

# **Software Requirements Specification**

**DYST<sub>uSS</sub>**

**Version 0**

**Rashil Gandhi  
18CS30036**

**Software Engineering Laboratory  
CS29006**

**May 24, 2020**

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Intended Audience and Reading Suggestions . . . . .	3
1.3	Project Scope and Need . . . . .	3
<b>2</b>	<b>Functional Requirements</b>	<b>4</b>
2.1	Profiles . . . . .	4
2.2	Recommender . . . . .	4
2.3	Posts . . . . .	4
2.4	Comments . . . . .	4
2.5	Appeal . . . . .	4
2.6	Search . . . . .	5
<b>3</b>	<b>Non-functional Requirements</b>	<b>6</b>
3.1	Performance Requirements . . . . .	6
3.2	Security Requirements . . . . .	6
3.3	User Interface/Experience Attributes . . . . .	6
3.4	Other Requirements . . . . .	6
<b>4</b>	<b>System Information</b>	<b>7</b>
4.1	Software . . . . .	7
4.2	Hardware . . . . .	7
<b>5</b>	<b>Auxiliary Information</b>	<b>8</b>
5.1	Tentative Timeline . . . . .	8
5.2	Marketing and Sales . . . . .	8
5.2.1	Usage . . . . .	8
5.2.2	Cost . . . . .	8

# 1 Introduction

## 1.1 Purpose

The purpose of this application is to provide an anonymous one-stop platform to enthusiasts to converse about what the entertainment industry has to offer. DYSTuss, pronounced like “discuss” but with ‘t’ instead of ‘c’, literally stands for *Did You See This - discuss*, indicating a discussion forum for film connoisseurs. Apart from discussion, the platform also enables users to get recommendations based on what they like.

## 1.2 Intended Audience and Reading Suggestions

This SRS is for developers, project managers, users and testers. Forthcoming content will provide all internal, external, functional and non-functional information about DYSTuss.

## 1.3 Project Scope and Need

DYSTuss creates a space for people to freely and anonymously discuss their ideas and thoughts about movies, shows and series, along with giving them the ability to maintain user profiles to keep track of what they have seen and their recommendations.

The inception of this idea kind of happened quite some time ago, mainly stemming for the need to have dedicated platform for having in-depth discussions about entertainment in general. A lot of my peers take active interest in watching Hollywood and Bollywood movies, TV shows, web series, and frequently discuss about them. It makes for a great pastime, and some of them even go as far as having in-depth discussions/arguments about various aspects of film-making such as directing styles, production decisions, acting finesse, editing expertise etc.

An important aspect of the platform is to provide anonymity to users, so that they can freely express themselves without fear of any repercussions - of course, in compliance with the moderation rules.

## **2 Functional Requirements**

### **2.1 Profiles**

The platform can only be accessed by registered users. Upon landing, people will be redirected to the Sign In page if they're not already signed in. If they're a new user, the Sign Up link can be used to make an account and register themselves. Each registered user will have a personal User Profile section. This will list the movies, shows and series that the user has already seen, and also a bucket list for yet-to-be-seen content.

### **2.2 Recommender**

The main feature of the User Profile section will be the recommendation system. This system will suggest new movies, shows, series to the users based on the two lists that the users maintain in their profile. Apart from suggesting content, this will also customize the user feeds to include related posts at the top.

### **2.3 Posts**

This is the crux of the platform - the discussion forum. Users will be able to create new posts about any topic related to entertainment, examples being, a director's work, an actor's legacy, a film review, an award show, film trivia, story-writing etc. The topic, it's category and sub-categories will be duly mentioned in the title of the post, along with relevant tags and metadata after the post itself. If the post is about a movie, the post sidebar can show links to IMDb ratings, Metacritic reviews etc. Below the post would lie the comments, encouraging healthy, engaging and constructive discussions.

### **2.4 Comments**

Comments are an integral part of any discussion. The platform will feature cascading-style nested comments, so that users can reply only to those comments which are relevant. This keeps the discussions precise and to-the-point.

### **2.5 Appeal**

Each post and individual comment will include with itself a grade - its appeal, indicating general popular opinion about what is mentioned in the content. The grade will start

at zero, and users will have the ability to either increment or decrement that grade, however, a user can only either increment it once, or decrement it once, or choose not to do anything.

## **2.6 Search**

Users can search through existing posts based on titles, categories, tags etc. to easily find relevant content. This will help in reducing duplication of posts and reviews along with streamlining the process of managing user feeds.

## **3 Non-functional Requirements**

### **3.1 Performance Requirements**

DYSTuss is going to be developed as a progressive web application, while giving it the ability to be platform-agnostic, i.e. it will run on Windows, Mac, Linux, iOS, Android or any device capable of running a web browser.

### **3.2 Security Requirements**

Since anonymity is integral to the usage of DYSTuss, stringent OAuth practices and API's would be placed to prevent access by unregistered users.

### **3.3 User Interface/Experience Attributes**

A principle focus of the development of this platform would be the implementation of Fluent Design language and Fluid elements.

### **3.4 Other Requirements**

DYSTuss would need constant maintenance as it is a long-term and continuously updating service. Further, refactoring the codebase might become necessary depending on scalability requirements.

## 4 System Information

DYSTuss is a desktop application, intended to be run on wide screens, optimised for use with both keyboard-mouse and touch-enabled input combinations.

The tech-stack to be used is as follows:

The front:

React.js, a JavaScript framework for developing beautiful and comprehensive web applications, along with a component library called React-UWP, would be used to develop the UI/UX of the platform.

The back:

Flask RESTPlus API, written in Python would be used to communicate with a custom backend using an SQL database. The backend would have Auth. implemented in it. Since the principle focus of this platform is UI/UX, this API would be a modified version of a backend used in one of my previous projects, so that the overhead of writing it again would be avoided.

The development environments are enumerated below:

### 4.1 Software

The development will primarily take place on a machine running Windows 10 v1909 (OS Build: 18363.657), with Visual Studio Code as the editor and Windows PowerShell as the terminal. For testing purposes, multiple devices with varied sizes and form factors would be used.

### 4.2 Hardware

The aforementioned software runs on a sufficiently advanced hardware configuration to give the best build speeds possible. The CPU is Intel i7-8750H Hexa-Core clocked at 2.20Ghz. It has 16GB of RAM and 128GB of Solid State memory. The GPU is Nvidia GTX 1060 clocked at 1.5 GHz. It has 6GB of VRAM, along with 1TB of hard-drive space.

## 5 Auxiliary Information

### 5.1 Tentative Timeline

Week-1: Lab-7	Design
Week-2: Lab-8	(cont.) Design
Week-3: Lab-9	ML Implementation
Week-4: Lab-10	API Integration
Week-5: Lab-11	Testing
Week-6: Lab-12	Demonstration
Week-7: Lab-13	(cont.) Demonstration

### 5.2 Marketing and Sales

#### 5.2.1 Usage

The following points would probably help advertise and extend DYSTuss use among the people. DYSTuss is a unique and very specific platform that just does (or rather helps people do) what it says:

- Promote healthy and fruitful discussions
- Provide a method to search for existing posts on a topic to include as many points-of-view as possible
- Provide a mechanism to evaluate credibility of posts/comments so the users know what to trust
- Provide an easy-to-use interface for getting recommendations both from the community and from the ML-trained computer models

#### 5.2.2 Cost

##### Cost to developer

- Time and effort required.
  - Cost of OMDb API.
  - Cost of Server for hosting the backend, ML implementation and the website.
- API charges will not be problematic initially as these service providers have free tiers for limited loads.

##### Cost to customer

The app will be free to use for the customer.