

# An Artificial Intelligence Based Virtual Assistant Using Conversational Agents

Mehdi Mekni

Department of Electrical and Computer Engineering and Computer Science (ECECS), Tagliatela College of Engineering (TCOE), Connecticut Institute of Technology (CIT), University of New Haven, New Haven, Connecticut, USA

Email: mmekni@newhaven.edu

**How to cite this paper:** Mekni, M. (2021) An Artificial Intelligence Based Virtual Assistant Using Conversational Agents. *Journal of Software Engineering and Applications*, 14, 455-473.

<https://doi.org/10.4236/jsea.2021.149027>

**Received:** January 22, 2021

**Accepted:** August 30, 2021

**Published:** September 2, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

Conversational agents are natural language interaction interfaces designed to simulate human conversations using Artificial Intelligence (AI). This paper explores current applications of these systems and raises the lack of their availability in education. To address this problem, we provide the design of a conversational agent system, which is efficient and time-saving in assisting student/college seeking information about curriculum, scheduling, teachers, classroom location at any time 24/7/365. To verify and validate the design and implementation of our proposed model, a pilot project has been set up involving three leading academic institutions. This platform is designed and developed to help universities provide continuous and instant assistance to their student, staff, and faculty communities.

## Keywords

Conversational Agent, Software Engineering, Virtual Assistant

## 1. Introduction

Customer experience plays a vital role in any organization or institution to grow and serve its users. Organizations now require round-the-clock service offerings to their users, keeping them informed regarding their services. One potential solution to this kind of offering is a chatbot. Chatbots can never replace human interactions, but they can reduce workloads drastically [1].

According to the education literacy report by United Nations International Children's Emergency Fund (UNICEF) [2], globally youth literacy rate has improved and it has increased from 83% to 91% [3]. There are currently 4.66 billion active internet users worldwide. 65.6% of the entire world's population has internet access and 81% of US adults go online on a daily basis [4]. This has also

given rise and access to online education.

College students must recurrently navigate a set of challenging tasks, such as building a graduation plan, learning about majors, retrieving valuable information about courses including the number of credits, students learning outcomes, sections and associated schedules, assigned professors, classroom locations among others. Without assistance and support, many students fail to raise these challenges. The lack of assistance and support causes an estimated 10% to 20% of college drop-out each year, with higher rates among low-income and first-generation college students [5] [6]. Common efforts to address the lack of assistance and support have supported students with additional individual counselor outreach [5] or through automated, customized text message-based outreach [7]. However, scaling these efforts requires significant resources because of the time needed for counsellors to address the specific questions and personal needs of each student [8].

With the global growth and acceptance of digital technology, our daily life is changing and replacing things around us quickly, similarly, conversational agents also called *chatbots* are revolutionizing businesses. As the NLP (Natural Language Processing) and AI (Artificial Intelligence) techniques are easily available and the concepts like conversational agents are now a reality, many small and medium-sized organizations are showing wider acceptability and implementation. The users are also preferring chatbots due to round-the-clock availability, reliability, accessibility, and instantaneous accurate response. As a consequence, world-class leading Information Technology & Communication (ITC) companies have launched their own conversational agents' system/chatbot time to time such as Google, Microsoft, Facebook, Amazon, IBM to name a few.

In this paper, we propose a smart virtual assistant using conversational agents technology to improve and enhance student assistance and support in academic institutions. The project has two objectives. The immediate objective is to free up resources from the mundane tasks that involve the regular updating/maintaining of digital resources and create a streamlined process of identifying pain points of students, FAQs, and feedback. The longstanding objective is to explore integration of an empowered version of the chatbot to become a part of a learning management system improving the student learning experience and playing the role of a virtual tutor.

Our project will be verified and validated by a group of candidate universities and colleges members of the Minnesota State Colleges and Universities (MnSCU). MnSCU also called Minnesota State System comprises 30 state colleges and 7 state universities with 54 campuses throughout the state of Minnesota. The system is the largest higher education system in Minnesota and the fourth largest in the United States, educating over 375,000 students annually. Our project envisions an incremental deployment approach. The first phase of our project involves a selection of three leading academic institutions in Minnesota; Century College; Metro State University; and St. Cloud State University. This pilot will help assess the project feasibility, manage risks, and confirm the expected out-

comes.

The remainder of this paper is organized as follows. Section 2 presents a comprehensive overview of conversational agents and the motivation behind their use. Section 3 details the Minnesota State Chatbot system. It presents the Software requirements analysis, design and architecture of the AI-Based virtual assistant using conversational agents. Section 4 discusses the various used technical choices and associated methodologies and concludes with future work.

## 2. Related Work

In this section, we first provide an overview of chatbots and the evolution of this concept over the years. Next, we detail our motivations to use chatbots to address the issue of supporting and assisting students in non-academic matters, including orientation, login information, regular check-ins, course registration, communicating graduation requirements and applying for graduation.

### 2.1. Overview on Conversational Agents

The term “Conversation Agent” has come to mean a wide variety of systems with varying capabilities and purposes, with the underlying assumption that the agent participates in a human-machine dialog. Licklider’s “Man-machine symbiosis” [9] was one of the earliest discourses from a Human Computer Interaction (HCI) perspective that visualized humans interacting with machines in a natural manner. Research in conversation agents started with messaging-based chatbots, whose purpose was to maintain a conversation with a human user.

In 1990, the Loebner Prize was instituted as an annual competition to award the most human-like chatbot. The first chatbot emerged in 1966 from MIT, called Eliza [10], which emulated a Rogerian psychotherapist. Eliza worked on simple declarative rules: If a certain keyword was identified in the user text, it responded with one or more predefined outputs. Subsequently, in the latter chatbots, the rules used for both natural language understanding and natural language generation were enriched. Ontologies were used to represent word meanings, reasoning was used to identify user intent, and memory was used to continue a contextual dialog [11] [12] [13]. The notable follow-up chatbots included MegaHAL [14], Alice [15], and Elizabeth [16]. Recent examples from the Loebner winners are Mitsuku and Rose. Popular chatbots that have recently emerged from the industry are Xiaoice, Tay and Zo from Microsoft [17].

In the last decade, conversational agents started focusing more on utility, with the goal of accomplishing specific tasks. Nowadays, conversational agents range across several modalities, including speech (such as Siri, Alexa, Cortana), text-messaging (such as Domino’s, CNN, Pandorabots, Burberry, etc. found on Messenger, Slack, and/or Skype platform), and as multimodal embodied agents. Embodied CAs have a graphical front-end as opposed to a robotic body, and attempt to be human-like by employing non-verbal behaviors, such as gestures and expressions, in addition to speech. Embodied agents are yet to reach the

wider population. On the other hand, the ease of development, familiarity of use, and the privacy afforded by purely messaging-based agents have ensured that most development efforts have centered on building conversational agents with no provision for gestures or speech. **Table 1** provides a summary of most popular text-messaging based CAs, called chatbots.

## 2.2. Motivations to Use Conversational Agents/Chatbots

On a general level, most users expect effectiveness and efficiency, through the use of chatbots, in conducting productivity tasks such as access to specific content or help with administrative chores, while other chatbots may be used for entertainment-based and social experiences. Moreover, successful chatbots seem to inform users about what to expect from the beginning. This means that they are transparent about who the users are having a conversation with—that they are interacting with a chatbot and not a human. Information about what chatbots are able (and not able) to deliver is another important factor to communicate to the user. However, while there is some existing research into people's uses and motivations for using media technology in general, there is a dearth of research on why people use chatbots or stop using them.

According to the uses-and-gratifications perspective, people's social and psychological factors produce reasons for their motivations for media use. People are found to use media technologies strategically by employing different media technologies for diverse purposes. Thus, media users select among media technologies based on how well a certain media form helps them meet specific needs or goals [19]. A fundamental notion in the uses and gratifications perspective [20] is that people are motivated by a desire to fulfill certain needs [21]. The key

**Table 1.** Timeline of conversational agents evolution [18].

Year	Conversational Agents/Chatbots
1950	Chatbots Revolution Concept of truly intelligent Machine
1966	Eliza-MIT-Simulate Human Conversation
1972	Parry-Added Conversational Strategy
1988	JABBERWACKY-Simulate natural human conversation
1992	Dr. SBAITSO-Speech Synthesis Program
1995	Alice-Artificial Linguistic Internet Computer Entity
2001	SMARTERCHILD-Precursor to Apple's SIRI
2006	IBM's WATSON-Natural Language Processing
2010	SIRI-Apple's IOS, Natural Language UI
2012	Google Now-Natural Language for google search
2015	Alexa-Amazon; using language processing Algorithms
2015	CORTANA-Bing Search; Natural Voice; Different Language
2016	Facebook user bots
2016	TAY-Microsoft to mimic the speech and habit of teenagers

is, therefore, not to ask how a particular media use influences users but how users' basic needs or requirements influence their particular media choices. These choices are found to be motivated by several basic needs such as entertainment, social connection, identity, and information.

Our Minnesota State Chatbot system has been designed while taking into consideration the characteristics of the students' community. Nowadays, students use more frequently their smartphones than any personal or instructional computers. The constant and continuous connection of students to the internet through institutionally provided WiFi or commercial internet service providers, increases the chances to engage students in interacting with our virtual assistant.

Typically, students have an assigned academic advisor or a counsellor while attending college. However, the availability of the advisor and the frequency of student-advisor interaction is particularly limited. Students should benefit from the continuous and systematic availability of our proposed virtual assistant which will become responsible for student support in all non-academic matters, including orientation, login information, regular check-ins, course registration, communicating graduation requirements and applying for graduation. Additionally, the virtual assistant could be trained on identifying a set of at-risk indicators, including consistently late assignments, technology challenges, lack of login activity and more. Our virtual assistant will coach students toward successful resolution of at-risk indicators.

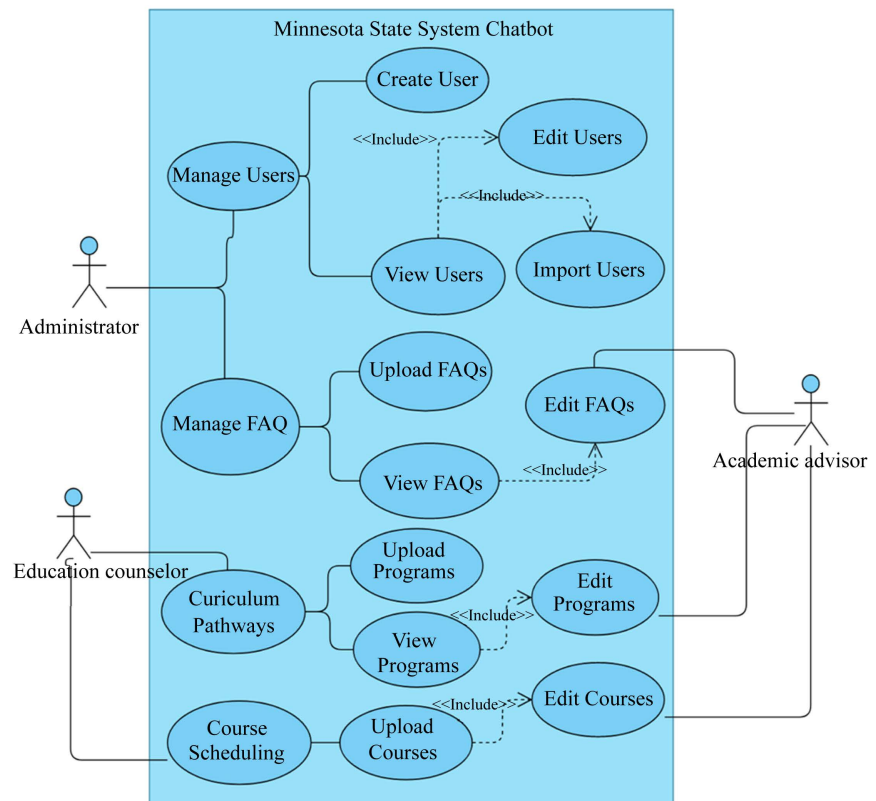
### 3. Minnesota State Chatbot System

In this section, we detail the followed steps to support the Software Development Life-Cycle (SDLC) [22]. First, we present the requirement engineering process and highlight the key system requirements. Next, we provide an overview on the system design and architecture. Finally, we outline the system security analysis of the Minnesota State Chatbot system.

#### 3.1. System Requirements Engineering

The Minnesota State Chatbot system has been designed to meet specific requirements that aim to support academic advising and academic counselling activities [23]. Therefore, two key actors have been identified: 1) Academic Advisor; 2) Education counsellor. Moreover, with respect to the distributed nature of the Minnesota State system involving several colleges and universities, each institution should independently and autonomously set up and control its instance of the chatbot by managing its users and associated resources. Hence a third actor has been added to the list; 3) Administrator. **Figure 1** presents the use case diagram of the Minnesota State Chatbot system.

Requirements describe the characteristics that a system must have to meet the needs of the stakeholders. These requirements are typically divided into functional and non-functional requirements. Functional Requirements [FR] describe how a software must behave and what are its features and functions [24].



**Figure 1.** Use case diagram of the conversational agent system.

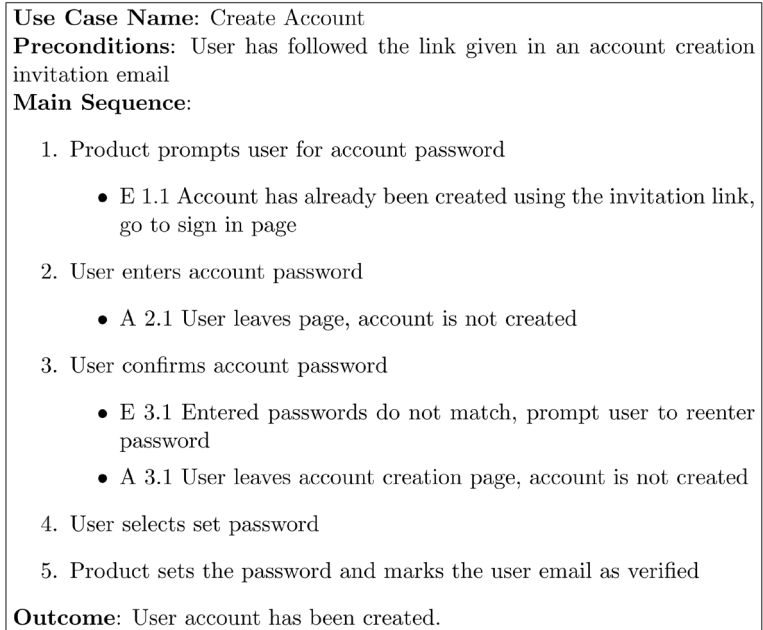
Non-Functional Requirements [NFR] describe the general characteristics of a system [25]. They are also known as quality attributes.

The following is a selection of functional requirements:

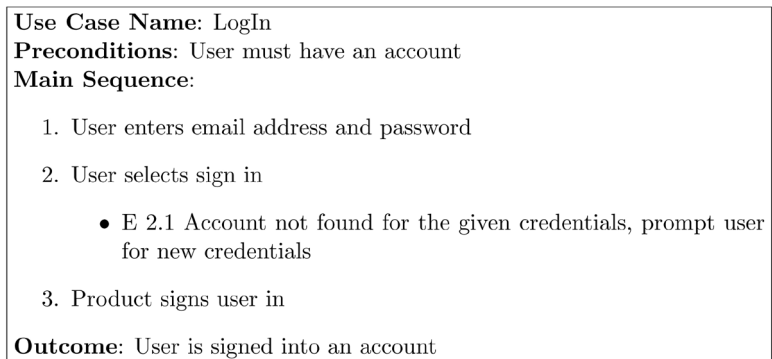
- **[FR1]** The system shall allow to create, view and edit users' accounts for the selected users who will be interacting with the system (see **Figures 2-7**);
- **[FR2]** The system shall allow extracting, pull, view and edit frequently asked questions;
- **[FR3]** The system shall allow to extract, pull, view, and edit existing curricula pathways;
- **[FR4]** The system shall allow to extract, pull, view and edit courses that are scheduled at each academic institution;
- **[FR5]** The system shall provide rich information about courses including sections, classrooms, timing, number of credits, description, student learning outcomes and assigned teachers;
- **[FR6]** The system shall allow various types of users each with specific roles and responsibilities in order to maintain the system integrity and ensure the control and protection of critical data.

The above listed functional requirements have been analyzed and validated with stakeholders and the following set of quality attributes (non-functional requirement) has been derived:

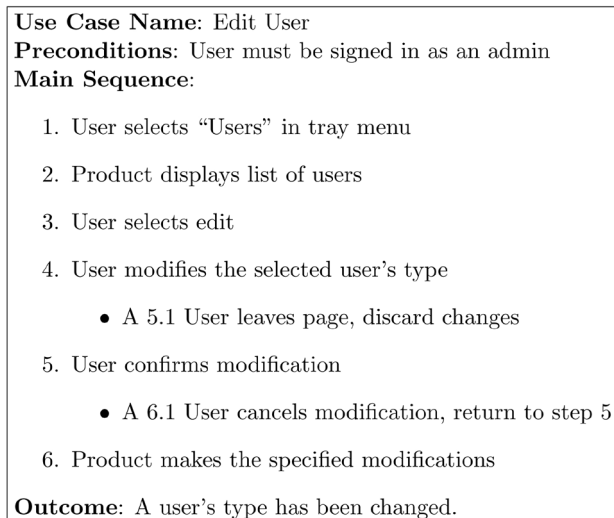
- **[NFR1]** Availability: the system shall be available 24/7/365;



**Figure 2.** Use case diagram *create account*.



**Figure 3.** Use case diagram *LogIn*.



**Figure 4.** Use case diagram *edit user*.

**Use Case Name:** Activate / Deactivate User  
**Preconditions:** User must be signed in as an admin  
**Main Sequence:**

1. User selects “Users” in tray menu
2. Product displays list of users
3. User selects a displayed user
4. User selects activate / deactivate user
  - A 4.1 User leaves page, discard changes
5. User confirms action
  - A 5.1 User cancels action, return to step 3
6. Product performs the specified action

**Outcome:** A user has been activated or deactivated.

**Figure 5.** Use case diagram *activate/deactivate user*.

**Use Case Name:** View Users  
**Preconditions:** User must be signed in  
**Main Sequence:**

1. User selects “Users” in tray menu
2. Product displays list of users and their details such as user add date, user creation date, user type, name, email, etc.

**Outcome:** Product has displayed a list of users.

**Figure 6.** Use case diagram *view users*.

**Use Case Name:** View FAQ  
**Preconditions:** User must be signed in the the system  
**Main Sequence:**

1. User selects “FAQ” in tray menu
2. User selects “Display FAQ” tab
3. Product displays a list of all material that has been imported to the FAQ
  - E 2.1 If nothing has been imported the product will display a message “Nothing has currently been imported to FAQ”

**Outcome:** FAQs in DialogFlow have been displayed.

**Figure 7.** Use case diagram *view FAQ*.

- **[NFR2]** Configurability: The system shall operate on separate data stores and repositories to ensure isolation from other existing information systems in each academic institution;
- **[NFR3]** Security: Users should be created and managed by the system with security features and techniques;
- **[NFR4]** Portability: The system shall be exportable and capable of using common and well-established messaging applications (*i.e.* Facebook messenger, Slack or Microsoft Skype).

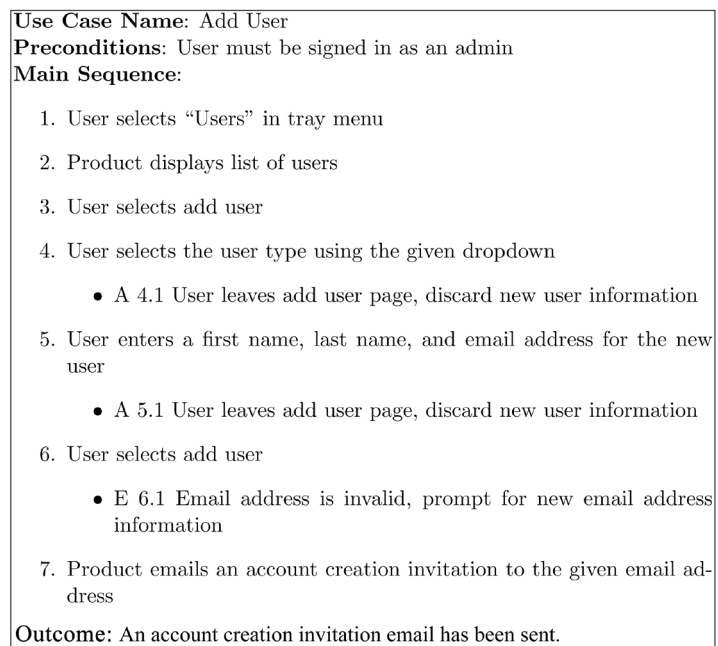
### Scenarios

The use case diagram depicted in **Figure 8** represents the actions that are required to meet system requirements. This use case has multiple “*paths*” that can be taken by any user at any one time. A scenario is a single path through the use case. It describes a real-world example of how one or more users interact with our system. A scenario describes the steps, events, and/or actions which occur during the interaction. Usage scenarios can be very detailed, indicating exactly how someone works with the user interface, or reasonably high-level describing the critical actions but not indicating how they’re performed. This section details the scenarios indicated in the use case diagram of the Minnesota State chatbot system.

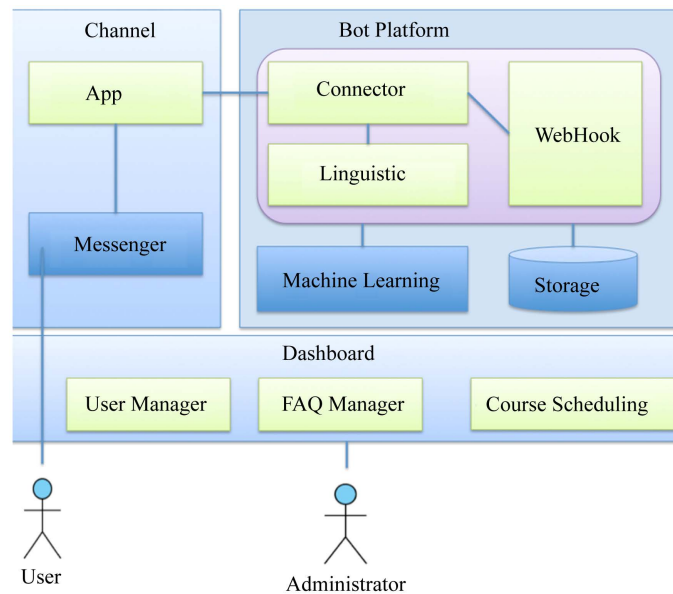
### 3.2. System Architecture

Any chatbot’s software architecture mainly encompasses the layer-architecture detailed in **Figure 9**.

*Channel* is the way of communication with users. It may be Slack, FaceBook Messenger, Skype or another application. The Application on the side of the channel needs to handle events to track incoming messages. It may require special permissions. The *Connector* is the service that interfaces with a bot platform. When integration configuration is done, the *App* will interact with the Connector in order to answer to the user. The Connector in its turn will use described linguistic logic and, if needed, will send requests to external services via web hooks. The logic service sends back results which the bot will then send to the user. Linguistic logic relies on rules and Machine Learning to recognize users speech. This recognition subsystem helps to extract important parameters to conduct requests to the external web services.



**Figure 8.** Use case diagram *add user*.



**Figure 9.** A chatbot typical software layered architecture.

In our Minnesota State Chatbot system, we explored several well-known platforms including:

- DialogFlow provided by Google;
- Wit.ai provided by Facebook;
- Watson Conversation provided by IBM;
- Microsoft Bot platform with LUIS;
- Amazon Lex.

We decided to adopt DialogFlow to implement our chatbot and we will discuss this decision in Sub-Section 4.1, in the following, we will introduce DialogFlow and its main characteristics.

### 3.2.1. DialogFlow

DialogFlow is a tool that supports NLP and is used to detect keywords and intents in a user's sentence. Its role is to help to build chatbots using Machine Learning algorithms.

**Agent:** DialogFlow allows for the implementation of NLU modules, called agents (basically the face of the bot). This agent connects to your backend and provides it with business logic.

**Intent:** An agent is made up of intents. Intents are simply actions that a user can perform on an agent. It maps what a user says to what action should be taken. They are entry points into a conversation. In short, a user may request the same thing in many ways, re-structuring their sentences. But in the end, they should all resolve to a single intent.

Examples of intents can be: "Where is Century College?" or "What does Dr. Baani teach?" It is possible to create as many intents as the business logic requires, and even co-relate them, using contexts. An intent decides what API to call, with what parameters, and how to respond back, to a user's request.

**Entity:** An agent wouldn't know what values to extract from a given user's input. This is where entities come into play. Any information in a sentence, critical to your business logic, will be an entity. This includes concepts like dates, distance, currency, etc. There are system entities, provided by DialogFlow for simple things like numbers and dates. And then there are developer defined entities.

**Context:** Context is what makes the bot truly conversational. A context-aware bot can remember things, and hold a conversation like humans do. Consider the following conversation: "Can I register for the course CS-200 if I did not take the course CS151?", "Sorry, you cannot register for CS-200 because the course CS-151 is a required prerequisite", "Okay, what about CS-100 then?", "That works! CS-100 does not require any prerequisites".

Let us analyse this short yet complex conversation. The first question is straightforward to parse. The registration request is about the course "CS-200", and the event, "no successful completion of CS-151". However, the second question, "Okay, what about CS-100 then?" doesn't specify anything about the actual event. It's implied that we're talking about "no successful completion of CS-151". This sort of understanding comes naturally to us humans, but bots have to be explicitly programmed so that they understand the context across these sentences.

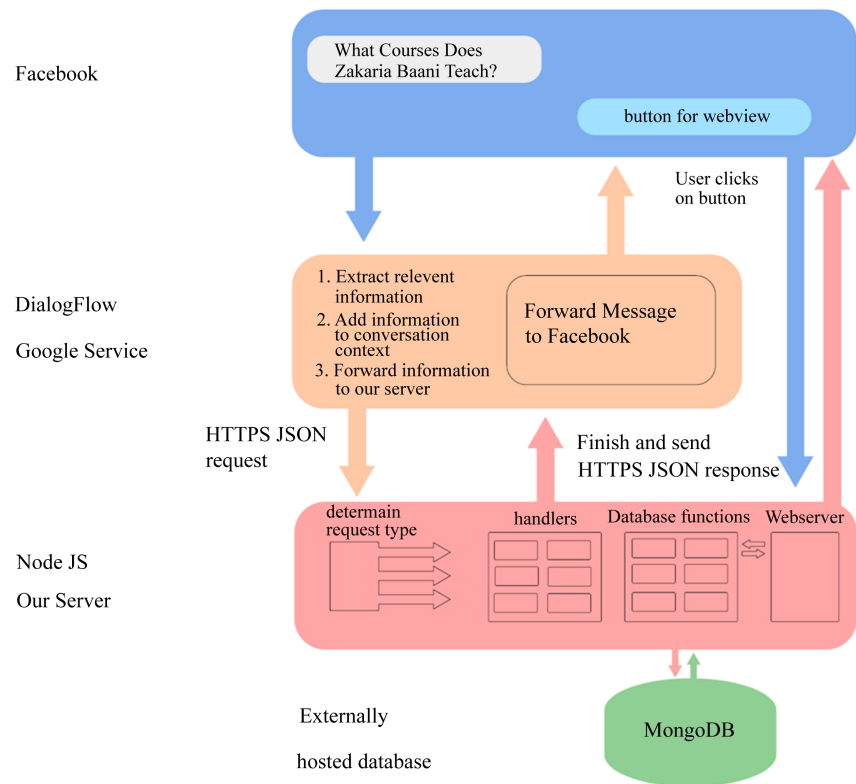
**WebHook:** One may ask "How do I create a useful chatbot with some complex actions?" That's where the webhook is handy. Every time DialogFlow matches an intent, it is possible to ask DialogFlow to send a request to a specific endpoint. An endpoint which will obviously have to be coded to perform a customized process.

The software design and architecture of the Minnesota State Chatbot system involves the following technologies; NodeJS [26], DialogFlow [27], MongoDB [28], and EJS [29]. The user interactions with the chatbot follows this sequence (see **Figure 10** for a graphical illustration):

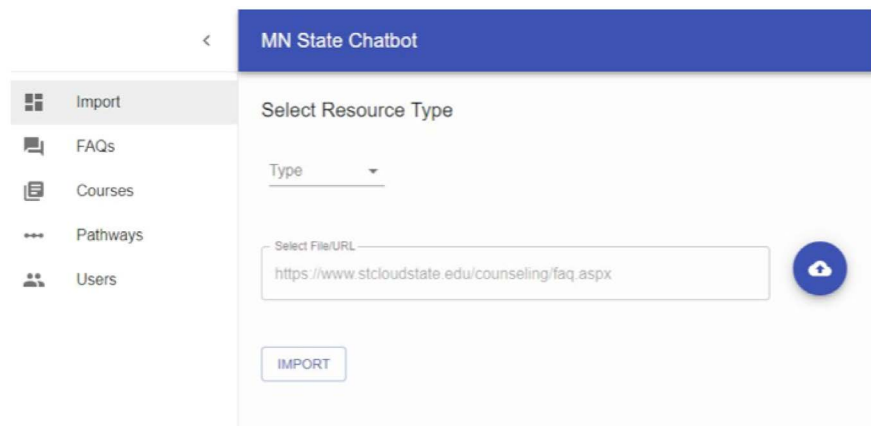
- 1) Facebook user sends question;
- 2) Question is parsed by DialogFlow for info;
- 3) DialogFlow sends extracted info to NodeJS;
- 4) NodeJS sends a response back to DialogFlow, may be a button;
- 5) DialogFlow forwards the message to Facebook;
- 6) When a button: clicking on it directly sends a message to NodeJS;
- 7) NodeJS directly responds to Facebook with generated HTML.

### 3.2.2. Chatbot Dashboard

The Dashboard is meant as a jumping-off point for the administrators to customize the Minnesota State Chatbot to fit the needs of their academic institution. Using the Dashboard, each institution is able to autonomously and independently manage its resources and users. **Figure 11** provides an overview of the Dashboard user interface. The left vertical menu highlights the currently supported features including, Resource management, Frequent Asked Questions management, Course Scheduling management, Pathways management, and



**Figure 10.** Chatbot collaboration workflow.



**Figure 11.** The dashboard system.

user management. Thanks to the dashboard system, each university or college will access the dashboard through a secured login process.

Administrators can create new user accounts. **Figure 12** illustrates the user interface allowing for the creation of new users. Notice the user type field which allows specifying the role of the new user. Two roles are supported: Admin and Viewer. Admin users are allowed to perform advanced actions including the edit of an account properties, the activation/deactivation of account as well as the physical delete of unwanted accounts. Unlike admins, viewers are only allowed to view user accounts and perform basic reporting and monitoring activities.

**Figure 12.** The new user creation page only available to admin users.

### 3.3. System Security

The main need for MNSCU Chatbot is to ensure secure network traffic between all the connections. Since The Minnesota State Chatbot system handles publicly accessible information, it does not require a lot of security layers to protect the public data. We have determined that the only security, the Chatbot, needs would be running Secure Sockets Layer (SSL) connections for all network traffic. Having Hypertext Transfer Protocol Secure (HTTPS) connections would be more than sufficient for what the chatbot is handling and would not cause any more overhead than necessary.

Facebook requires any server attempting to connect to its developer servers to be a secure HTTPS connection. Any traffic from chatbot servers to facebook will be secure because of their regulation. The main connections from NodeJs to DialogFlow to MongoDB would all be using an HTTPS connection as its one and only layer currently. The connection from Dashboard to DialogFlow will also use a simple HTTPS connection which should be more than enough. Another potential layer would be an access token in and out of Dashboard servers since it deals with the chatbot resources.

Dashboard uses an email provider called SendGrid which sends new users a welcome email. SendGrid [30] uses Transport Layer Security (TLS) encryption for all data in transit so the traffic to and from their server should be secured on their end.

## 4. Discussion and Future Work

### 4.1. Evaluation of DialogFlow

DialogFlow enables developers to enhance their application's interaction features

for their users through AI-powered text and voice discussions. DialogFlow helps save the developers' time by simplifying the coding process. The system has a built-in, inline code editor where developers can perform all their code-related tasks.

The machine learning technology from Google is now supporting and powering DialogFlow which enhanced its capability. With this, developers are given the platform with which they can train their agents. They also have access to several templates that are pre-built and can be used as a foundation for chatbot development.

Chatbots created with the aid of DialogFlow can have the capability to engage in natural language conversations. When the customers will be talking to an application program, usually to ask for support or assistance, they would receive in-context replies. The chatbot wouldn't feel too robotic or mechanical.

DialogFlow suffers from several limitations. First, it only supports web-based applications which excludes desktop as well as iOS or Android based applications. Moreover, it only supports online customer support and not phone-based. One more limitation deals with the types of users of DialogFlow. DialogFlow has been adopted mainly by small businesses and it does not seem to meet the expectations nor the requirements of medium-size and enterprise-level businesses. In addition, in its current version, is it impossible to block the matching of an intent if a context is present. Also, the training of the machine learning algorithm section is still in beta.

## **4.2. Next Generation Learning Management Systems**

The paper acknowledges the importance and growing trends of chatbots in user's convenience. Chatbots are effective in resolving problems and providing information accurately to the user and at the same time provide major analytics to better understand the user behaviors.

Learning Management System (LMS) is software used to deliver education and training courses by organizing details, creating, managing and delivering the courses. LMS can be used for all learning activities whether it's an employee training, orientation, and knowledge retention or learning in school and higher education institutions. The LMS has also revolutionized the learning sector worldwide through its utilities to students, teachers and administrators at their own choices.

The Next Generation LSM (NGLMS) is expected to include chatbots as an approach towards a more systematic and user-friendly environment to seek right information at right time through the effective usage of natural language and Artificial intelligence. The use of chatbots is expected to effectively reduce the time a student spend to locate his information, and allows the teacher to monitor the usage for effective teaching. NGLMSs with the chatbots will enable the continuous learning among the students even between the short and long breaks due to ease of access, connectivity and relevance of information. The bots will keep updating the question bank and the answers thereof to reach the maximum accuracy. The user need not to understand the complex NGLMS interface to re-

solve his doubts by peeping in his NGLMS but will get the quick access to the intelligent search.

Participating academic institutions will get a chance to closely observe the data related to content, learning habits, usefulness, and ways to deliver the content to make the learning more successful through the planned scripting, data analysis and content creation for the learner-centred model. The chatbots in NGLMS will enable institutions to know what the students need to learn, how they learn, what is being learned and when to make the learning more effective.

### 4.3. Future Work

In this paper, we presented the analysis, design and implementation of the Minnesota State Chatbot system that can be customized for each academic institution member of the Minnesota State system. Although the implemented features of the chatbot are quite important and mainly focus on answering students' questions that are institution-specific like tuition, course schedule, events, and programs/degrees (Pathways), there are other value-added features that are still missing.

It could be interesting to integrate the Minnesota State Chatbot system with a Learning Management System (LMS) [31]. LMS is a cloud-based technology used by faculty to deliver content, monitor student participation, and assess student performance. It is also used by students to check course material, policy, deliver assignments, and check grades. Our Minnesota State Chatbot system would be useful to help to identify a set of at-risk indicators, including consistently late assignments, technology challenges, lack of login activity and more.

Moreover, affordability of higher education is a growing issue for students and among the factors contributing to this issue is the cost of academic resources such as textbooks. To overcome this issue, colleges started adopting Open Educational Resources (OER) textbooks. Open Educational Resources (OER) are teaching, learning, and research resources that reside in the public domain. Our Minnesota State Chatbot system would play a key role in allowing academic institutions to add OER textbooks into the chatbot's knowledge base. This will add value to OER textbooks and students will be able to ask the chatbot questions about different topics from the textbook and get valuable and helpful answers (Figure 13 and Figure 14).

Finally, adding voice command capability to our Minnesota State Chatbot system is yet another innovative feature that could be implemented. Voice command is more and more popular and widely adopted user interface to control and command software such as chatbots.

By implementing the features above, the Minnesota State Chatbot system will help keep students connected to their learning environment inside and outside of the classroom. For example, by integrating the chatbot with LMS, students will be able to ask questions (voice or text) like: "When is assignment 1 of course CS-100 due?" Students will hence receive notifications about assignments approaching due dates, reminders of class announcements, school events, etc. (Figure 15).

MN State Chatbot

Charlie Tran

Search user by name

+ ADD USER

EMAIL SELECTED

Users

<input type="checkbox"/>	Name	Email	User Type	Last Login	Account Info	Edit	Delete
<input type="checkbox"/>	Charlie Tran	cntran@stcloudstate.edu	admin	July 31st 2019, 1:49 pm			
<input type="checkbox"/>	Samuel Testing	samuelbosire60@gmail.com	viewer	July 24th 2019, 1:35 pm			
<input type="checkbox"/>	SCSU Admin	scsuvizlab@gmail.com	admin	July 24th 2019, 1:34 pm			
<input type="checkbox"/>	Marty Filzer	martyfilzer@gmail.com	admin	July 24th 2019, 1:14 pm			

**Figure 13.** The user management page. All users' attributes are listed including name, email, type, last login data such as last login time, ID, IP address and OS of the used terminal for monitoring purposes.

MN State Chatbot

FAQ IMPORT

FAQ DISPLAY

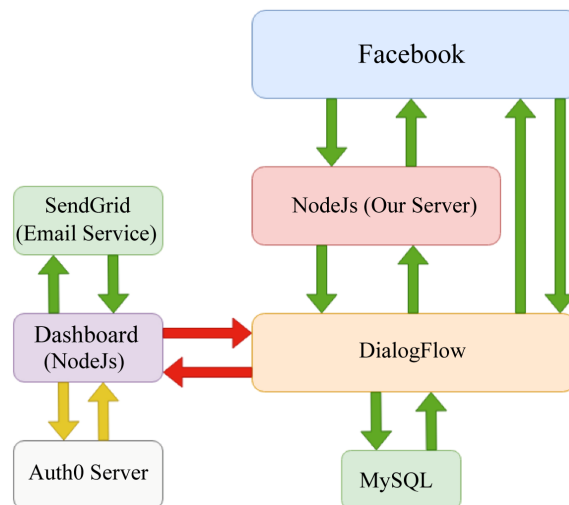
Select FAQ Location

Select File/URL

<https://www.stcloudstate.edu/counseling/faq.aspx>

IMPORT FAQ

**Figure 14.** The automated Frequent Asked Questions (FAQ) loading page.



**Figure 15.** The network flow of the Minnesota state chatbot system.

## Acknowledgements

This research was supported by The Minnesota State Colleges & Universities (MNSCU). Sincere thanks to my outstanding software engineering students at *St. Cloud State University*: Raymond Fitzner, Samuel Ondieki, Joshua Blair, Charlie Tran, Patrick Van Landshoot who provided valuable efforts that greatly benefited this research project. We thank our collaborators Scott Schanke, Eli Krumholz and Mark C. Gill, who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper. We would also like to show our gratitude to our talented students who actively participated in the development of the conversational agent System: Aaron Lawrence from Metro State University and Mark Derosier from Century College.

## Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

## References

- [1] Lin, Z., Madotto, A., Winata, G.I., Xu, P., Jiang, F., Hu, Y., Shi, C. and Fung, P. (2021) Bitod: A Bilingual Multi-Domain Dataset for Task-Oriented Dialogue Modeling. arXiv Preprint, arXiv:2106.02787.
- [2] World Health Organization (WHO) and United Nations International Children Emergency Fund (UNICEF) (1990) Integrated Community Case Management (ICCM): An Equity-Focused Strategy to Improve Access to Essential Treatment Services for Children. World Health Organization (WHO) and United Nations International Children Emergency Fund (UNICEF), Geneva.
- [3] United Nations Children's Emergency Fund (UNICEF) (2019) Education-Literacy Data. United Nations Children's Emergency Fund, New York.
- [4] Abubakar, A.M. and Al-Zyoud, M.F. (2021) Problematic Internet Usage and Safety Behavior: Does Time Autonomy Matter? *Telematics and Informatics*, **56**, Article ID: 101501. <https://doi.org/10.1016/j.tele.2020.101501>
- [5] Arnold, K.D., Chewning, A., Castleman, B. and Page, L. (2015) Advisor and Student Experiences of Summer Support for College-Intending, Low Income High School Graduates. *Journal of College Access*, **1**, Article No. 3.
- [6] Bettinger, E.P., Long, B.T., Oreopoulos, P. and Sanbonmatsu, L. (2012) The Role of Application Assistance and Information in College Decisions: Results from the H&R Block Fafsa Experiment. *The Quarterly Journal of Economics*, **127**, 1205-1242. <https://doi.org/10.1093/qje/qjs017>
- [7] Castleman, B.L. and Page, L.C. (2015) Summer Nudging: Can Personalized Text Messages and Peer Mentor Outreach Increase College Going among Low-Income High School Graduates? *Journal of Economic Behavior & Organization*, **115**, 144-160. <https://doi.org/10.1016/j.jebo.2014.12.008>
- [8] Clarizia, F., Colace, F., Lombardi, M., Pascale, F. and Santaniello, D. (2018) Chatbot: An Education Support System for Student. 2018 *International Symposium on Cyberspace Safety and Security*, Amalfi, 29-31 October 2018, 291-302. [https://doi.org/10.1007/978-3-030-01689-0\\_23](https://doi.org/10.1007/978-3-030-01689-0_23)

- [9] Licklider, J.C. (1960) IRE Transactions on Human Factors in Electronics. Institute of Electrical and Electronics Engineers, New York.  
<https://www.worldcat.org/title/ire-transactions-on-human-factors-in-electronics/oclc/804151195>
- [10] Weizenbaum, J. (1983) Eliza—A Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the ACM*, **26**, 23-28. <https://doi.org/10.1145/357980.357991>
- [11] Bohus, D. and Rudnicky, A.I. (2003) Ravenclaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda. *8th European Conference on Speech Communication and Technology*, Geneva, 1-4 September 2003.
- [12] Rosenfeld, R., Olsen, D. and Rudnicky, A. (2001) Universal Speech Interfaces. *Interactions*, **8**, 34-44. <https://doi.org/10.1145/384076.384085>
- [13] Wen, T.-H., Gasic, M., Mrksic, N., Su, P.-H., Vandyke, D. and Young, S. (2015) Semantically Conditioned Lstm-Based Natural Language Generation for Spoken Dialogue Systems. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, 17-21 September 2015, 1711-1721.  
<https://doi.org/10.18653/v1/D15-1199>
- [14] Hutchens, J.L. (1996) How to Pass the Turing Test by Cheating. School of Electrical, Electronic and Computer Engineering Research Report TR97-05. University of Western Australia, Perth.
- [15] Abu Shawar, B. and Atwell, E. (2015) ALICE Chatbot: Trials and Outputs. *Computation y Sistemas*, **19**, 625-632. <https://doi.org/10.13053/cys-19-4-2326>
- [16] Shawar, B.A. and Atwell, E. (2002) A Comparison between Alice and Elizabeth Chatbot Systems. Research Report 2002, No. 19, University of Leeds, School of Computing, Leeds.
- [17] Abdul-Kader, S.A. and Woods, J. (2015) Survey on Chatbot Design Techniques in Speech Conversation Systems. *International Journal of Advanced Computer Science and Applications*, **6**. <https://doi.org/10.14569/IJACSA.2015.060712>
- [18] Bieliauskas, S. and Schreiber, A. (2017) A Conversational User Interface for Software Visualization. *2017 IEEE Working Conference on Software Visualization (VISOFT)*, Shanghai, 18-19 September 2017, 139-143.  
<https://doi.org/10.1109/VISOFT.2017.21>
- [19] Mekni, M. and Haynes, D. (2020) Smart Community Health: A Comprehensive Community Resource Recommendation Platform. *13th International Conference on Health Informatics, HEALTHINF 2020-Part of 13th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2020*, Valletta, 24-26 February 2020, 614-624.
- [20] Brandtzaeg, P.B. and Følstad, A. (2017) Why People Use Chatbots. *2017 International Conference on Internet Science*, Thessaloniki, 22-24 November 2017, 377-392.  
[https://doi.org/10.1007/978-3-319-70284-1\\_30](https://doi.org/10.1007/978-3-319-70284-1_30)
- [21] Katz, E., Blumler, J.G. and Gurevitch, M. (1973) Uses and Gratifications Research. *The Public Opinion Quarterly*, **37**, 509-523. <https://doi.org/10.1086/268109>
- [22] Mahalakshmi, M. and Sundararajan, M. (2013) Traditional SDLC vs SCRUM Methodology—A Comparative Study. *International Journal of Emerging Technology and Advanced Engineering*, **3**, 192-196.
- [23] Loucopoulos, P. and Karakostas, V. (1995) System Requirements Engineering. McGraw-Hill, Inc., New York.
- [24] Shah, J.J. and Rogers, M.T. (1988) Functional Requirements and Conceptual Design

- of the Feature-Based Modelling System. *Computer-Aided Engineering Journal*, 5, 9-15. <https://doi.org/10.1049/cae.1988.0004>
- [25] Glinz, M. (2007) On Non-Functional Requirements. *15th IEEE International Requirements Engineering Conference*, Delhi, 15-19 October 2007, 21-26. <https://doi.org/10.1109/RE.2007.45>
- [26] Farrell, E. and Winner, N. (2016) CreateSpace Independent Publishing Platform.
- [27] Singh, A., Ramasubramanian, K. and Shivam, S. (2019) Introduction to Microsoft Bot, Rasa, and Google Dialogow. In: *Building an Enterprise Chatbot*, Springer, Apress, 281-302. [https://doi.org/10.1007/978-1-4842-5034-1\\_7](https://doi.org/10.1007/978-1-4842-5034-1_7)
- [28] Kanade, A. and Kanade, S. (2021) A Study of MongoDB Data Models and a Novel Hybrid Data Modeling Approach. *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, 3-5 June 2021, 1554-1562. <https://doi.org/10.1109/ICOEI51242.2021.9452962>
- [29] Carlsson, D. and Ko, D. (2015) System and Method for Javascript Based Html Website Layouts. US Patent No. 8959427.
- [30] Twilio SendGrid (n.d.) Security. <https://sendgrid.com/policies/security/>
- [31] Mekni, M., Baani, Z. and Sulieman, D. (2020) A Smart Virtual Assistant for Students. *Proceedings of the 3rd International Conference on Applications of Intelligent Systems*, Las Palmas, 7-9 January 2020, Article No. 15. <https://doi.org/10.1145/3378184.3378199>