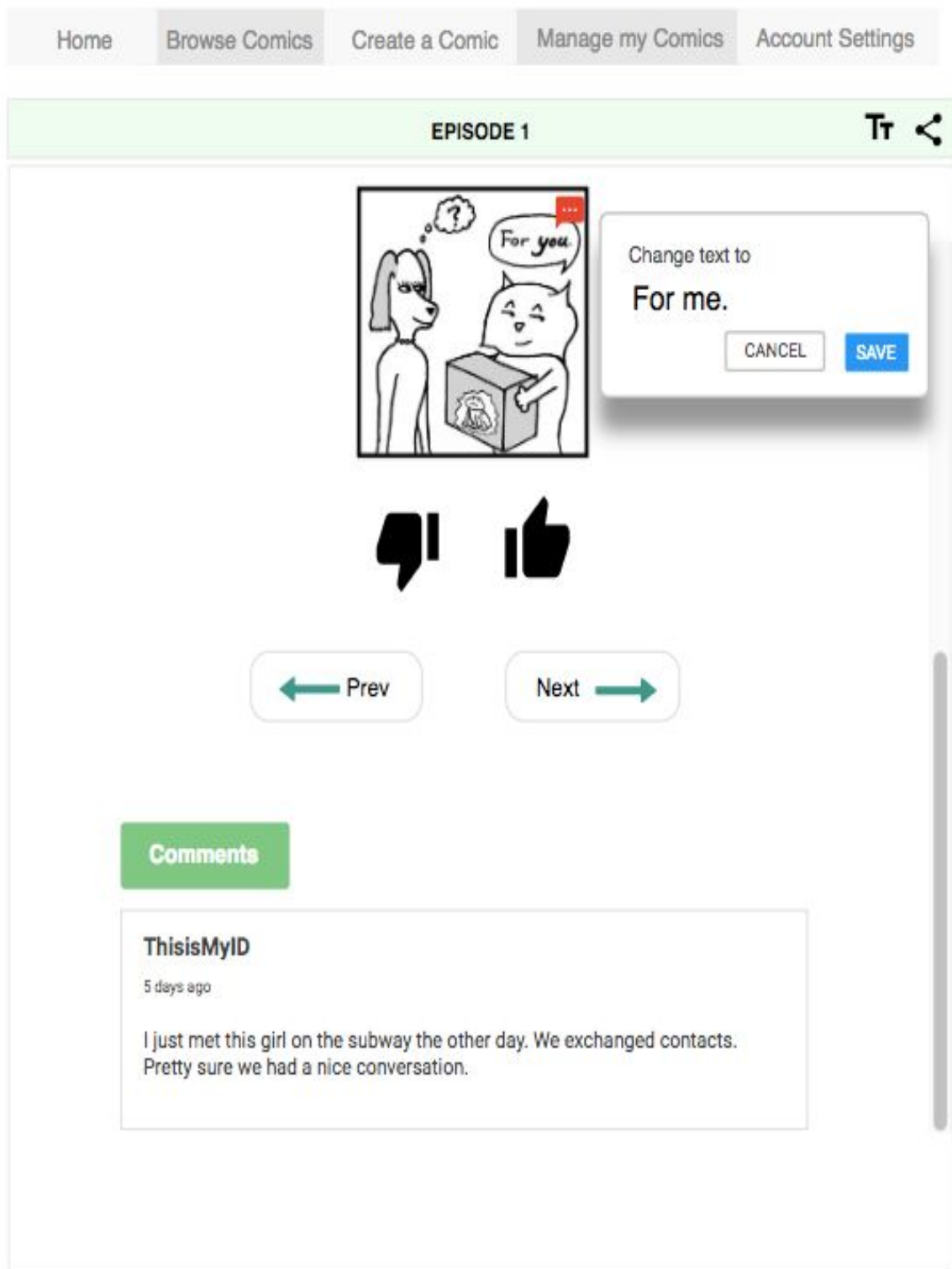
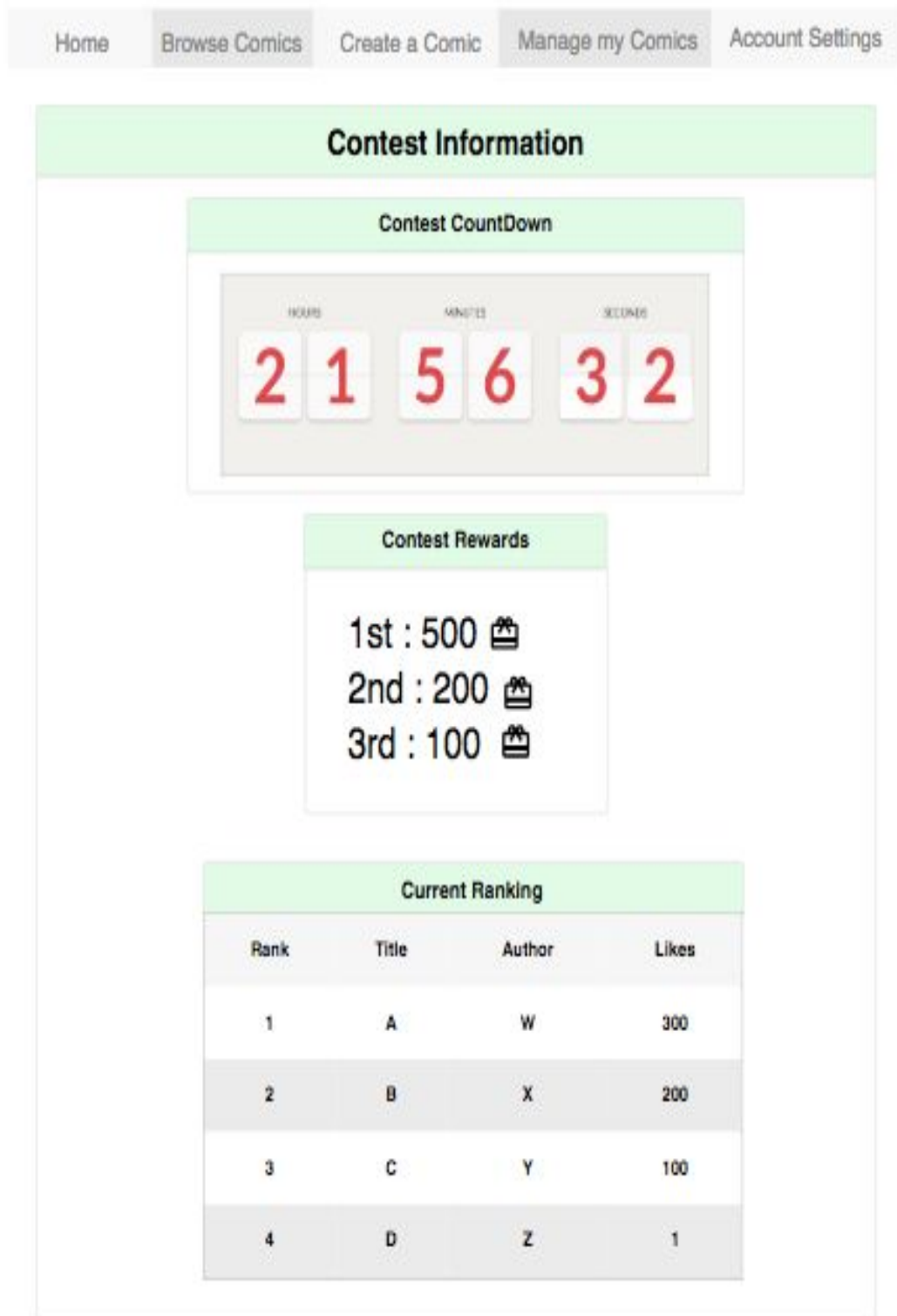


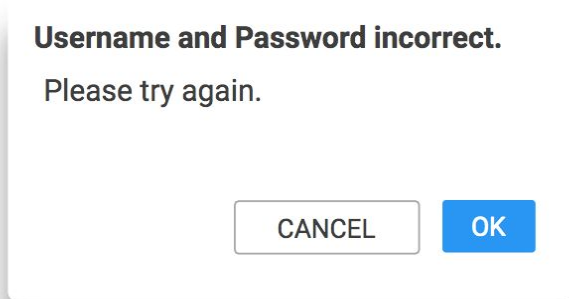
Figure 3.16 Contest Information



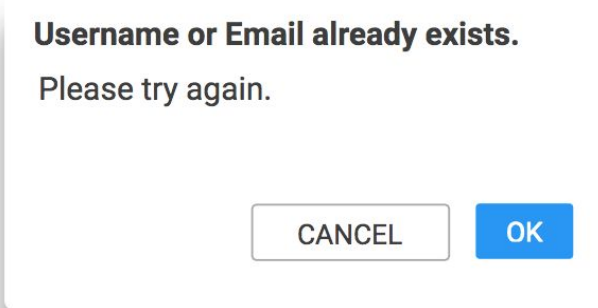
### 3.2 Functions

Our web application will provide feedback with appropriate messages to the user. User will see a dialog box of type confirmation, alert or error.

**Figure 3.17 Login Error Dialog**

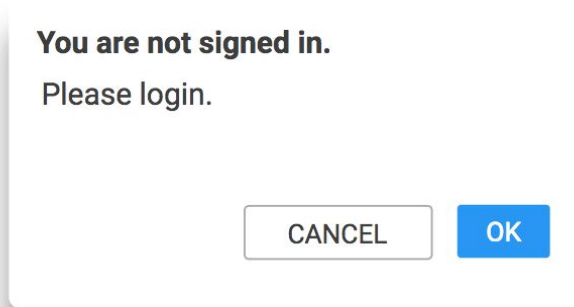


**Figure 3.18 Sign up Error Dialog**

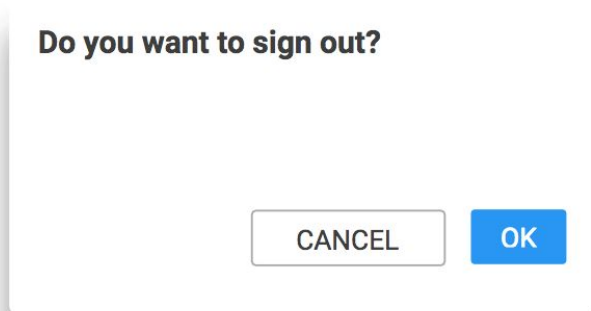


**Figure 3.19 Access denied Error Dialog**

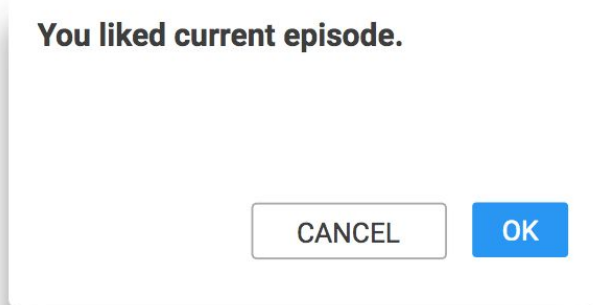




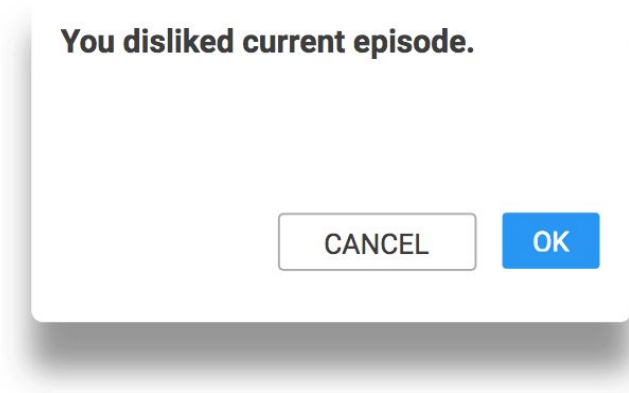
**Figure 3.20 Sign out Confirmation Dialog**



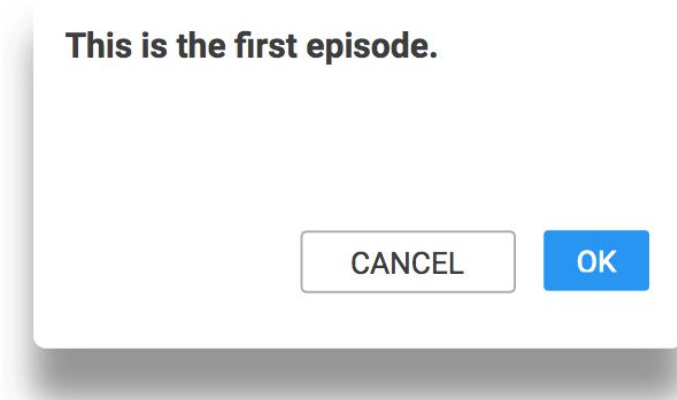
**Figure 3.21 Like Confirmation Dialog**



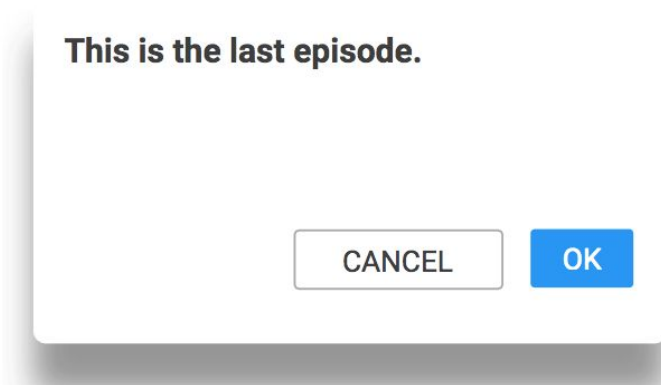
**Figure 3.22 Dislike Confirmation Dialog**



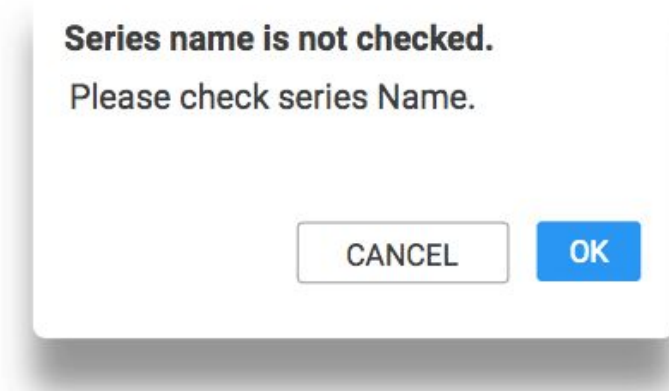
**Figure 3.23 Previous Episode Alert Dialog**



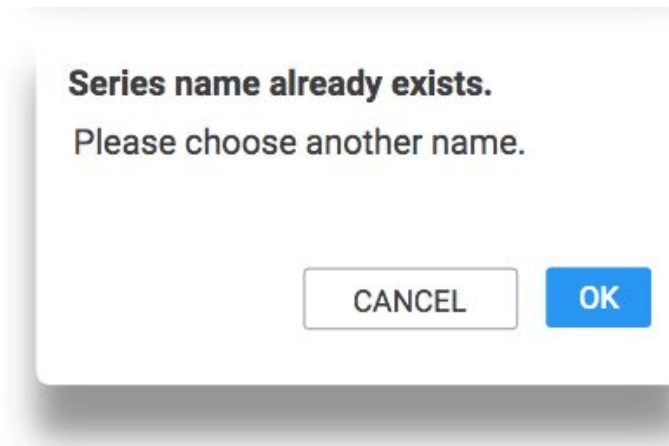
**Figure 3.24 Next Episode Alert Dialog**



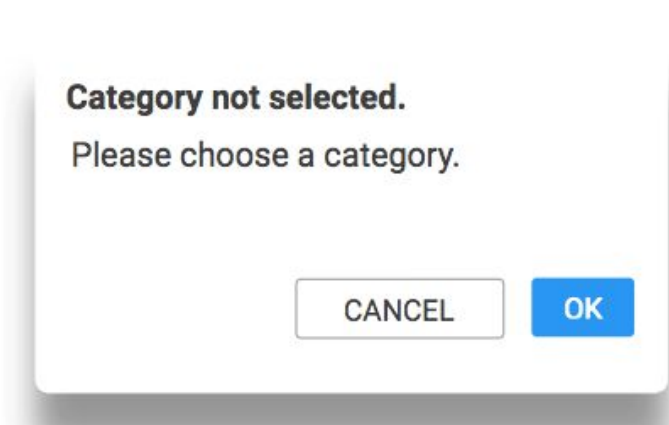
**Figure 3.25 Create Series Error Dialog**



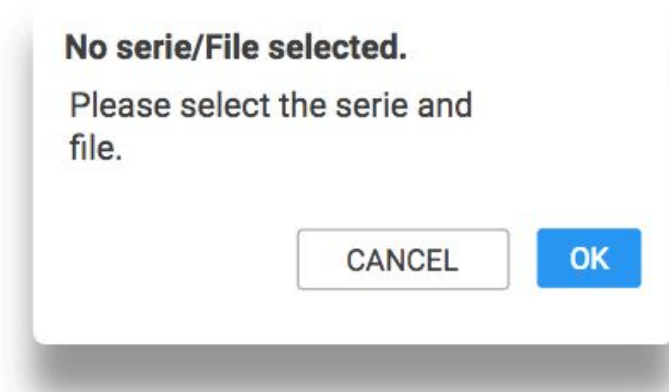
**Figure 3.26 Create Series Error Dialog**



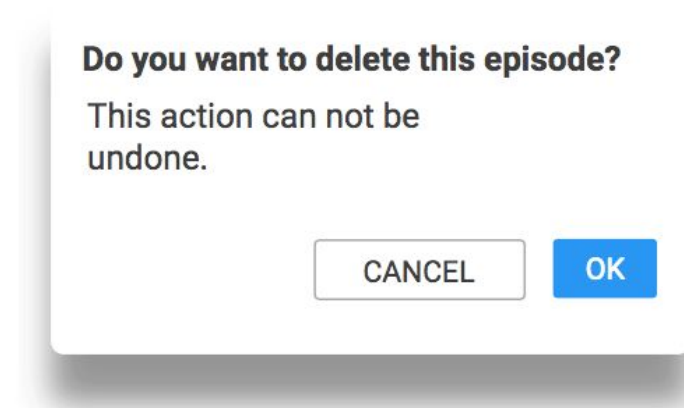
**Figure 3.27 Create Series Error Dialog**



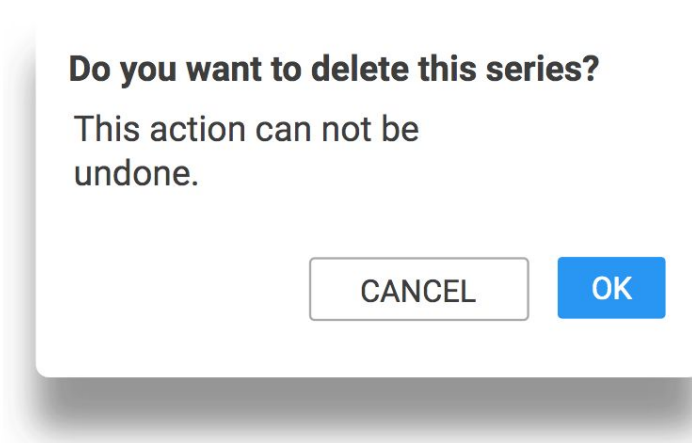
**Figure 3.28 Upload Episode Error Dialog**



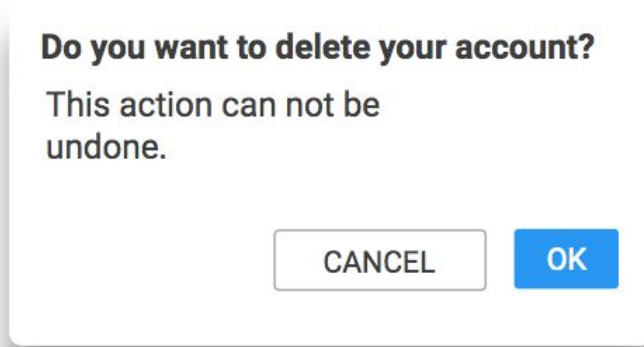
**Figure 3.29 Delete Episode Confirmation Dialog**



**Figure 3.30 Delete Series Confirmation Dialog**



**Figure 3.31 Delete Account Confirmation Dialog**



### **3.3 Performance requirements**

The program should never crash. Subscriptions and unsubscriptions should happen immediately. Drawing tools and text changing features should be responsive.

### **3.4 Logical database requirements**

N/A

### **3.5 Design constraints**

The assumption is that the user will have a screen and mouse. We will not assume any other devices.

### **3.6 Software system attributes**

As professionals, all members of this project must take this project seriously. In order to achieve this level of quality, we should build a web application with the following properties in mind:

**3.6.1 Reliability** – The program should be carefully designed, implemented and tested such that it behaves flawlessly for the end user with minimal instructions.

**3.6.2 Availability** –Customers may visit the website and use the application for free.

**3.6.3 Security** – All security mechanisms will be addressed by future revisions.

**3.6.4 Extensibility** –Drawing features and file formats and should be carefully considered such that the app can be easily extended to social medias.

**3.6.5 Portability** – To start with, the app will target all platforms that support Java.

**3.6.6 Maintainability** – Update mechanisms will be addressed by future revisions.

### **3.7 Organizing the specific requirements**

The specific requirements for this application already fit neatly into the sections listed in the IEEE's recommended SRS format.

### **3.8 Additional comments**

N/A

## **4 Supporting Information**

Note that this document should serve as a reference for the designers and coders in the future stages of the development process.

### **4.1 Table of contents**

1. Introduction
  - 1.1. Purpose
  - 1.2. Scope
  - 1.3. Definitions, acronyms, and abbreviations
  - 1.4. References
  - 1.5. Overview
2. Overall description
  - 2.1. Product Perspective
    - 2.1.1 System Interfaces
    - 2.1.2 User Interfaces
  - 2.2. Product functions
  - 2.3. User characteristics
  - 2.4. Constraints
  - 2.5. Assumptions and dependencies
3. Specific requirements
  - 3.1. External Interfaces
  - 3.2. Functions
  - 3.3. Performance requirements
  - 3.4. Logical database requirements
  - 3.5. Design constraints
  - 3.6. Software system attributes
  - 3.7. Organizing the specific requirements

- 3.8. Additional comments
- 4. Supporting Information
  - 4.1. Table of contents
  - 4.2. Index
  - 4.3. Appendices

## 4.2 Index

1.	Introduction	1
1.1.	Purpose	1
1.2.	Scope	2
1.3.	Definitions, acronyms, and abbreviations	2
1.4.	References	2
1.5.	Overview	2
2.	Overall description	3
2.1.	Product Perspective	3
2.1.1	System Interfaces	4
2.1.2	User Interfaces	6
2.2.	Product functions	8
2.3.	User characteristics	30
2.4.	Constraints	30
2.5.	Assumptions and dependencies	30
3.	Specific requirements	30
3.1.	External Interfaces	30
3.2.	Functions	47
3.3.	Performance requirements	52
3.4.	Logical database requirements	52
3.5.	Design constraints	52
3.6.	Software system attributes	52
3.7.	Organizing the specific requirements	53
3.8.	Additional comments	53
4.	Supporting Information	53
4.1.	Table of contents	53
4.2.	Index	54
4.3.	Appendices	54

## Appendices

N/A

# ***The ComicVerse™***