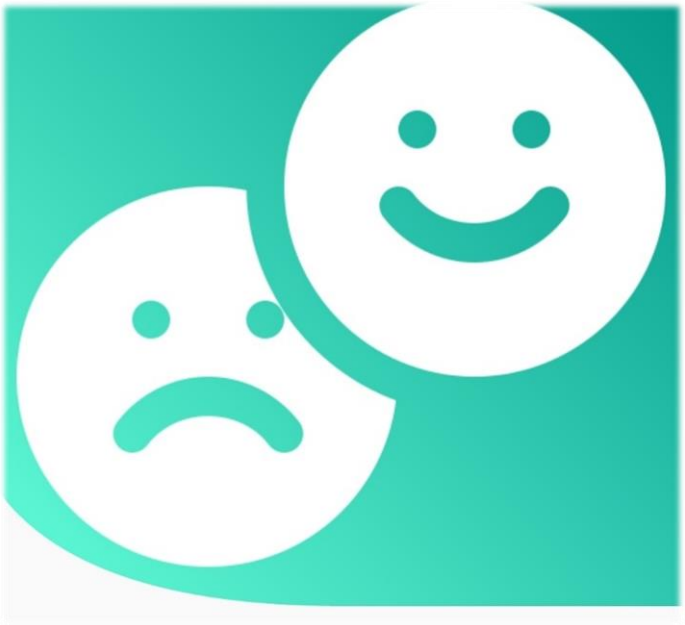
Software Requirement Specification



Project: Sentimento

“Social media assisting platform with sentiment analysis”

Table of Contents

Introduction

The SRS document is a non-technical report clarifying the features/capabilities of the software, system or application. It contains detailed information on functional and non-functional requirements of the project. This document is prepared identifying and evaluating the client's requirement. If the project is not being built for specific client, the survey feedbacks of the potential users of the system is used to prepare the software requirement specification document (Sommerville, 2011).

The qualities of a clear and in-depth SRS document are as follows (GeeksforGeeks, 2021):

i. The requirements stated by client or feedbacks from surveys are clearly addressed in the document.
ii. All the must have functional and non-functional requirements are properly attached.
iii. Preciseness over ambiguousness is another essential quality for a good SRS document.
iv. It is better for a SRS document to be easily modifiable as requirements of a project in SCRUM of Agile are prone to have changes or improvements in the further iterations of the project.
v. The SRS document is also made for client or non-technical viewers. It is a good practise to make this document easily understandable with use of simpler language and terms.

Purpose

Primary objective of this SRS document is to provide substantial information about the functional and non-functional requirements of the Sentimento platform. This document clearly highlights must include features of the platform to function. The other tool and technologies requirements for the development of the project is also defined in this document.

The core purpose of this mobile application is to provide a user-friendly platform where non-technical and technical (with understanding of Data Analytics) users can perform sentiment analysis within same interface.

Intended Audiences of this document

This document is primarily intended for the development team of the Sentimento project. Developers from algorithm team, backend and frontend including other team members engaged in technical and non-technical areas are the key readers/viewers of this document. The client of the project or potential users can also read this document to verify their demands are included in the document.

Why this document for the developer team?

Viewing all the features to include in the platform, developers can identify and plan their moves accordingly. The vocal hearing from the client is not consistent and official but this document with client's approval provides clear vision on what to do and how to implement those essential features in the platform. This document can also be a proof of record of tasks to carry out for both the parties (client and development team).

Contents

User's need:

From the feedbacks of pre-survey, a mobile application with easy-to-use interfaces is to be developed where they can just provide video URL or topic of tweet. And just with this little information, users should be able to do their sentiment analysis task. The demand of a social media related job vacancies posting/viewing feature is also considered to include in the system.

Assumptions and dependencies:

The development work of this project is first divided into four sections and carried out sequentially. The four sections are as follows:

i. Sentiment analysis functionality

Two functions are to be made to extract comments of a chosen video of YouTube and tweets using each of the respective third-party documentation. After that, a function to clean the data is made and using Multinomial Naive Bayes functionality of scikit-learn and training data models, sentiment analysis module is developed.

ii. Backend development

Flask framework will be used to make the application programming interface (API) of the system. The CRUD operations and sentiment analysis module will be integrated with this part of the development and it interacts with the frontend part of the system to complete the overall usefulness of the system.

iii. Frontend development

Flutter, cross-platform application framework is to be used to effectively developed all the features of the designed platform. For state management task, Business and Logic Unit (BLOC) will be used. The packages from the Flutter community will be utilized to perform basic functionalities quickly and efficiently. The folder structure will be maintained as recognized by Flutter developer community. The user interfaces will be broken down into smaller components to make the particular code easily accessible when needed to make any changes.

iv. Hosting and taking to production level

Heroku, a cloud hosting service is to be used to host the Python backend. The free account with zero payment to make service will be utilized to make the platform's API live.

The mobile application is estimated to publish on Google Play Store if the developed application at the end fulfils all the policy requirements of publishing application on Play Store.

Requirements

Functional requirements:

i. Primary Features

<table>
<tr><th>Req.ID</th><th>Requirement Description</th></tr>
<tr><td>FR. 1</td><td>User can register to system with unique and valid registration detail
<table>
<tr><th colspan="2">System Requirement</th></tr>
<tr><td>SR. 1</td><td>User can provide their correct registration information</td></tr>
<tr><td>SR. 2</td><td>Backend responds with registration success or fail according to user provided information</td></tr>
</table></td></tr>
<tr><td>FR. 2</td><td>User can login to system with their credentials used on registration
<table>
<tr><th colspan="2">System Requirement</th></tr>
<tr><td>SR. 3</td><td>User can provide their correct login credentials</td></tr>
<tr><td>SR. 4</td><td>Backend responds with login fail in case of invalid information or else user will be forwarded to dashboard</td></tr>
</table></td></tr>
<tr><td>FR. 3</td><td>User can perform sentiment analysis on YouTube comments and tweets
<table>
<tr><th colspan="2">System Requirement</th></tr>
<tr><td>SR. 5</td><td>User can provide YouTube video URL or topic of tweet to perform sentiment analysis</td></tr>
<tr><td>SR. 6</td><td>Backend responds with sentiment report if valid input is given by user.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>FR. 4</td><td>User can view open job vacancies and skill offering from other users of platform<table><tr><th colspan="2">System Requirement</th></tr><tr><td>SR. 7</td><td>User can choose either job vacancies or freelancing services to view</td></tr><tr><td>SR. 8</td><td>Backend responds with currently available vacancies posted by other users of platform</td></tr></table></td></tr>
<tr><td>FR. 5</td><td>User can post social media related job vacancies and their skill offering in the platform<table><tr><th colspan="2">System Requirement</th></tr><tr><td>SR. 9</td><td>User can provide information about their vacancy or freelancing skill.</td></tr><tr><td>SR. 10</td><td>Backend stores the data in the database that is accessible by other users later</td></tr></table></td></tr>
<tr><td>FR. 6</td><td>User can view their detailed profile<table><tr><th colspan="2">System Requirement</th></tr><tr><td>SR. 11</td><td>Profile viewing REST API request can go from frontend to backend with user’s authentication detail</td></tr><tr><td>SR. 12</td><td>Backend responds with user’s profile data if authentic request is provided.</td></tr></table></td></tr>
<tr><td>FR. 7</td><td>User can save and view their past sentiment analysis reports<table><tr><th colspan="2">System Requirement</th></tr><tr><td>SR. 13</td><td>User can have the option to save sentiment report for further uses.</td></tr><tr><td>SR. 14</td><td>Backend stores the report in the database.</td></tr></table></td></tr>
</table>

ii. Authentication

Bearer authentication tokens will be used by the system to authenticate user's request to backend.

iii. Reporting/Documentation requirements

Use case diagram, system architecture, flowchart and similar diagrams are designed before the start of development of the project.

Non-functional requirements:

i. Performance: To reduce the memory usage and shorten sentiment analysis algorithm processing time, the machine learning model is to be serialized into pickle file.

ii. Scalability: The backend of the project is to be hosted in cloud for larger scale accessibility of the project.

iii. Security: For user's security reasons, the platform should not store third party access tokens. Considering it, the tokens will be locally saved in user's device.

Approval

The platform is designed and developed to fit all the potential users like data analyst, marketing and news media professionals and social media related job seeker/provider. It clarifies that this project is not made for any particular client. The platform is developed keeping in mind to fulfil the professional needs of potential users. The user interface and database architecture are structured for all the general users of the platform.

This project is designed to suit the interest of multiple or large-scale users, there is no any particular client of this project to approve this software requirement specification. Instead, our supervisors of the project have approved the proposed system to carry out the development works of this project.

Bibliography

GeeksforGeeks, 2021. *Software Engineering* | *Quality Characteristics of a good SRS.* [Online]
Available at: https://www.geeksforgeeks.org/software-engineering-quality-characteristics-of-a-good-srs/
[Accessed 9 December 2021].

Sommerville, I., 2011. Software Requirement Documents. In: M. H. Marcia Horton, ed. *SOFTWARE ENGINEERING.* Hagerstown: Pearson, pp. 91-98.