BINFO INFO BOT

Software Requirements Specification

v0.3

March 22, 2023

Fardous Sadeq
Pol Kihn

Prepared for
Software Engineering Project – BINFO, 4th semester

Revision History

Date	Description	Author	Comments
15.03.23	v0.1	Pol, Fardous	Initial Draft
21.03.23	v0.2	Pol, Fardous	Functional Requirements
22.03.23	v0.3	Pol, Fardous	Non-Functional Requirements

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date

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 OVERVIEW 1

2. GENERAL DESCRIPTION **1**

2.1 PRODUCT PERSPECTIVE 1
2.2 PRODUCT FUNCTIONS 1
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 2
3.1.1 User Interfaces *2*
3.2 FUNCTIONAL REQUIREMENTS 2
3.2.1 Natural Language Processing in English using machine learning *2*
3.2.2 Access to information on Uni.lu website *3*
3.2.3 Manage multiple simultaneous users *4*
3.3 NON-FUNCTIONAL REQUIREMENTS 4
3.3.1 Performance *4*
3.3.2 Reliability *4*
3.3.3 Availability *5*
3.3.4 Security *5*
3.3.5 Maintainability *5*
3.3.6 Portability *5*
3.4 INVERSE REQUIREMENTS 5

1. Introduction

1.1 Purpose

This document is intended for both students, as developers, and the teacher, as the client, to define the scope and features of the Software Engineering Project and document any changes to these during the sprints.

1.2 Scope

The scope of the project comprises a server-side, AI-driven chatbot written in Python, which will answer questions about the BINFO study program, to promote said program to prospective students. The data on which these answers are based will be solely derived from the uni.lu website, on which the bot is then trained.
Any conversation outside of the topic of BINFO are out of scope.

1.3 Definitions, Acronyms, and Abbreviations

AI	–	Artificial Intelligence
BINFO	–	Bachelor in applied INFOrmation technology
NLP	–	Natural Language Processing

1.4 Overview

Part 2 of the SRS addresses the requirements from a client's point of view, while point 3 will elaborate on, and guide the actual developing process, providing concrete quality criteria by which the end-product should be judged.

2. General Description

2.1 Product Perspective

The deliverable will be a specialized, standalone chatbot, without a dedicated user interface. It is intended to run on a server, receiving and sending in- and output over real or simulated network connections.

2.2 Product Functions

The chatbot will receive text-based input, containing questions about BINFO, to which it will output a comprehensible, factually correct text-based reply.

2.3 User Characteristics

The intended users are prospective students, or any other person interested in facts about BINFO, who does not want to spend a lot of time collecting all the necessary information from the different parts of the uni.lu website.

2.4 General Constraints

The limiting factors of the project are the hard, non-negotiable deadline, as well as the developers' relative inexperience with Python, chatbot design principles, and server-side programming requirements.

2.5 Assumptions and Dependencies

The project assumes the availability of Natural Language Processing libraries, to be used with the AI driven aspect of the chatbot. Failing this, the chatbot will have to revert to a rules-based structure.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

There will be no dedicated UI. All input and output will be handled via console.

3.2 Functional Requirements

This set of functional requirements will enumerate all the features of the chatbot, and describe the way they're intended to work.

3.2.1 Natural Language Processing in English using machine learning

3.2.1.1 Introduction

Since the chatbot will be designed to interact with humans in a free form conversation expected to be in English, as opposed to a multiple choice, tree form structure. It therefore needs to have systems in place to both interpret natural language input, as well as structure the response output in a grammatically and syntactically meaningful way. This process will use machine learning to gradually improve the quality of the generated answers.

3.2.1.2 Inputs

The input will be in the form of a string of characters entered by the user, that we assume to be in plain English. We will consider this input to be non-malicious and sanitized for security purposes, as this would be handled by the mobile messaging interface, whose development is out of scope for this project.

3.2.1.3 Processing
The input will be processed into word-stem tokens, which will then be passed to a neural network for interpretation using a sequence-to-sequence transformation to generate an output. The neural network will need access to a database of English conversations to be trained on for general conversation, for this transformation to be possible.

3.2.1.4 Outputs
After scoring a multitude of possible outputs against a predefined list of quality criteria, the bot will randomly pick from among the outputs tied for highest score and pass this as a response to the user.

3.2.1.5 Error Handling
Error handling will try to catch any formatting issues during the processing of the user input for the neural network.

3.2.2 Access to information on Uni.lu website

3.2.2.1 Introduction
The chatbot will need access to the information presented on the uni.lu website about the BINFO study program and any information relevant to undergrad students in the BINFO program, in order to accurately relay this information to the user upon request.

3.2.2.2 Inputs
The content of the website will need to be collected, so that the bot will be able to access it. This can be done by manually, or possibly automatically scraping the website. If the process can be automated, the information accuracy will remain high across changes to the program or to university regulations over the semesters, as well as the changing academic calendar each year.

3.2.2.3 Processing
The data is then processed, categorized and stored in a file format accessible to the bot, so that the neural net may be trained on this data.

3.2.2.4 Outputs
The output is the database to be loaded by the bot to help it formulate a response to a query by a user.

3.2.2.5 Error Handling
The error handling will try to catch badly formatted datapoints during processing to maintain the integrity and readability of the database.

3.2.3 Manage multiple simultaneous users

3.2.3.1 Introduction

This bot is intended to interaction with users over a network connection. This means that users do not have their own instance of the bot running on their device, but rather connect to a server running the bot and pass the queries to the bot over these connections. They would then receive an answer via that same connection. This implies that more than one user could interact with the bot at the same time, and the bot must be designed in a way to handle this.

3.2.3.2 Inputs

The user inputs will be sent over a network connection. The bot must distinguish between the different users and keep their inputs separated.

3.2.3.3 Processing

While generating a response the bot must not conflate the different user inputs or have a response be influenced by a differing input. The response must be user specific. There is however a no need to retain a persistent record of user interactions, as every query should be independent.

3.2.3.4 Outputs

The generated response should be sent to the respective user only. To this end the bot must retain which user sent which input and match the response accordingly.

3.3 Non-Functional Requirements

These non-functional requirements will provide a set of quality criteria to judge the performance of the program at deployment.

3.3.1 Performance

Since the scope of the conversation with the bot is limited to mostly short question and answer type exchanges, 90% of the responses should be generated within 1 second, with no response exceeding 5 seconds. Should the question be out of scope, as by our definition earlier in this document, we will not consider the performance of the bot relevant.

3.3.2 Reliability

While the mean time between failures of the program is not as interesting as a metric for this project at deployment, the accuracy of the bot's statements is. We aim for, at the very least, an accuracy of 85% at deployment, meaning that only three out of twenty responses may contain false information to any degree. This does not consider the relevance of the answer. Should the bot provide a non sequitur, the user may ask again. Should the bot lie, however, the user might be satisfied with this wrong answer, which we consider a catastrophic failure of the system. Should the accuracy of the bot be below 85%, we cannot consider the bot viable for providing information.

3.3.3 Availability

The chatbot should be able to handle to handle 5 simultaneous users. As this bot is intended to be deployed on a server and interact with users over networked mobile devices, a minimum of 5 simultaneous users seems appropriate, and could be scaled up in further iteration. Since the development of the mobile interface is out of scope, the connections will have to be simulated.

3.3.4 Security

Security will not be a concern for this project, as both input sanitation, and guarding against denial-of-service attacks are out of scope. The bot and its adjoining databases will not contain any confidential information, which would need to be secured.

3.3.5 Maintainability

The database containing information about the BINFO program, and the university itself, must be able to be updated or replaced to maintain accuracy in the bot's responses. Whether there is an automated process in place to do this remains optional.

3.3.6 Portability

Since we are going with an interpreted programming language, Python, the bot should be able to run on any machine capable of running a Python interpreter.

3.4 Inverse Requirements

While not undesirable, we do not need the chatbot to be able to carry a meaningful conversation over multiple exchanges.