

Table of Contents

Abstract	3
Report Revision History	4
Changes in Version 1.5	4
Problem Statement	5
Business Background	5
Needs	5
Objectives	5
Requirements	6
User Requirements	6
Glossary of Relevant Domain Technology	6
User Groups	6
Functional Requirements	7
Project Scope	7
User Scenarios	8
List of User Functional Requirements	16
Non-Functional Requirements	20
Product: Performance Requirements	20
Product: Dependability/Reliability/Security	21
Organizational: Development Requirements	21
System Requirements	22
Functional Requirements	22
List of System Functional Requirements	22
System Behavior	30
Data Requirements	39
Non-Functional Requirements	39
Product: Performance Requirements	39
Product: Dependability/Reliability/Security	40
Organizational: Development Requirements	40
Requirements Trace Table	41
Exploratory Studies	42
Relevant Techniques	42
Relevant Packages/Products	43
Broader Impacts	44
System Design	45
Architectural Design	45
System Implementation	46

Programming Languages & Tools	46
Coding Conventions	46
Code Version Control	46
System Testing	47
Challenges & Open Issues	48
Challenges Faced in Requirements Engineering	48
System Manuals	49
Instructions for System Development	49
How to setup development environment	49
Notes on system further extension	49
Conclusion	50
References	51

1. Abstract

Watson uses “natural language processing and machine learning” to reveal insights from large amounts of unstructured data^[1]. We will use Watson to create a structured analysis of possible academic careers related to the CSSE majors at Penn State Behrend.

We seek to enable students to ask questions relating to these majors and to provide valuable feedback, promoting better decision making about academic and professional careers. Our tools will also assist advisors in preparing relevant and unique advice to each student seeking their guidance.

In this report, we arrange our goals as user and system requirements, showing the engineering process of this project. UML Diagrams are also provided for further detail and explanation of this process.

2. Report Revision History

2.1. Changes in Version 1.5

- Reformatted the document to look nicer
- Removed voice-to-text use cases, requirements, and sequence diagrams
- Modified Ask Question sequence diagram to not include voice-to-text reference
- Modified descriptions of all images
- Added more definitions to glossary
- Reworded the Abstract and Problem Statements
- Added more examples to the Exploratory Studies Techniques
- Added in more references
- Created in-text citations
- Added in picture for architectural design

3. Problem Statement

3.1. Business Background

IBM Watson's services, provided by Bluemix's APIs and other services from 3rd parties or developers, can be utilized to conduct textual analysis and output a numerical scale of performance factor. Web Experience Management (WEM) can be trained to answer many open-ended questions. The question we are trying to answer is whether we can create a system to assist with the enrollment process, through use of Bluemix and WEM.

3.2. Needs

To increase the effectiveness of the IBM Watson services, a larger domain is needed. Additionally, students are often unsure of which fields would coincide with their interests and talents.

3.3. Objectives

The objective of this project is to increase Watson's domain to include Penn State Behrend's academic information related to the CSSE majors, such as recommended courses, FAQs, and advisor information. The project will enable students to make a better decision as to which careers they might be interested in pursuing and what each path would entail. It will also help advisors to accurately guide the students.

4. Requirements

4.1. User Requirements

Glossary of Relevant Domain Technology

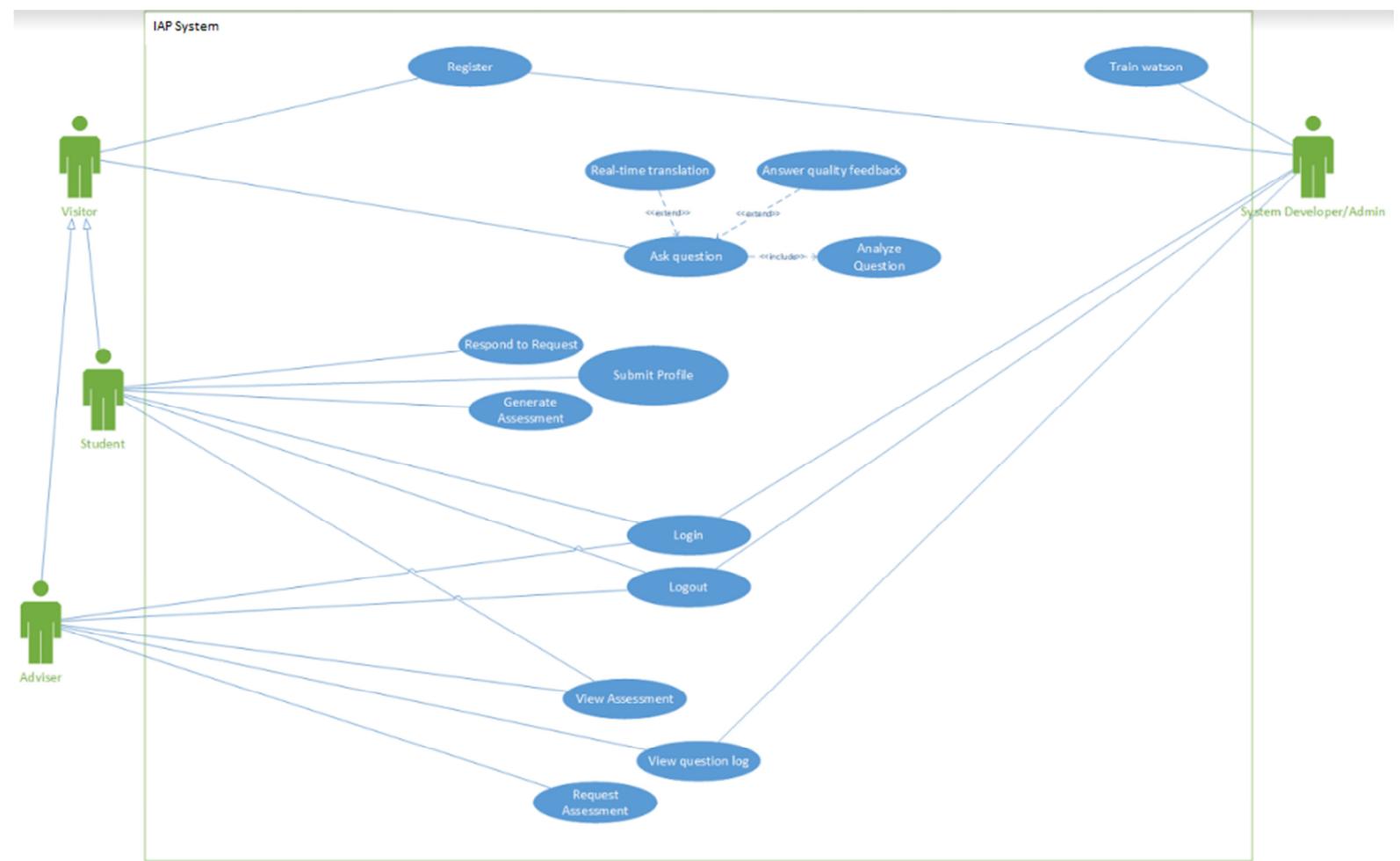
- **Watson** - An IBM supercomputer that combines artificial intelligence (AI) and sophisticated analytical software for optimal performance as a “question answering” machine.
- **Big Data Analysis** - The process of examining large datasets to uncover hidden patterns, unknown correlations, customer preferences
- **Textual Analysis** - A research method that requires the researcher to closely analyze the content of communication rather than the structure of the content.
- **Web Experience Management** - A process of managing the all-round experience of the web user across various touch points in the journey through an organization's web presence.
- **Use Case Diagram** - A representation of all of the functionalities the system is expected to have and what functionalities a specific user has access to.
- **Use Case** - Communication between a user and the system to perform a specific functionality that is represented in the Use Case Diagram.
- **Sequence Diagram** - A diagram that explains the expected flow of the system once the functionality has been implemented.
- **Data Crawling** - In this context, data crawling refers to the collection of specific data from our own resources, such as our database.
- **Natural Language Processing** - The field of study concerned with the interactions between computers and natural human languages.
- **Machine Learning** - A branch of artificial intelligence in which a computer generates rules underlying or based on raw data that has been fed into it.
- **Supervised Learning** - The machine learning task of inferring a function from labeled training data.
- **Unsupervised Learning** - A type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses.

User Groups

- Visitors
- Students
- Advisors
- System Developers

Functional Requirements

Project Scope



This is the Use Case Diagram of our system (Intelligent Academic Planner (IAP) System). Student and Advisor share the options that Visitor has (represented by the arrows). Each blue bubble represents a functionality that is present in the system, and will be gone into detail in each use case. The functionality of Real-time translation and Answer quality feedback are sometimes used in the Ask question functionality, and Analyze question is always used. This is represented by the “extend” arrows for situational functionality, and “include” arrows for full-time functionality done by the system.

User Scenarios

Project Name:	Intelligent Academic Planner								
Use Case ID:	UC-001								
Use Case Name:	Respond to Request								
User Goal:	User approves of advisor viewing assessment								
Scope:	IAP System								
Level:	Primary task								
Relevant User Reqs:	UF-E								
Relevant System Reqs:	SF-E-05								
Primary Actor:	Student								
Precondition:	User has a request awaiting approval								
Minimal Guarantee:	User's assessment are non-view-able								
Success Guarantee:	User's assessment becomes view-able for advisor that requested								
Trigger:	User requests to respond to request								
Success Scenario:	<table> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The user requests to respond to request</td> </tr> <tr> <td>2</td> <td>The system asks for user's response</td> </tr> <tr> <td>3</td> <td>The user responds</td> </tr> </tbody> </table>	Step	Actions	1	The user requests to respond to request	2	The system asks for user's response	3	The user responds
Step	Actions								
1	The user requests to respond to request								
2	The system asks for user's response								
3	The user responds								
Extensions:	Branching Scenarios								
3A	<p>Condition: If request is declined</p> <table> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The system sends notification to advisor</td> </tr> <tr> <td>2</td> <td>Exit out of functionality</td> </tr> </tbody> </table>	Step	Actions	1	The system sends notification to advisor	2	Exit out of functionality		
Step	Actions								
1	The system sends notification to advisor								
2	Exit out of functionality								
3B	<p>Condition: If request is accepted</p> <table> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The system sends notification to advisor</td> </tr> <tr> <td>2</td> <td>The system allows advisor to view assessment of student</td> </tr> </tbody> </table>	Step	Actions	1	The system sends notification to advisor	2	The system allows advisor to view assessment of student		
Step	Actions								
1	The system sends notification to advisor								
2	The system allows advisor to view assessment of student								

Acknowledgment: Generated from the CapStone process management system ©2015

UC-001: This use case explains the communication between the user and system to perform the functionality of responding to a request. Only students have access to this functionality, and it is performed when a student receives a request from an Advisor. This allows the user to either accept or deny an advisor's request for their assessment.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-004
Use Case Name:	Register
User Goal:	To be recognized by the system.
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-F
Relevant System Reqs:	SF-F-01
Primary Actor:	Visitor, System Devleoper
Precondition:	User is viewing program
Minimal Guarantee:	User is not recognized by system
Success Guarantee:	User is recognized by system
Trigger:	User requests to register
Success Scenario:	Step Actions
	1 The user requests to register
	2 The system asks for registration information
	3 The user inputs registration information
	4 The system validates information
	5 The system accepts user registration
Extensions:	Branching Scenarios
4A	Condition: Information is invalid
	Step Actions
	1 The system notifies user of invalid information
	2 Return to step 2

Acknowledgment: Generated from the CapStone process management system ©2015

UC-004: This use case explains the communication between the user and system to perform the functionality of registering. Anyone that accesses the program has access to this functionality. It is triggered when the user requests to register. This allows the user to login.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-005
Use Case Name:	View assessment
User Goal:	User is able to view assessment
Scope:	IAP System
Level:	Subfunction
Relevant User Reqs:	UF-E
Relevant System Reqs:	SF-E-03
Primary Actor:	student, advisor
Precondition:	The user is logged in
Minimal Guarantee:	System does not display student's assessment
Success Guarantee:	system display student's assessment
Trigger:	User requests to view assessments
Success Scenario:	Step Actions
	1 The user requests to view assessments
	2 The system displays assessments that can be viewed
Extensions:	Branching Scenarios

Acknowledgment: Generated from the CapStone process management system ©2015

UC-005: This use case explains the communication between the user and the system to perform the functionality of viewing an assessment. This function is only available to logged in users, specifically students and advisors, and is performed when the user requests to view an assessment on their profile. This allows the user to view an assessment that they have generated.

Project Name:	Intelligent Academic Planner										
Use Case ID:	UC-006										
Use Case Name:	View question log										
User Goal:	User is able to view the question log										
Scope:	IAP System										
Level:	Primary task										
Relevant User Reqs:	UF-G										
Relevant System Reqs:	SF-G-01										
Primary Actor:	Adviser, System Developer										
Precondition:	User is logged in										
Minimal Guarantee:	System does not display question log										
Success Guarantee:	System displays question log										
Trigger:	User requests to view question log										
Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The user requests to view question log</td> </tr> <tr> <td>2</td> <td>The system asks for filter information</td> </tr> <tr> <td>3</td> <td>The user enters filter information</td> </tr> <tr> <td>4</td> <td>The system displays all questions based on filter information</td> </tr> </tbody> </table>	Step	Actions	1	The user requests to view question log	2	The system asks for filter information	3	The user enters filter information	4	The system displays all questions based on filter information
Step	Actions										
1	The user requests to view question log										
2	The system asks for filter information										
3	The user enters filter information										
4	The system displays all questions based on filter information										
Extensions:	Branching Scenarios										

Acknowledgment: Generated from the CapStone process management system ©2015

UC-006: This use case explains the communication between user and system to perform the functionality of viewing the question log. This functionality is only available to advisors and system developers. It is triggered when the user requests to view the question log. This allows the user to view a list of question that have been asked.

Project Name:	Intelligent Academic Planner																										
Use Case ID:	UC-007																										
Use Case Name:	Train Watson																										
User Goal:	User be able to train Watson with new questions that user asked but hasn't been answered																										
Scope:	IAP System																										
Level:	Primary task																										
Relevant User Reqs:	UF-H																										
Relevant System Reqs:	SF-A-01																										
Primary Actor:	System Developer																										
Precondition:	User is logged in																										
Minimal Guarantee:	Watson is not further trained																										
Success Guarantee:	Watson is further trained																										
Trigger:	User requests to train Watson																										
Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The user request system export new questions</td> </tr> <tr> <td>2</td> <td>The system export new question list as a text file</td> </tr> <tr> <td>3</td> <td>The user request system export incorrect answered questions</td> </tr> <tr> <td>4</td> <td>The system export downvotes answered questions</td> </tr> <tr> <td>5</td> <td>The user request view answer quality</td> </tr> <tr> <td>6</td> <td>The system display the statistical analysis of the answer for each of the questions</td> </tr> <tr> <td>7</td> <td>The user request view the feedback of answer from user</td> </tr> <tr> <td>8</td> <td>The system display the most up voted help feedback for the answer</td> </tr> <tr> <td>9</td> <td>The user request the analysis of a specific question</td> </tr> <tr> <td>10</td> <td>The system display analysis of a question by keyterm</td> </tr> </tbody> </table>	Step	Actions	1	The user request system export new questions	2	The system export new question list as a text file	3	The user request system export incorrect answered questions	4	The system export downvotes answered questions	5	The user request view answer quality	6	The system display the statistical analysis of the answer for each of the questions	7	The user request view the feedback of answer from user	8	The system display the most up voted help feedback for the answer	9	The user request the analysis of a specific question	10	The system display analysis of a question by keyterm				
Step	Actions																										
1	The user request system export new questions																										
2	The system export new question list as a text file																										
3	The user request system export incorrect answered questions																										
4	The system export downvotes answered questions																										
5	The user request view answer quality																										
6	The system display the statistical analysis of the answer for each of the questions																										
7	The user request view the feedback of answer from user																										
8	The system display the most up voted help feedback for the answer																										
9	The user request the analysis of a specific question																										
10	The system display analysis of a question by keyterm																										
Extensions:	<table border="1"> <thead> <tr> <th colspan="2">Branching Scenarios</th> </tr> </thead> <tbody> <tr> <td>1A</td> <td>Condition: User request system export all new questions from last time export</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The system export question list that is new from last time</td> </tr> </tbody> </table> </td></tr> <tr> <td>1B</td> <td>Condition: User request system export all new questions by major</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> </table> </td></tr> <tr> <td>1C</td> <td>Condition: User request system export all new questions other condition</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The system export top 10 unanswered FAQ</td> </tr> <tr> <td>2</td> <td>The system export top 10% unanswered FAQ</td> </tr> </tbody> </table> </td></tr> </tbody> </table>	Branching Scenarios		1A	Condition: User request system export all new questions from last time export		<table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The system export question list that is new from last time</td> </tr> </tbody> </table>	Step	Actions	1	The system export question list that is new from last time	1B	Condition: User request system export all new questions by major		<table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> </table>	Step	Actions	1C	Condition: User request system export all new questions other condition		<table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The system export top 10 unanswered FAQ</td> </tr> <tr> <td>2</td> <td>The system export top 10% unanswered FAQ</td> </tr> </tbody> </table>	Step	Actions	1	The system export top 10 unanswered FAQ	2	The system export top 10% unanswered FAQ
Branching Scenarios																											
1A	Condition: User request system export all new questions from last time export																										
	<table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The system export question list that is new from last time</td> </tr> </tbody> </table>	Step	Actions	1	The system export question list that is new from last time																						
Step	Actions																										
1	The system export question list that is new from last time																										
1B	Condition: User request system export all new questions by major																										
	<table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> </table>	Step	Actions																								
Step	Actions																										
1C	Condition: User request system export all new questions other condition																										
	<table border="1"> <thead> <tr> <th>Step</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The system export top 10 unanswered FAQ</td> </tr> <tr> <td>2</td> <td>The system export top 10% unanswered FAQ</td> </tr> </tbody> </table>	Step	Actions	1	The system export top 10 unanswered FAQ	2	The system export top 10% unanswered FAQ																				
Step	Actions																										
1	The system export top 10 unanswered FAQ																										
2	The system export top 10% unanswered FAQ																										

Acknowledgment: Generated from the CapStone process management system ©2015

UC-007: This use case explains the communication between user and system to perform the functionality of training Watson. This functionality is only available to the system developer and is performed when the user requests to train Watson. This allows the user to view data to assist with the training of Watson in the future.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-009
Use Case Name:	Real-time translation
User Goal:	User is able to see input/output in english
Scope:	IAP System
Level:	Subfunction
Relevant User Reqs:	UF-C
Relevant System Reqs:	SF-C-02
Primary Actor:	Visitor, Student, Advisor
Precondition:	User is asking a question
Minimal Guarantee:	Input/output is not translated
Success Guarantee:	System displays translated text
Trigger:	User requests translation
Success Scenario:	Step Actions
	1 The user input non-english input
	2 The system determines what language is being used
	3 The system translates text to english
	4 The system displays english answer
Extensions:	Branching Scenarios

Acknowledgment: Generated from the CapStone process management system ©2015

UC-009: This use case explains the communication between user and system to perform the functionality of real-time translation. This functionality is available to all users and is performed when the user requests a translation of their question. This allows the user to translate text from well-known languages to english. This is called occasionally when the user asks a question, and is present in the Ask Question use case.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-010
Use Case Name:	Answer quality feedback
User Goal:	User is able to provide feedback on question response.
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-H
Relevant System Reqs:	SF-H-01
Primary Actor:	Visitor, Student, Advisor
Precondition:	User is asking a question
Minimal Guarantee:	Feedback is not stored
Success Guarantee:	Feedback is stored
Trigger:	User requests to submit feedback
Success Scenario:	Step Actions
	1 The user requests to submit feedback
	2 The system allows user to enter feedback
	3 The user submits feedback
	4 The system stores feedback
Extensions:	Branching Scenarios

Acknowledgment: Generated from the CapStone process management system ©2015

UC-010: This use case explains the communication between user and system to perform the functionality of sending answer quality feedback. This functionality is available to all users and is performed when the user requests to submit feedback on their response. This allows the user to submit feedback on the response to their question. This is called occasionally when the user asks a question, and is present in the Ask Question use case.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-011
Use Case Name:	Ask question
User Goal:	User is able to ask questions
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-C
Relevant System Reqs:	SF-C-01,SF-C-02,SF-C-03,SF-C-04,SF-C-05
Primary Actor:	Visitor, Student, Advisor
Precondition:	User is viewing program
Minimal Guarantee:	Question is not logged
Success Guarantee:	Question is logged
Trigger:	The user requests to ask a question
Success Scenario:	Step Actions
	1 The user requests to ask a question
	2 The system requests user's question
	3 The user enters question and requests answer
	4 The system analyzes question <<Analyze Question>>
	5 The system displays answer
	6 The system logs question to question log
	7 The system asks user if they would like to submit feedback on answer quality
Extensions:	Branching Scenarios
3A	Condition: If user requests translation
7A	Step Actions
	1 The system translates text <<Realtime Translation>>
	Condition: If user requests to provide feedback
7A	Step Actions
	1 The system requests feedback information <<Answer Quality Feedback>>

Acknowledgment: Generated from the CapStone process management system ©2015

UC-011: This use case explains the communication between user and system to perform the functionality of asking a question. This functionality is available to all users and is performed when the user requests to ask a question. This allows the user to submit a question and receive a response. This use case occasionally uses the functionality of Real-time Translation and Answer Quality Feedback use cases, and always uses the functionality of the Analyze Question use case.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-012
Use Case Name:	Analyze Question
User Goal:	System determines answer for user
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-C,UF-D
Relevant System Reqs:	SF-C-01,SF-C-03,SF-C-04,SF-C-05,SF-D-01
Primary Actor:	N/A
Precondition:	User asks a question
Minimal Guarantee:	Question is not analyzed
Success Guarantee:	Question is analyzed
Trigger:	User submits a question
Success Scenario:	Step Actions
	1 The user submits a question
	2 The system performs textual analysis on the question
	3 The system determines if another question could be asked to clarify answer
	4 The system displays answer
Extensions:	Branching Scenarios
1A	Condition: If another question could be asked
1A	Step Actions
	1 The system asks user the question
	2 The user responds to question
	3 Return to step 2 in main scenario using response given

Acknowledgment: Generated from the CapStone process management system ©2015

UC-012: This use case explains the communication between user and system to perform the functionality of analyzing a question. This functionality is performed solely by the system. This allows the system to perform a textual analysis on the question that has been asked. This use case is always used in the Ask Question use case.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-013
Use Case Name:	Submit Profile
User Goal:	User can create profile
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-E
Relevant System Reqs:	SF-E-01,SF-E-02,SF-E-03,SF-E-04,SF-E-05
Primary Actor:	student
Precondition:	User is logged in
Minimal Guarantee:	Profile is not stored
Success Guarantee:	Profile is stored
Trigger:	User requests to update profile
Success Scenario:	<p>Step Actions</p> <p>1 The user requests to update profile 2 The system verify the profile 3 The system upload the profile to server</p>
Extensions:	<p>Branching Scenarios</p> <p>2A Condition: If system detect invalid format of the profile</p> <p>Step Actions</p> <p>1 The system notify user of problem 2 Return to step 1</p>

Acknowledgment: Generated from the CapStone process management system ©2015

UC-013: This use case explains the communication between user and system to perform the functionality of submitting a profile. This functionality is available only to students and is performed when the user requests to update their profile. This allows the user to update their profile information.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-015
Use Case Name:	Generate Assessment
User Goal:	User receives assessment of profile information
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-E
Relevant System Reqs:	SF-E-04,SF-E-05
Primary Actor:	Student
Precondition:	User has submitted information to profile
Minimal Guarantee:	No assessment generated
Success Guarantee:	System generate assessment and display to user
Trigger:	User requests to generate assessment
Success Scenario:	<p>Step Actions</p> <p>1 The user requests to generate assessment 2 The system analyzes profile and generates assessment 3 The system displays assessment</p>
Extensions:	<p>Branching Scenarios</p> <p>3A Condition: keep assessment on user's record</p> <p>Step Actions</p> <p>1 The user request system save the assessment 2 The system upload assessment to server and relate to user account</p> <p>3B Condition: The user request assessment to be viewable by advisers</p> <p>Step Actions</p> <p>1 The user request assessment to be viewable by advisers 2 The system mark user's assessment as public and save to user's account</p>

Acknowledgment: Generated from the CapStone process management system ©2015

UC-015: This use case explains the communication between user and system to perform the functionality of generating an assessment. This functionality is available only to students and is performed when the user requests to generate an assessment. This allows the user to generate an assessment.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-016
Use Case Name:	Login
User Goal:	User is able to log in
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-A
Relevant System Reqs:	SF-A-01
Primary Actor:	Student, Advisor, System Devel
Precondition:	User is registered
Minimal Guarantee:	User is not logged in
Success Guarantee:	User is logged in
Trigger:	User requests to log in
	Step Actions
Success Scenario:	1 The user request to login to the system
	2 The system verifies user's login credential
	3 The system logs in user
Extensions:	Branching Scenarios
2A	Condition: login credential doesn't match account info
	Step Actions
	1 The system notifies user of problem
	2 Return to step 1
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>	

UC-016: This use case explains the communication between user and system to perform the functionality of logging in. This functionality is available only to registered users and is performed when the user requests to login. This allows the user to log in to their own session and view their personal information.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-018
Use Case Name:	Logout
User Goal:	User is able to log out
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-B
Relevant System Reqs:	SF-B-01
Primary Actor:	Student, Advisor, System Devel
Precondition:	User is logged in
Minimal Guarantee:	User is not logged out
Success Guarantee:	User is logged out
Trigger:	User requests to log out
	Step Actions
Success Scenario:	1 The user requests to logout
	2 The system verifies all information is saved
	3 The system logs users out
Extensions:	Branching Scenarios
2A	Condition: If some information is unsaved
	Step Actions
	1 The system checks if user still wants to log out
	2 The user responds
	3 BRANCH - If user responds no exit functionality. Else - Continue on.
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>	

UC-018: This use case explains the communication between user and system to perform the functionality of logging out. This functionality is available only to logged in users and is performed when the user requests to log out. This logs the user out of their personal session.

Project Name:	Intelligent Academic Planner
Use Case ID:	UC-019
Use Case Name:	Request Assessment
User Goal:	System notify student that an adviser wants to see his assessment
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-E
Relevant System Reqs:	SF-E-05
Primary Actor:	Advisor
Precondition:	User is logged in
Minimal Guarantee:	No request is sent
Success Guarantee:	Request is sent
Trigger:	User requests to send request to student
Success Scenario:	Step Actions
	1 The user requests to send request to student
	2 The system asks for student information
	3 The user inputs student information
	4 The system validates student information
	5 The system sends request to student
Extensions:	Branching Scenarios
4A	Condition: If invalid student information
	Step Actions
	1 The system notifies user of problem
	2 Return to step 2

Acknowledgment: Generated from the CapStone process management system ©2015

UC-019: This use case explains the communication between user and system to perform the functionality of requesting an assessment. This functionality is available only to advisors and is performed when the user requests to ask a student for their assessment. This allows the user to send a request to a student for their assessment.

List of User Functional Requirements

Project Name:	Intelligent Academic Planner				
Requirement ID:	UF-A	Type	Functional	Non-Functional	
Creation:	Sep 20 2016 01:25 PM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:50 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user can log in.				
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Refined Into:	SF-A-01				
Justify why UF-A can be completely covered by SF-A-01	If a user can login within 5 seconds, they can log in properly.				
Traceability:	Use cases cf.	UC-016			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

UF-A: This user requirement requests that we include functionality in the system for users to log in. High priority was given to this requirement since making individual sessions is required to begin work on the profile.

Project Name:	Intelligent Academic Planner				
Requirement ID:	UF-B	Type	Functional	Non-Functional	
Creation:	Sep 20 2016 04:17 PM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:50 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user can log out.				
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Refined Into:	SF-B-01				
Justify why UF-B can be completely covered by SF-B-01	If a user can log out within 5 seconds, they can log out properly				
Traceability:	Use cases cf.	UC-018			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

UF-B: This user requirement requests that we include functionality in the system for users to log out. High priority was given to this requirement since a user should be able to log out if they can login, and log in has High priority.

Project Name:	Intelligent Academic Planner				
Requirement ID:	UF-C	Type	Functional	Non-Functional	
Creation:	Sep 20 2016 04:18 PM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Modification:	Sep 20 2016 04:58 PM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user can ask the system questions.				
Priority:	✓ Highest	High	Medium	Low	Lowest
This Req. is Refined Into:	SF-C-01, SF-C-02, SF-C-03, SF-C-04, SF-C-05, SF-C-07				
Justify why UF-C can be completely covered by SF-C-01, SF-C-02, SF-C-03, SF-C-04, SF-C-05, SF-C-07	If the system is able to perform the functionality in SF-C-01 through SF-C-06 then a user will undoubtedly have been able to ask the system questions. The system would not be able to perform these functions without a user first asking it a question.				
Traceability:	Use cases cf.	UC-008, UC-009, UC-011, UC-012			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

UF-C: This user requirement requests that we include functionality in the system for users to ask the system questions. Highest priority was given to this requirement since our first priority is to allow Watson to answer questions both accurately and uniquely. In addition, since any user can ask a question, logging in and registering is not a requirement to begin work on this.

Project Name:	Intelligent Academic Planner				
Requirement ID:	UF-D	Type	Functional	Non-Functional	
Creation:	Sep 23 2016 12:52 PM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:51 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user should receive multiple responses to a question.				
Priority:	Highest	High	Medium	✓ Low	Lowest
This Req. is Refined Into:	SF-D-01				
Justify why UF-D can be completely covered by SF-D-01	By requiring a minimum of 1 response to be given, it is given that multiple responses are given to a question.				
Traceability:	Use cases cf.	UC-012			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

UF-D: This user requirement requests that we include functionality in the system for users to receive multiple responses to a question. This means that if the user asks a question that does not have a clear answer, Watson should ask a question to clarify what is being asked to make the answer clearer. Low priority was given to this requirement since it requires we first implement asking a question, which was given highest priority, and in some cases the second question may require personal information to ask, requiring the profile to be complete which is medium priority.

Project Name:	Intelligent Academic Planner				
Requirement ID:	UF-E	Type	Functional	Non-Functional	
Creation:	Oct 04 2016 11:51 PM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 12:06 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user can create a profile				
Priority:	Highest	High	✓ Medium	Low	Lowest
This Req. is Refined Into:	SF-E-01, SF-E-02, SF-E-03, SF-E-04, SF-E-05				
Justify why UF-E can be completely covered by SF-E-01, SF-E-02, SF-E-03, SF-E-04, SF-E-05	All of the system requirements associated with this cover what can be put into their profile.				
Traceability:	Use cases cf.	UC-001, UC-005, UC-013, UC-015, UC-019			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

UF-E: This user requirement requests that we include functionality in the system for users to create a profile. Medium priority was given to this requirement since it requires we first implement registering, logging in, and logging out (all High priority) before we can set up user-specific profiles.

Project Name: Intelligent Academic Planner					
Requirement ID:	UF-F		Type	Functional	Non-Functional
Creation:	Oct 05 2016 12:15 AM		User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modification:	Oct 05 2016 01:51 AM		System	<input type="checkbox"/>	<input type="checkbox"/>
Description:	A user can register				
Priority:	Highest	<input checked="" type="checkbox"/> High	Medium	Low	Lowest
This Req. is Refined Into:	SF-F-01				
Justify why UF-F can be completely covered by SF-F-01	If a user can register within 5 seconds, they can register properly.				
Traceability:	Use cases cf.	UC-004			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

UF-F: This user requirement requests that we include functionality in the system for users to register. High priority was given to this requirement since making individual sessions is required to begin work on the profile.

Project Name: Intelligent Academic Planner					
Requirement ID:	UF-G		Type	Functional	Non-Functional
Creation:	Oct 05 2016 12:36 AM		User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modification:	Oct 05 2016 01:52 AM		System	<input type="checkbox"/>	<input type="checkbox"/>
Description:	A user can view a log of asked questions.				
Priority:	Highest	<input checked="" type="checkbox"/> High	<input checked="" type="checkbox"/> Medium	Low	Lowest
This Req. is Refined Into:	SF-G-01				
Justify why UF-G can be completely covered by SF-G-01	If advisors and system developers can view the question log, then users can view the question log.				
Traceability:	Use cases cf.	UC-006			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

UF-G: This user requirement requests that we include functionality in the system for users to view a log of asked questions. Medium priority was given to this requirement because it can be easily implemented after completing the ask a question requirement, and because it will assist with increasing the accuracy of Watson.

Project Name:	Intelligent Academic Planner				
Requirement ID:	UF-H	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 02:12 AM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 02:16 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user can provide information to improve accuracy of the system.				
Priority:	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
This Req. is Refined Into:	SF-H-01				
Justify why UF-H can be completely covered by SF-H-01	If answer quality feedback is submitted, accuracy of responses can be increased.				
Traceability:	Use cases cf.	UC-007, UC-010			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

UF-H: This user requirement requests that we include functionality in the system for users to improve the accuracy of the system by providing feedback. Medium priority was given to this requirement because it can be easily implemented after completing the ask a question requirement, and because it will assist with increasing the accuracy of Watson.

Non-Functional Requirements

Product: Performance Requirements

Project Name:	Intelligent Academic Planner				
Requirement ID:	UP-03	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 01:36 AM	User	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Modification:	Oct 05 2016 01:54 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user should receive a quick response after asking a question		Product (sub-type below)		
			Performance Requirements		
Priority:	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
This Req. is Refined Into:	SP-03-01				
Justify why UP-03 can be completely covered by SP-03-01	By specifying performance requirements, it is ensured that the question will be answered in a quick manner.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

UP-03: This user requirement requests that when we create the functionality for asking a question, the system should respond quickly. Low priority was given to this requirement because we are focusing first on implementing features and then focusing on quality and performance of the features.

Product: Dependability/Reliability/Security

Project Name:	Intelligent Academic Planner				
Requirement ID:	UP-01	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 12:10 AM	User	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Modification:	Oct 05 2016 01:53 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user's profile should be secure.				
	Product (sub-type below)				
	Dependability/Reliability/Security				
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Refined Into:	SP-01-01				
Justify why UP-01 can be completely covered by SP-01-01	Ensures only specific people can view a user's profile, making it secure.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

UP-01: This user requirement requests that when we create the functionality for creating a profile, the system should ensure that the profile is secure. High priority was given to this requirement because it should be done while creating the functionality of the profile and we were encouraged to keep security in mind.

Organizational: Development Requirements

Project Name:	Intelligent Academic Planner				
Requirement ID:	UO-01	Type	Functional	Non-Functional	
Creation:	Sep 23 2016 12:42 PM	User	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Modification:	Oct 05 2016 01:53 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user's session should be managed.				
	Organizational (sub-type below)				
	Development Requirements				
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Refined Into:	SO-01-01				
Justify why UO-01 can be completely covered by SO-01-01	Ensures that a user can only be logged in for 1 hour, managing their session.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

UO-01: This user requirement requests that when we create the functionality for registering, logging in, and logging out, the system should ensure that the session is managed. High priority was given to this requirement because it should be done while creating the functionality of logging in, logging out, and registering.

4.2. System Requirements

Functional Requirements

List of System Functional Requirements

Project Name:	Intelligent Academic Planner					
Requirement ID:	SF-A-01	Type	Functional	Non-Functional		
Creation:	Sep 23 2016 12:22 PM	User	<input type="checkbox"/>	<input type="checkbox"/>		
Modification:	Oct 05 2016 01:41 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Description:	The system should log-in a user within 5 seconds.					
Priority:	Highest	High	Medium	✓ Low	Lowest	
This Req. is Engineered From:	UF-A					
Justify why meeting SF-A-01 can contribute to the fulfilment of UF-A	Provides performance requirement for logging in.					
Traceability:	Use cases cf.	UC-007, UC-016				
	Test cases cf.	Yet to be completed in test case worksheet!				
Acknowledgment	Generated from the CapStone Process Management System ©2015					

SF-A-01: This system requirement requests that when we create the functionality for logging in, the system should log the user in within 5 seconds. Low priority was given to this requirement because we are focusing first on implementing features and then focusing on quality and performance of the features.

Project Name:	Intelligent Academic Planner					
Requirement ID:	SF-B-01	Type	Functional	Non-Functional		
Creation:	Sep 23 2016 12:32 PM	User	<input type="checkbox"/>	<input type="checkbox"/>		
Modification:	Oct 05 2016 01:41 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Description:	The system should log-out a user within 5 seconds.					
Priority:	Highest	High	Medium	✓ Low	Lowest	
This Req. is Engineered From:	UF-B					
Justify why meeting SF-B-01 can contribute to the fulfilment of UF-B	Explains performance requirement.					
Traceability:	Use cases cf.	UC-018				
	Test cases cf.	Yet to be completed in test case worksheet!				
Acknowledgment	Generated from the CapStone Process Management System ©2015					

SF-B-01: This system requirement requests that when we create the functionality for logging out, the system should log the user out within 5 seconds. Low priority was given to this requirement because we are focusing first on implementing features and then focusing on quality and performance of the features.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-C-01	Type	Functional	Non-Functional	
Creation:	Sep 20 2016 04:22 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:42 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should conduct textual analyses.				
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Engineered From:	UF-C				
Justify why meeting SF-C-01 can contribute to the fulfilment of UF-C	In order to provide an answer to questions asked, system must be able to perform this function.				
Traceability:	Use cases cf.	UC-011, UC-012			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

SF-C-01: This system requirement requests that when we create the functionality for asking a question, the system should conduct a textual analysis. High priority was given to this requirement because this is one of the key requirements we are focusing on.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-C-02	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 01:59 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 02:00 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should be able to handle input from multiple well-known languages.				
Priority:	Highest	High	Medium	Low	✓ Lowest
This Req. is Engineered From:	UF-C				
Justify why meeting SF-C-02 can contribute to the fulfilment of UF-C	Allows users to ask a question in a variety of ways.				
Traceability:	Use cases cf.	UC-009, UC-011			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

SF-C-02: This system requirement requests that when we create the functionality for asking a question, the system should be able to handle input from multiple well known languages. Lowest priority was given to this requirement because it is a bonus feature that we only intend to implement if time provides.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-C-03	Type	Functional	Non-Functional	
Creation:	Sep 20 2016 04:31 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:44 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should recommend majors suitable for the user based on the personality assessment.				
Priority:	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
This Req. is Engineered From:	UF-C				
Justify why meeting SF-C-03 can contribute to the fulfilment of UF-C	This allows questions to be answered more accurately.				
Traceability:	Use cases cf.	UC-011, UC-012			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

SF-C-03: This system requirement requests that when we create the functionality for asking a question, the system should be able to recommend majors suitable for the user based on their profile information. Medium priority was given to this requirement because it requires the profile to be complete.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-C-04	Type	Functional	Non-Functional	
Creation:	Sep 20 2016 04:34 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:44 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should gather data unique to each user.				
Priority:	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
This Req. is Engineered From:	UF-C				
Justify why meeting SF-C-04 can contribute to the fulfilment of UF-C	This allows questions to be answered more accurately.				
Traceability:	Use cases cf.	UC-011, UC-012			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

SF-C-04: This system requirement requests that when we create the functionality for asking a question, the system should be able to gather data unique to each user. Medium priority was given to this requirement because it requires register, logging in, and logging out to be complete.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-C-05	Type	Functional	Non-Functional	
Creation:	Sep 20 2016 04:35 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:44 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should recommend courses based on the recommended majors.				
Priority:	Highest	High	✓ Medium	Low	Lowest
This Req. is Engineered From:	UF-C				
Justify why meeting SF-C-05 can contribute to the fulfilment of UF-C	This allows questions to be answered more accurately.				
Traceability:	Use cases cf.	UC-011, UC-012			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

SF-C-05: This system requirement requests that when we create the functionality for asking a question, the system should be able to recommend courses based on their recommended major. Medium priority was given to this requirement because it requires the profile to be complete and the system should already be able to recommend majors.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-D-01	Type	Functional	Non-Functional	
Creation:	Sep 23 2016 12:54 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:46 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should show a minimum of 1 related search/question.				
Priority:	Highest	High	Medium	✓ Low	Lowest
This Req. is Engineered From:	UF-D				
Justify why meeting SF-D-01 can contribute to the fulfilment of UF-D	This allows responses to be structured and more accurate.				
Traceability:	Use cases cf.	UC-012			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

SF-D-01: This system requirement requests that when we create the functionality for responding with a question, the system should ask at least one question in response. Low priority was given to this requirement because creating the functionality for responding with a question is also Low priority.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-E-01	Type	Functional	Non-Functional	
Creation:	Oct 04 2016 11:56 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:46 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should allow between 100 and 600 words to describe a user's academic and professional interests.				
Priority:	<input checked="" type="checkbox"/> Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	UF-E				
Justify why meeting SF-E-01 can contribute to the fulfilment of UF-E	Allows user to enter information about themselves to their profile				
Traceability:	Use cases cf.	UC-013			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

SF-E-01: This system requirement requests that when we create the functionality for creating a profile, the system should allow the user to enter between 100 and 600 words of academic and professional interests. Highest priority was given to this requirement because this is what will be to determine questions and answers related to the user.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-E-02	Type	Functional	Non-Functional	
Creation:	Oct 04 2016 11:56 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:47 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should allow a user to submit 100 words of self-description about their personality.				
Priority:	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
This Req. is Engineered From:	UF-E				
Justify why meeting SF-E-02 can contribute to the fulfilment of UF-E	Allows user to enter personality information on their profile				
Traceability:	Use cases cf.	UC-013			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

SF-E-02: This system requirement requests that when we create the functionality for creating a profile, the system should allow the user to enter 100 words of self-description. Medium priority was given to this requirement because this will be used to determine question and answers, but will be taken into account after professional and academic interests.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-E-03	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 12:00 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:47 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should allow a user to view their personality assessments.				
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Engineered From:	UF-E				
Justify why meeting SF-E-03 can contribute to the fulfilment of UF-E	Allows a user to learn about themselves based on profile information.				
Traceability:	Use cases cf.	UC-005, UC-013			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

SF-E-03: This system requirement requests that when we create the functionality for creating a profile, the system should allow the user to view personality assessments. High priority was given to this requirement because this is one of the main resources advisors will use.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-E-04	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 01:43 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:48 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should create a personality assessment unique to each user based on the data gathered.				
Priority:	Highest	High	✓ Medium	Low	Lowest
This Req. is Engineered From:	UF-E				
Justify why meeting SF-E-04 can contribute to the fulfilment of UF-E	Allows user to view information about themselves on their profile that they did not input.				
Traceability:	Use cases cf.	UC-013, UC-015			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

SF-E-04: This system requirement requests that when we create the functionality for creating a profile, the system should allow the user to generate personality assessments. Medium priority was given to this requirement because it requires the 100 words of self-description to be complete.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-E-05	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 01:45 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:45 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should summarize this data to be used by an advisor directing the student.				
Priority:	Highest	High	✓ Medium	Low	Lowest
This Req. is Engineered From:	UF-E				
Justify why meeting SF-E-05 can contribute to the fulfilment of UF-E	Allows user to get assistance from advisors based on their profile.				
Traceability:	Use cases cf.	UC-001, UC-013, UC-015, UC-019			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

SF-E-05: This system requirement requests that when we create the functionality for creating a profile, the system should summarize data for advisors. Medium priority was given to this requirement because it requires the profile to be fully complete before a summarization can be created.

Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-F-01	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 12:16 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:49 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should register the user within 5 seconds.				
Priority:	Highest	High	✓ Medium	Low	Lowest
This Req. is Engineered From:	UF-F				
Justify why meeting SF-F-01 can contribute to the fulfilment of UF-F	Places a performance requirement on registration.				
Traceability:	Use cases cf.	UC-004			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

SF-F-01: This system requirement requests that when we create the functionality for registering, the system should register the user within 5 seconds. Medium priority was given to this requirement because we are focusing first on implementing features and then focusing on quality and performance of the features but this will be one of the first features users encounter, so it has a slightly higher priority than login and logout performance requirements.

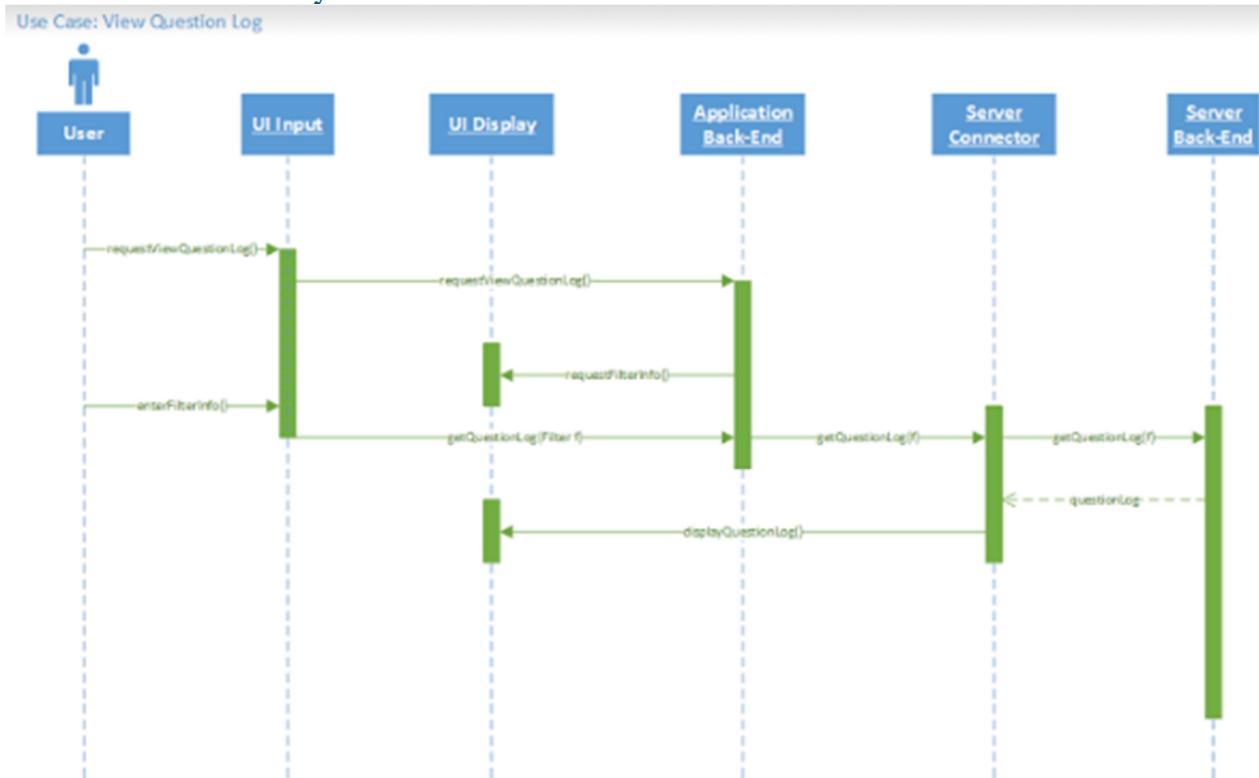
Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-G-01	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 12:37 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 12:37 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should only allow Advisors and System Developers to view the question log.				
Priority:	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
This Req. is Engineered From:	UF-G				
Justify why meeting SF-G-01 can contribute to the fulfilment of UF-G	Adds security to the question log.				
Traceability:	Use cases cf.	UC-006			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

SF-G-01: This system requirement requests that when we create the functionality for viewing the question log, the system should only allow advisors and system developers to view it. Medium priority was given to this requirement because the functionality of the question log is also medium priority.

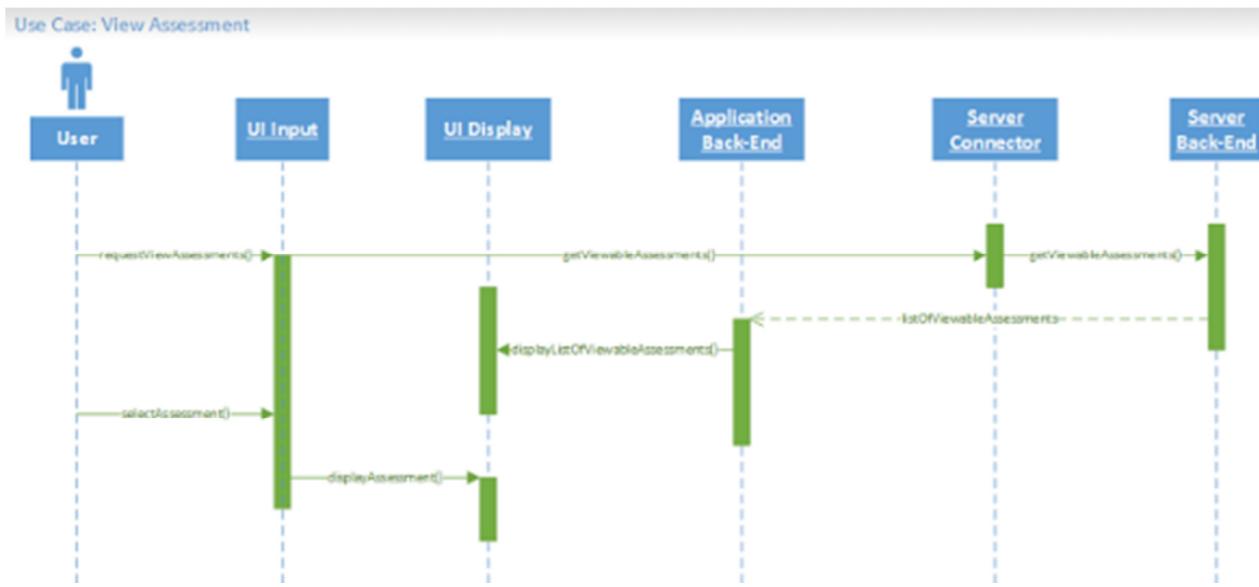
Project Name:	Intelligent Academic Planner				
Requirement ID:	SF-H-01	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 02:15 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 02:15 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	A user can provide answer quality feedback after asking a question.				
Priority:	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
This Req. is Engineered From:	UF-H				
Justify why meeting SF-H-01 can contribute to the fulfilment of UF-H	Allows feedback to be submitted.				
Traceability:	Use cases cf.	UC-010			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

SF-H-01: This system requirement requests that when we create the functionality for providing feedback, the system should ask for the feedback after asking a question. Medium priority was given to this requirement because the functionality of providing feedback is also medium priority.

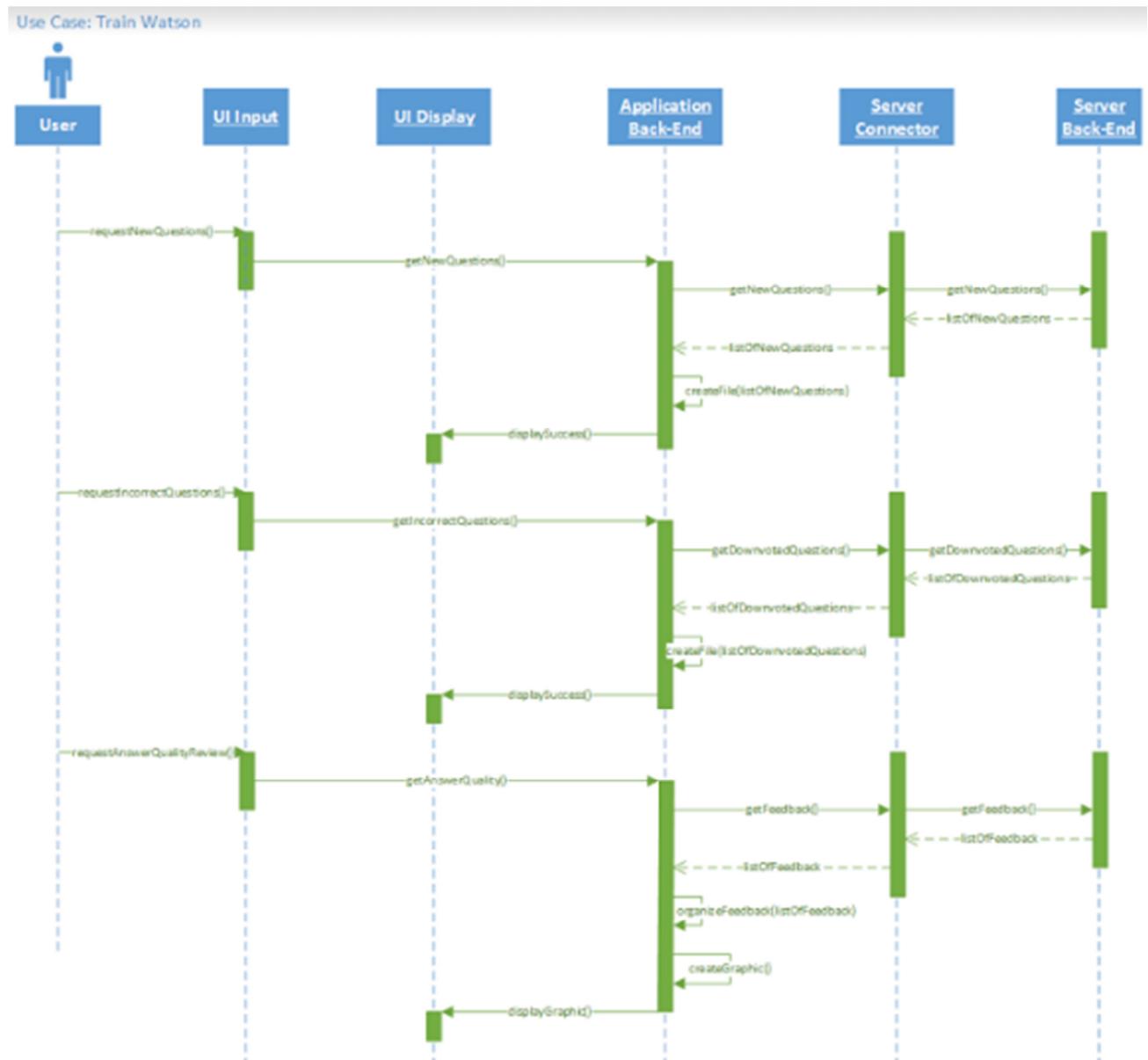
System Behavior

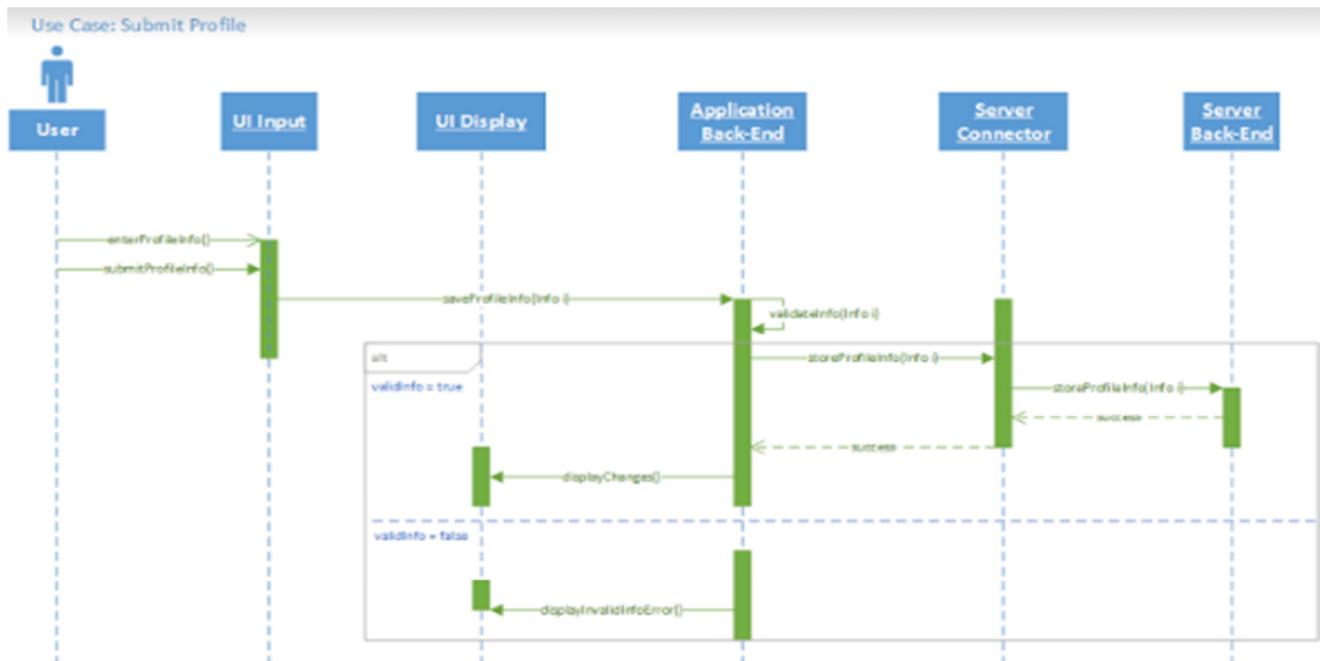


View Question Log Sequence Diagram: After requesting to view the question log, the system will ask for filter info and then display the question log for viewing to the user after receiving the questions from the server's back-end.

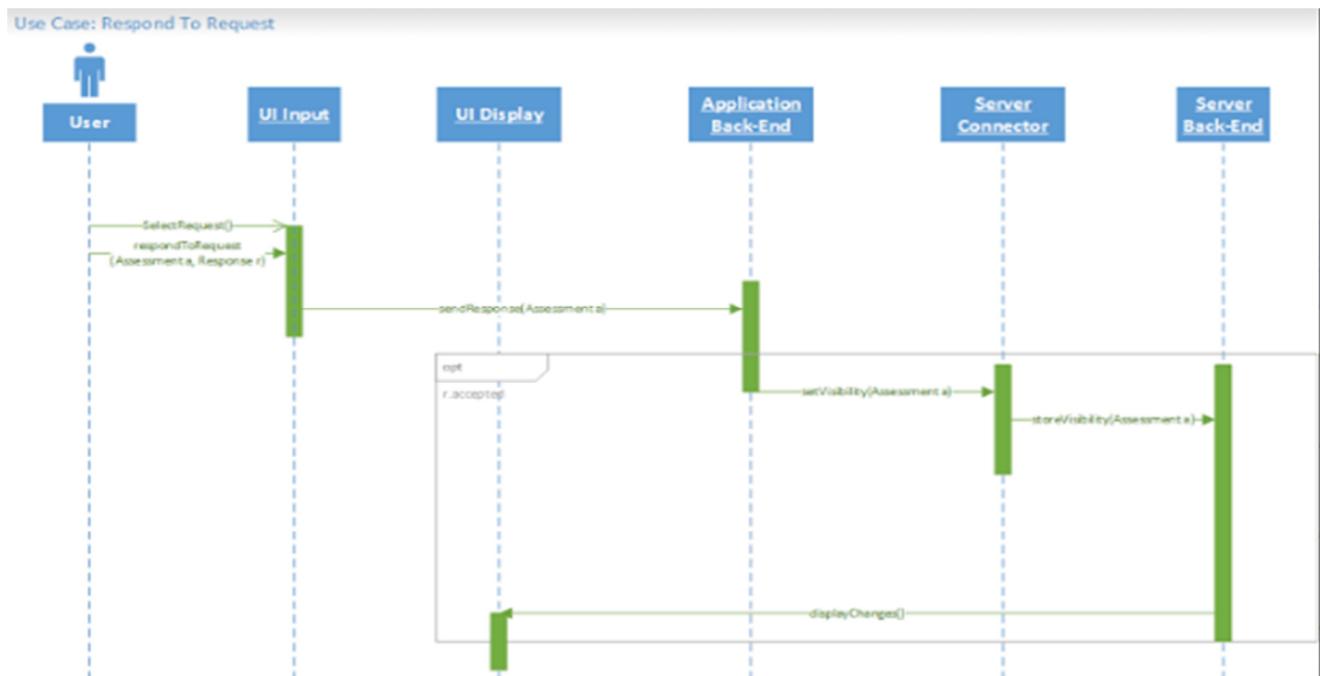


View Assessment Sequence Diagram: After requesting to view an assessment, the system will display a list of viewable assessments. Once the user selects which assessment they wish to view, the system will display the assessment information on the screen.

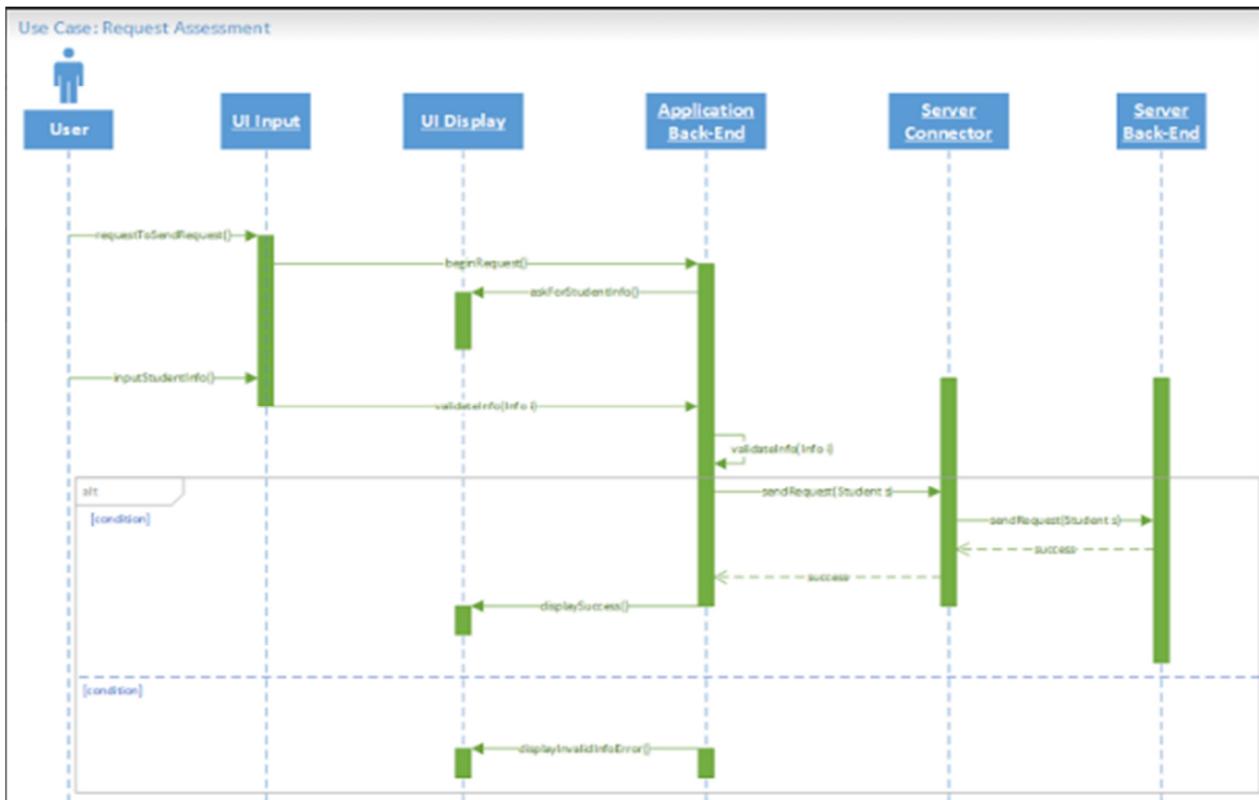




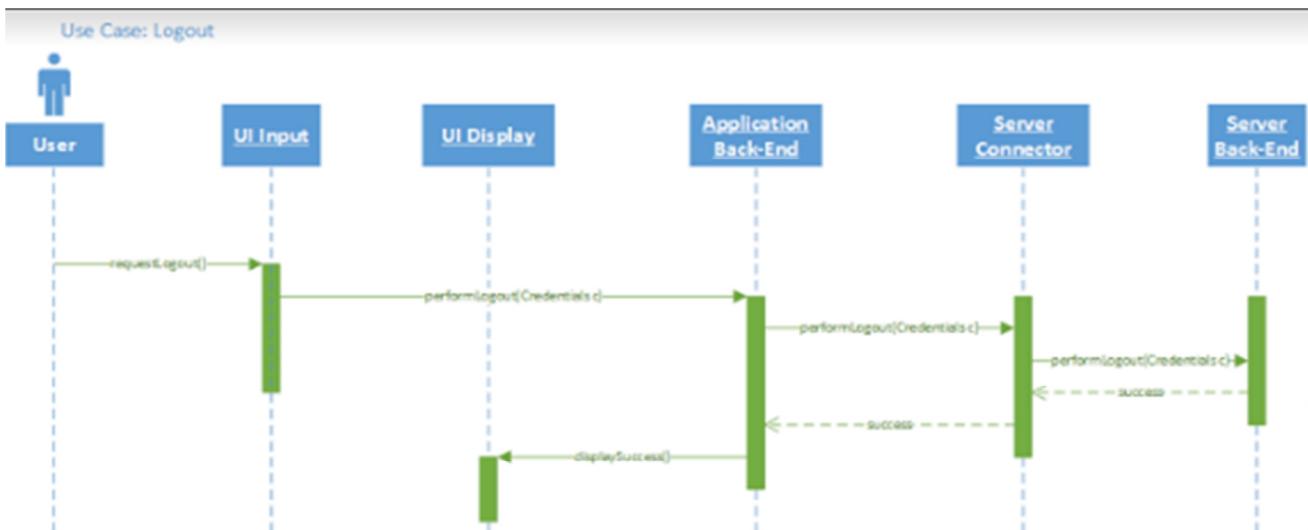
Submit Profile Sequence Diagram: After a user inputs their profile information and requests to save, the system validates the information to ensure that there is no invalid or dangerous input. If it passes validation, the information is saved and a message stating that it was saved is displayed. Otherwise, a message displaying that there was a problem is displayed.



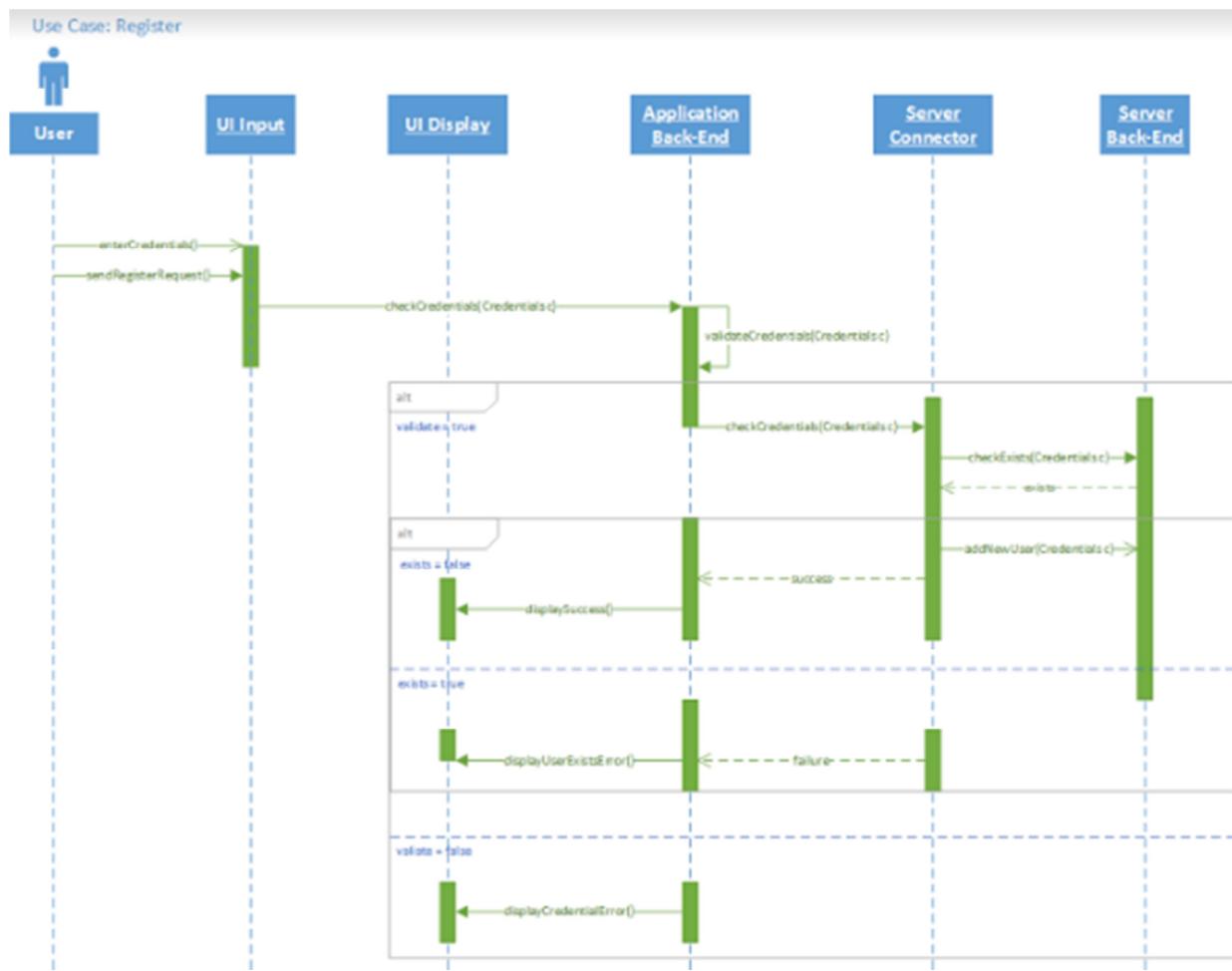
Respond to Request Sequence Diagram: After the user selects the request they wish to respond to and submits their response, the system sends a message back to the requester, sets the visibility of the selected requested assessment



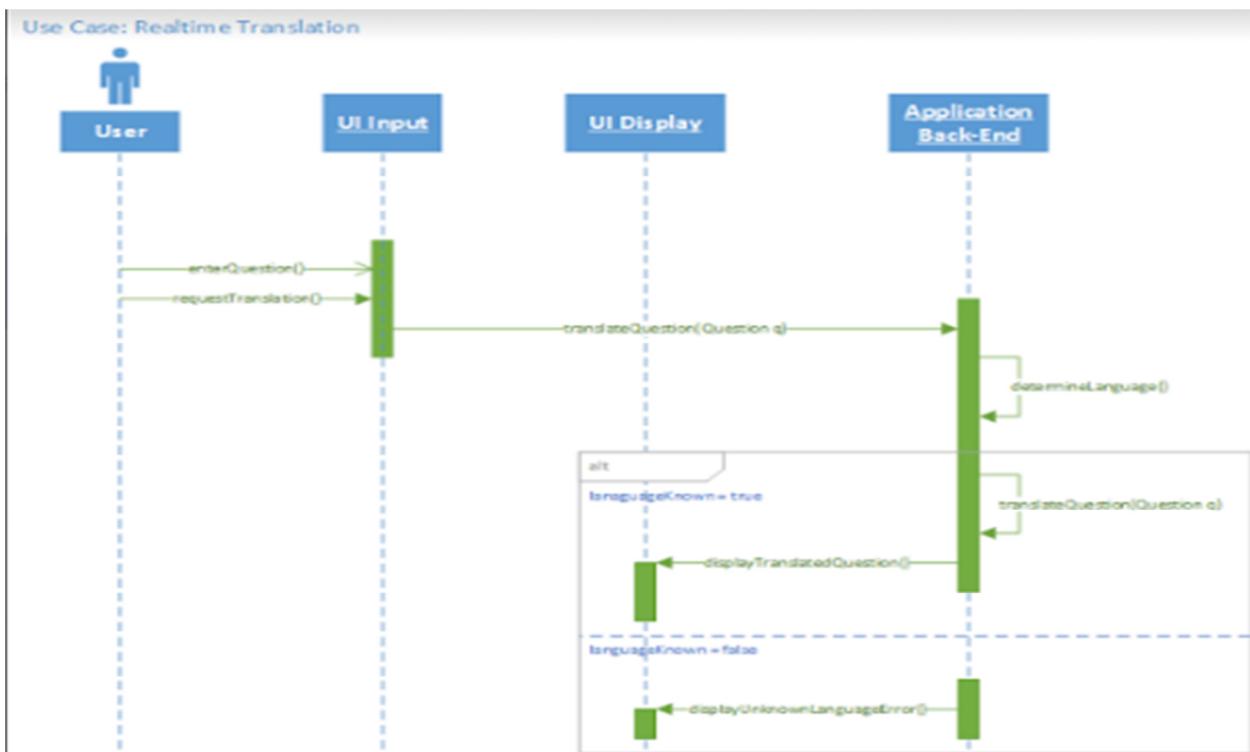
Request Assessment Sequence Diagram: After the user requests to send a request, the system will ask for the student information. Once student information is inputted, it is validated to see if the student exists. If the student exists, the request is sent to the student and a success message is displayed. Otherwise, a message is displayed that the student could not be located.



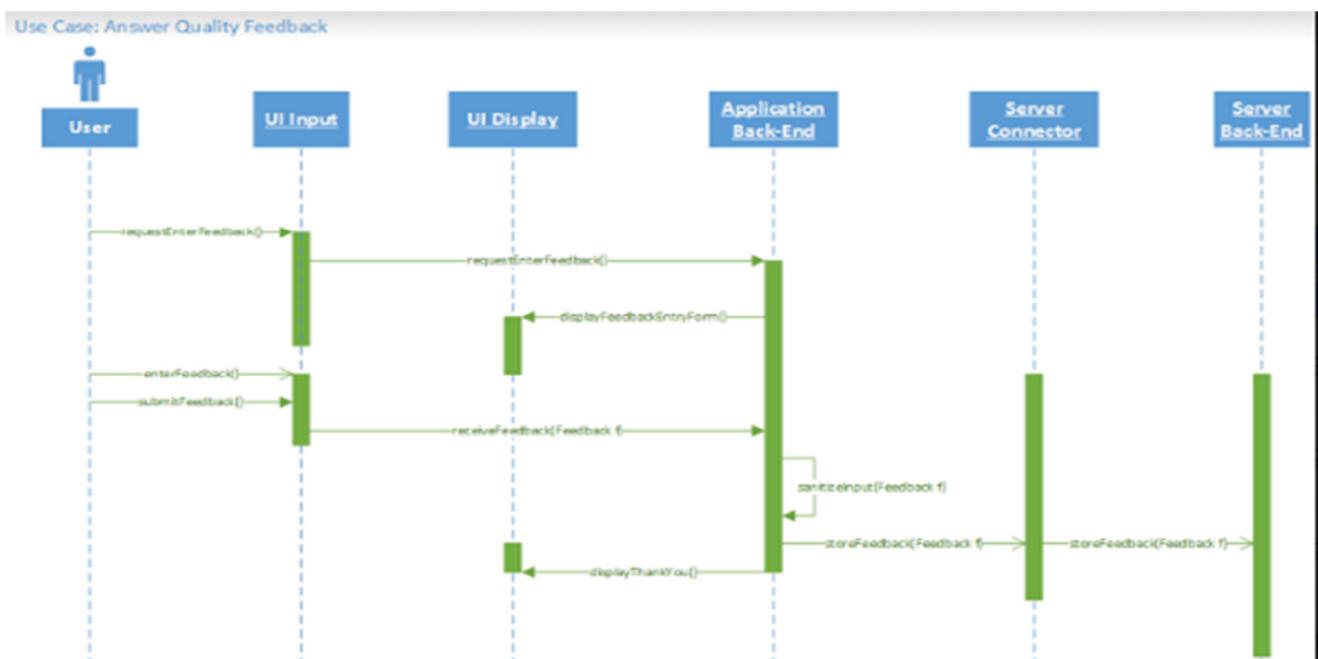
Logout Sequence Diagram: After a user requests to logout, the system logs the user out of the server and displays success.



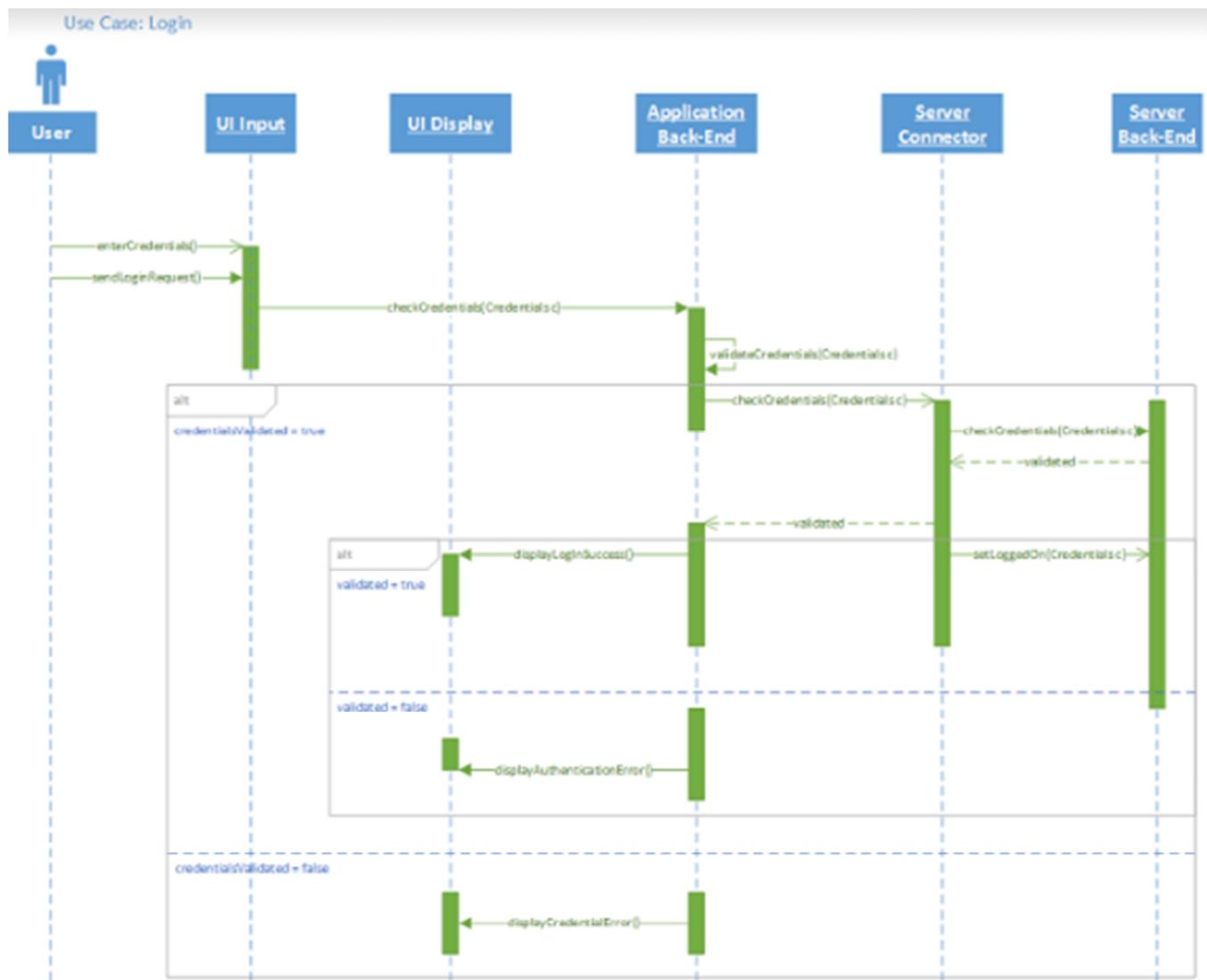
Register Sequence Diagram: After the user enters credentials and requests to register, the system checks the credentials to see if they meet the requirements (email format, correct number of characters, etc). If valid, the system checks if a user with the same email already exists. If a user does not exist, the system adds a new user to the database and displays success. If a user with the same email already exists or the credentials do not meet requirements, an error message is displayed.



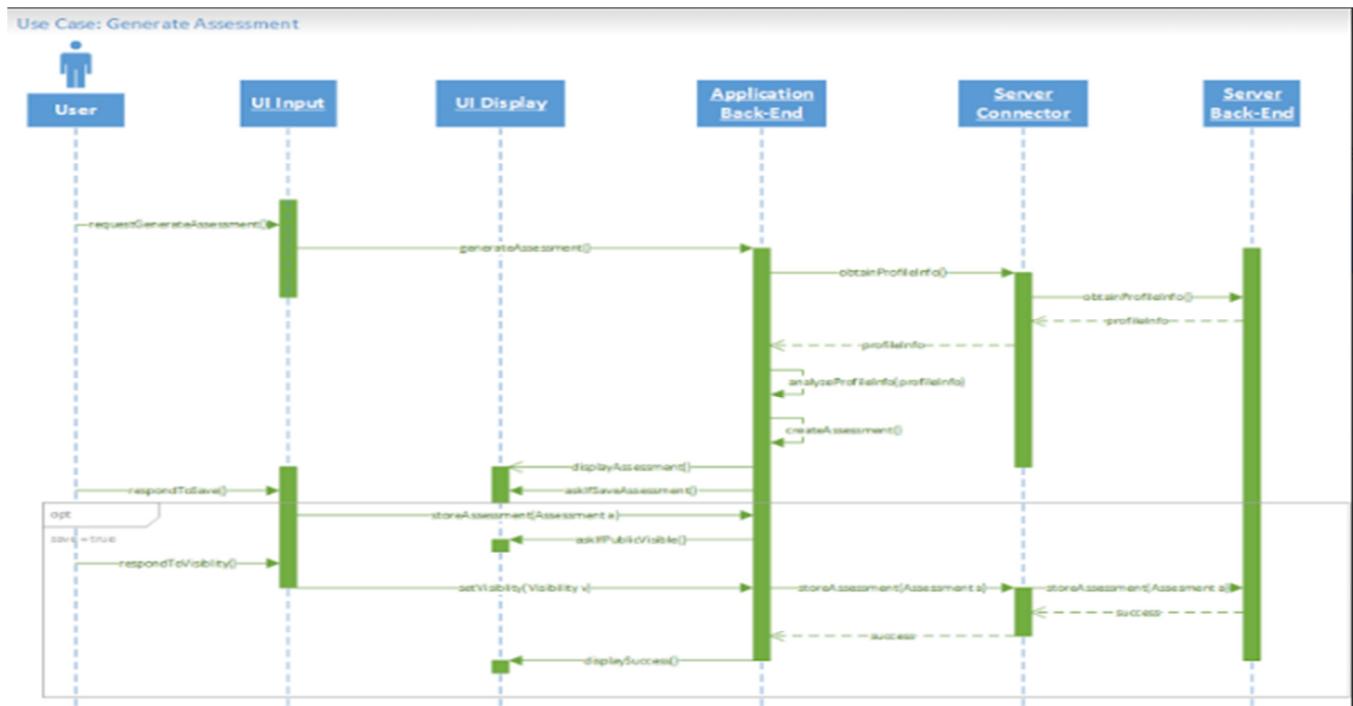
Real-time Translation Sequence Diagram: After the user enters the question and requests a translation, the system determines the language of the question. If it can be translated, the translated text is then displayed. Otherwise, an error message is shown.



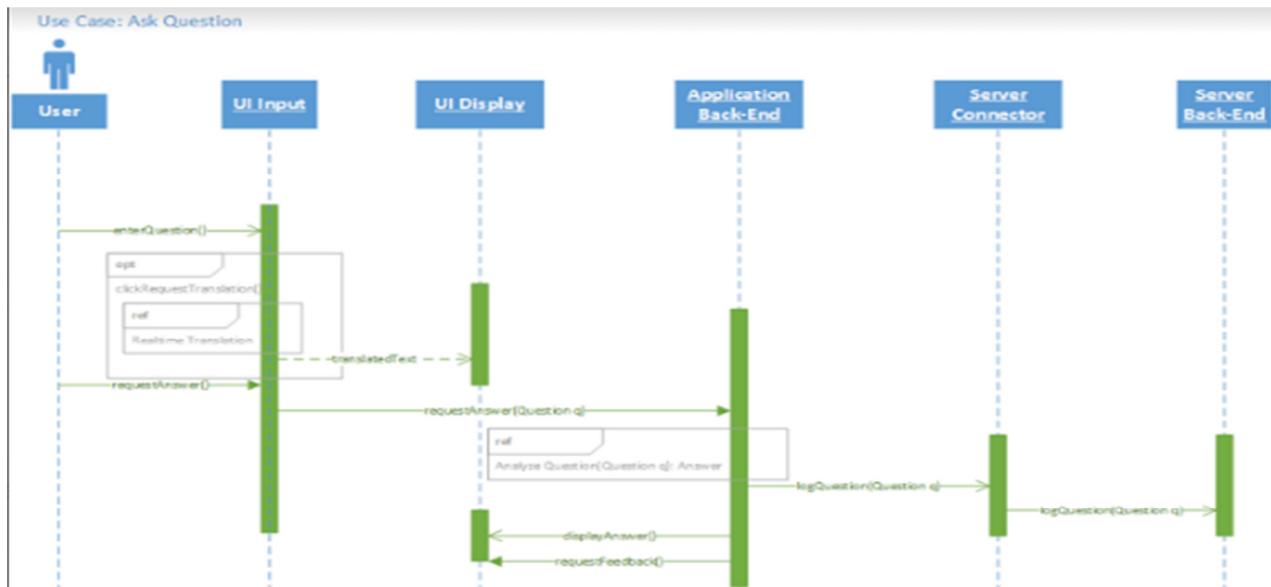
Answer Quality Feedback Sequence Diagram: After a user requests to enter feedback, an entry form is displayed for the user to input the feedback. Once submitted, the system checks to make sure the input is not dangerous, then stores it in the database, and finally displays a thank you message to the user.



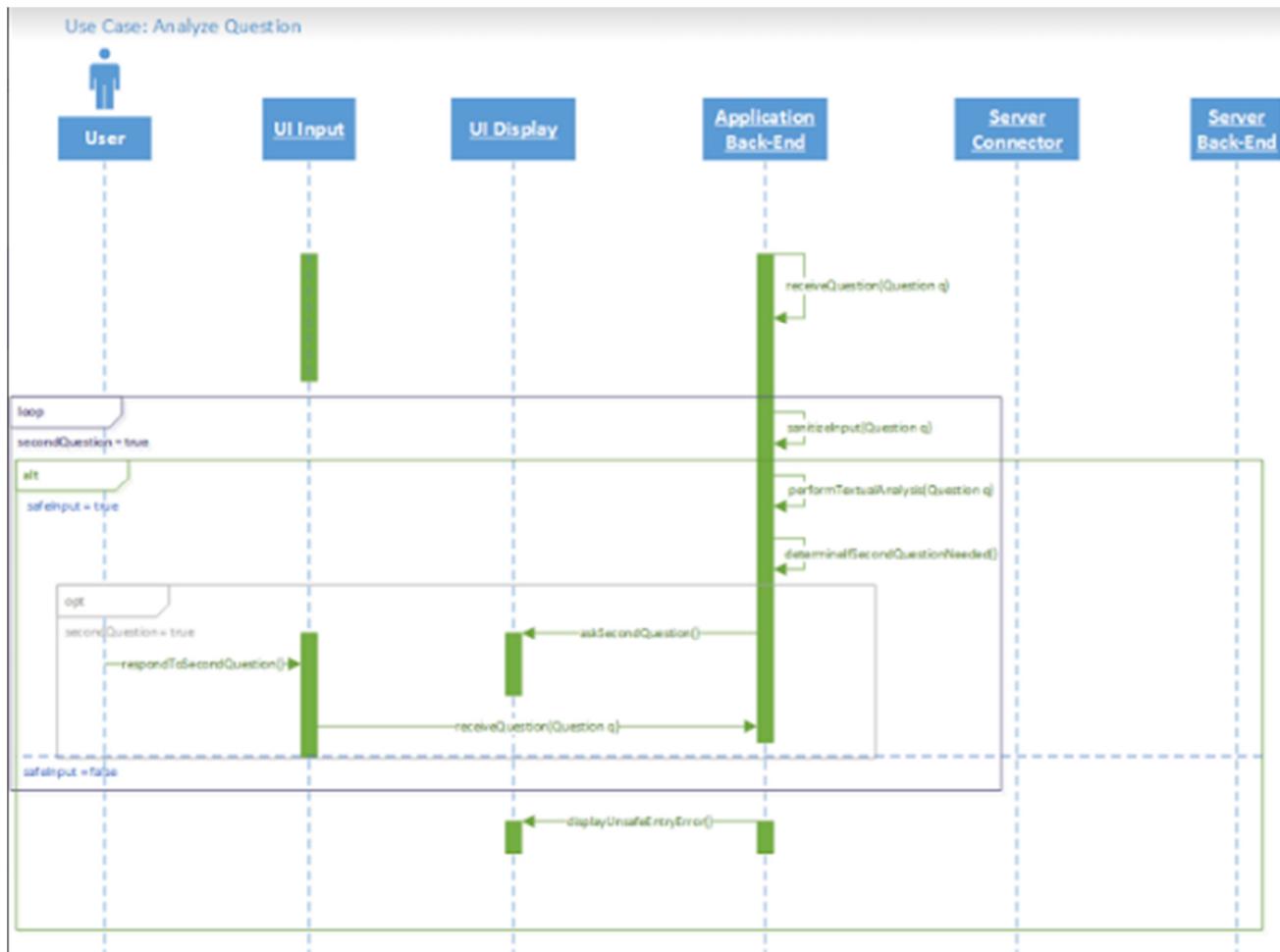
Login Sequence Diagram: After the user enters their credentials and sends a login request, the system checks the credentials to see if they meet requirements first, and then checks to see if they are valid. If the credentials meet requirements and are valid, the user is logged on and a success message is displayed. Otherwise, an error message is displayed.



Generate Assessment Sequence Diagram: When the user requests to generate an assessment, the system obtains their profile information and analyzes it. Once analyzed, an assessment is created and displayed to the user. The system then asks if the user would like to save the assessment, if they do want to save then the system asks if they want it publicly visible. The publicly visible answer(true or false) is sent back with the assessment to the server to be saved.



Ask Question Sequence Diagram: When the user enters a question, they can request translation before requesting their answer. Once an answer is requested, the system calls to Analyze Questions and performs the functionality and returns the answer. The question is then logged and the answer is displayed. After the answer is displayed, the system asks for feedback.



Analyze Question Sequence Diagram: The system receives a question from the Ask Question functionality. First, the system checks to see if the question is dangerous. If it is not dangerous, a textual analysis is performed. After the textual analysis, the system determines if it needs to ask a second question. If a second question is needed, it displays it to the user and then awaits input. The process above is repeated until another question no longer needs to be asked.

Data Requirements

- **UC001** - Input: User whose assessment is being requested.
- **UC004** - Input: User email address and real name for registration and login.
- **UC004** - Input: User password for registration and login.
- **UC006** - Input: Filter information for useful question log (eg. major, minor, courses).
- **UC007** - Output: List of newly generated questions exported to a text file.
- **UC007** - Output: List of unanswered frequently asked questions.
- **UC009** - Input: Untranslated text.
- **UC009** - Output: Translated text.
- **UC010** - Input: User feedback pertaining to the relevance of the responses to their questions.
- **UC011** - Input: User questions.
- **UC011** - Output: Responses to user questions.
- **UC013** - Input: User profile.
- **UC015** - Output: User assessment based on their profile and search history.
- **UC019** - Input: The advisor who is requesting the student's assessment.

Non-Functional Requirements

Product: Performance Requirements

Project Name:		Intelligent Academic Planner						
Requirement ID:	SP-03-01			Type	Functional	Non-Functional		
Creation:	Oct 20 2016 01:53 AM			User	<input type="checkbox"/>	<input type="checkbox"/>		
Modification:	Oct 20 2016 01:54 AM			System	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
Description:	The system should provide an answer to a question within 5 seconds.		Product (sub-type below)		Performance Requirements			
Priority:	Highest	High	Medium	✓ Low	Lowest			
This Req. is Engineered From:	UP-03							
Justify why meeting SP-03-01 can contribute to the fulfilment of UP-03	Ensures an answer is quickly given by adding a performance requirement.							
Traceability:	Use cases cf.	N/A						
	Test cases cf.	Yet to be completed in test case worksheet!						
Acknowledgment	Generated from the CapStone Process Management System ©2015							

SP-03-01: This system requirement requests that when we are ensuring that an answer is providing quickly, we should test that it answers within 5 seconds. Low priority was given to this requirement because we are focusing first on implementing features and then focusing on quality and performance of the features.

Product: Dependability/Reliability/Security

Project Name:	Intelligent Academic Planner				
Requirement ID:	SP-01-01	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 12:12 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 12:14 AM	System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Description:	The system should only display information that the user knows is being displayed.				Product (sub-type below)
					Dependability/Reliability/Security
Priority:	Highest	<input checked="" type="checkbox"/> High	Medium	Low	Lowest
This Req. is Engineered From:	UP-01				
Justify why meeting SP-01-01 can contribute to the fulfilment of UP-01	Makes the user's profile more secure by allowing the user to specify which information is visible.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

SP-01-01: This system requirement requests that when we are ensuring that a profile is secure, we should test that it only displays information that the user has set to be public. High priority was given to this requirement because it should be done while creating the functionality of the profile and we were encouraged to keep security in mind.

Organizational: Development Requirements

Project Name:	Intelligent Academic Planner				
Requirement ID:	SO-01-01	Type	Functional	Non-Functional	
Creation:	Sep 23 2016 12:42 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 20 2016 01:59 AM	System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Description:	The system should log the user out after 1 hour of inactivity.				Organizational (sub-type below)
					Development Requirements
Priority:	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
This Req. is Engineered From:	UO-01				
Justify why meeting SO-01-01 can contribute to the fulfilment of UO-01	Ensures a user is not logged on for too long, managing their session.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

SO-01-01: This system requirement requests that when we are ensuring that a user's session is managed, we should test that it logs the user out after 1 hour of inactivity. Low priority was given to this requirement because while it is connected to logging in, logging out, and registering, it is not critical to other functionality of the system.

4.3. Requirements Trace Table

Project Name: Intelligent Academic Planner			
User Requirements		System Requirements	
Req ID	Description	Req ID	Description
UF-A	A user can log in.	SF-A-01	The system should log-in a user within 5 seconds.
UF-B	A user can log out.	SF-B-01	The system should log-out a user within 5 seconds.
UF-C	A user can ask the system questions.	SF-C-01	The system should conduct textual analyses.
		SF-C-02	The system should be able to handle input from multiple well-known languages.
		SF-C-03	The system should recommend majors suitable for the user based on the personality assessment.
		SF-C-04	The system should gather data unique to each user.
		SF-C-05	The system should recommend courses based on the recommended majors.
UF-D	A user should receive multiple responses to a question.	SF-D-01	The system should show a minimum of 1 related search/question.
UF-E	A user can create a profile	SF-E-01	The system should allow between 100 and 600 words to describe a user's academic and professional interests.
		SF-E-02	The system should allow a user to submit 100 words of self-description about their personality.
		SF-E-03	The system should allow a user to view their personality assessments.
		SF-E-04	The system should create a personality assessment unique to each user based on the data gathered.
		SF-E-05	The system should summarize this data to be used by an advisor directing the student.
UF-F	A user can register	SF-F-01	The system should register the user within 5 seconds.
UF-G	A user can view a log of asked questions.	SF-G-01	The system should only allow Advisors and System Developers to view the question log.
UF-H	A user can provide information to improve accuracy of the system.	SF-H-01	A user can provide answer quality feedback after asking a question.
UO-01	A user's session should be managed.	SO-01-01	The system should log the user out after 1 hour of inactivity.
UP-01	A user's profile should be secure.	SP-01-01	The system should only display information that the user knows is being displayed.
UP-03	A user should receive a quick response after asking a question	SP-03-01	The system should provide an answer to a question within 5 seconds.

Acknowledgment: Generated from the CapStone process management system ©2015

5. Exploratory Studies

5.1. Relevant Techniques

- Surveys
 - ◆ Qualitative Data: Open-ended questions “can lead to the discovery of new initiatives or problems that should be addressed.”^[8]
 - If your major/minor has changed, what was it and why did it change?
 - What are the best and worst features of your field of study?
 - Why did you choose to study at Behrend?
 - What questions/concerns did you have when deciding on your major and school?
 - ◆ Quantitative Data: Closed-ended questions “allows researchers to categorize respondents into groups based on the options they have selected.”^[17]
 - What year are you? (Freshman, Upper)
 - What is your major and minor?
- Data Crawling
 - ◆ Question Log
 - ◆ Suggested Feedback: answers for unclear questions
 - ◆ Analysis Data: data gleaned from user's input
- Natural Language Processing^[9]
 - ◆ Generative Models for Parsing
 - Parse Trees
 - Part-of-Speech
 - Useful Relationships
 - Context-Free Grammar
 - ◆ Log-linear Taggers
 - Information Extraction
 - Named Entity Recognition
 - Relationships between Entities
 - Named Entity Extraction as Tagging
- Document Curating^[10]
 - ◆ Exact match search
 - ◆ Wildcard search
 - ◆ Proximity search
- Machine Learning Algorithms^[11]
 - ◆ Supervised Learning
 - Linear Regression
 - Logistic Regression
 - Decision Tree
 - ◆ Unsupervised Learning
 - K-Means
 - Apriori

5.2. Relevant Packages/Products

→ NodeJS^[12]

- ◆ Server application type
- ◆ We decided to use this because most of the Bluemix APIs are designed for Node.JS handling and Node.JS is an advanced, efficient server application.
 - Watson Platform
- ◆ Conversation^[5]
 - Mainly used to classify the question input from user and determine if the question is in our domain or not.
- ◆ Retrieve and rank^[4]:
 - Core functionality of the project
 - Uses machine training and NLP(Natural language process) to study the training dataset and provide feedback on the answer with an accuracy measurement.
- ◆ Real time translation^[15]
 - Allows instant translation between common languages.
 - Displays desired language based on the user's account preference.
- ◆ Personality analysis^[16]
 - Assessment functionality to generate personality analysis for user. This will allow us to give more specific answers to a user based on their personality.
- MongoDB^[14]
- ◆ Main non-relational DB for the application, since we will handle a lot of unstructured data.
- ◆ MongoDB has good features to store and sort.
 - Natural language processing
- ◆ Monkey learn^[2]
 - A mature public accessible API provider to do NLP analysis
- ◆ Natural^[9]
 - Equivalent NodeJS library of NLTK(Natural language toolkit) in python. It mainly focuses on customized NLP classification.
- ◆ Alchemy^[3]
 - Similar to Monkey learn, but is managed by IBM. It will provide feedback on the detailed analysis of a sentence, so we can analyze parts of sentences to determine the intent of the question.

→ Hadoop

- ◆ The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce. Hadoop splits files into large blocks and distributes them across nodes in a cluster.

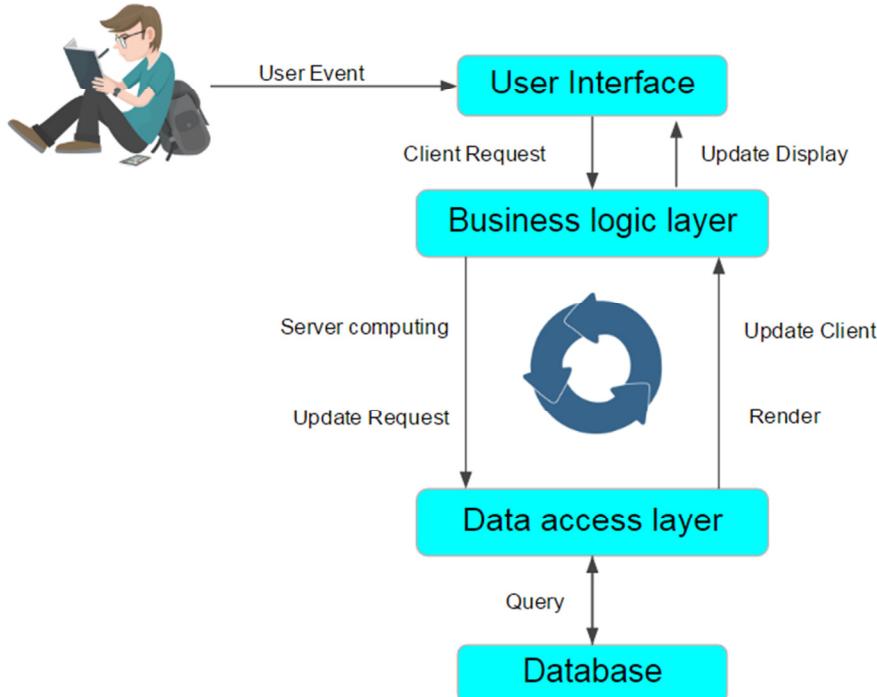
5.3. Broader Impacts

This project can be expanded to include other majors within Behrend or branches of Penn State University. On a grand scale, this type of tool would be a beneficial tool for high school juniors and seniors, as well as college freshman.

6. System Design

6.1. Architectural Design

- Layered Architecture
 - ◆ Client-Server-Database model
 - ◆ Server handles all computation and updates
 - ◆ Server query with DB
 - ◆ Feedback data to client side



7. System Implementation

7.1. Programming Languages & Tools

- Bluemix (conversation, retrieve and rank, alchemyAPI,...)^{[3][4][5]}
- MongoDB^[14]
- Node.js^[12]

7.2. Coding Conventions

- N/A

7.3. Code Version Control

- GitHub^[13]

8. System Testing

This section will be gone over in future reports.

9. Challenges & Open Issues

9.1. Challenges Faced in Requirements Engineering

- Initially, we struggled to determine the scope of the project. Watson has many features, therefore we had to be very specific about which of these features would make the most sense for our project's initial development.
- Understanding the domain, or what sort of questions we wanted Watson to answer.
- Be able to benchmark the accuracy of answering system.
- Be able to automate the training/learning system.

10. System Manuals

10.1. Instructions for System Development

How to setup development environment

The Intelligent Academic Planner project is a web application, that will be accessible publicly on Heroku, a Node.JS environment host platform. The source code will be stored by GitHub, for version control purposes. And the master version will always be automatically deployed and run on Heroku. There is no restriction on what IDE will be used by each group member.

Notes on system further extension

- Automate question and feedback log DB
- Natural language extension^[9]
- Analytical library
- Visual recognition on building^[7]
- Attitude analysis on question
- Campus direction utility
- Course info helper (location, material, etc...)
- News/feeds utility^[17]

11. Conclusion

This section will be gone over in future reports.

12. References

- [1]. I. B. M. What is IBM Watson? <http://www.ibm.com/watson/what-is-watson.html> (accessed Oct 5, 2016).
- [2]. MonkeyLearn <http://docs.monkeylearn.com/> (accessed Oct 5, 2016).
- [3]. AlchemyAPI <http://www.alchemyapi.com/> (accessed Oct 5, 2016).
- [4]. Watson Developer Cloud <https://www.ibm.com/watson/developercloud/retrieve-rank.html> (accessed Oct 5, 2016).
- [5]. Watson Developer Cloud
<https://www.ibm.com/watson/developercloud/conversation.html> (accessed Oct 5, 2016).
- [6]. Watson Developer Cloud <https://www.ibm.com/watson/developercloud/speech-to-text.html> (accessed Oct 5, 2016).
- [7]. Watson Developer Cloud <https://www.ibm.com/watson/developercloud/visual-recognition.html> (accessed Oct 5, 2016).
- [8]. 3 Types of Survey Research, When to Use Them, and How they Can Benefit Your Organization! <https://fluidsurveys.com/university/3-types-survey-research-use-can-benefit-organization/> (accessed Oct 21, 2016).
- [9]. Machine Learning Methods in Natural Language Processing
http://www.cs.columbia.edu/~mcollins/papers/tutorial_colt.pdf (accessed Oct 21, 2016).
- [10]. IBM Knowledge Center
<https://www.ibm.com/support/knowledgecenter/SSR99/kc/watsoncurator.html> (accessed Oct 21, 2016).
- [11]. Essentials of Machine Learning Algorithms (with Python and R Codes)
<https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/> (accessed Oct 21, 2016).
- [12]. NodeJS <https://nodejs.org/en/> (accessed Oct 21, 2016).
- [13]. Github <https://help.github.com/> (accessed Oct 21, 2016).
- [14]. MongoDB <https://docs.mongodb.com/> (accessed Oct 21, 2016).
- [15]. Language Translator-Translate and publish content in multiple languages.
<https://www.ibm.com/watson/developercloud/language-translator.html> (accessed Oct 21, 2016).
- [16]. Personality Insights-Uncover a deeper understanding of people's personality characteristics, needs, and values to drive personalization.
<http://www.ibm.com/watson/developercloud/personality-insights.html> (accessed Oct 21, 2016).
- [17]. Comparing Closed-Ended and Open-Ended Questions
<http://fluidsurveys.com/university/comparing-closed-ended-and-open-ended-questions/> (accessed Oct 21, 2016).