Software Requirements Specification

for

Alice Academic Chat-Bot

Version 1.0

Prepared by

Group Name: ***<place your group name here>***

Ujjwal Arora	**140911330**	**arora.ujjwal@gmail.com**
Saurav Prakash Vasudev	**140911442**	**saurav1608@gmail.com**
Vikalp Shukla	**140911374**	**shukla.vikalp4@gmail.com**

Instructor:	**Mr. Rajesh K.**
Course:	**ICT 2213 – Software Engineering Lab**
Lab Section:	***B2***
Date:	**March 19, 2016**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Ujjwal Arora	First Draft	17/03/16

1 Introduction

<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>

1.1 Document Purpose

This document includes software requirements for Alice Academic Chatbot, release number 1.0. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2 Product Scope

Alice is an artificially intelligent chat bot which the students and faculty can text just for fun or in order to get various kind of information regarding their college life.

The proposed chat bot will use various algorithms to understand the user's message, which is typed in natural language, and return back the reply. There is no specific format to talk and the replies will be as if they are answered by a real person. Alice will also be familiar with the user's details and will be able to provide information pertaining to that student, such as his/her class schedule and volunteer duties. Alice will also be able to update the student and the staff with ongoing cultural activities, as well as just make conversation, as well as answer regular questions with the help of online search engine APIs. Hence, this bot will be greatly beneficial to both students and the faculty, saving both time and effort.

1.3 Intended Audience and Document Overview

This document is intended for:

Developers: in order to be sure they are developing the right project that fulfills requirements provided in this document.

Testers: in order to have an exact list of the features and functions that have to respond according to requirements and provided diagrams.

Users: in order to get familiar with the idea of the project and suggest other features that would make it even more functional. The users are the students and teachers.

Documentation writers: to know what features and in what way they have to explain. What security technologies are required, how the system will response in each user's action etc.

Advanced end users and system administrators: in order to know exactly what they have to expect from the system, right inputs and outputs and response in error situations.

The rest of this document is divided into chapters for better understanding.

Chapter 2: an overall description of the system is provided. First, product perspective is presented with product features and main functions. Then follow user classes and characteristics, operating environments that the system supports as well as design and implementation constraints. After all that user documentation is presented and will provide more details about each feature's technology.

Chapter 3: most important features are presented with detailed description, use cases and requirements, along with user, software, and communication interfaces.

Chapter 4: requirements about safety and performance are presented, as well as software quality attributes.

1.4 Definitions, Acronyms and Abbreviations

Definitions:

MD5	The **MD5 message-digest algorithm** is a widely used cryptographic hash function producing a 128-bit hash value, typically expressed in text format as a 32-digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic application.
User	Someone who interacts with the system.

Acronyms:

AACB	Alice Academic Chat-Bot
API	Application Program Internface
GUI	Graphic User Interface
Webapp	Web Application

2 Overall Description

2.1 Product Perspective

The system described is a new, self contained product. AACB consists of an internal “database”, with help of which all the query questions are answered. Every user will be provided with a password beforehand which they will use to gain access to the system.

In the diagram below there are the main components of the system, subsystem interconnections and external interfaces to help you understand the main idea of AACB. All of them are analyzed with more details in this document.

2.2 Product Functionality

- Respond to users’ queries pertaining to him/her with high accuracy.
- Ability to casually chat to the user and learn from the user’s replies.
- Authentication of the user.
- User is able to edit his/her details and change his password.
- Students and Teachers are able to ask the bot for their class schedule, calendar, etc.
- The system can use the web to answer general and mathematical queries.
- Administrators can add new information to the system.
- The users can submit bug reports which the developers can view.
- Users can change settings about the software.
- Users can view basic information about themselves as soon as they login.

2.3 Users and Characteristics

Casual Users: users which will use the system for entertainment purposes.
Students: users who will use the system for academic information related to students.
Teachers: users who will use the system for academic information related to teachers.
Advanced Users: who will employ extensively the artificial intelligence of the system.
System Administrators: administrators will modify the content available on the platform.
Developers: who will be able to view the bug reports submitted by the other users.

Most important users are the students and teachers, while the less important to satisfy are the casual users.

2.4 Operating Environment

The software is operating system independent since it is written in java, and hence will run on all major operating systems including Windows, Linux and Mac OSX.

All new release contains

Filename	Architecture	Type

AACB-x.x-Setup.exe	X86	.exe (32-bit Windows Executable)
AACB-x.x-Src.zip	Platform Independent	.zip (Source)
N/A	N/A	Webapp

Minimum requirements: Intel Core i3 processor, 2GB ram, 100 MB free HDD space

2.5 Design and Implementation Constraints

Timing Requirements: The system should be built such that it is always up and running, even during upgradation and maintenance.
Language Requirements: The system as well as all the documents are in English only.
Memory Requirements: For extensive pattern matching, some hard disk space is needed.
Specific Technologies: AIML is to be used for artificial intelligence.
Security Considerations: For authentication of users, passwords are to be stored as md5 hashes.

2.6 User Documentation

The software ships with a manual containing basic phrases to get started with for regular users, and extensive description of queries and learning for advanced users.

2.7 Assumptions and Dependencies

- No of users simultaneously using the system are only in hundreds.
- The devices used by users are powerful enough to handle the software application.
- The users have access to the internet.
- The system should be online 24/7
- Users are familiar with basic knowledge of computer systems and chatting.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

One of the key challenges for AACB is to design a GUI which is user-friendly, avoids duplication of functionality and gives a consistent, integrated view of the system.

The GUI will be built on java swing. The screens consist of, but not limited to, an authentication screen, a screen where the user can chat with the bot, a screen with basic user information, an account management screen, and a settings screen, a screen for administrator to add new information, a bug report screen, and an About Us screen.

3.1.2 Software Interfaces

- Java Virtual Machine version 8
- AIML version 2.0

3.1.3 Communications Interfaces

The communication between the different parts and users of the system is important since they depend on each other. This communication is achieved by deploying the system as a webapp which users can use simultaneously.

The passwords used for authentication are encrypted as MD5 hashes.

3.2 Functional Requirements

1. **Authentication**
    - 1.1. **Digest password to MD5**
    - 1.2. **Verify password**
2. **View basic user details**
3. **Edit user details**
    - 3.1. **Change password**
    - 3.2. **Change name**
    - 3.3. **Change section**
4. **Chat to System**
    - 4.1. **Query about academic and events**
    - 4.2. **Query which uses the web**
    - 4.3. **Casual chatting**
    - 4.4. **Learn from users' replies**
5. **Bug report**
    - 5.1. **Submit bug report**
    - 5.2. **View bug report**
6. **Add new data to system**
7. **Change software settings**
    - 7.1. **Change Theme**
    - 7.2. **Change Font**

4 Other Non-functional Requirements

4.1 Performance Requirements

1. There should never be a delay of more than 4 seconds between the user’s message and the system’s reply.
2. High accuracy of information would be maintained at all costs.
3. The system is expected to be online 24/7 as it is relays information vital to the users.
4. The software should be stable and not susceptible to crashes.
5. In case of not being able to reach the Internet, the system must be able fall back on the backup after a timeout of 2 seconds.

4.2 Safety and Security Requirements

1. The user can only use the system if he has a username and password has agreed to the underlying terms and conditions.
2. The user’s password and details will always remain confidential and will never be conveyed to a third party. Password hashing would be used to ensure safety of the passwords.
3. Any information provided by the user shall not be used against him/her in any way.
4. The Chat-Bot will never reply in a manner which will demean the user.
5. All activities carried out by the system would be in accordance with the Information Technology Act, 2000.

4.3 Software Quality Attributes

4.3.1 Correctness

The information provided by Chat-Bot should be in accordance with the actual data (dates, schedules etc.) maintained by the University and should not mislead the user in any way. This is achieved by regularly updating the database.

4.3.2 Availability

AACB should be available as and when needed by the users as many users may depend on the system to view their schedule.

4.3.3 Portability

The system should be portable and hence the same software should run in different environments. This is achieved as the software is written in java which is platform independent.

4.3.4 Responsiveness

Delay in the input output loop may be a major cause of frustration among users and must be reduced to a minimum level. This is done by optimizing the process to eliminate wasteful output from algorithm and enabling the software to regularly learn the correctness of the output by a rating system. Keeping track of the frequently used keywords may also help in reducing the latency.

4.3.4 Understandability

AACB shall understand the natural language and respond in the natural language without any spelling errors or grammatical errors.

Appendix A - Group Log

23rd February 2016: Met at the library to decide the project and create the synopsis.

8th March 2016: Discussed the project details.

15th March 2016: Started writing the SRS

***** End of the SRS *****