virtusa
Accelerating Business Outcomes

Architecture Document for *Content Moderation using AI*

.....

Team Members:
Koyya Khushhal Reddy
Aryamaan Srivastava
Preeti Ranjan Panda
Naman Sharma
Satyanand Prasad

TABLE OF CONTENTS

Note: These documents are strictly for specific Virtusa use only during the Jatayu initiative. They shall not be shared with an external party other than the client concerned. This category also covers client intellectual property where Virtusa has a non-disclosure agreement with the client.

Important note: The document is divided into two parts: **Part-A** and **Part-B**. Part-A shall be completed within 10 days of commencement and submitted to your Virtusa mentors. Part-B shall be completed and submitted at the end of the Jatayu initiative along with your solution. Part-A at first submission can be in draft. Early submission of Part-A can ensure that the mentors can determine if your team are on track and should there needs be course correction.

1 Introduction – PART-A

1.1 Purpose

This document provides a comprehensive architectural overview of the **Content Moderation system using AI**, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions, which have been made on the system.

1.2 Scope

The areas affected by this project include all the platforms which involve any transmission of data in the form of text, audio, image or video. This can be either on E-learning platforms like Microsoft Teams or Google Meets, on company platforms which provide support to the customer, on online second hand markets or on social media platforms.

However, there are some areas which are beyond the scope of this project. As stated in the features of the project, the system does not filter the data from coming from your connections. It can be your friends whom you follow on social media or the teachers in the online class. It thus enables seamless flow of data from the above-mentioned users.

1.3 Definitions, Acronyms and Abbreviations

Acronyms:

- **CNN**: Convolutional Neural Network.
- **BERT**: Bi-Directional Encoders Representations from Transformers.
- **OSS**: Open Source Softwares

Definitions:

- **PyTorch**: PyTorch is an open source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Facebook's AI Research lab.
- **Neural Machine Translation**: Neural machine translation (NMT) is an approach to machine translation that uses an artificial neural network to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model.
- **Neural Networks**: A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

1.4 References

- Dive into Deep Learning, Release 0.15.0, Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics.
- Hate Speech Identification using CNN and BERT. <link>
- Neural Network for Hate Speech classification, Office of Scholarly communication, Harvard Library. <link>
- Flask Web Development, Web development using Flask, Miguel Grinberg
- Exploring Deep Multimodal Fusion of Text and Photo for Hate Speech Classification <link>
- Deep Learning with PyTorch by Eli tevens, Luca Antiga, Thomas Viehmann

2 Architectural Goals and Constraints - – PART-B

1.5 Reusability

The model can be integrated with any existing application which facilitates transferring of different media files, for content moderation and can be reused by all the users of the application i.e. the core of the UI system is reusable.

1.6 Scalability

The system extends its service and makes it scalable enough to provide it according to the scale and size of the application. The model provides its seamless service, even at a higher scale of users and increased size of the application interface.

1.7 Customizability

Our project comes with support for customization of UI at the user's end and enables them to personalize their UI according to their wish. The usage and functionalities of the model remain the same, even after changes are made at the user interface level, supporting customization to any display the users wishes to keep.

1.8 Extendibility

The technologies used while building the project are constantly being updated and evolved to support new features which can further be expected from the application. If suppose, sharing of a new media type is to be added as a core feature, then the media file will be passed onto the model and can be classified as offensive or not.

Further, modular development is highly encouraged while working with the project as it will make it easier to work with different media files.

1.9 Use of Existing Business Logic

The model can be easily integrated with the existing logical aspects of the business and will work with the existing business logic of the application by enhancing the flow of data through content moderation and can together be piled up as a reusable component.

1.10 Time to Market

With the models having high accuracy and low margin of error, the project is at its last stage of release. With the deployment of API, the product is ready to be available in the market.

1.11 Portability

The project extends its support to various Operating Systems and database frameworks. The use of the model is independent of any OS and hence is easy to port from one OS to other. It equally supports all the database frameworks and can be switched from one to another.

1.12 Availability

The project provides a highly fault-tolerant content moderation model and also promises higher availability of the product. The system with low downtime indicating higher availability of the model in a reliable operating manner, ensures high uptime i.e. higher stability and reliability of the system.

2 Productization Assessment – PART-B

1.13 Re-Usable Components

The API in the project is the component which renders the service of moderating the content and is reusable for every type of media file and can be reused by the users. The API is made in such a way, that it supports all the available devices like android, PC and is also independent of the OS and functions equally well in Windows, MacOS, Linux. It can be easily integrated with all the platforms which supports the use of REST API and is also compatible with the various database frameworks.

1.14 Analyze Architectural Frameworks in Repository

The frameworks used in the Repository include the following:

- Tensorflow- A Deep Learning framework used to train the violence model.
- Scikit-Learn- A Machine Learning Module used to build the Spam and Offensive Text Classifiers.
- Native Android Development Framework- Used to build an android application to test the working of the API on a mobile device.
- Flask- A Web Development Framework used to build a Website to test the working of the API on a website.

1.15 Identify and Analyze Open Source and COTS Products

- *SightEngine : The leading API to moderate photos, videos and livestreams. Instantly detect nudity, violence, offensive content.*
- *urlExtract : Module to extract links and URLs from text.*
- *Googletrans : Module to implement Google Translate functions in our project.*

3 System Architecture – PART-A/B

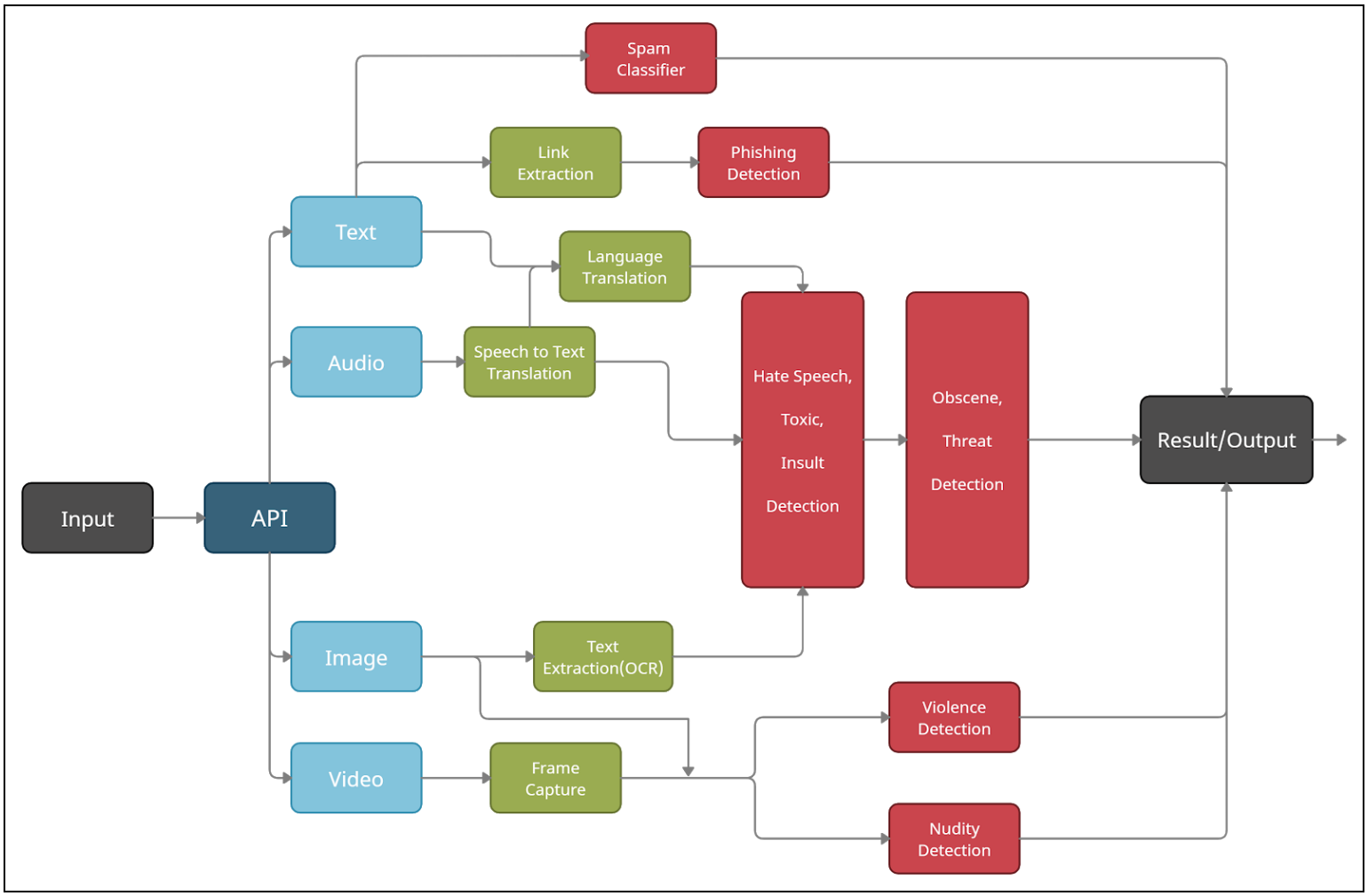
1.16 Overview - – PART-A

The application consists of two models, one that recognizes inappropriate text and one that recognizes inappropriate images.

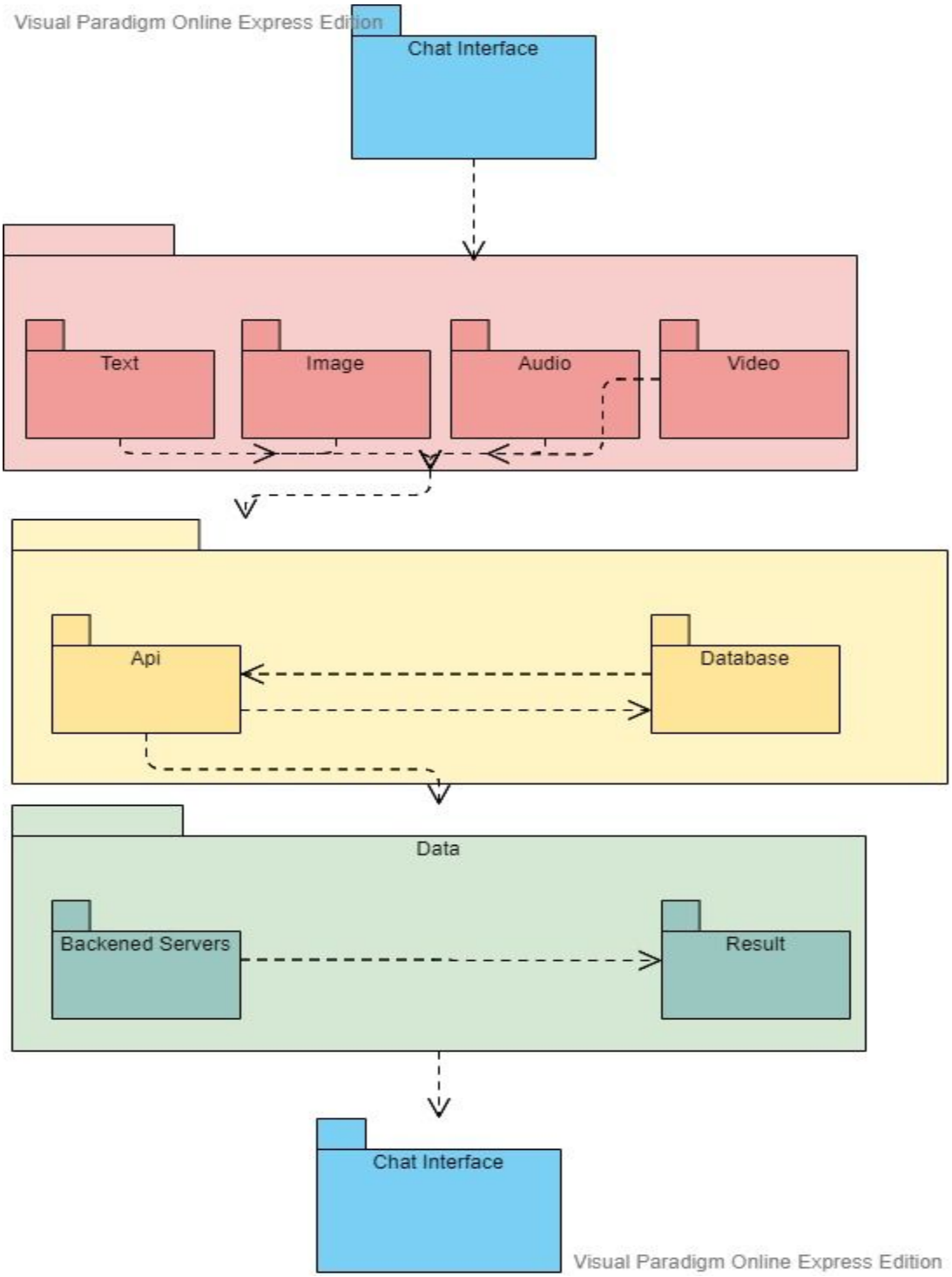
Images, text and audio are first processed to extract text from them and the acquired text is passed through the text model.

Images are also passed through the Image model to detect any inappropriate content or features.

Once results are received from both the models the appropriate response is sent back to the user.

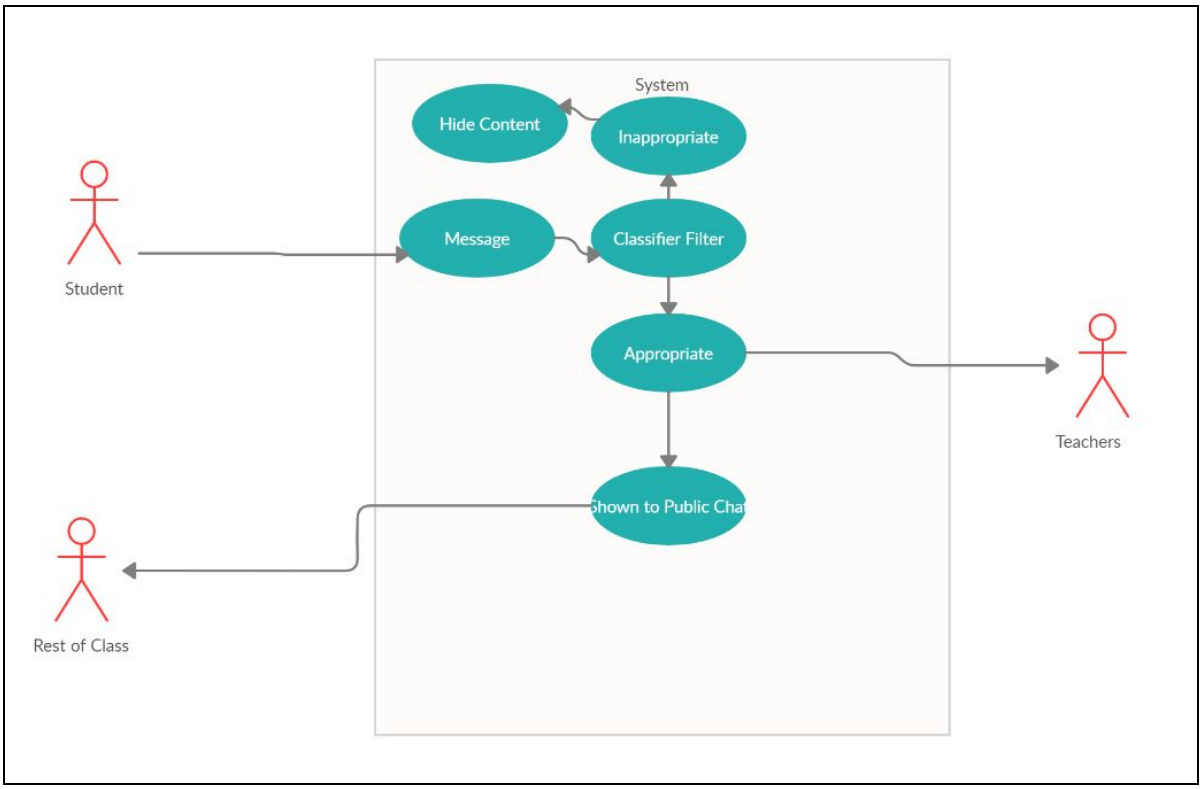


1.17 Logical/Functional View - – PART-A

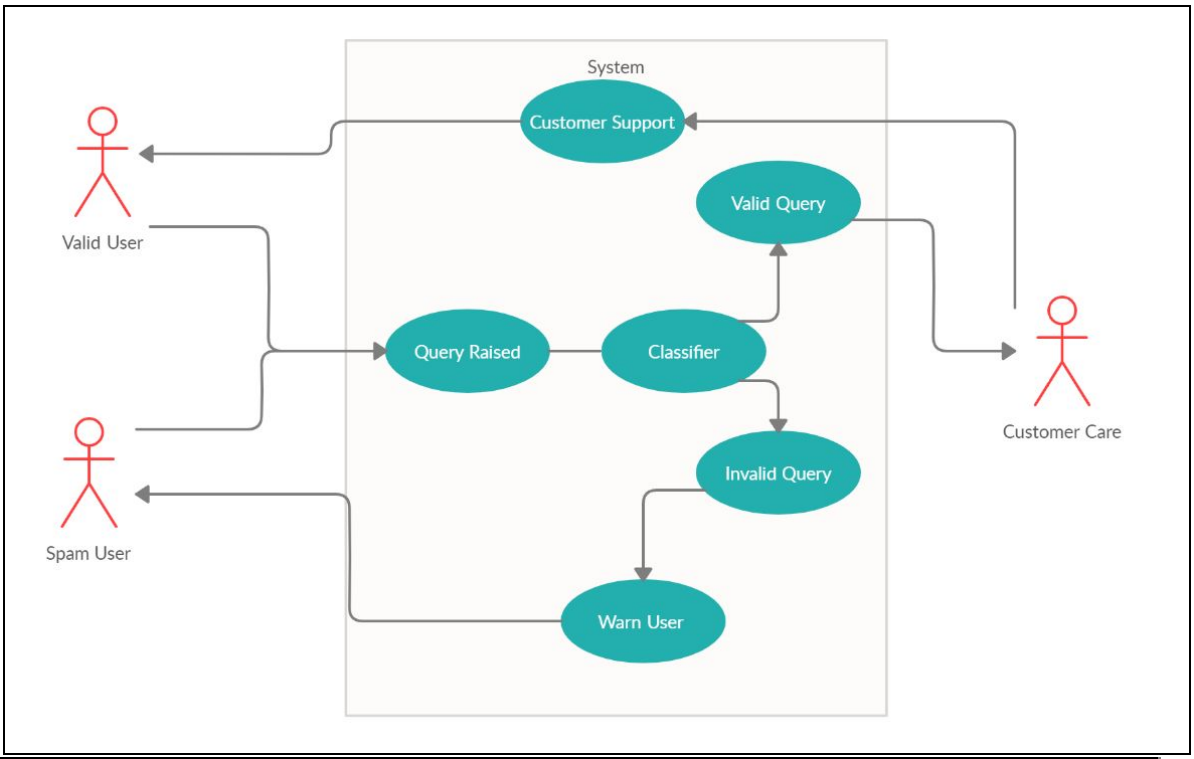


1.18 Use Case View – PART-A

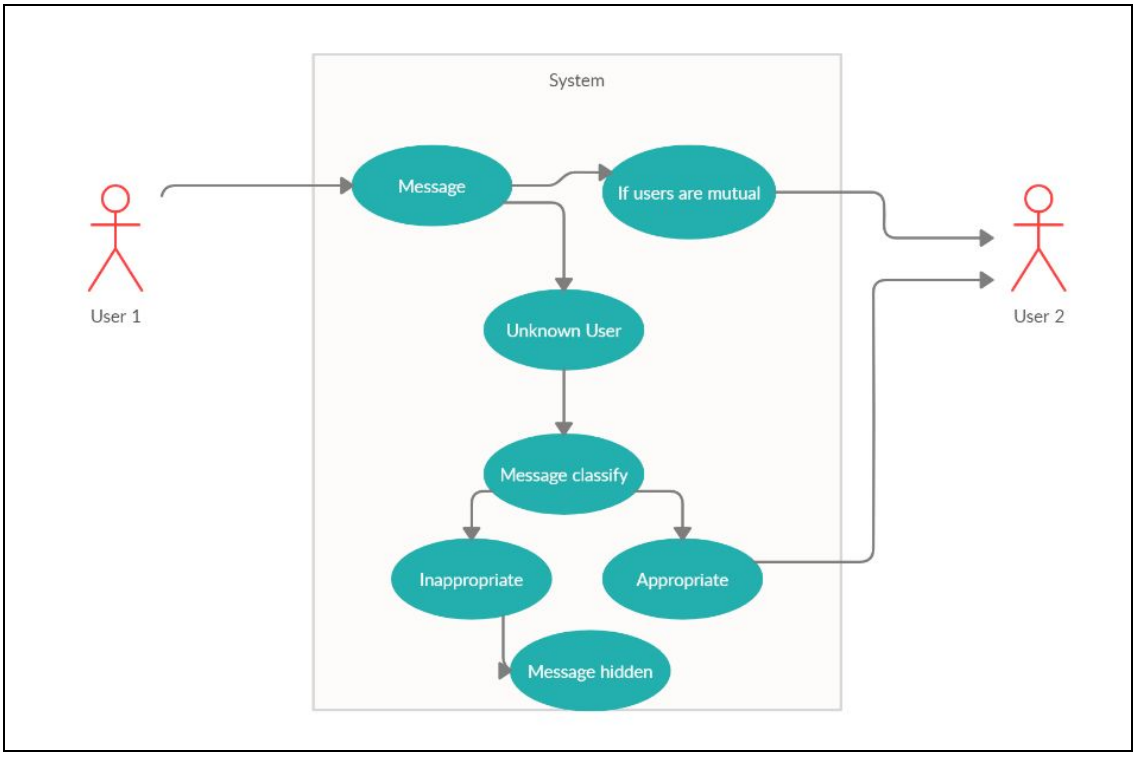
Example diagram: Online Education



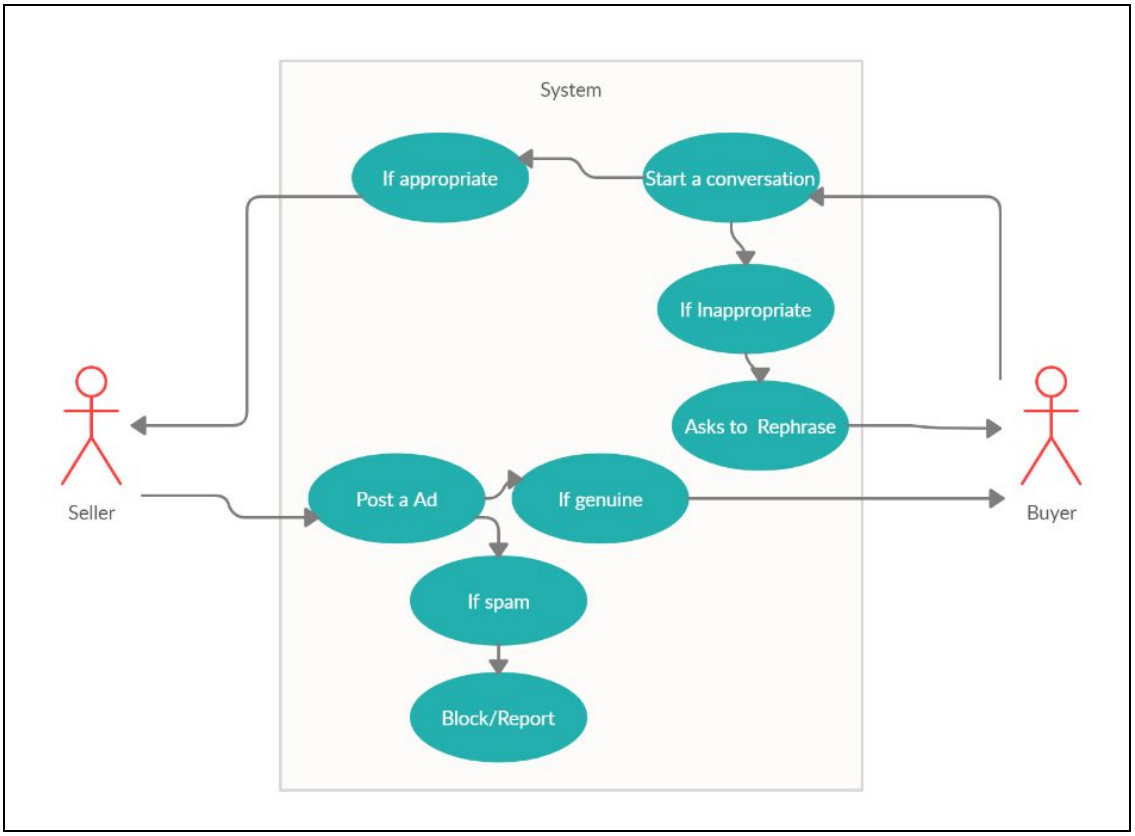
Example diagram: Customer Support



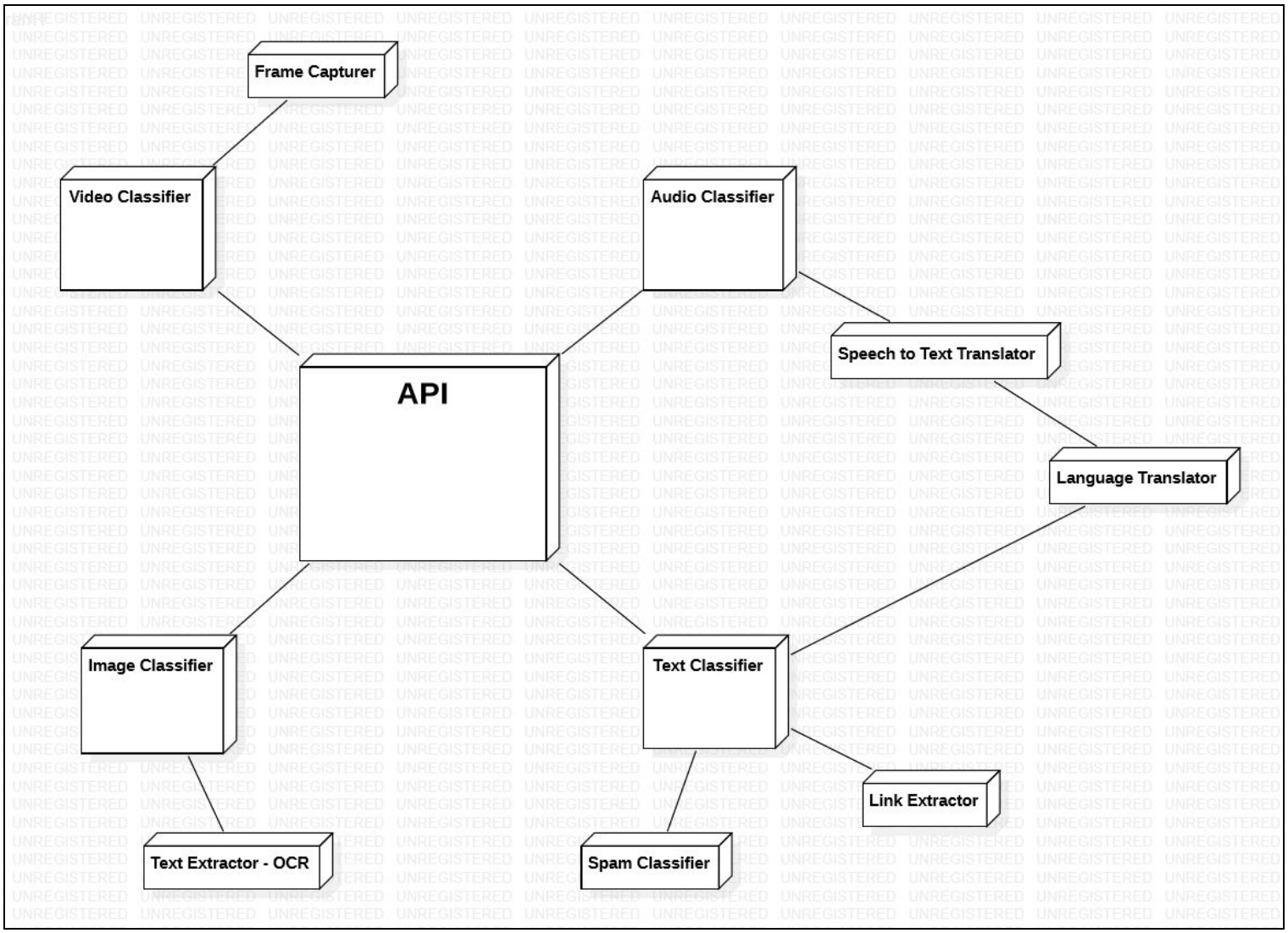
Example diagram: Social Media



Example diagram: Second Hand Online Business



1.19 Deployment View – PART-B



4 Alternative Solutions Considered – PART-B

1)Classification based on different Languages:

A text based model trained on different dataset of different languages could be used in order to make a hate speech classifier model for all the languages.

2)Deployment of the model via Chrome Extension:

The built model can be deployed to Real Time via Chrome extension. Extensions could be built using beginner level technologies like HTML, CSS and Javascript. Extensions can be easily used with any web applications as it provides an add on over the web page rather than actually running with the website.

Why we selected our current solution:

1) Collecting dataset for different languages is a heavy job and moreover dataset for many languages are not even available. Additionally, training a model on such a large dataset will increase the computational time. Hence, We have chosen to train a Uni-language model with good accuracy.

2) We have deployed our model using an API. Using API will provide a larger accessibility by provide a way to embed the model in both Web based and Mobile Based application, while a chrome extension is only limited to Web based applications