# Software Requirements Specification (SRS) Document

***SCRAPSHUT***

*Team No 24*

*Avani Gupta, Apoorva Tirupati, Parv Joshi, Siddharth Jain*

# Brief problem statement

To develop a web platform for ScrapShut where users can come and rate different websites and get reviews of other users on a particular article/website. We train our ML model which will predict the genuineness of the websites based on the majority of user ratings, and scraped data from the website. Users further would be asked to tell which words or things in a website made then feel its fake. This data would be used to further refine our predictions on the genuineness of websites. We aim to provide users with data on which websites are real and which are not.
When a user wants to check the genuineness of any website, he will come to our web platform, enter its link and we'll show him the result based on user reviews collected, whether it's real/fake along with reviews on it. In this way, we aim to provide a hallmark in providing users with reviews on websites and hereby also stopping fake content on the internet.

# System requirements

* Python
* Django
* Vue.js/Angular 4
* Machine Learning
* TensorFlow/Pytorch
* AWS/ Heroku Hosting

# Users profile

*Common users using the internet who want to get reviews on a site/article hosted online.
* Users doubtful about the genuineness of a website and who want to check whether they are'nt entering a fraud website.

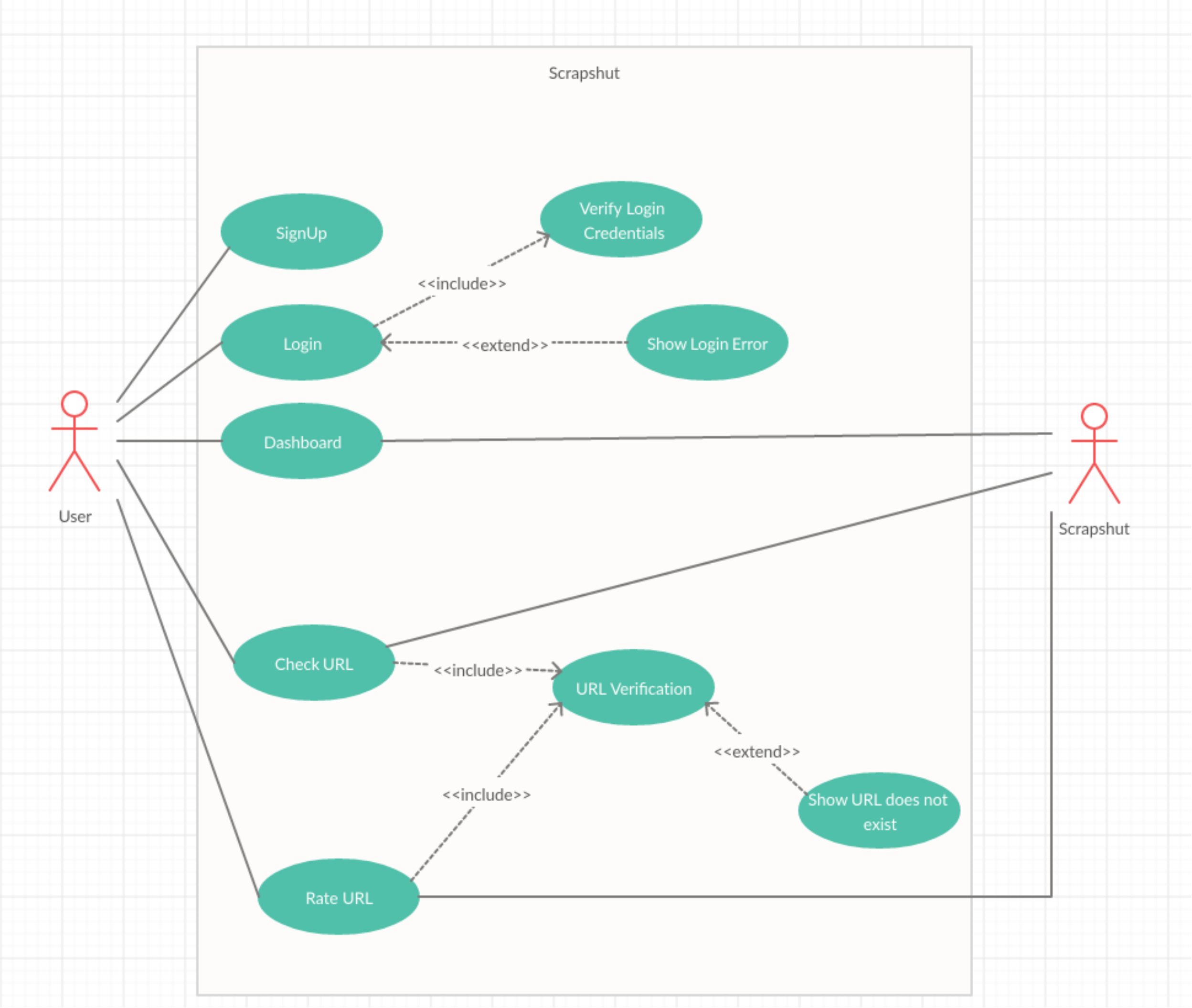
# Feature requirements (described using use cases)

| No | Use Case Name | Description | Release |
|---|---|---|---|
| 1 | Webplatform | Web platform for users to rate things.<br>•Using Django, Angular/Vue.js, Django Rest framework. converting Srapshut's database into JSON format<br>• Connecting Scrapeshut's backend django with frontend(Angular/Vue.js) with API | |
| 2. | Checking Entry | Web platform for users to check genuineness of articles<br>1. Using Django, Angular/Vue.js, Django Rest framework. | |

| | | | |
|---|---|---|---|
| | | 2. If rating greater than threshold then article is genuine<br>3. Else checking whether article is fake using ML algorithms in Fake Article Database(AWS). | |
| 3. | Transmitting Data To AWS | Whatever rating and review is provided by user, it will be transmitted to databased on AWS into 3 separate categories.<br>1) Rating>=3 -> Genuine Section<br>2) Rating <3 -> User is asked what was wrong whether article was fake/ some other reason(eg. slow website/bad UI)Fake Section -> If review says article is fake<br>a) Else it goes | |

| | | | |
|---|---|---|---|
| | | in other section | |
| 4. | Login/signup(Creating User Profiles) | User Data and account will saved in AWS.<br>It will contain all reviews and rating he is provided. | |

## Use case diagram



## Use case description

| | |
|---|---|
| **Use Case Number:** | 1 |
| | Get reviews of article/website |

| | |
|---|---|
| **Use Case Name:** | |
| **Overview:** | When a user wants to get reviews of an artcile/website posted online get reviews of others on that article. |
| **Actors:** | User |
| **Pre condition:** | Reviews of other users must be present on that link. |
| **Flow:** | Main Flow:<br>1. User logins<br>2. Posts link of article he wants to get review<br>3. Gets reviews of other users |

| | |
|---|---|
| | Alternate Flows: |
| **Post**<br>**Condition:** | That article link can be stored in database and suggested to toher users using it |

| | |
|---|---|
| **Use Case Number:** | **2** |
| | |

| | |
|---|---|
| **Use Case Name:** | Give reviews of article/website |
| **Overview:** | Users can come to our website and give reviews |
| **Actors:** | User |
| **Pre condition:** | The link which user wants to rate and review must be public |
| **Flow:** | Main Flow:<br>1. User logins<br>2. Posts link of article he wants to give review |

| | |
|---|---|
| | 3. Rates the artcile/website/webpage.<br>4. If rating < 3<br>● He is prompted to give reason for it |
| | |
| **Post Condition:** | That article link can be stored in database and suggested to other users using it |