UE18CS390A – Capstone Project Review #3
(High Level Design and Proposed Methodology)

Project Title : Web Page Classification for Safer Browsing

Project ID : 18

Project Guide: Dr. Mehala N

Project Team :

Name	SRN	Section
Manav Agarwal	PES2201800025	F
Rishab Kashyap	PES2201800065	F
Shreya Yuvraj Panale	PES2201800117	E
Shreya Venugopal	PES2201800688	F

Outline

- Abstract
- Additional Literature Survey
- Summary of Literature Survey
- Suggestions from Review – 2
- Proposed Methodology / Design Approach
- Architecture
- Design Description
- Technologies Used
- Project Progress
- References

Abstract

▪Well defined problem statement.

Malicious websites are responsible for a majority of the cyber-attacks and scams today. These URLs are delivered to unsuspecting users via emails, text messages, etc. We aim to identify such URLs posing a threat to browsing and web searches using a combination of various website detection schemes.

▪Provide a basic introduction of the project

The internet today proves to be a most formidable place to retrieve or put up information and hence becomes a breeding ground for dangerous websites and malicious users who misuse the internet. The project focuses on the study of malicious URLs and their respective websites, and thereby fabricate a model that allows us to detect dangerous websites and URLs, and thus eliminate future recurrences of such incidents from occurring in our systems. Multiple approaches are used to establish the following method with the use of various concepts related to the browser search engines, machine learning, pattern analysis and so forth.

Additional Literature Survey

- Along with the research done from the previous review, a deeper study on the outlook and performance of each method was surveyed and reported to create a case study.
- We added in a few extra papers that allowed us to create a basic idea about the workflow we shall be following for this as well as clarity on the approaches we are taking.
- Link to the Survey Paper.

Summary of Literature Survey in Review 2

- It is seen that each of the papers share the common trait of blocking out phishing or malicious URLs.
- They vary in their methodologies, each of which prove to be successful in their own way
- In conclusion, we aspire to create a working application by using multiple features from some of the selected literature to improve the efficiency and build a model to perform malicious URL detection.

Suggestions from Review – 2

- Look into the positives of the literature in order to determine the approach
- We have taken this feedback into account while modelling the approach.

Design Details

Design Goals

- The current system of checking blacklists is a slightly slower process, and it's efficiency reaches a limit when it is unable to find a URL in the list even if it is malicious.
- Furthermore, as mentioned above, the URLs have a short lifespan and won't exist for too long and hence the prediction must be done quickly to block it before any damage is done
- Our goal is to minimize the dependency of such lists and create a more flexible and robust system to be placed in the environment for smoother and better functioning.

Design Details

Interoperability requirements

- System must be connected to a browsing system or must be connected to a source that gives URls as inputs.
- Might have issues with the interface of the browsing system; Needs to be compliant with all of them and run smoothly or any of them.

Interface/protocol requirements

- Requires a basic computer system with any browser installed over it or a data source that allows it to retrieve URLs to classify then successfully.
- Might lead to memory issues on smaller systems, thereby affecting the overall performance

Data repository and distribution requirements

- Must have a cluster of URLs fed into it as a testing set
- Needs to be upgraded in order to allow single URLs at a time

Performance related issues

- Efficient in terms of classification and trustworthy in terms of safety
- Needs to be configured to suit the system and reduce overall memory utilization to improve performance

Design Details

End-user environment.

- Must have a browsing system and a system that supports enough RAM to accommodate the system.
- Can also have an alternate data source to feed the system data for classification of URLs

Availability of Resources.

- We assume that the URLs and other inputs are available to the system whenever required
- Initially in the form of batches or clusters, and later on in a dynamic format for real time classification

Hardware or software environment

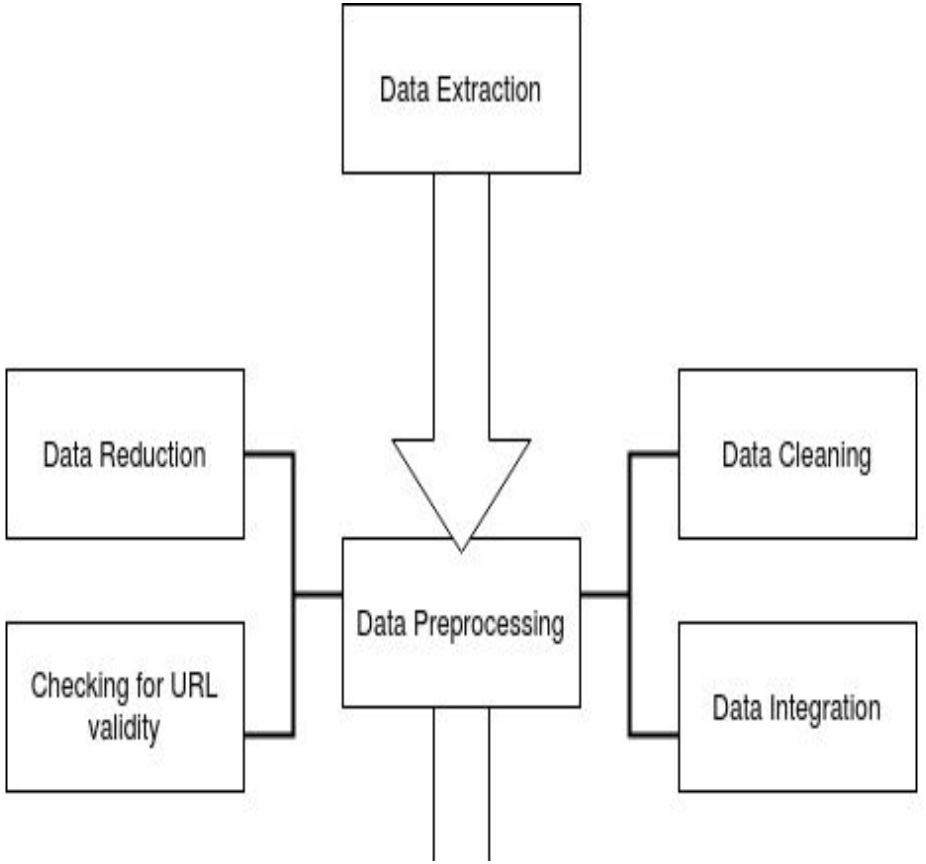
- Any computer system with a good amount of RAM and memory capacity to store URLs and allow the system to freely classify the test data.
- A browser system to generate these URLs while searching online

Issues related to deployment in target environment, maintainability, scalability, availability, etc.

- Might have issues with compatibility of the system with the browser

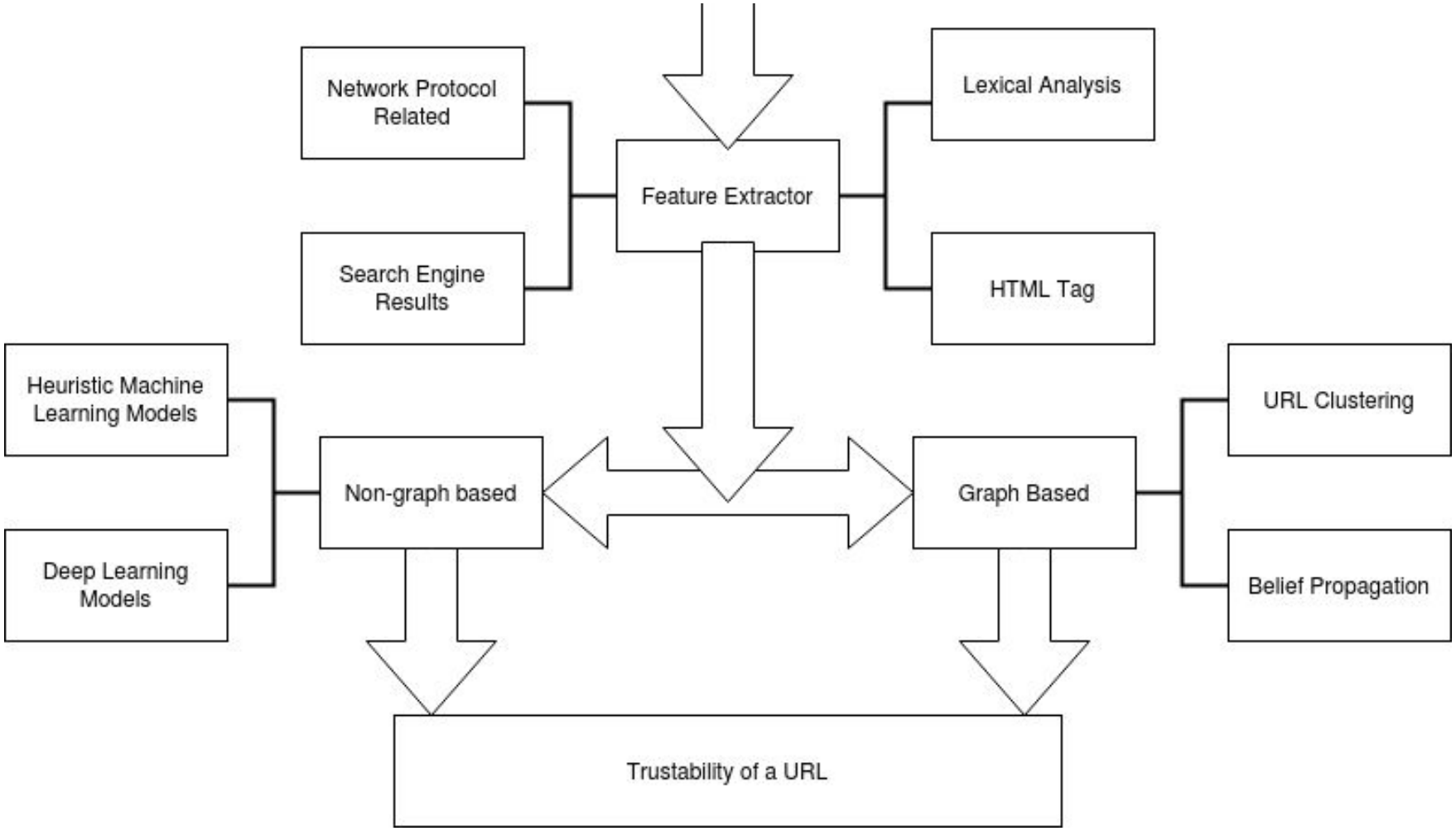
Proposed Methodology / Approach

Basic Approach



Proposed Methodology / Approach

Basic Approach



Proposed Methodology / Approach

Is there a need to change the approach?

- Too much time taken for classification leading to a crash
- If the dataset is not sampled properly the results may be inaccurate or biased
- Access time to some attributes are limited
- May have more time and space complexity than existing methods
- Complete feature extraction may lead to a sparse dataset leading to inaccuracies
- Heuristics or rules generated may not be consistent with the current malicious URLs

Benefits

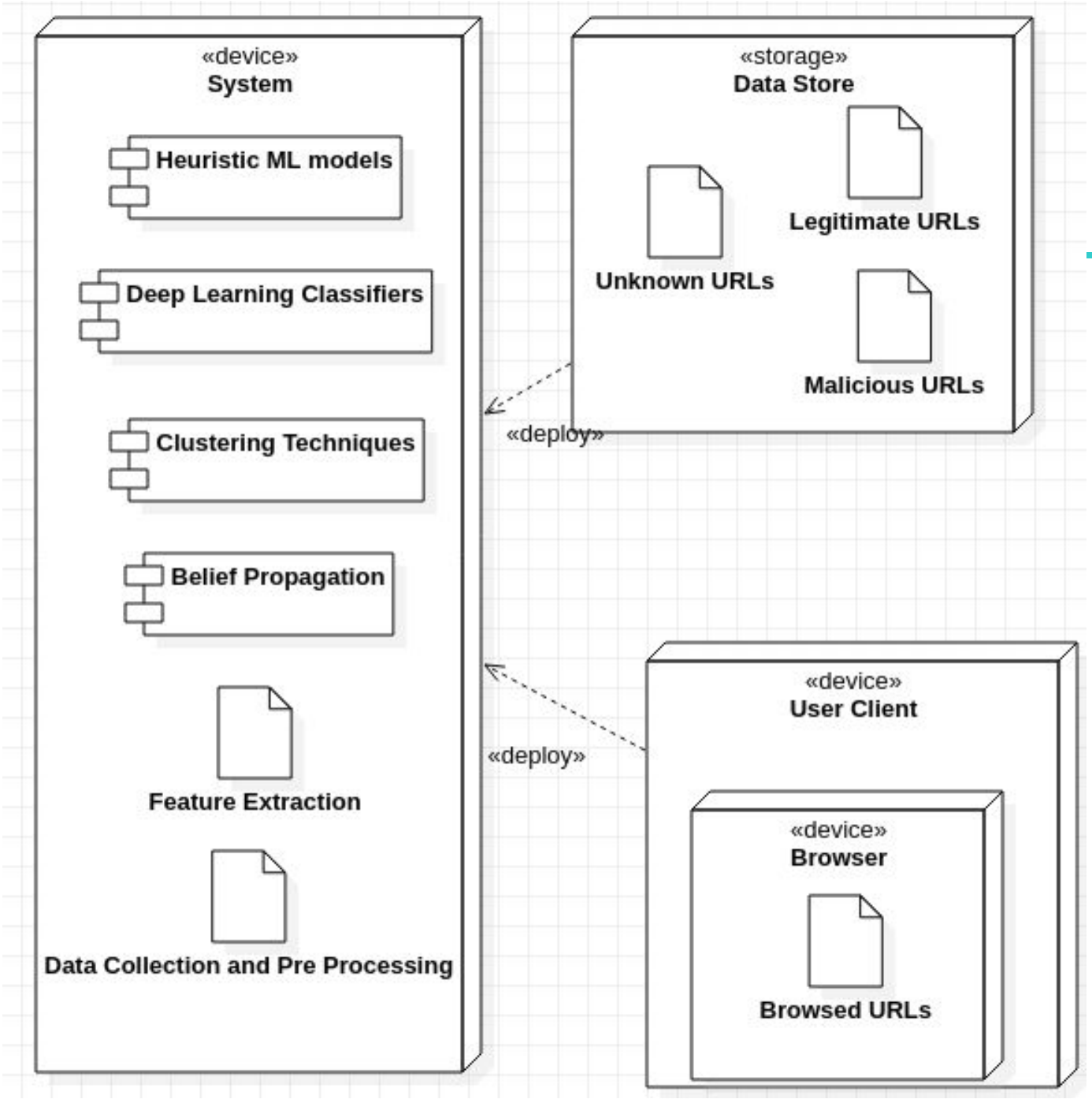
- Cleaner method of studying each URL, thus making a thorough checkup before the actual classification
- Usage of high efficiency models such as neural networks and page ranking algorithms for better performance
- Levels of testing and clarification for accurate classification
- Will identify new URLs that do not belong to the conventional BlackList

Proposed Methodology / Approach

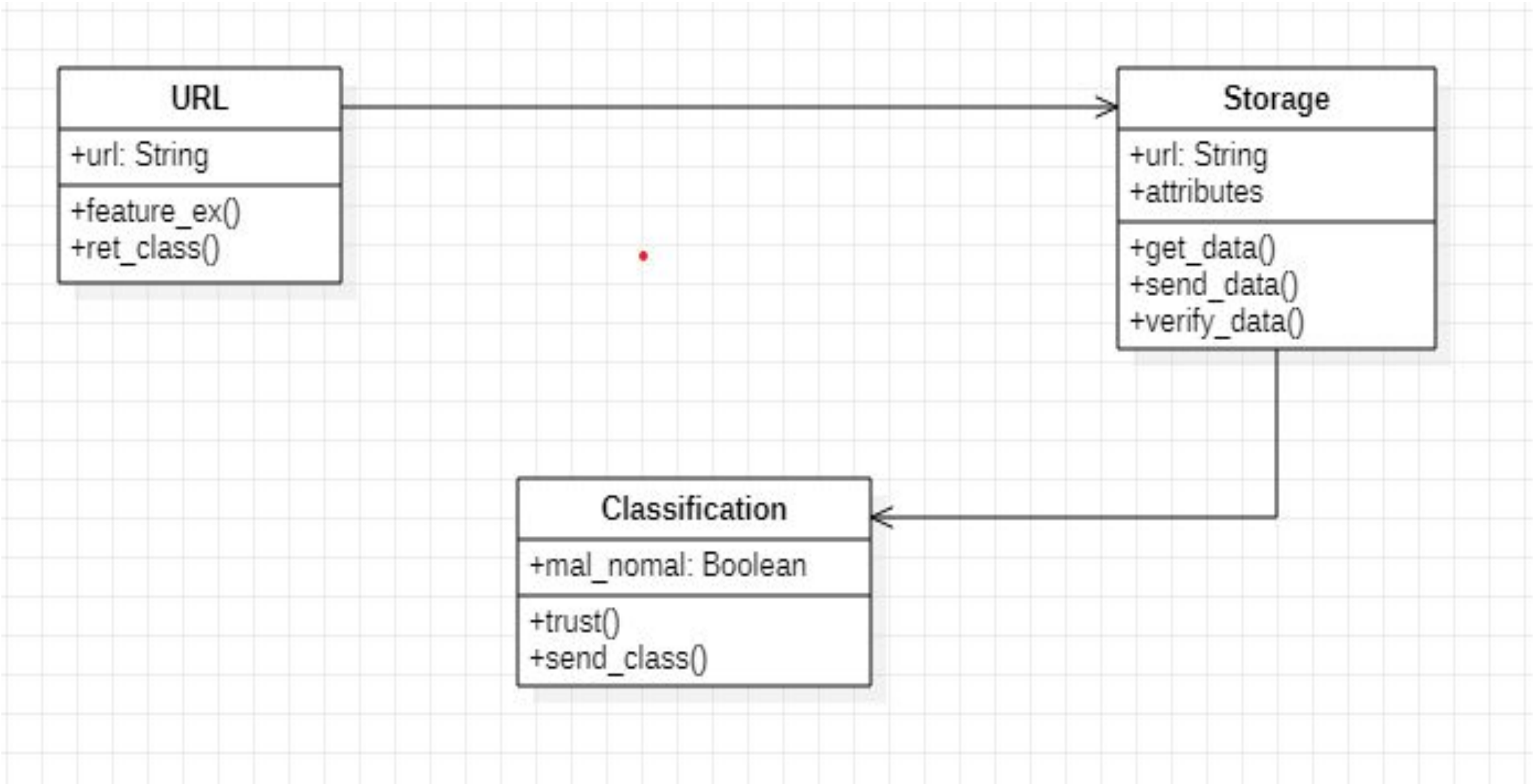
Disadvantages

- Costly process timewise due to a number of methods being used
- Memory utilization is higher due to the use of large modules for the system

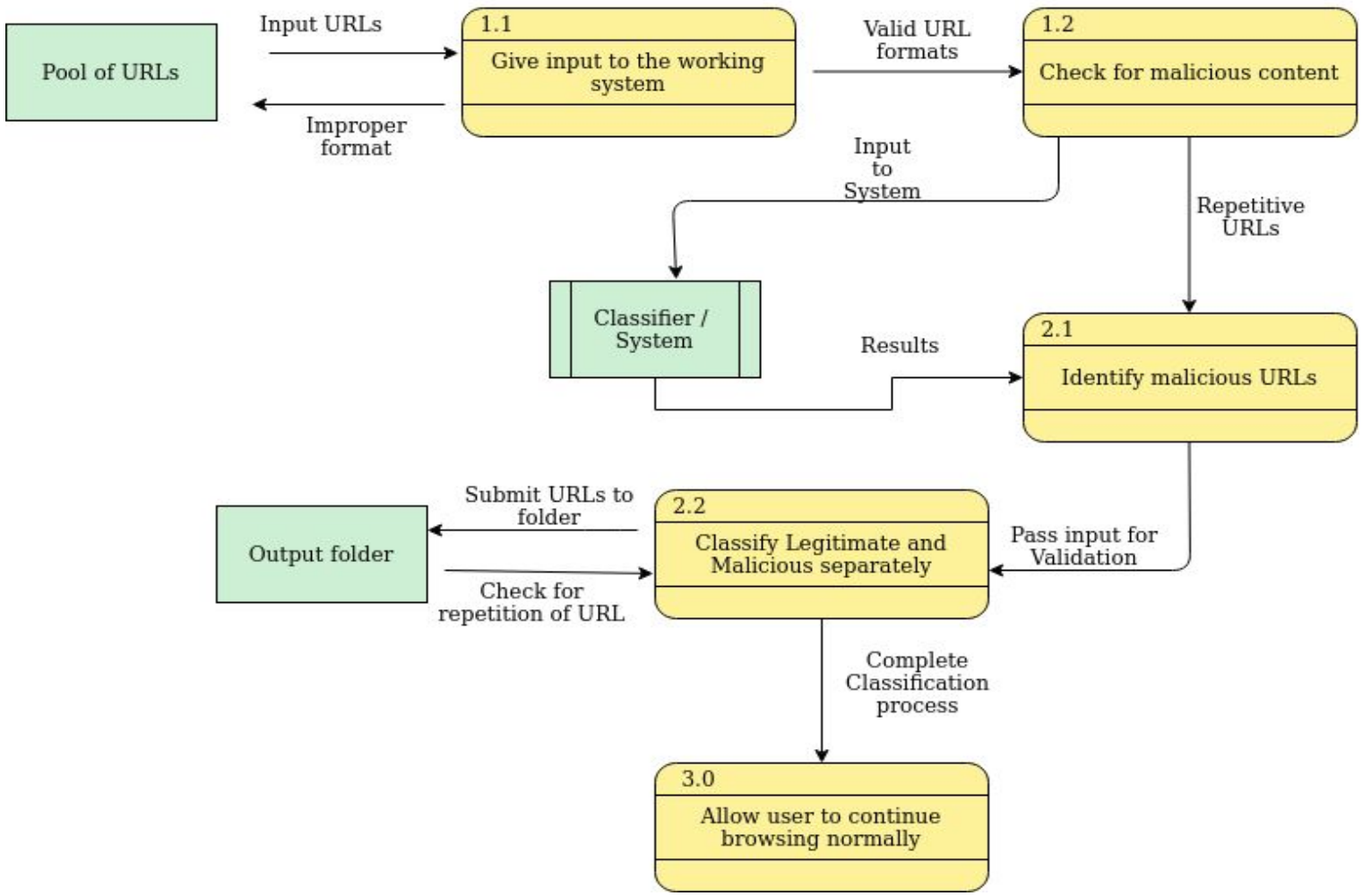
Architecture



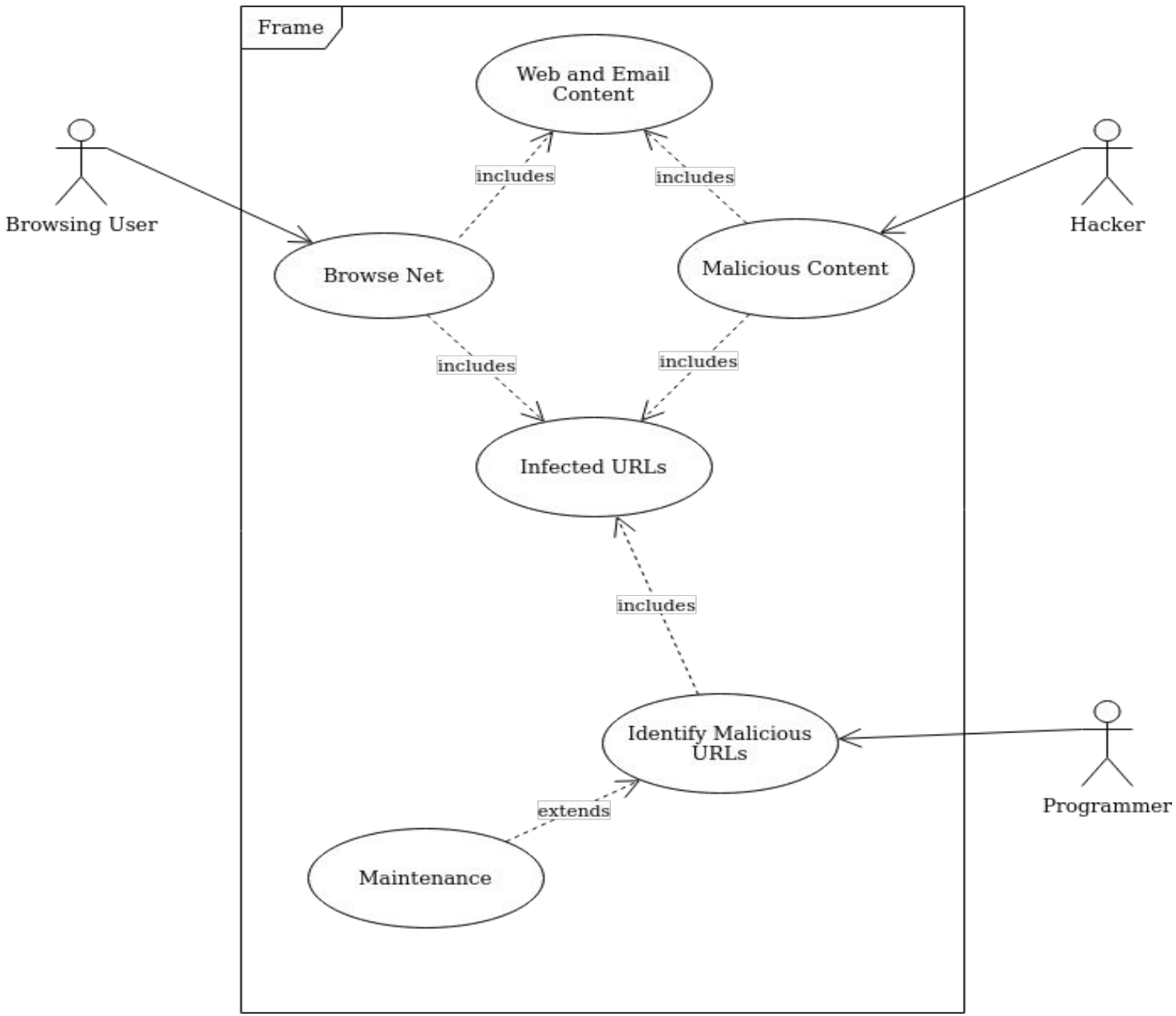
Master Class Diagram



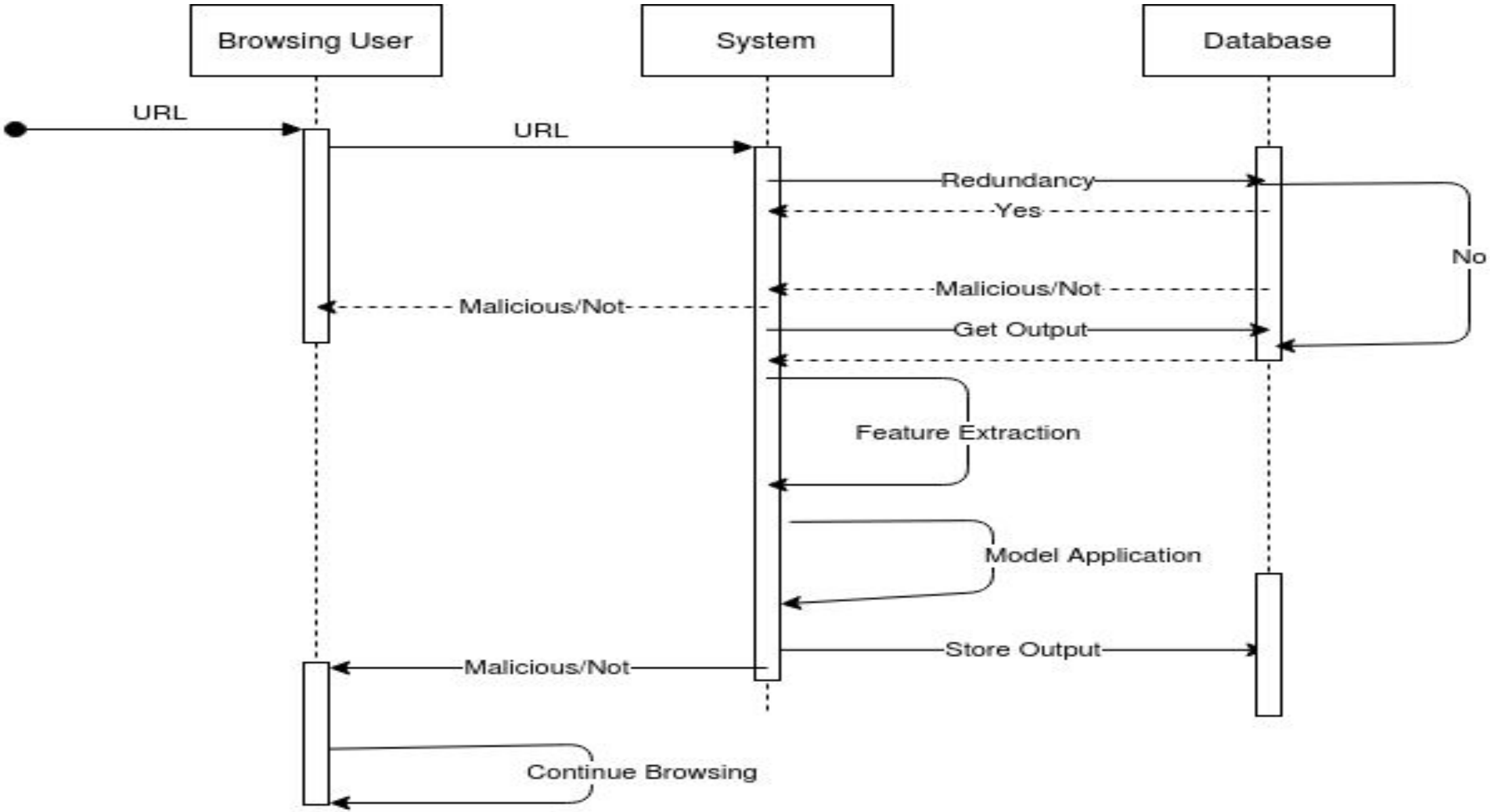
Design Description



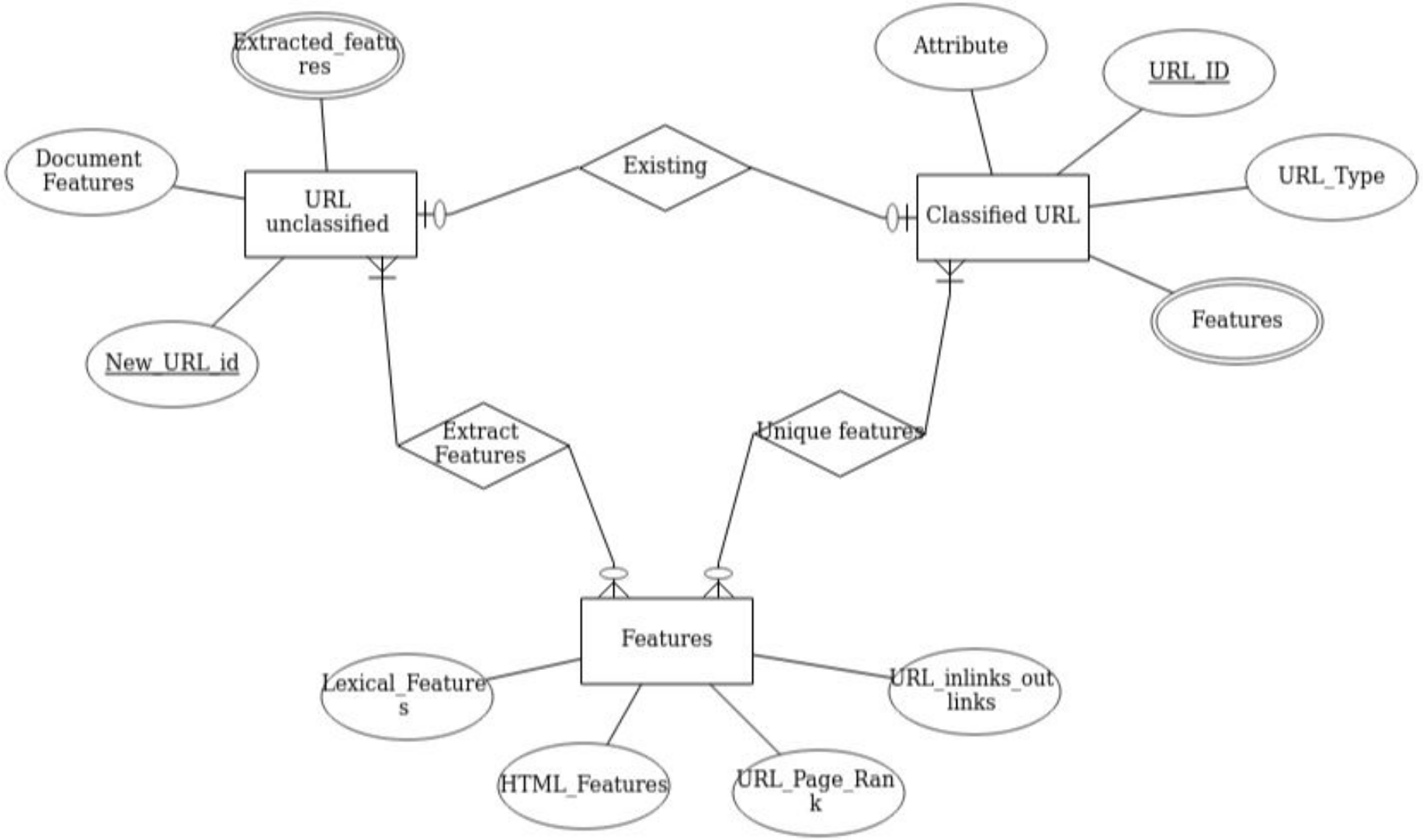
Use-Case Diagram



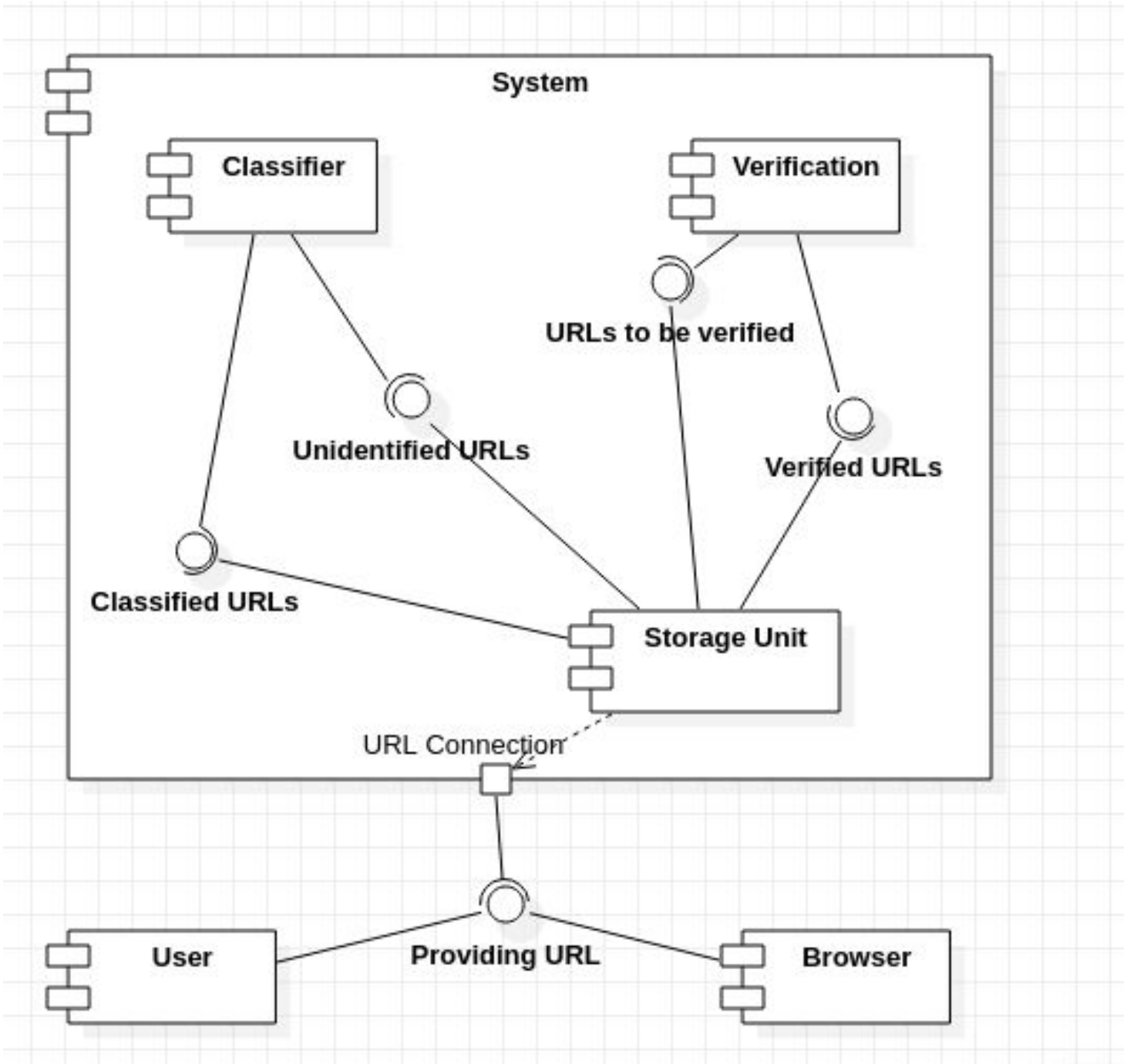
User Interface Diagram



ER Diagram


Extracted_features
Document Features
URL unclassified
Existing
Attribute
URL_ID
URL_Type
Classified URL
Features
New_URL_id
Extract Features
Unique features
Features
Lexical_Features
URL_inlinks_outlinks
HTML_Features
URL_Page_Rank

External Interfaces



Feature Selection

Correlation Coefficient	**Number of Examples Available**	**Computational Resources Required**	**VIF**	**Can more features be derived from it?**
Defines the level to which two features are correlated to each other. Features must have a strong correlation and must not result in spurious values	Adequate number of examples with a good variation in them should be provided in order to get desirable results	The features must be computationally feasible to use for a comparison and should not be	Variance inflation Factor is used to determine the covariance between the features. Lower the value, better the results	This is an advantage if we have many features to test against as it allows us to introduce uniqueness in the classification thus improving on the accuracy.

Data Extraction

Content of URL Accessible	Popularity of Dataset(if dataset)	Language of URL Content(Rejected if not English)
It is important to ensure that we have the URL as well as the contents available so as to perform feature extraction for the same.	Higher the popularity of the dataset, the more reliable it becomes to use as it has been considered a genuine source.	It is impossible to combine multiple languages as the models will not be able to understand them.

Grouping Criteria for validity of a Train and Testing Split

Cross-Validation	Data Variety	Data volume
Resampling technique that detects overfitting, ie, failing to generalize a pattern	Higher the variety in both the testing as well as training datasets, the better the outcome in terms of accuracy since we reduce biased values.	Volume plays an important role in deciding the ratio in which we split the data. Larger the size of the dataset, smaller the ratios

Choice of model for Neural Network

Number of tunable parameters	**Linear Separability of Data**	**Memory Requirements**
Improves the precision and accuracy of the data	The kind of scatter plot obtained for data and whether it can be classified with a line	A neural network that performs similar with lesser memory usage will be preferred

Choice of Model for Graph Clustering

Fowlkes-Mallows scores	**Mutual information based scores**	**Completeness**	**Homogeneity**	**V-measure**	**Time Complexity**
The Fowlkes-Mallows Score is an evaluation metric to evaluate the similarity among clusterings obtained after applying different clustering algorithms.	This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way. This metric is furthermore symmetric	Completeness: A perfectly complete clustering is one where all data-points belonging to the same class are clustered into the same cluster. Completeness describes the closeness of the clustering algorithm to this perfection	Homogeneity describes the closeness of the clustering algorithm to this perfection.	Developed by the combination of the previous 2 terms	Graph related algorithms take a lot of time as it has to explore every relevant node. Hence the time complexity involved is important

Technologies Used

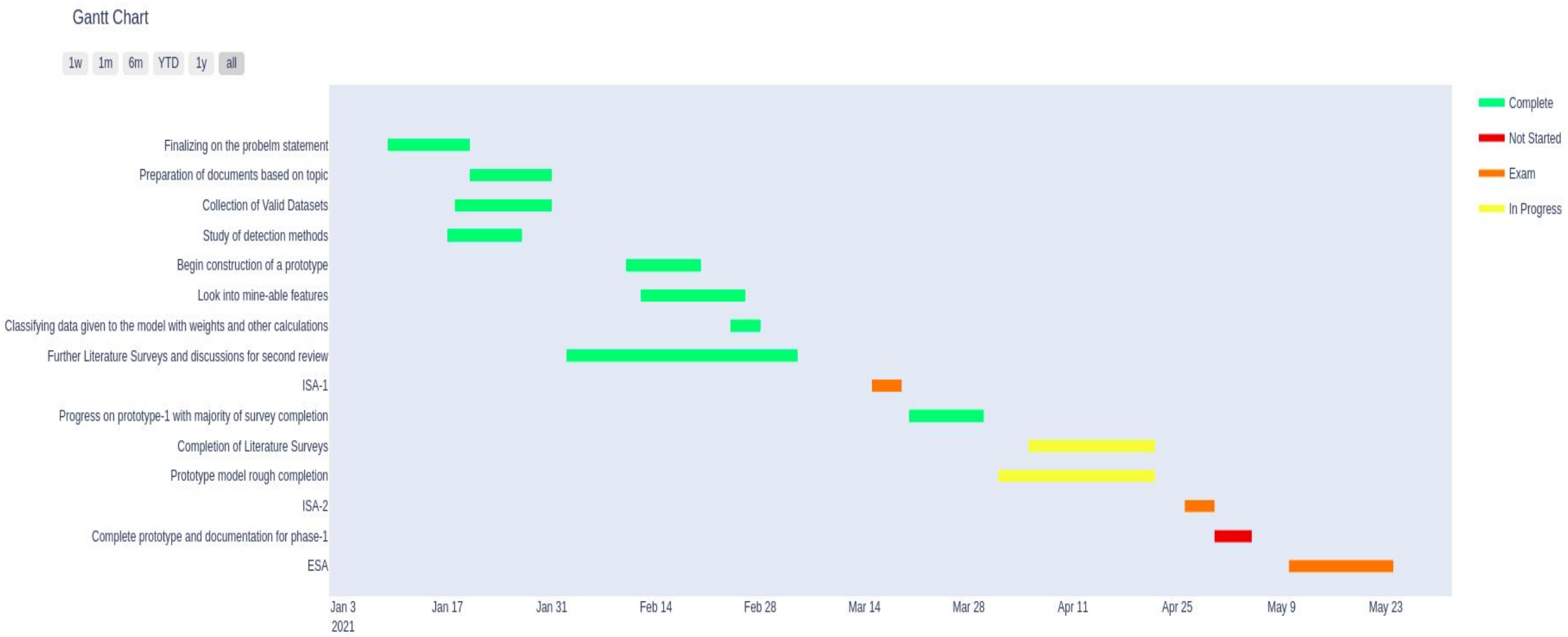
- Operating systems:
  - Windows
  - Unix and Linux
  - Mac OS
- Language Compatibility:
  - Python 3.6 and above
- Libraries used (Major):
  - Pandas
  - Numpy
  - Keras
  - NLTK
  - Scikit Learn and Scipy
- Device Compatibility:
  - Desktops
  - Laptops
  - Phones and Tablets

Project Progress

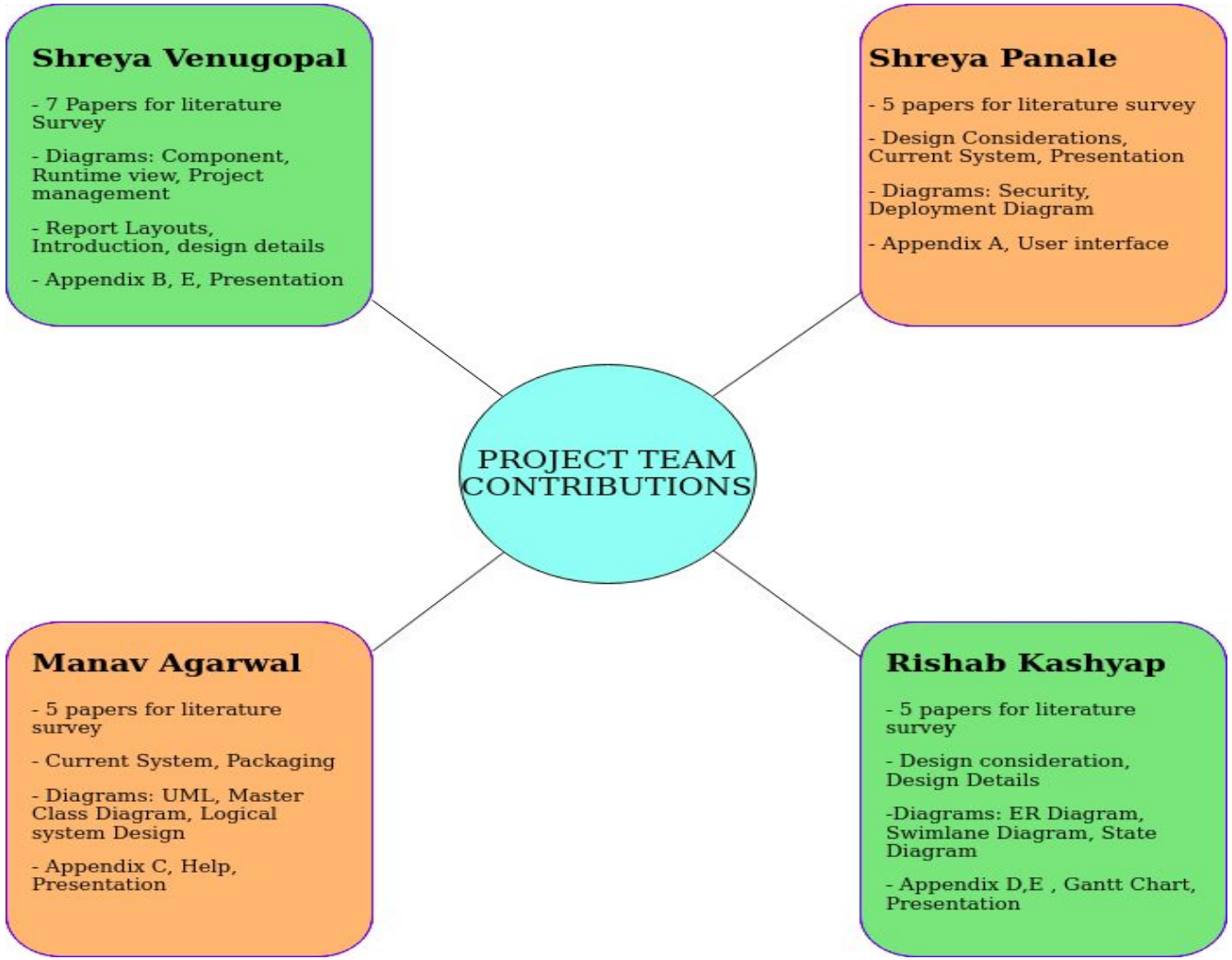
- Prepared Project Requirements Specification, System Architecture
- Identified some datasets and plan on integrating them
- Tested in terms of implementation some Python libraries that would be required for implementation

The Design Phase of the project and about 5% of the implementation is complete.

Capstone (Phase-I & Phase-II) Project Timeline



The plan in terms of efforts by individuals in the team.



Conclusion

1.Basic Approach is to extract features from the URL and use heuristic based machine learning methods for non-graph based features and graph related methods on the graph based features

2. Criteria for extracting the dataset, selecting the features, selecting the models will be based on getting better performance in terms of speed, space complexity and accuracy

3. Since the literature survey and basic design has been completed the project is ready to move to the implementation phase.

References

- Link to the Survey Paper
- Link to the PRS
- Link to the High Level Design Document HLD
- Literature survey papers:
  -

Thank You