REQUIREMENTS DOCUMENT

1. PROJECT INFORMATION

Project Name: Abstractive Conversation Summarizer With Open Source AI Tools
Date of Submission: 3/22/2023
Version: 1.0
Client: Impact Intell

2. PROJECT TEAM

Team Member	Role	Document Responsibilities
Akshat Baranwal	Product Owner	Functional-Requirements, Data Management.
Mark O'Connor	Developer	System, Design, and Development Constraints, System Environment
Steven Warner	Scrum Master	External Interface Requirement
Cameron Caruso	Developer	Change Control
Parth Dalwadi	Developer	Non-Functional Requirements
Michael Provenzano	Developer	Project Scope and Definition

3. PROJECT SCOPE AND DEFINITION

The scope of this project is to create an AI powered text summarization tool using open-source machine learning libraries and functions in order to modify input as conversations from a client's database and convert it to a brief paragraph displaying the major highlights.

This tool will allow police officers to communicate effectively with members of their department as well as other departments. The necessity of this tool was realized when a major problem of Criminals using information silos to hide from law enforcement was understood. There was a lack of such a tool in law enforcement to allow officers to effectively communicate across cases or departments. The product deliverable consists of, but not limited to, a machine learning algorithm that can produce human-like summaries of text conversations. This algorithm will also be implemented on the client's

Python Django based web server. The algorithm will also be able to retrieve text conversations based on a user's given data range.

Expected Progress by Increment -

- The first increment is a data cleaning Python script that works by removing stop words such as 'the' or 'a' and by tokenizing words into a readable format for the machine learning algorithm.
- The second increment is a general machine learning model that takes generalized input in number format and is untrained.
- The third increment will be a trained machine learning model that will take text input and give a text response.

- The fourth increment will allow the trained machine learning model to be uploaded and integrated into the front end server application.
- The fifth increment will have the machine learning algorithm take text input based on a given date range. The deliverables will also be fully tested and bug-free.

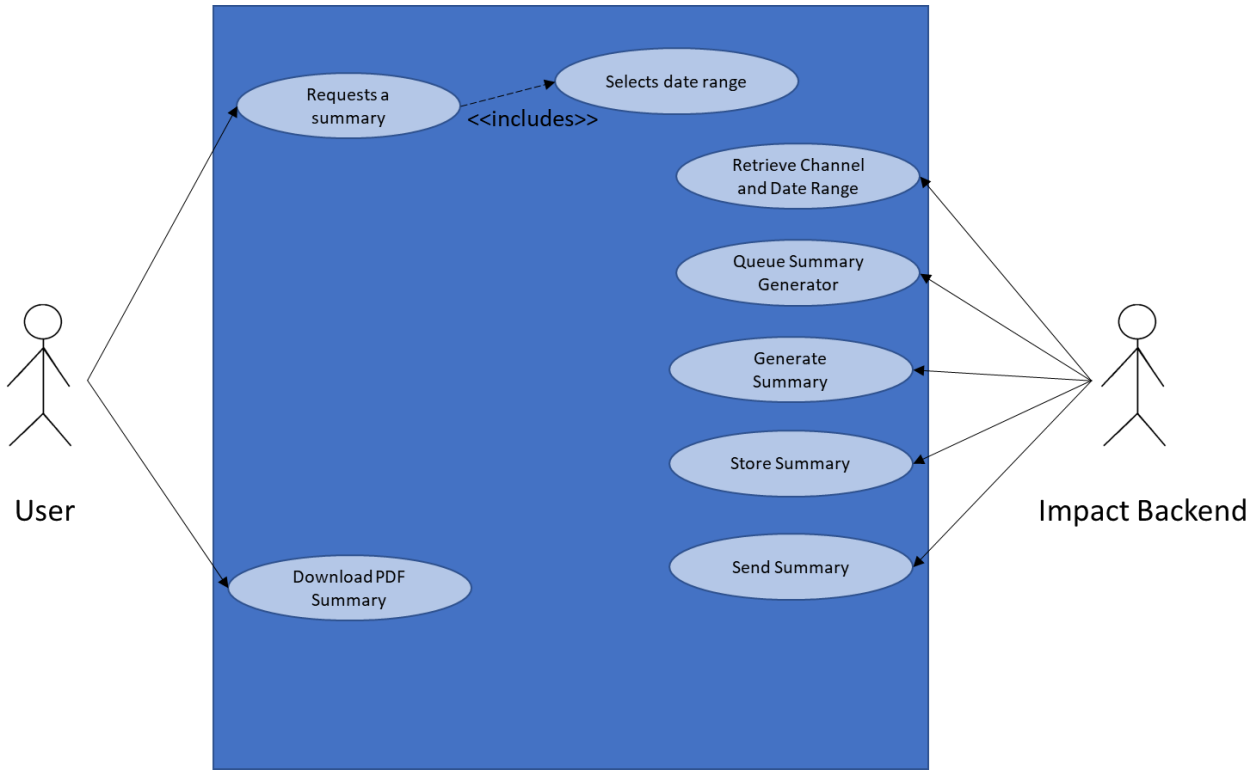
The major constraint is data cannot leave the server when the algorithm is running. The data used to either train the algorithm, or to be given as input is regarded as highly classified and needs to be secured.

The most important feature of the product is that it delivers a summary that is readable and accurate. The program should also be able to store the summaries? quoted information without paraphrasing it. The program will also be able to have limits placed on its input in the form of data ranges.

4. SYSTEM ENVIRONMENT: USE CASE DIAGRAM

- The user starts the scenario by requesting a summary from the Impact app.
- When they request a summary it is required that the user also selects a date range.
- From there the Impact backend will retrieve the selected channel and date range. It will queue the summary generator function.
- The summary generator function will run and the result will be stored in the server.
- The backend server will send the summary to the user.
- The user will then select to download the summary as a PDF.

Use Case Diagram-



5. FUNCTIONAL REQUIREMENTS

Functional Requirement	Conditions [User Story]
Accepts requests: The product must act on the request of the user on generating summary.	Given the request by the user, the product must generate a summary of text conversations after validating the date range requested by the user.
Input Processing: The tool must be able to read the conversation data in a required format.	In this case, the input is a text **.csv** file. The product should be able to extract relevant information from the input source and preprocess it for analysis.
Natural Language Processing: The tool must have the ability to understand natural language and extract key information from it.	The product uses techniques such as Sentiment Analysis to identify important concepts, entities, and emotions.
Summarization: The primary function of the tool is to summarize the conversation.	The product should be able to generate a concise and coherent summary that captures the main points of the discussion.
Output Generation: The tool should be able to present the summary in a readable format.	The product must display output as a text file which can also be available as .pdf format.

6. NON-FUNCTIONAL REQUIREMENTS

Non-Functional Requirement	Description
Performance Requirements	The tool should be able to process the conversation and generate a summary in a timely manner.
Security	The product is designed with the help of open-source python libraries which will not send data to any external server, rather keeping it on the local server which makes it a lot more secure.
Compatibility	The program must work on a Linux Ubuntu 22.04 OS at minimum.
Working Environment	This application is to be integrated within our client's existing product. So it must work in a Django framework on a Ubuntu server.
Availability	As this program will ultimately be integrated into our client's servers, and because it is the main feature of their servers, it will always be available as long as our client's servers are available.
Accuracy	The product must be trained on text input and target summary using a transformer model, so that it does not omit the useful information during the time of generating summary.
Usability	The application should be simple to use. Users should only need to select a chat room and date range. The application should give results from that.

7. EXTERNAL INTERFACE REQUIREMENTS

The interface for the users will consist of a separate option to request a summary on the frontend part of the app. Entering that option, the user will be prompted to select a range of dates. Once the date is selected, the user will automatically get a summary if the date range is valid, otherwise will be getting another prompt to select the date again.

The product inference must work on an "Ubuntu Server". This server works under a python based framework, Django. Along with this, the product will use multiple open-source python libraries such as pytorch and transformers.

8. SYSTEM, DESIGN, AND DEVELOPMENT CONSTRAINTS

Privacy concerns are a main constraint to our product. The product will have access to confidential and sensitive information. To meet this constraint, all python tools used

must be open source and secure. The server environment will dictate that all deliverables will have to be written in python within a Django web server.

Amount of available data will impose a constraint on the deliverable. Machine learning models require large corpuses of data, especially tasks that use natural language processing, and we will not have access to.

9. DATA MANAGEMENT

The required data set of conversations are to be stored in a folder at the root directory in the form of .csv files. The stored files of the decision tree and the transformer will be in a folder in the main directory. The program and dependencies are to be saved in the client's server at a specific location, the "frl" directory. The data is stored and managed by a provided database. This database is already integrated into the Django application.

10. CHANGE CONTROL

All project changes are discussed and considered by the development team and client. The team uses the guidance of the Impact sponsors, running some of the more important changes being made by the sponsors to ensure the product will end up as desired. Additionally, a Trello board is used to help the team stay on track, so that the product can be completely planned out step by step and can be broken up into sprints. If changes are made, the Trello board is then updated to accommodate the changes being made.