Software Project Management Plan

Group No. 28 : Prevention of Cyber Troll & Sarcasm System on Social Networking using Machine Learning with Bilingual Analytics

Tejas Karia, Priya Mane, Jeet Mehta, Pratik Merchant

October 2020

Contents

1 Introduction **2**
1.1 Project Overview 2
1.2 Project Deliverables 2

2 Project Organization **2**
2.1 Software Process Model 2
2.2 Roles and Responsibilities 3
2.3 Tools and Techniques 4

3 Project Management Plan **4**
3.1 Tasks 4
3.1.1 Requirement Analysis 4
3.1.2 Software Requirement Specification 5
3.1.3 APIs 6
3.1.4 Classification Models 6
3.1.5 User interface 7
3.1.6 Integration 7
3.1.7 Testing 8
3.2 Assignments 8
3.3 Timetable 9

1 Introduction

1.1 Project Overview

This project will help to deal with online social media hate speech and automate the process of blocking such malicious accounts. The current process for the social media platforms are manual and there are no automated processes. Since the process is manual it becomes very difficult to keep track of such users who are habitual offenders. There are several categories of cyberhate and each of these are interpreted differently. The project has broken this down to mainly 2 categories: offensive and sarcastic depending on the sentiment and bilingual sentiment analysis on Hinglish comments. Our main target for developing this tool is to empower influencers who do not have the time to tackle the hate speech and thus they have to keep a social media manager who has to manually delete such malicious comments. Primarily, all the comments will be retrieved from the user created posts and then those will be classified using sentiment analysis. An automated response will be generated to the comments.

1.2 Project Deliverables

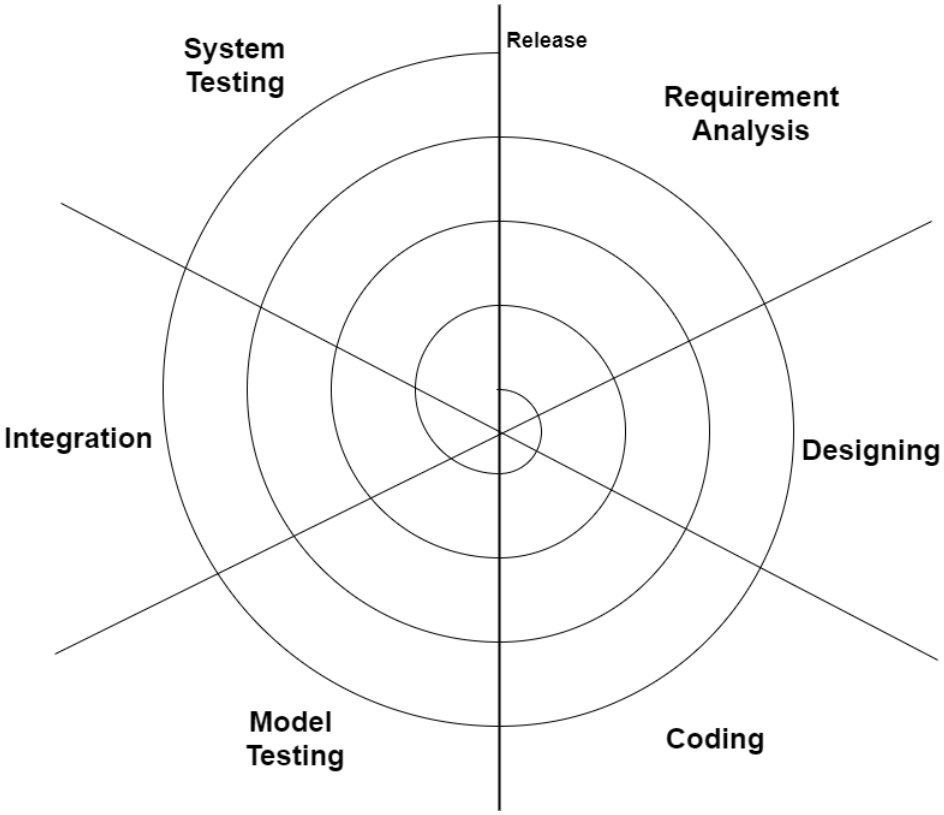
- Requirement Analysis - Mid September
- Software Requirement Specification Document - Mid October
- Algorithm for Troll Detection - Mid October
- Software Design Document - Mid October
- Software Test Document - October End
- Demonstrative User Interface - October End
- Algorithm for Sarcasm Detection - October End
- Project Synopsis - Mid November
- Algorithm for Bilingual analytics (Hinglish) - January End
- Complete User Interface and Database Connectivity - Mid February
- Integrated Web Application - February End
- Addition of Security Features - February End
- Tested Web Application - Mid March
- Final Project Report - Mid April

2 Project Organization

2.1 Software Process Model

The iterative spiral model approach would be used for this project. This model allows us to iterate through each process framework communication, planning, modelling, construction and delivery.

Figure 1: Process Model



2.2 Roles and Responsibilities

1. Tejas Karia (1714088) - Project Manager
The project manager will look after the overall functioning of the project and assign tasks to each of the members. He/She will act as the face of the project and will lead all the activities.
2. Pratik Merchant (1714093) - Designer
The designer will design the overall system and will lead all the activities of designing the system in the best and most efficient way possible.
3. Priya Mane (1714091) - Developer
The developer will define how the design is converted to reality by defining the implementation details of the various modules and successfully developing the same.
4. Jeet Mehta (1714092) - Tester
The tester will lead all the activities of testing each and every module that has

been implemented to determine the errors and will hence inform the developer to overcome the same. If a design flaw has been spotted, the designer would be informed first and then the developer.

2.3 Tools and Techniques

• The application will be developed for desktop usage. It would be compatible across any standard browsers such as Google Chrome, Mozilla Firefox, Internet Explorer, etc.
• SQLite database will be used to store user login credentials and details for the web application.
• The algorithms will be implemented using Python.
• For developing demo UI / Wireframes , a wireframing tool would be used.
• OAuth authentication service will be used to get special rights for performing operations on behalf of the user.
• The respective APIs will be used for accessing data from social media platforms.
• The web based application will be developed using the Django framework for backend and HTML,CSS for frontend development.
• For development synchronisation purposes, version control system would be used.
• Git will be as version control system as it can be easily used in android studio and simple to understand.

3 Project Management Plan

3.1 Tasks

3.1.1 Requirement Analysis

3.1.1.1 Description Requirement Analysis would be done to know the exact expectations of the client from the product. The functionalities and working of the product would also be clear by doing sufficient and effective requirement analysis.

3.1.1.2 Deliverables and Milestones Effectively communicate with all the actors involved in the working of the web application. By the end of this process, the design and development team will be sure of the functioning of the web application.

3.1.1.3 Resources Needed For effective requirements analysis, multiple meetings would have to be conducted with the stakeholders of the web application. Brainstorming sessions or joint discussions must be organised for effective communication and information gathering.

3.1.1.4 Dependencies and Constraints Task cannot be completed without conducting meetings with the stakeholders and knowing the expectations from the web application.

3.1.1.5 Risks and Contingencies Risks: The risk involved would be only failure to communicate with the organisations involved and users. Another issue could be miscommunication between the stakeholders and developers/designers.
Contingency: Can be tackled by having multiple sessions and creating well defined SRS and getting it approved by the client.

3.1.2 Software Requirement Specification

3.1.2.1 Description The users and the client get a brief idea about the software while in the initial stages. The purposes and intentions as well as the expected results are properly defined. It hence lays the outline for software design. The desired goals are defined thereby easing off the efforts of the developers in terms of time and cost. It forms a basis for the agreement between the client and the developer. It becomes easier while transferring and using the solution elsewhere or with new customers as the basis of functioning of the software is mentioned. It acts as a material for reference at a later stage. It acts as a basis for reviews.

3.1.2.2 Deliverables and Milestones The document focuses on briefing all the members of the team as well as the client about the specifications and functionalities of the software project.

3.1.2.3 Resources Needed Meetings with stake-holders and brainstorming sessions or joint discussions must be organised for requirement gathering.

3.1.2.4 Dependencies and Constraints
Task cannot be completed without conducting meetings with the stakeholders and knowing the expectations from the web application.

3.1.2.5 Risks and Contingencies Risks: Matter of risk mainly revolves around communication and necessary documentation. If the SRS isn't well defined and well addressing each and every aspect of the project, then major miscommunication and false information transfer could take place. Clients may face issues on being on the same page as the developing team. The expectations and deliverables would have explosive differences between them.
Contingency: Can be tackled by having multiple sessions and creating well defined SRS and getting it approved by the client.

3.1.3 APIs

3.1.3.1 Description Using APIs of social media platforms, comments will be retrieved and automated responses would be generated. The flagged comments will be deleted or reported using the API calls.

3.1.3.2 Deliverables and Milestones The API will facilitate the retrieval of comments on an user's post. Based on the sentiment of the comment, then, an automated response will be provided. Deletion or reporting of flagged comments.

3.1.3.3 Resources Needed User login credentials and user authentication services.

3.1.3.4 Dependencies and Constraints The task cannot be completed without getting authentication from the respective social media platform regarding the API usage.

3.1.3.5 Risks and Contingencies Risk - The risk involved is with the user authentication for each social media platform, the authentication must allow access through some authentication service for the application to carry out tasks like posting replies and blocking users or deleting user comments based on the results of sarcasm detection and troll detection.
Contingency - Proper authentication rights can be acquired by certain services offered by the particular social media platform. Exploring these services can help overcome the risk mentioned above.

3.1.4 Classification Models

3.1.4.1 Description A classification model for classifying offensive or sarcastic comments algorithms will be coded. It will include pre-processing, training and testing the data.

3.1.4.2 Deliverables and Milestones The model should be able to classify offensive or sarcastic comments with utmost accuracy and not overfit the data.

3.1.4.3 Resources Needed Huge amount of labelled dataset, Google Collaboratory for training and testing the models.

3.1.4.4 Dependencies and Constraints The accuracy of the model depends highly on the dataset and suitable processing power for executing the algorithms. The labelled dataset should be accurate and also from the same distribution for better results.

3.1.4.5 Risks and Contingencies Risks: Dataset not being labelled correctly and not from same distribution. Model overfitting or underfitting the data or have high bias and/or variance.
Contingencies: Select dataset with accurate labelling and from the same distribution. High bias can be solved with more data and high variance can be solved by dropout method.

3.1.5 User interface

3.1.5.1 Description This unit will be the most crucial part of the project since the target audience includes social media users and hence the ease of using this application must be comparable if not better, to the existing social media applications. The home screen will describe the application features and login/signup buttons. Once the user signs up for the application and logs into his/her account, the user is taken to a dashboard. The dashboard will have different tabs or sections for various social media platforms. For each application, there will be an authentication for requesting access to the user's account with rights to perform required operations. Once authentication is done, users will be displayed with the details of the comments and further analysis after applying algorithms for sarcasm detection, troll detection and bilingual analysis.

3.1.5.2 Deliverables and Milestones Easy to use and intuitive user interface.

3.1.5.3 Resources Needed Wireframing tool for creating demonstrative user interface web pages. HTML and CSS will be used primarily for developing the User interface.

3.1.5.4 Dependencies and Constraints Without adequate feedback from the client regarding ease of navigation and usage on the website, it would be difficult to design it effectively.

3.1.5.5 Risks and Contingencies Risk - The primary risk could be difficulty for a new user to navigate through the website or difficulty in comprehending the results of the analysis displayed on the dashboard.
Contingency - The risk can be mitigated by seeking continuous feedback from users.

3.1.6 Integration

3.1.6.1 Description Integration of various social media platforms, their services and algorithms for sarcasm and troll detection on the website to make the user dashboard.

3.1.6.2 Deliverables and Milestones Fully developed web-based application with insightful user interface.

3.1.6.3 Resources Needed The framework used for web development will be Django for backend and HTML, CSS for the frontend. SQLite database for storing user credentials.

3.1.6.4 Dependencies and Constraints All social media platform APIs must function smoothly under a single application.

3.1.6.5 Risks and Contingencies Risk: If any API fails, the user won't be able to have a wholesome experience. Integration may result in unforeseen bugs.
Contingency: The system integration can be tested by conducting rigorous integration tests to mitigate bugs.

3.1.7 Testing

3.1.7.1 Description Once the design and final software is developed, the application goes for testing. Testing is based on different criterias related to efficiency, bugs, performance, response time, correct functionality, etc.

3.1.7.2 Deliverables and Milestones The web application must be able to perform all the tasks successfully and give all expected results.

3.1.7.3 Resources Needed Dummy data and some users would be required to test the web application.

3.1.7.4 Dependencies and Constraints The step will be incomplete without the web application being handled by real users and getting their feedback.

3.1.7.5 Risks and Contingencies Risk: The web application might not be satisfactory or the functions/code might cause some anomalies and give wrong results.
Contingency: If the web application gives faulty results, it can be corrected by debugging the app and again presenting it to the customer.

3.2 Assignments

Task 1: Requirement Analysis - Tejas, Pratik
Task 2: Software Requirement Specification - Pratik, Jeet
Task 3: APIs - Tejas, Priya, Jeet, Pratik
Task 4: Algorithms - Tejas, Priya, Jeet, Pratik
Task 5: User interface - Tejas, Priya
Task 6: Integration - Tejas, Priya, Jeet, Pratik
Task 7: Testing - Jeet, Priya

3.3 Timetable

Title	T/M	Start	End
Requirement Analysis	T	07/09/2020	14/09/2020
Software Requirement Specification Document	T	15/09/2020	15/10/2020
Algorithm for Troll Detection	T	25/09/2020	15/10/2020
Software Design Document	T	01/10/2020	15/10/2020
Software Test Document	T	15/10/2020	23/10/2020
Demonstrative User Interface	T	10/10/2020	30/10/2020
Algorithm for Sarcasm Detection	T	15/10/2020	30/10/2020
Project Synopsis	T	01/11/2020	15/11/2020
Algorithm for Bilingual analytics (Hinglish)	T	01/01/2021	30/01/2021
Complete User Interface and Database Connectivity	T	31/01/2021	15/02/2021
Integrated Web Application	T	08/02/2021	28/02/2021
Addition of Security Features	T	15/02/2021	28/02/2021
Tested Web Application	T	01/03/2021	15/03/2021
Final Project Report	T	16/03/2021	15/04/2021

Gantt Chart for Timetable: