

Smart Assistant

by

Parker Hughes, Peter Bengel and Abel Berhan

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2019 Parker Hughes, Peter Bengel and Abel Berhan

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

Parker Hughes

Parker Hughes, Project Manager

04/15/2018

Date

Peter Bengel

Peter Bengel, Software Developer

04/15/2018

Date

Abel Berhan

Abel Berhan, Network Specialist

04/15/2018

Date

Abdou Fall

Abdou Fall, Faculty Advisor

04/15/2018

Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2019

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
Abstract_____	1
1. Problem Statement_____	2
1.1 Introduction_____	2
1.2 Project Description_____	2
1.3 Problem_____	2
1.4 Solution_____	3
2. Project Management_____	3
2.1 User Profile_____	3
2.2 Use Case Diagram_____	5
2.3 Major Project Milestones_____	6
2.4 Project Schedule_____	7
2.5 Methodology/Technical Approach_____	10
2.6 Application Architecture_____	11
2.7 Gantt Chart_____	14
2.8 Budget_____	16
2.9 Test Plan_____	17
2.10 Future Recommendations_____	22
3. Conclusion_____	24
3.1 Fall Semester_____	24
3.2 Spring Semester_____	24
Appendices	
Appendix A: Tech Expo Poster_____	A
Appendix B: References_____	B

LIST OF ILLUSTRATIONS

TABLES

<u>No.</u>		<u>Page</u>
Table 1.	User Profile_____	3-4
Table 2.	Project Objectives/Deliverables Due Dates_____	7
Table 3.	Work Breakdown Structure_____	7-9
Table 4	Budget_____	17
Table 5	Stability Testing_____	21
Table 6	Launch Testing_____	21
Table 7	User Interface Testing_____	22
Table 8	Functionality Testing_____	22

FIGURES

<u>No.</u>		<u>Page</u>
Figure 1.	Use Case Diagram_____	5
Figure 2:	Technical Architecture_____	11
Figure 3:	QnA Bot Test_____	12
Figure 4:	Question/Response Example_____	12
Figure 5:	QnA Bot Response_____	14
Figure 6:	Assignment Plan Gantt Chart_____	14
Figure 7:	Development Plan Gantt Chart_____	15
Figure 8:	Test and Documentation Plan Gantt Chart_____	16
Figure 9:	IT Expo Poster_____	A

ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programing Interface
GSA	Google Search Appliance
LUIS	Language Understanding
QA	Quality Assurance
UC	University of Cincinnati
UCIT	University of Cincinnati's Information Technology Department

ABSTRACT

The current Google search tool worked when searching for keywords but cannot answer specific questions well, leaving a void and many confused users. The current solution to this problem was to programmatically put in solutions for common questions, but many still go unanswered. This caused users to put a strain on the website owner's helpdesk for problems it faces. The Smart Assistant was a powerful QnA bot that provided more accurate search results for to UC-related questions by leveraging Azure's powerful AI system, tracking users when he/she interacts with the bot and remembering what solutions he/she selects for similar questions by future users. It was built on the Azure Cognitive Services platform and featured language recognition for similarly worded questions, a robust cloud network that can be scaled up/down as needed, and self-learning technology that required little maintenance by administrators.

1. PROBLEM STATEMENT

1.1 Introduction

University of Cincinnati's search engine uses Google's Search Appliance and Google will be ending support for it in 2019 and switching to cloud services. Users searching the GSA will get results based on the keywords he/she is searching for. Users who are looking for quick answers will not find what he/she is looking for right away. For this reason, we have come up with a Smart Assistant tool that give users the ability to find the exact answers he/she is searching for.

1.2 Project Description

Develop a QnA tool for any organization that provides accurate solutions for searches and campus questions. Utilizing Microsoft Azure Cognitive Services' APIs, machine learning bots will provide more accurate search results based on tracking what other users have found for that solution. If the answer cannot be found using our QnA service, the service will provide the standard search. All the services will be hosted in the cloud. The user will also be able to provide information to the bot as well for an even more enhanced solution.

1.3 Problem

According to Jeremy Martin, director of Digital Communicans at UCIT, their current search tool works well finding specific keywords but does not have the capability to answer questions. Currently, the solution is for staff to set the correct answers programmatically for common questions, but more complex ones go unanswered. There is a gap where a tool is needed that alleviates an organization's work and allows the user to ask complex questions and expect an accurate response. This was the main reason the Smart Assistant was created.

1.4 Solution

Using the Azure tools, we will be able to get through a lot of our procedures and accomplish most of our objectives. Smart Assistant will utilize Azure to provide more accurate search results to the user. Through machine learning, it will develop and track what users click on after searching and learn from that information what it thinks future results will be for that query. It will allow users limited input to enhance the search results for future users. It will develop a multi-part design that starts user off with QnA chat bot, then transitions to Bing Search if no answer is found.

2. PROJECT MANAGEMENT

2.1 User Profile

The targeted users are mainly regular users looking to interact with the Smart Assistant for questions and UCIT Admins looking to view metrics of the system. *Table 1: User Profile* presents the user profile of these different users, their expected knowledge, and design objectives that would enhance these group's experience using the Smart Assistant.

PROJECT
Smart Assistant – an AI-powered QnA bot
POTENTIAL USERS
• Technologically adept users with background in IT or familiar with organization applications • Users who aren't technologically adept • IT Admins looking to manipulate data for QnA bot
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE
All users should have experience interacting with any search tool or chat bot frequently found in websites like Google or Facebook. This project is aimed to be simple enough for less technologically savvy people to be able to interact with and

accomplish their objective. Admins should have experience viewing and understanding the data structures involved in search engines.
TASK EXPERIENCE At the initial visit to the application’s Web page, users will interact with the Smart Assistant which scans UC’s various Web pages looking for keywords that relate to the question the user asked. It will either return with a result it believes to be the answer or let the user know the result couldn’t be found. • Users should be comfortable using their mobile phone or a browser to interact with the assistant. • Users should also be comfortable having their answers tracked for assistance to future users Admins will interact with the QnA bot through the Azure console and view the metrics of the bot. They will also be able to change the results for specific keywords if need be.
FREQUENCY OF USE This application will be used when users begin searching for something on the UCIT Web page. This could occur daily for people such as IT faculty or up to yearly for outside users looking to interact for the first time. Admins might interact with the bot on a more frequent basis, viewing and changing results to fit their needs.
KEY PROJECT DESIGN REQUIREMENTS THAT PROFILE SUGGESTS • Easy to navigate and understand UI that resembles other chat bot services to familiarize users • Language recognition feature for the application to understand less technologically savvy users’ needs • Ensure technical documentation is thorough and easy to read for IT admins to be able to use tool faster and understand it more. • Ensure application is section 508 compliant for users with various disabilities can interact with the smart assistant

Table 1: User Profile

2.2 Use Case Diagram

Figure 1: Use Case Diagram presents the Use Case Diagram for Smart Assistant. The application will be interacted with by two different groups, regular users consisting of students, faculty, alumni, and other persons who wish to engage with UC's Web site as a demonstration. Regular users will interact with the bot by conducting searches with the system and providing feedback on whether the results were correct or not. Admins will interact with the console seeing what users are looking up and what he/she is selecting, view the number of questions being asked at a given period, and edit the results of specific keywords.

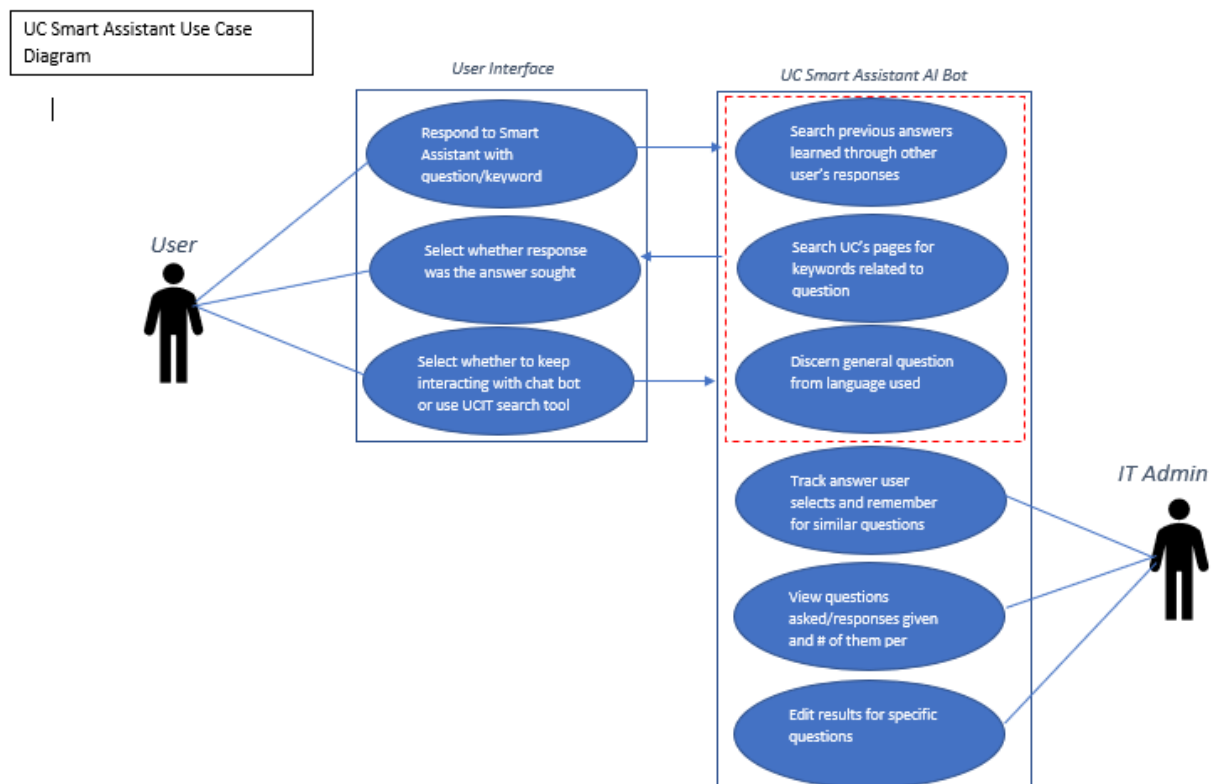


Figure 1: Use Case Diagram

2.3 Project Deliverables

Our project milestones include the features needed for the minimum viable product of the Smart Assistant Product. The goals focus on creating a comprehensive QnA bot that stands out by adding user tracking features and providing features exclusive to Google and Bing products to other developers.

One of the biggest problems faced is how to make the bot understand the questions being asked. After some research through the Microsoft documentation, LUIS was the best fit for the project. LUIS is a machine learning based service to build natural language into apps. It helps the bot become smarter and more efficient by understanding the language used in the question being asked.

Tracking user results and adding a ratings system to the project is a key part that will make it stand out from other QnA bots. This feature will mimic how companies like Google currently track user clicks to improve their search results. This feature will allow the bot to improve its search results without oversight overhead from the IT Admins. Some features planned originally such as adding a Bing Search to the project was scrapped after talking to UCIT and understanding their needs for the project and the features the Smart Assistant should focus on to provide the best service to an organization like UCIT. The project was pivoted to focus on what was originally going to be a component that was going to be lesser to the Bing Search, the QnA bot. The project is also planned to be launched to Github so other companies will be able to use the features included.

Table 2: Major Project Milestones shows the project deliverables and deadlines for the Fall and Spring semesters.

Major Project Milestones (Deliverables)			
Fall 2018 Milestones			
Build basic QnA Bot	10/11/18	Integrating into Web sites	10/29/18
Luis Integration	11/11/18	Track User Results	11/26/18
Spring 2019 Milestones			
Add Styling Options	1/21/19	Display Metrics	2/4/19
UI and Load Testing	2/18/19	Technical Documentation	3/18/19

Table 2: Major Project Milestones

2.4 Project Schedule

Table 3: Project WBS shows our project schedule and major milestones for the Fall and Spring semesters. Listed are the due dates for the course required assignments, the developmental phase of our project as well as the testing and the documentation of our project.

Task Name	Start Date	End Date	Duration (DAYS)	Owner
WBS of Smart Assistant				
Assignment Plan	8/27/18	4/29/19	242	
Establish Team and Complete Assignment 0	8/27/18	9/3/18	7	Parker
Assignment 1: Team Contract	9/17/18	9/24/18	7	Parker
Assignment 2: Project Abstract	10/12/18	10/15/18	3	Parker
Prepare Elevator Pitch	10/12/18	10/22/18	10	Abel
Assignment 3: Team Contract Resubmission	10/11/18	10/15/18	4	Parker

Assignment 4: User Profile	10/15/18	10/22/18	7	Parker
Assignment 5: Use Case Diagram	10/15/18	10/22/18	7	Parker
Assignment 6: Draft Report	10/29/18	11/5/18	7	Parker
Prepare for Final Presentation	11/5/18	11/19/18	14	Abel
Assignment 7: Final Fall Semester Report	11/12/18	11/26/18	14	Parker
Assignment 1: Test Plan/Report	1/28/19	2/11/19	14	Abel
Assignment 2: Abstract Resubmission	2/11/19	2/18/19	7	Parker
Assignment 3: Draft Tech Expo Poster	2/18/19	3/4/19	14	Parker
Assignment 4: Final Poster	3/11/19	3/18/19	7	Parker
Prepare Final Presentation	3/18/19	4/1/19	14	Abel
Prepare for Tech Expo	3/18/19	4/9/19	22	Parker
Assignment 5: Final Report	4/9/19	4/15/19	6	Parker
Assignment 6: Safe-Assign of Final Report	4/14/19	4/15/19	1	Parker
Assignment 7: Library Copy	4/15/19	4/29/19	14	Parker
Development	9/3/18	4/9/19	218	
Research Azure Cognitive Services	9/3/18	9/24/18	21	Abel
Build small knowledge base with test data	9/24/18	10/10/18	16	Peter
Build Azure Bot	9/24/18	10/10/18	16	Peter
Train bot in test data	9/24/18	10/10/18	16	Peter
Add QnA Maker Service	9/24/18	10/1/18	7	Peter
Create an MVC site Using C# to intergate the Azure QNA	10/1/18	10/10/18	9	Parker
Meet with Brian Verkamp and UCIT Communication Team	10/11/18	10/11/18	0	Parker
Adjust project scope after meeting	10/11/18	10/15/18	1	Parker
Further build QnA service	10/15/18	10/29/18	14	Peter

build basic tracking functionality to QnA Service	10/29/18	11/12/18	14	Peter
Enhance tracking feature of service	11/19/18	11/26/18	7	Peter
Increase size of knowledgebase	11/26/18	12/3/18	7	Peter
Build cloud network with ability to scale	12/3/18	12/10/218	7	Abel
Add language recognition feature	12/10/18	1/21/19	35	Peter
Add ability to customize appearance of QnA Service	12/10/18	1/21/19	35	Parker
Develop APIs for outside use	1/21/19	2/4/19	14	Peter
Build new demo for UCIT Communication Team	2/18/19	3/4/19	14	Peter
Adjust project after meeting feedback	3/4/19	3/18/19	14	Peter
Adjust demo for tech expo/final presentation	3/18/19	4/9/19	22	Peter
Testing and Documentation	10/15/18	3/18/19	154	
Research Section 508 compliance and other UCIT requirements	10/15/18	10/22/18	7	Abel
Enusre project is Section 508 compliant	10/22/18	11/5/18	14	Abel
Conduct alpha QA test	11/5/18	11/12/18	7	Abel
Fix critical bugs related to QA test	11/12/18	11/19/18	7	Peter
Build framework diagram of current project	11/19/18	12/3/18	14	Abel
Begin development of technical documents for UCIT	12/10/18	1/21/19	35	Abel
Begin alpha QA test of current features	1/21/19	2/4/19	14	Abel
Optimize code after QA test	2/4/19	2/18/19	14	Peter
Conduct load test of project and review results	2/4/19	2/18/19	14	Parker
Review and adjust framework diagram	2/4/19	2/18/19	14	Abel
Optimize code after load test	2/18/19	2/25/19	7	Peter
Complete technical documentation of project	2/18/18	3/18/19	14	Abel

Table 3: Project WBS

2.5 Methodology/Technical Approach

Design requirements

We started are design requirements based on are initial meeting with UC. A simple QnA service with a knowledge base as its backend. However, upon learning more about the different services Azure could add in to our initial simple QnA, we decided to expand our project by using the services we discovered. We wanted to make sure that any question a user wanted to know could be answered by our system. Also, we wanted to make sure that any company who wanted this system could implement it in their systems easily and without having a developer constantly maintain it. We wanted to give a good search system to the company, but also make sure the price of the application was low as well. Otherwise, companies would not use the system. We understood having just a few knowledge bases probably would not be enough, so we used Bing search through Azure. Any Azure search function could be added. However, if they did not need that function then they could take it out. This whole system had to be easily customizable to fit a company's needs. Therefore, everything in the application had to be dynamic.

Procedures

When a user asks a question to our search service, the main procedure is to return a response by answering their question. The response can come from multiple different places such as a SQL Server database, a knowledge base, or an Azure Bing search. It will run through the SQL database first and try to find anything related to the question asked. If something is found it will return a response. However, if nothing is found it will do the same thing for the knowledge base and search for an answer there. If it finds something it

will return a response however, if it does not it will search Bing. If it finds no response, it will return with a message that nothing can be found and to try their inquiry again.

2.6 Technical Elements

Architecture

Figure 2: Technical Architecture details how the project looks through a technical view. All the project services are in the Azure Cloud which allows the project to remain hardware-free.

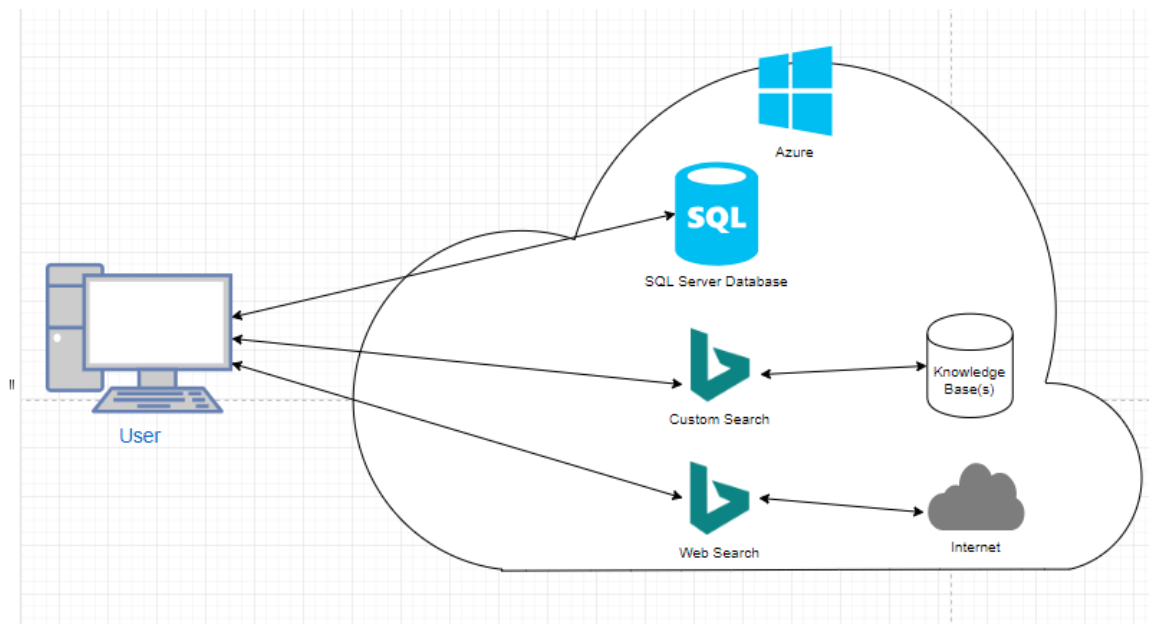


Figure 2: Technical Architecture

Network

The project is hosted on the Azure Cloud by Microsoft. It requires no physical servers hosted by us and is hosted on a pay-as-you-go need plan. Each service uses Microsoft servers in their data center. We use Microsoft Azure SQL Server for the database to store the user ratings feature for the project.

Application

All tools used will be mostly from Microsoft and its Azure Cognitive Services products. Azure will be used to get the services setup. Azure provided .ai sites will enhance the QnA bot like LUIS as well as QnA Maker. Some coding will be required for some of the tasks and Visual Studio 2017 will be used to accomplish that.

Figure 3: QnA Bot Test shows a screenshot where testing is done for the bot. It's a search bar where the user will type in their question.

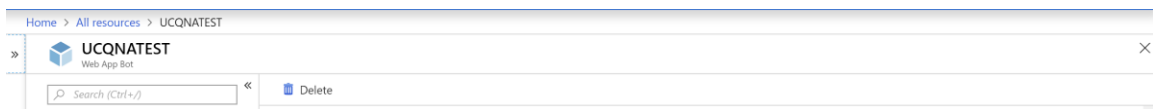


Figure 3: QnA Bot Test

The QnA bot will be hosted in the cloud so no onsite server hosting is required. Microsoft will also store the questions in the cloud as well, so they can be accessed anywhere. The front end to creating questions is on <https://www.qnamaker.ai/>.

Figure 4: Question/Response Example shows a screenshot of the database and how the questions and answers will be stored. This will simplify the process of editing and adding new information as needed.

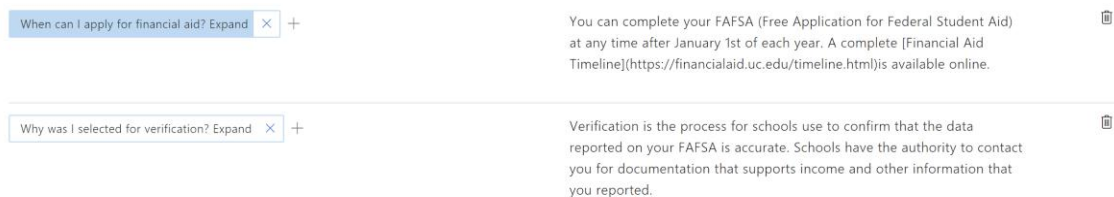


Figure 4: Question/Response Example

Using a bot and questions can make a simple QnA program. The QnA program can be embedded through any site using an iframe. Azure will automatically create the Iframe

that can be embedded on any Web site. The only thing that would be required is adding on some CSS to customize the layout of the QnA program.

The QnA maker will be enhanced through a Microsoft product called LUIS. LUIS is designed to identify valuable information in conversations, Language Understanding interprets user goals (intents) and distills valuable information from sentences (entities), for a high quality, nuanced language model. Using LUIS will make the QnA program machine learning. Making the tool machine learning will make it stand out from any regular QnA program that a site might have.

The last Azure tool that will be added is a Bing Web Search API. If there are no results to the query that was asked, the QnA maker will browse Bing and return results of the query, if there is any at all. The key point is to keep people using the tool and on the site. In this regard, the project will not have to rely on people leaving the site so they can Google search. The point of the program is to keep people on the site to help find the answers they need and give them the best possible experience while doing so.

Overall these three tools should make this QnA maker stand out from the rest of any regular QnA tool. The goal is to make sure that any question that is asked on the site will be answered on the site and they can either navigate to the page that has more information or leave the site once their query is answered. The biggest advantage is that all the tools are through Microsoft and are hosted in the cloud, so everything can be easily scalable and the setup for it will be quite easy because it does not need to be installed on any server that would risk security. Everything would be kept on Microsoft's secure servers.

Figure 5: QnA Bot Response shows a screenshot of how the QnA will look like in a working condition. A question is being asked and a response is given with the answer.

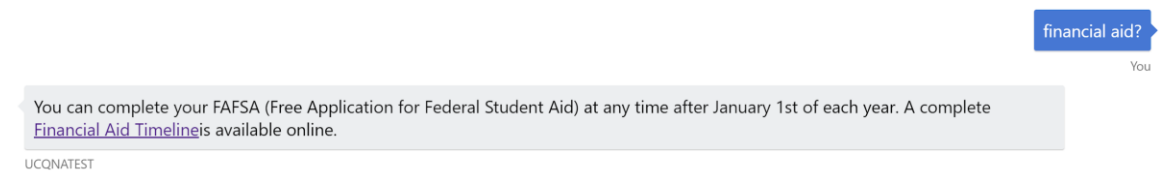


Figure 5: QnA Bot Response

2.7 Gantt Chart

Figure 6: Assignment Plan Gantt Chart details how the group plans to do the various assignments for the Senior Design class and the estimated length it is believed to complete these assignments.

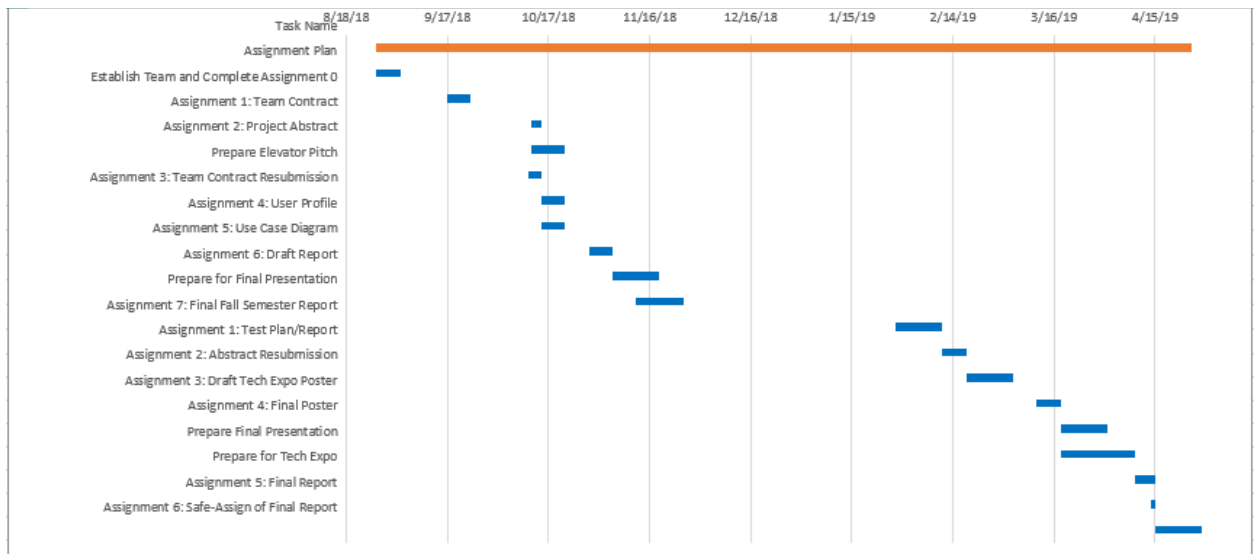


Figure 6: Assignment Plan Gantt Chart

Figure 7: Development Plan Gantt Chart which handles the technical sides of the project, discusses the various components needed for the project to be complete, again with estimation about the time needed.

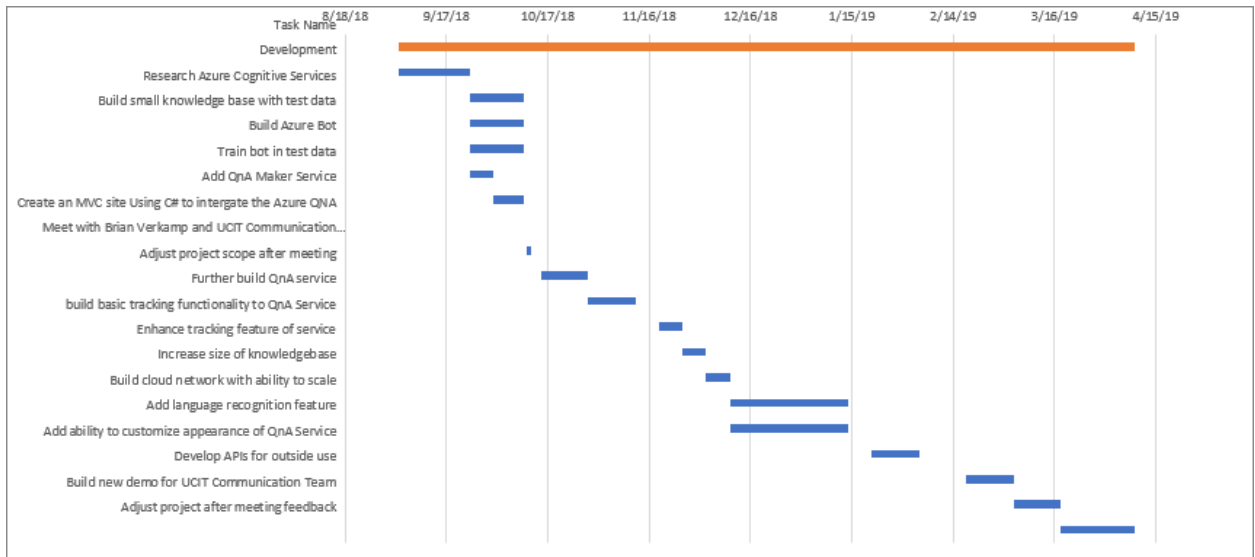


Figure 7: Development Plan Gantt Chart

Figure 8: Test and Documentation Plan Gantt Chart handles both the documents produced outside of the Senior Design class requirements related to the project and the tests that will be handled to ensure the project meets any organization compliance with their technology standards.

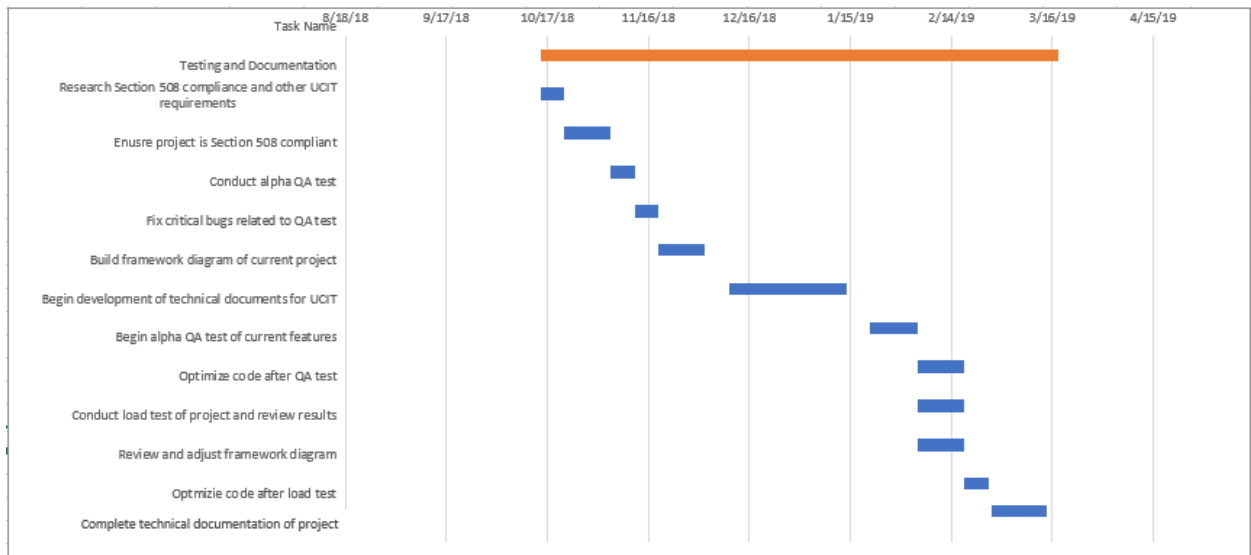


Figure 8: Test and Documentation Plan Gantt Chart

2.8 Budget

The Smart Assistant application is hosted on the Microsoft Azure cloud and does not require physical servers. This aspect is where most of the cost of the project is placed. The simulated cost to use the Azure cloud was calculated assuming it was used by an organization which has users interact with it. The cost of the database is calculated using the Azure Cloud pricing cost of \$15.00 per month for 7 months (1), the duration of our project which comes out to \$105.00. The cost of the Azure Bot Services also uses the Azure website and we assume there are 100,000 messages per month of the page for 7 months, which comes out to \$350.00, with a \$0.50 cost per 1,000 messages. For wages, we assume that each person on the team is earning \$25.00 per hour and contributing 10

hours to the project every week. With 7 months of work, this totals out to \$21,000.00.

Table 4: Budget outlines the project budget for the Smart Assistant with the total cost calculated from the estimated cost if this were a retail project for a company. The total simulated cost of the project would total to \$29,662.53.

Category	Sub-Category	Item	Description	Estimated Cost	Actual Cost
Hardware	Actual Storage Costs	Azure Bot Services	The actual cost of the project taking into account Azure cloud offers 10,000 free messages.	\$0.00	\$0.00
		Azure SQL Database	The database needed to store user ratings.	\$0.00	\$0.00
	Simulated Storage Costs	Azure Bot Services	The QnA bot feature for the project.	\$350.00	\$350.00
		Azure SQL Database	The database needed to store user ratings.	\$105.00	\$105.00
Labor	Wage Costs	Actual Wages Costs	The actual wages for completion of project.	\$0.00	\$0.00
		Simulated Wage Costs.	The simulated wages for the project	\$21,000.00	\$21,000.00
Total				\$21,455	\$21,455

Table 4: Budget

2.9 Test Plan

Overview

This section will cover the testing methodology for the Smart Assistant. This reference can be used by the following individuals:

- Project Manager
- Software Developer

- Network Specialist
- Quality Assurance Personnel

Scope

The scope of testing is to ensure the overall functionality of the Smart Assistant. The tests will be organized based on the requirements of the application.

Objective

The objective of testing is to verify the product is working as intended by the developers, to validate the database is operational and the correct data is being pulled by the user interface, and to verify that there are no errors on the user interface when searching.

Bug Reporting

All bugs found during testing will be logged and reported. The bugs will then be discussed amongst the team and either ignored or fixed on order of severity and how much it affects the product.

Entry and Exit Criteria

Entry Criteria:

- Testable code is available and ready to be tested
- Test case table is setup correctly
- Test environment is available

Exit Criteria:

- Ensuring all critical test cases have passed
- Majority of functionality is completed
- Bugs are properly documented, and high-priority bugs are fixed.

Testing Procedures

The necessary steps needed to test the Smart Assistant application are preparing a table detailing the test cases and providing a separate document for errors found, which will then be discussed among the team.

The headers of the test will list:

- The specified test being run as a student using the Smart Assistant platform
- The date the test occurred
- The Item number under each section.
- The expected results of the test
- The actual results encountered in the test
- Whether the test passed or failed
- The specific bug, if possible

Below are the tests utilized:

1. Stability testing, which is based on being stable in a local environment and website,
2. Launch testing, making sure the program spins up/down as intended.
3. User-Interface Testing, handling that everything appears and is readable.
4. Functionality Testing, handling the bot responses and making sure data is stored correctly.

Pass/Fail Conditions

If any test case fails, the team will review the severity of the bug and fix it based on that order of severity.

Lessons Learned

During the project we learned a lot about many different aspects of developing a project with a group. We learned the most important lesson to keep the group together is communication. Everyone needs to know where we are at and how to get to the finish line. Working together can get us to the finish line faster and it takes off the stress on one person if it was individual project. We also learned how to work with members who came from completely different backgrounds. Two of the people in our group are software developers. One is in networking. It has been a great learning experience bringing in all our skill sets together and making this project. In the future we know that we will be working with people across many different fields. This project was a great step in bringing all of us together. We also learned the skills of planning. This is the biggest project we have ever worked on at school and it takes place over two semesters. We learned how necessary it was to plan to meet all of our deadlines together and to successfully deliver and show a great project to people.

Stability Testing Cases and Results

This test is made to make sure the project is stable in a local environment and website. ***Table 5: Stability Testing*** displays the individual tested for, the date, the expected and actual result, and whether the test passed/failed.

Tester	Date	Item #	Expected	Actual	Pass/Fail	Bug
Student	2/10/2019	1	Runs on Azure Emulator	Is able to run without issue	Pass	
Student	2/10/2019	2	Runs on an IFrame on the website	Is able to run without issue	Pass	

Table 5: Stability Testing

Launch Testing Cases and Results

This test ensures the program spins up/down as needed and connects to the necessary knowledge bases. ***Table 6: Launch Testing*** displays the individual tested for, the date, the expected and actual result, and whether the test passed/failed.

Tester	Date	Item #	Expected	Actual	Pass/Fail	Bug
Student	2/11//2019	1	Connects to Azure	Is able to connect without issue	Pass	
Student	2/11//2019	2	Connects to Financial Aid KnowledgeBase	Is able to connect without issue	Pass	
Student	2/11//2019	3	Connects to CO-OP KnowledgeBase	Is able to connect without issue	Pass	
Student	2/11//2019	4	Connects to luis	Is able to connect without issue	Pass	
Student	2/11//2019	5	Shows the default start message	Shows message	Pass	

Table 6: Launch Testing

User Interface Testing Cases and Results.

This test focused on making sure the user interface has the necessary components and is readable. ***Table 7: User Interface Testing*** displays the individual tested for, the date, the expected and actual result, and whether the test passed/failed.

Tester	Date	Item #	Expected	Actual	Pass/Fail	Bug
Student	2/11//2019	1	Loads all buttons	Loads all buttons	Pass	
Student	2/11//2019	2	Loads on IFrame	Loads on IFrame	Pass	
Student	2/11//2019	3	Asks user if they are finding what they are looking for	Asks user	Pass	
Student	2/11//2019	4	Loads in an emulator	Loads in an emulator	Pass	
Student	2/11//2019	4	Questions are readable in the form	Questions appear in grey boxes and look readable	Pass	

Table 7: User Interface Testing

Functionality Testing Cases and Results

This test is making sure the data is stored correctly and the response feedback.

Table 8: Functionality Testing displays the individual tested for, the date, the expected and actual result, and whether the test passed/failed.

Tester	Date	Item #	Expected	Actual	Pass/Fail	Bug
Student	2/11//2019	1	Receive answer when asking a question	Receive answer	Pass	
Student	2/11//2019	2	Receive answer within 30 seconds	Sometime on the first question it can take more	Fail	
Student	2/11//2019	3	Return response when nothing is found	Returns response	Pass	
Student	2/11//2019	4	Able to rate the system	Able to rate the system	Pass	
Student	2/11//2019	5	Able to get answers from the different Knowledge Bases	Got the answers	Pass	

Table 8: Functionality Testing

2.10 Future Recommendations

If the project was done over again, it would look more like the standard Google search tool, as the team believes this would be a better showcase of the language understanding and user ratings features and would ‘wow’ the judges at the IT Expo. A lot

of feedback from the Spring presentation and Expo was displaying multiple results at the same time which if done on the QnA bot format would flood the screen and confuse the user too much.

If the project had more time, the team would build and add more QnA features such as taking in Audio and converting that to a question and an option to hear questions out loud, both targeted towards disabled users ensuring the project is as accessible as possible to anyone with a question.

One suggestion from the one of the IT Expo judges was to focus on making sure the knowledge bases utilized in the project were up-to-date and contained accurate information, so that is another area the team would expand upon if given time, perhaps taking into account the date a result was posted if there are 2 similar results and the bot needs to choose which one to display first.

As of right now, besides writing out some documentation, there are no plans to take the project to market or build it out more. The team is content with the work done for the project and it is built in a way that companies can pick and choose the features they want. The project can integrate with Cortana and Slack if necessary, giving it further functionality.

3. CONCLUSION

3.1 Fall Semester

The biggest lesson learned so far is that the group learned a lot more about Azure Cognitive Services now compared to the beginning phase of the project. It is a great tool to use on any website. Azure bot with QnA and Bing search services are both used for the Smart Assistant. While there are a lot more services available with Azure, the few services that we are using have demonstrated a lot of potential.

All team members developed a better understanding of how AI and the cloud work through this project. Even though members had various experience with these ideas, all members now have a general understanding of how chatbots operate and the different components needed in order to make one. All team members have enhanced their understanding of C#, the language used to write the project, especially Peter Bengel, the software lead of the team. Each team member has done individual research on tools that can be integrated into the project and how those tools work.

For the spring semester, the plan is to fully build out the QnA service and the user tracking feature, allowing the bot to provide more comprehensive results. Metrics will be provided outside of the azure console for technical documentation will make it easy to export the bot, so any company can easily start using it. Most of the heavy testing will be done next semester as well. All team members will be taking part of the testing stage for QA to make sure the project is operating and functioning as originally intended.

3.2 Spring Semester

The major focus for the Spring semester was the completion of the ratings system necessary to rank individual search results. Bing Web Search was also integrated as well,

developing a three-tier solution for the application ensuring that any UC-related search question will be answered by the Smart Assistant.

All team members grew a deeper understanding of the software's project lifecycle and how one is run. The project manager, Parker, further developed communication skills necessary to articulate ideas and drive the project vision forward. The network specialist, Abel Berhan, learned more about testing methodology and built on his knowledge of Azure services. The software developer, Peter Bengel, built on his knowledge of C# and Azure services as well.

The team successfully completed a lot of the objective from the Fall semester conclusion to build out the QnA service. The bot is able to integrate with services such as Cortana and Slack allowing the Smart Assistant to be exported if necessary. Features in the application were designed to be turned on/off as needed so companies can tailor the application to fit their needs. Testing was done to ensure the bot met standards.

What was learned from the 2019 IT Expo was considering that some pages might have old/conflicting information that would affect the search results for that given answer, as pointed out by one of the judges.

APPENDIX A: TECH EXPO POSTER



University of
CINCINNATI

Smart Assistant

Team 1: Parker Hughes, Peter Bengel and Abel Berhan
Technical Advisor – Abdou Fall

College of Education, Criminal Justice, and Human Services - School of Information Technology



About

The Smart Assistant is a powerful QnA bot that provides more accurate search results for UC-related questions by leveraging Azure's powerful AI system.

- Tracks users when they interact with the bot and remembers solutions selected for similar questions by future users.
- Built on the Azure Cognitive Services platform.
- Features language recognition for similarly worded questions
- Scalable cloud network.
- Self-learning technology that requires little maintenance by admins.

Technical Elements



Implementation

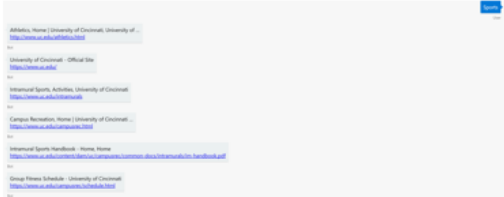
The Smart Assistant utilizes Microsoft's Azure Cognitive Services and consists of the following features:

- 1. LUIS – Language Understanding** interprets user goals (intents) and distills valuable information from sentences (entities).
- 2. QnA Maker API** – tool used to create a knowledgebase of questions and answers.
- 3. Bing Web Search API** – tool used to search outside of the QnA responses.

Problem

UC's current search tool works well on finding specific keywords but does not have the capability to answer questions.

Screenshot



Acknowledgements

Special Thanks to Abdou Fall and the University of Cincinnati IT Department.

Solution

Smart Assistant utilizes Azure to provide more accurate search results to the user.

Figure 9: IT Expo Poster

APPENDIX B: REFERENCES

- [1] “Microsoft Azure,” 2018. [Online]. Available: <https://azure.microsoft.com/en-us/>
- [2] “Luis,” 2018. [Online]. Available: <https://www.luis.ai/home>
- [3] “Microsoft,” 2018. [Online]. Available: <https://www.microsoft.com/en-us/>



Smart Assistant

TEAM 1

Parker Hughes: Project Manager
Peter Bengel: Software Developer
Abel Berhan: Network Specialist

AGENDA

- PROBLEM
- SOLUTION
- TECHNICAL ELEMENTS
- DEMO
- BENEFITS
- CONCLUSION

PROBLEM

- Lack of QnA bot in a lot of different organizations
- No machine learning integration
- Complex questions go unanswered

SOLUTION

1. Developed a QnA bot using Azure's Cognitive Services QnA Maker API
2. Implement Machine Learning capability using Microsoft's Language Understanding (LUIS)
3. Survey System where:
 - ▶ Users can provide limited input on the usefulness of the results
 - ▶ Highest rated results will be displayed first
4. Provides functionality seen in Google and Microsoft products to outside groups
 - ▶ Embed the Smart Assistant in any website using an Iframe, using CSS to customize the appearance

TECHNICAL ELEMENTS



DEMONSTRATION

BENEFITS

- Admins will not have to input results physically as much
- More accurate results with more users
- Less stress on helpdesk system
- Cloud solution that can be scaled up/down as necessary

CONCLUSION

QUESTIONS?