Chatbot Web app

Software Requirements Specification

Version 1

10th November 2021

Jerish Bovas

Sanjay Kannan

Naveen Jose

Jesse Hughes

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

3. Specific Requirements 2

3.1 External Interface Requirements 3

3.1.1 User Interfaces 3

3.1.2 Hardware Interfaces 3

3.1.3 Software Interfaces 3

3.1.4 Communications Interfaces 3

3.2 Functional Requirements 3

3.2.1 <Functional Requirement or Feature #1> 3

3.2.2 <Functional Requirement or Feature #2> 3

3.3 Use Cases 3

3.3.1 Use Case #1 3

3.3.2 Use Case #2 3

3.4 Classes / Objects 3

3.4.1 <Class / Object #1> 3

3.4.2 <Class / Object #2> 3

3.5 Non-Functional Requirements 4

3.5.1 Performance 4

3.5.2 Reliability 4

3.5.3 Availability 4

3.5.4 Security 4

3.5.5 Maintainability 4

3.5.6 Portability 4

5. Change Management Process 5

# Introduction

The project is to create AI/ML Chatbot for the company named IT Clan Consulting services. The Chatbot will be acting as the first line of communication with the users.

## Purpose

The purpose of this document is to present sufficient information about software requirements for the successful completion of the project.

## Scope

1. Chatbot web app.
2. The project is to create a Chatbot with the ability of AI and ML.
3. The main objective of the Application is to reduce the wait time of the customers waiting for customer service and increase customer satisfaction.

## Definitions, Acronyms, and Abbreviations

AI – Artificial Intelligence.

ML – Machine Learning.

Chatbot - Chatbots are designed to simulate how a human would behave in a conversation. These pre-programmed automated interfaces communicate on text-based channels, such as Facebook Messenger, Skype, Slack, WeChat, and more. The bots are programmed to pull replies that match the greatest number of keywords or the most similar wording patterns based on what has been composed by the customer. This creates a quick, but a generic response to a customer’s questions.

## References

<http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf>

[www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc](http://www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc)

## Overview

The rest of the document will contain the General description, Specific requirements and the external Interface requirements and Analysis models.

# General Description

The efficiency of the application largely depends on the quantity of the information available on the previous questions and the enquiries the customers had. And the proper replays the Customer service representatives provided them with (If it is right)

## 2.1 Product Perspective

The perspective of this product is to provide instant service to the customers with accurate answers to their queries using Machine Learning and AI

## 2.2 Product Functions

Analyze the cloud data and get relevant answers for the customer’s queries.

Offer some pickup questions for the customers.

## 2.3 User Characteristics

Product interface themes are affected by users' preferences.

## 2.4 General Constraints

The design of the system must be easier to use. Must provide accurate answers and reduce wait time for the response.

## 2.5 Assumptions and Dependencies

If the chatbot server is not available, it should try to connect the customer with the customer service agent or provide an error message with contact info.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

1. Front-end software: React

2. Back-end software: SQL

### Hardware Interfaces

A browser that supports CGI, HTML & Javascript.

### Software Interfaces

|  |  |
| --- | --- |
| 1. **Software used** | **Description** |
| Browser | Our application runs in a browser. |
| Database | To save the response answers, we are using a database. |
| React | To implement the project, we have chosen React as our frontend framework. |

### 3.1.4 Communications Interfaces

This project supports all types of web browsers. We are using Single Page Application which is implemented by React.

## 3.2 Functional Requirements

### 3.2.1 Feature: Make questions from words

3.2.1.1 Introduction

The system analyses the customer’s input of jumbled words and forms a fully constructed question to accurately answer users.

3.2.1.2 Inputs

The input will be from users, which will be in the form of some keywords from the questions that they intended to ask

3.2.1.3 Processing

We configure the system using the existing libraries to analyze the input and match them to possible questions.

3.2.1.4 Outputs

Find answers to the processed question and give it back to user.

3.2.1.5 Error Handling

In case of error, ask the user to try different keyword, saying, the provided keyword doesn’t match the database.

### 3.2.2 Function: Greet User with current weather

3.2.2.1 Introduction

Out application will greet the user with the current day, time and weather and provide customer a pleasant experience

3.2.2.2 Inputs

Customers open the chatbot web app

3.2.2.3 Processing

Application processes the system’s current time, day and get weather data from API services

3.2.2.4 Outputs

Show the result in users application interface

3.2.2.5 Error Handling

In case of error or any insufficient information, it shows the default greeting message.

## 3.3 Use Cases

### 3.3.1 Use Case #1

Customer interacting with the system.

A screenshot of a computer

Description automatically generated with low confidence

### 3.3.2 Use Case #1

Chatbot response to the user

A picture containing text, compact disk, businesscard

Description automatically generated

## 3.4 Classes / Objects

### 3.4.1 User

3.4.1.1 Attributes

Firstname

Lastname

Address

Email

Phone

3.4.1.2 Functions

isActive

sendMessage

changeInfo

toString

## 3.5 Non-Functional Requirements

The steps involved to perform the implementation of ML, AI and backend database

### 3.5.1 Performance

The performance of the system depends on the backend server, database and the user’s internet and it does not depend on the system.

### 3.5.2 Reliability

The reliability of the system is maintained by using cloud computing, thus providing a reliable service

### 3.5.3 Availability

By using cloud computing model, we are aiming to achieve full available system.

### 3.5.4 Security

Our application is secured by using the security tools available with the webserver hosting services

### 3.5.5 Maintainability

Our system needs to be maintained in a frequent basis, based on customer needs

### 3.5.6 Portability

This system works on all operating systems, given a basic updated web browser.

# 4. Change Management Process

In case of changes or new additions to the SRS, it can be proposed by anyone involved in the system, but the approval will be done by the team lead.