SOFTWARE REQUIREMENTS SPECIFICATIONS

SEHPAATHI- STUDENT HELP BOT

MADE BY	
NAME	SID
PIOUS ANNIE	20103049
MANTEG SINGH	20103127
SANJEEV SINGH RAWAT	20103128
NIKHILESH VARDHAN GUPTA	20103129

TABLE OF CONTENTS

1 INTRODUCTION

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyse and give an in-depth insight of the complete Sehpaathi- student help software system by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of the Sehpaathi- student help software system is provided in this document.

There are millions and billion pieces of data available, but making the right information accessible when needed is very important. Getting the right documents to read and further getting a direct answer to one's question from the set of documents is a challenging task. Even with the abundance of resources it often becomes quite difficult for students to find quality answers to questions while studying. Students often get caught up juggling between websites and are still left with their heads scratching.

1.1 PURPOSE

The purpose of the document is to collect and analyse all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

1.2 SCOPE

Primarily, the scope pertains to the student features for making Sehpaathi project live. It focuses on the project, the stakeholders and applications, which allow for a conversational chat bot and a question answering system.

The following SRS contains the detail product perspective from different stakeholders. It provides the detail product functions of Sehpaathi- student help software system with user characteristics permitted constraints, assumptions and dependencies and requirements subsets.

The software system aims to solve the problems faced by college students using an artificial intelligence enabled question answering bot that helps students find quality answers and study materials to their questions.

This SRS is also aimed at specifying requirements of software to be developed. The standard can be used to create software requirements specifications directly or can be used as a model for defining an organization or project specific standard. It does not identify any specific method, nomenclature or tool for preparing an SRS.

Future scope allows for a speech-to-text feature to be implemented in the software system. Along with image to text recognition to provide for a seamless user experience and enhance the scope of utility of the software system.

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

While the software requirement specification (SRS) document is written for a more general audience, this document is intended for individuals directly involved in the development of Sehpaathi- student help software system. This includes software developers, project consultants, and team managers. This document need not be read sequentially; users are encouraged to jump to any section they find relevant. Below is a brief overview of each part of the document.

1.4 DEFINITION AND ACRONYM

Term	Definition
Admin	Person responsible for the upkeep, configuration, and reliable operation of the system
Users	Anyone using the system
DFD	Data Flow Diagram
API	Application Programming Interface
React	JavaScript framework for frontend UI development.
Python	A programming language which will be used for any machine learning aspect of application.
JS	JavaScript is a programming language which is used to implement complex features on web pages.
Flask	Python based backend code development library used to communicate with the frontend and the database.
NLP	Natural language processing is the ability of a computer program to understand human language as it is spoken and written
HTML-CSS(Frontend)	Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS) are two of the core technologies for building Web pages
SQL	Structured Query Language that will permanently store the records in global database.

1.5 OVERVIEW

The remaining sections of this document comprises of the following 4 sections:

- Section 2 provides a general description, including characteristics of the users of this project, the functional and non-functional and data requirements of the product. General description of the project is discussed in section 2 of this document.

- Section 3 gives the functional and non-functional requirements, data requirements and constraints and assumptions made while designing the software system. It also gives the user viewpoint of product. Section 3 also gives the specific requirements of the product. Section 3 also discusses the external interface requirements and gives detailed description of functional requirements.

- Section 4 is for supporting information.

2 OVERALL DESCRIPTION

This document contains the problem statement that the current system is facing which is hampering the college students. It further contains a list of the stakeholders and users of the proposed solution. It also illustrates the needs and wants of the stakeholders that were identified in the brainstorming exercise as part of the requirements. It further lists and briefly describes the major features and a brief description of each of the proposed system.

The following SRS contains the detail product perspective from different stakeholders. It provides the detail product functions of Sehpaathi- student help software system with user characteristics permitted constraints, assumptions and dependencies and requirements subsets.

The software system aims to solve the problems faced by college students using an artificial intelligence enabled question answering bot that helps students find quality answers and study materials to their questions.

Automated Answering is the process of computing answers by computers where answering models are learned using web scraping and natural language processing It is a method of educational assessment and an application of natural language processing (NLP). A question answering system is concerned with building systems that automatically answer questions posed by humans in a natural language.

A question answering system is generally programmed to pull answers from a structured database or an unstructured collection of natural language document/s. There are two types of question answering. One is Closed-domain question-

answering system which basically deals with questions under a specific domain and the other is Open-domain question answering which is concerned with questions about nearly anything. Our solution also incorporates an image processing model that helps extracting texts from images and can further be used in our question answering model. This allows us to expand the utility of our software in our day-to-day lives.

2.1 PRODUCT PERSPECTIVE

The perspective of the software system is to allow users the ability to ask questions and get automated answers in a chat bot type format from an integrated open and closed domain answer bank scraped from the web. Thus, enabling the correct information accessible from the millions of answers available online.

2.2 PRODUCT FUNCTIONS

The functions of the software system are as follows-

- Student has to configure a username (Student ID) and password for the server and login to the application.
- Users can upload any college related query and get instant automated answers from the database.
- Users can upload corpus to the question answering tab and get the required answer from the database to the related query.

2.3 USER REQUIREMENT

The user requirements of the software system are as follows-

- User must be a verified Punjab Engineering College student registered under one of the available courses.

- System requirements-

	Windows requirements	Mac requirements	Linux requirements
Operating system	Windows 8 or later	macOS Sierra 10.12 or later	64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+
Processor	Intel Pentium 4 or later	Intel	Intel Pentium 4 or later
Memory	2 GB minimum, 4 GB recommended		
Screen resolution	1280x1024 or larger		
Application window size	1024x680 or larger		
Internet connection	Required		

2.4 PROCESS CONSTRAINTS

- System software Understand Human Context

  The AI-powered smart-bots can understand the general context, but most cases are related to the broad context hence they are unable to understand human context. Probably, the biggest drawback that conversational AI suffers is that it cannot handle complex queries or hold conversations with humans—it can handle only basic questions. Bots can respond to humans only to an extent they have been trained. If a user query is beyond what the machine has been trained for, it cannot understand or respond, which can frustrate users.

- System software Can't Make Decisions

  Another limitation is that they lack decision-making. They don't have the right know-how to differentiate between the good and the bad. Machines can neither take decisions nor help users take decisions. The lack of the ability to discern between good and bad can have serious consequences. One of the classic examples is that of a bot that Microsoft built for Twitter. Within a day, based on the content received from the users, the bot became rogue and racist, as it could not decide what was good and bad.

- System software may Have the Same Answer for a Query
  Machines are trained to provide standard answers to user queries. In a situation when a user does not find the answer satisfactory and rephrases the question, the machine still provides the same answer. This is a give-away to the users that they are in fact interacting with a machine and not a live human being. This can prove irritating and prevent users from proceeding any further. Suppose you are asking something to a bot that is not available in the data server so that you will get an apology. The same is the case with other queries; no matter how many different questions you ask, it will deliver you with the same apology, which is quite irritating. System software can also not identify human typos.

3 SPECIFIC REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

FUNCTIONAL FEATURES	REMARKS
Authentication	Authenticate users by using their SID
Create Environment	Setting up a new profile and establishing a new session uploading corpus and question answering
Answering	Send an automated response to the respective query
Homework help	Answer homework questions using uploaded corpus
Deadline alert	Alerts user of due deadlines

3.2 NON-FUNCTIONAL REQUIREMENTS

3.2.1 PERFORMANCE REQUIREMENTS

The product shall be based on web and has to be run from a web server. The product shall take initial load time depending on internet connection strength which also depends on the media from which the product is run. The performance shall depend upon hardware components of the client/user.

3.2.2 SECURITY REQUIREMENTS

- DATA TRANSFER

  The system shall use secure sockets in all transactions that include any confidential user information. The system shall automatically log out all users after a period of inactivity. The system shall not leave any cookies on the user's computer containing the user's password. The system shall not leave any cookies on the user's computer containing any of the user's confidential information.

- DATA STORAGE

  The user's web browser shall never display a user's password. It shall always be echoed with special characters representing typed characters. The system's back-end servers shall never display a user's password. The user's password may be reset but never shown. The system's back-end servers shall only be accessible to authenticated administrators. The system's back-end databases shall be encrypted.

3.2.3 USABILITY REQUIREMENTS

- GRAPHICAL USER INTERFACE

  The system shall provide a uniform look and feel between all the web pages. The system shall provide a digital image for each product in the product catalogue. The system shall provide use of icons and toolbars.

- ACCESSIBILITY

  The system shall provide handicap access. The system shall provide multi language support.

3.2.4 INTERFACE REQUIREMENTS

- The protocol used shall be HTTP.
- The Port number used will be 80.
- There shall be logical address of the system in IPv4 format.

- USER INTERFACES

The user interface for the software shall be compatible to any browser such as Internet Explorer, Mozilla or Netscape Navigator by which user can access to the system. The user interface shall be implemented using any tool or software package like Java Applet, MS Front Page, EJB etc.

3.3 CONSTRAINTS

- Flask (A python runtime environment)
- SQL (Database system)
- React.js (JavaScript library for frontend development)
- HTML (Technology for building web pages)