1. What is the uniqueness/novelty added by you to the defined problem statement?

The application was designed primarily to search the StackOverflow site using a query, or a question, and obtain a specific number of relevant questions available online, and their most relevant answer, using their public API.

The application's tasks are:

1. To identify most relevant questions to a query – This has been achieved by extraction of keywords and then using them to query StackOverflow.
2. Identify the matching tags and pick top relevant questions from stack overflow – This has been achieved by using Keyword Graph and Edit Distance based ranking.
3. To identify top k solutions of the problem – This has been achieved by ranking of answers based on the the result of sentiment analysis of the answer's comments, the score of the answer received from StackOverflow and whether the answer has been accepted or not.

In addition to these, the application was designed to be usable through a HTML-based User Interface, which is accessible through any web browser, and to run on various Cloud Platforms with minimum changes.

The application was designed as a micro-service, such that it can be integrated into a larger application, using the Flask Library.

2. How is the proposed solution impacting the business? How are the business processes simplified or bringing value over the existing process?

The application can potentially enrich the learning experience of the inexperienced developers, by picking efficient solutions from StackOverflow and thus contribute to business growth of any Information Technology services based organization, by saving time and energy of the new developers required to search StackOverflow.

3. Architectural flow of the proposed solution, with the mention of technologies to be used in developing the solution.

This section describes the architecture of the proposed solution. It uses the StackAPI, NetworkX Library, Natural Language ToolKit and IBM Watson Natural Language Understanding API for obtaining data and determining relevant solutions. The application was designed as a Flask application, which can be run on Cloud Platforms, like Amazon Web Services and IBM Cloud Platform. The architecture of the application primarily comprises of 5 tasks – (a) Obtaining Keywords using IBM Watson Natural Language Understanding API, (b) Obtaining questions using keywords of the query, using StackAPI, (c) Graph and Edit Distance-based question ranking using NetworkX and NLTK, (d) Sentiment analysis and ranking based answer picking, using IBM Watson Natural Language Understanding API and (e) Deployment as a Flask-based Web Application. Fig. - 3.1 describes the architecture of the proposed solution.

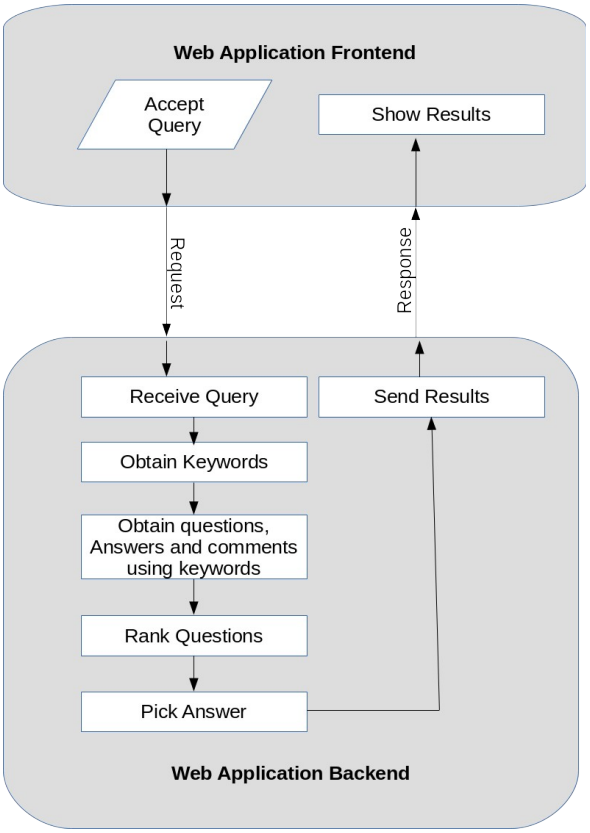

Fig. - 3.1 The architecture of the proposed solution.

4. Define the scope of work to be implemented in the project with modules etc.

The scope of work of the application will include implementation of the processes of - (a) Obtaining Keywords (b) Obtaining questions using keywords of the query, (c) Graph and Edit Distance-based question ranking, (d) Sentiment analysis and ranking based answer picking, and (e) Deployment as a Web Application.

The solution shall be able - (a)To identify most relevant questions to a query, (b) To Identify the matching tags and pick top relevant questions from stack overflow, (c)To identify top k solutions of the problem.

The solution to the problem is tightly coupled and hence, cannot be broken down into modules.