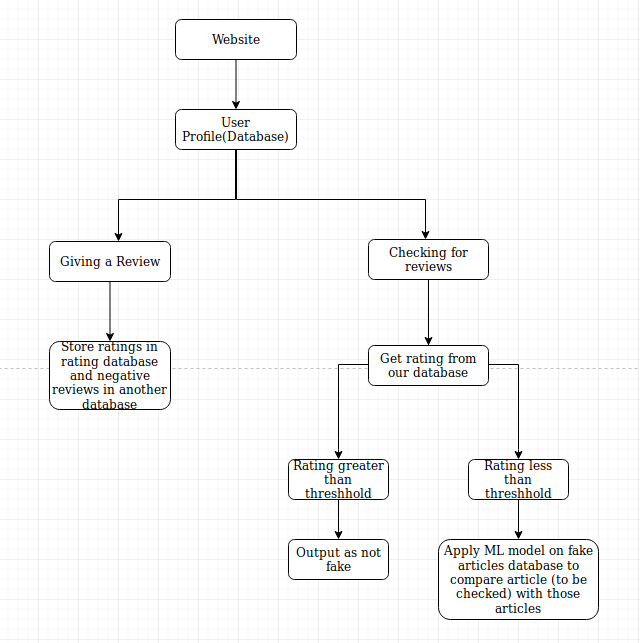
## Product Design Document

|  |  |
| --- | --- |
| **Team** | **24**  **Siddharth Jain**  **Parv Joshi**  **Avani Gupta**  **Apoorva Tirupathi** |

# Design Overview

## Architectural design

Dividing system into independent functional modules.



## System interfaces

### User Interface

Users can use this system in two ways.

1)Checking authenticity of an article

2)Writing Reports/Giving Reviews/Ratings

## Model

|  |  |
| --- | --- |
| User | Responsible for all the functions and data related to the user object, which includes their login credentials, their reviews/reports against articles/news sources & their saved articles   * The methods an object of the user class can implement are very simple: * 1. Can report an Article via the Report class's object * 2. Can view the reports related to an article * 3. Can run the DL model on an unseen article they're doubtful about |
| Article | Class state   * What information is the class responsible for maintaining? * Content of Article * Newspaper that published it * Article writer * Lists of reports and reviews on it   Class behavior   * Add to DL training set |
| NewsSource | Class state   * Credibility Rating wrt News Section * Reviews written on the News Channel |
| Report | Class state   * Explanations as to why they found it fake * Sources/Citations as proof to base their opinions if possible |

.

# Sequence Diagram

A close up of a map

Description automatically generated

# Design Rationale

1. We initially had to take a call on whether our Deep Learning Model that detects fake articles will be one that trains on-the-fly or in-batches. After discussion with the client and considering the cost of one-the-fly versus batch-trained, we arrived at choosing batch-trained as it would be less expensive over of having a constant training occurring at all times, instead on attaining an additional 100 new samples, we'd retrain the model.

2. Another major reason for choosing batch-training is for consistency in results. With on-the-fly training happening in the background at all times (like a perpetual learning machine), the results are subject to variability and volatility at the smallest intervals, this cannot be explained for, and makes the user question the site's credibility.

3. There was a lot of discussion on which frontend framework to use. We had initially started out using vue.js but realised later into the project that a framework like angular.js can more tend to the web apps' needs, as angular.js was more enterprise ready, from routing, templates to testing utilities in its package. We then But, we faced a few stumbles on the path to learning it, as it way more complicated than vue.js or react.js, ones we had experience with from our assignments.

4. We initially used a basic word-to-vec single-layered neural network, on a simple fake-not fake annotated news dataset, to learn how to implement deep learning and get our basics clear and then shifted to a more complicated CNN, using Glove, and trained on a better more credible Kaggle Dataset.

5. We've learnt a lot in the process about ML & Neural Networks and we finally built 4 models, LSTM, CNN, xgboost & a passive aggressive classifier. We learnt a lot from the various results, and decided to make an ensemble model based on the 4 results. But after multiple trials, decided not to, as it would give a skewed understanding and felt 4 individual results would have more use, as they explain in 4 different aspects.