Syracuse Chatbot

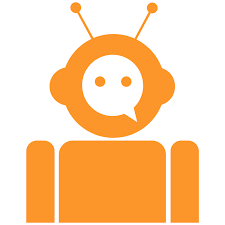
System Requirements Specification

**Team:**

**Bharath Karumudi <bhkarumu@syr.edu>**

**Haixin Chang <hchang04@syr.edu>**

**Version: 0.4**



Syracuse

Chatbot

Table of Contents

1. Revision History 3

2. Introduction 3

2.1 Purpose 3

2.2 Project Scope 3

2.3 Overview of Document 4

3. Description 4

3.1 Product Perspective 4

3.2 Product Functions 4

3.3 Constraints 5

3.4 System Architecture 5

3.5 System Models 7

4. Requirements 9

4.1 Functional User Requirements 9

4.2 Functional System requirements 9

4.3 Non-functional requirements 10

5. System evolution 11

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Comment** | **Version** |
| Bharath Karumudi | 02/11/2019 | Document Structure | 0.1 |
| Haixin, Bharath | 02/13/2019 | Added content to sections 1 to 4. | 0.2 |
| Haixin, Bharath | 02/17/2019 | Added use case and sequence diagrams | 0.3 |
| Haixin, Bharath | 03/21/2019 | Added section 5 and modified section 3 | 0.4 |

# Introduction

## 2.1 Purpose

This document will provide all the requirements for the project Syracuse Chatbot. It will serve as a reference for both developers and customers during the development of the final version of the system.

## 2.2 Project Scope

Syracuse Chatbot is an AI Chatbot that receives questions from users, tries to understand the question, and provides appropriate answers. The application does this by converting an English sentence into a machine-friendly query, then going through relevant data to find the necessary information, and finally returning the answer in a natural language sentence. In other words, it answers your questions like a human. For example, when it receives the question "What is my term fee balance?", it will give a response “You owe $3500.”

The main objective is creating a Web based API, and sample web, mobile interfaces (through Slack and Facebook messengers) that demonstrate the use of the API.

The goal is to provide Syracuse students a quick and easy way to have their questions answered.

## 2.3 Overview of Document

1. Revision History: Provides the date of, reason for, and people who were involved with the modification of this document.  
2. Introduction: Provides an overview of the application, explain the objectives and goal of the project and describe the document structure.   
3. Description: Provides the specification of the system model, the classes model and the main constraints.  
4. Requirements: Provide the analysis of the requirements by feature and also provides some other constraints that affect performance, safety and security.  
5. Glossary: Definitions of terms used.

# Description

## 3.1 Product Perspective

Chatbot is a web-hosted application, developed based on the current bot technology. This application acts an intermediate media between users and databases. A user can interact with Chatbot via simple English sentences to request and update information from certain databases. These English sentences are analyzed by a Language Understanding Intelligent Service (LUIS) which is integrated with the Chatbot.

There will be four main units to the system working together to understand the question and return an appropriate answer:

**Generic question construction** - capable of taking a natural language question and making it more generic.   
**Generic answer construction** - capable of taking a generic question template and providing a generic answer template.   
**Generic answer population** - capable of taking a generic answer template and populating it with information from the database to form an answer.   
**Information extraction** - capable of finding information available from the database.

## 3.2 Product Functions

Syracuse Chatbot shall be able to query on:

* Student profile information
* Classes and schedules
* Payment information

## 3.3 Constraints

* + 1. **Limited Question Scope**

Creating a Chatbot able to answer every single question about Syracuse is not possible to implement within the duration of the project, so the system will be able to answer questions about limited topics.

* + 1. **Language**

The system will only support questions in standard English.

## 3.4 System Architecture

**3.4.1 System Architecture Diagram**

****

**3.4.2** Reused application systems

**3.4.2.1** Language Understanding Intelligent Service (LUIS): provide by Azure Microsoft

**3.4.2.2** Facebook messenger

**3.4.2.3** Slack

**3.4.2.4** SQL database: provided by Azure Microsoft

**3.4.2.5** Reused application framework: Microsoft Bot Framework provided by Azure Bot service

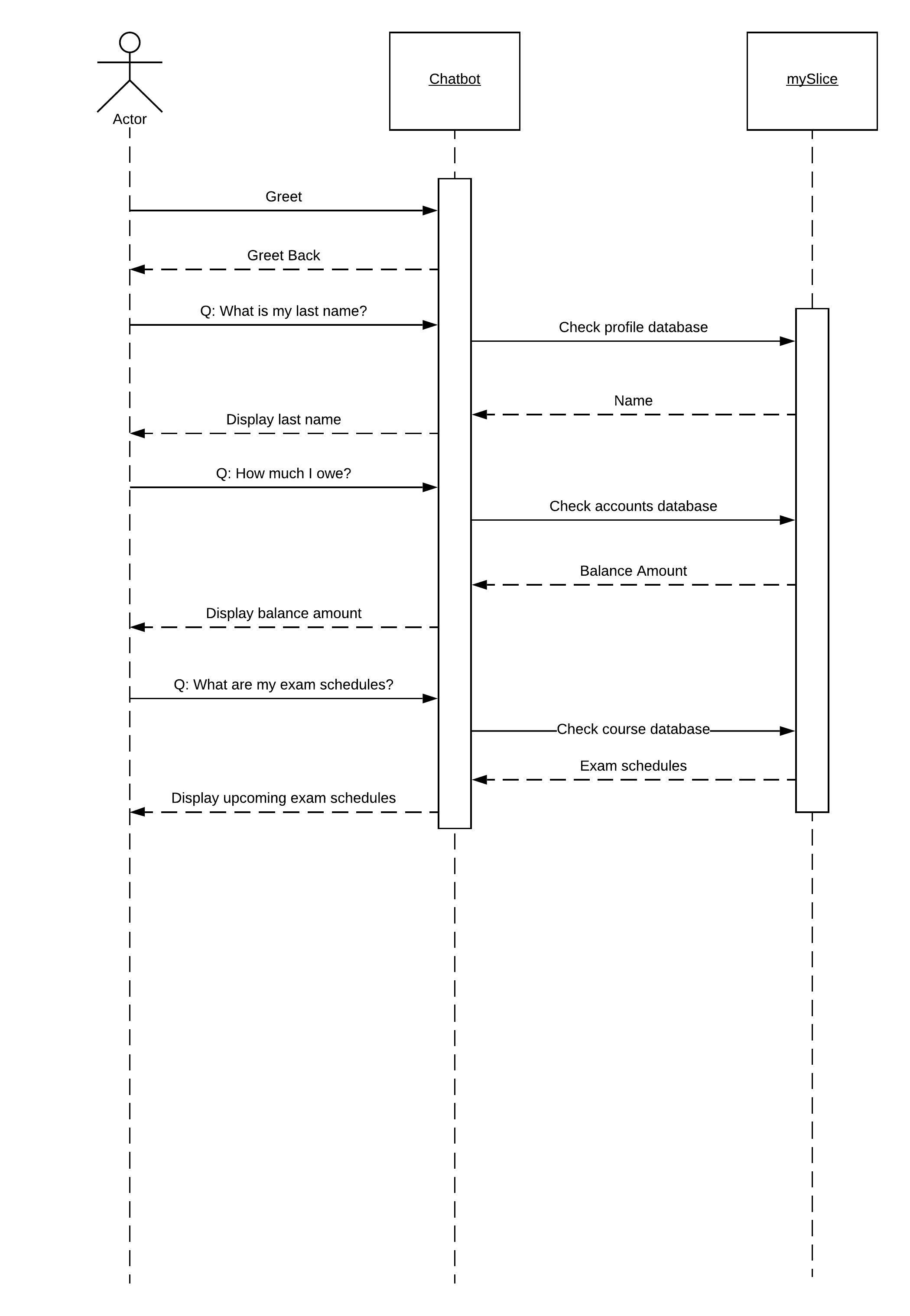
## 3.5 System Models

**3.5.1 Use case diagram**

A picture containing text

Description automatically generated

**3.5.2 Sequence diagram**

****

# Requirements

## 4.1 Functional User Requirements

1. A student shall be able to check his/her profile details such as Name, Email, phone number, address on the records.
2. A student shall be able to update his/her phone number and Address details.
3. A student shall be able to check for the availability of classes.
4. A student shall be able to enroll for the classes.
5. A student shall be able to check term fee and balance.
6. A student shall be able to check his/her class schedule for current date.
7. A student be able to check his/her exam schedules.
8. The user shall be able to use Chatbot from Web, Facebook messenger and Slack.

## 4.2 Functional System requirements

1. The bot shall be able to understand the question and provide the appropriate answers.
2. When a student greets the bot, the bot shall greet the user back and should display a fun fact, if there is any available, about Syracuse University.
3. When a student asks the bot to show his/her records, it shall pull only his/her first name, last name, email, address and phone.
4. When a student asks to update either address or phone, the bot shall able to take the input from the student and update the MySlice database and shall provide a confirmation to the student.
5. When a student asks the bot to enroll for a course, the bot shall first check the seat availability for the course. If available, the student shall be enrolled; if no seats are available, it shall show no seats are available message.
6. When a student asks the bot to provide the term fee details, it shall check the MySlice database and shall display the balance due. If nothing is due, it shall display No payment is required.
7. When a student asks for his/her class schedules, the bot shall check all the enrolled courses and return their class schedules for next seven days.
8. When a student asks for the upcoming exam schedules, the bot shall check all the enrolled courses and return their upcoming exam schedules.
9. The bot should be integrated with Slack and Facebook messenger.
10. A standalone web-based application shall be available.

## 4.3 Non-functional requirements

1. The Chatbot shall be able to respond to users’ request within one second.
2. The Chatbot shall be easy to use. Normally a user should be able to learn to use within 30 minutes.
3. The system shall support questions in standard English.
4. The bot shall able to handle concurrent users with no limit.
5. The return answers should be in an English language sentence.   
   For example, when the bot receives the question "What is my term fee balance?", it should give a response “You owe $3500 and the due date is March 14, 2019.”
6. The application shall be developed in node.js.
7. The application shall be hosted in Azure cloud environment.
8. The bot should be available to the users all the time. Scheduled maintenance is acceptable.
9. A code base shall be maintained in GitHub.
10. The development should use Continuous Integration and Continuous deployment.

# System evolution

Bot technology has been widely used in many aspects of people’s lives. Along with the development of Artificial Intelligence (AI) technology, we expect the bot services will be more prevalent and provide more functions. Specifically our bot would be able to connect more databases to carry more functionality and handle more complicated tasks. For example, our bot could place an order for the user such as a textbook, based on the course enrollment. A voice recognition may be added so user can directly talk to the bot.