



# Intelligent Academic Planner

**Daria Cook: Major in SE  
Krystal Elliott: Major in SE  
Xiaoyu Zhou: Major in CS**

Course Instructor: Xiaocong Fan  
Faculty Advisor: Meng Su

Industry Sponsor: PSU Behrend CSSE Faculty  
Project Mentor: Adriano Cavalcanti  
Assistant Professor of CSSE

A capstone project report submitted to the faculty of  
The Computer Science and Software Engineering Department  
Penn State Erie, The Behrend College

**February 2017  
(Version 3.0)**

*CSSE Technical Report Series: CSSE-BD-Class2017-003*



# Table of Contents

<b>Abstract</b>	4
<b>Report Revision History</b>	5
Changes in Version 1.5	5
Changes in Version 2.0	5
Changes in Version 2.5	5
Changes in Version 3.0	5
<b>Problem Statement</b>	7
Business Background	7
Needs	7
Objectives	7
<b>Requirements</b>	8
User Requirements	8
Glossary of Relevant Domain Technology	8
User Groups	8
Functional Requirements	9
Project Scope	9
User Scenarios	10
List of User Functional Requirements	12
Non-Functional Requirements	13
Product: Performance Requirements	13
Product: Dependability/Reliability/Security	13
Organizational: Development Requirements	13
System Requirements	14
Functional Requirements	14
List of System Functional Requirements	14
System Behavior	16
Data Requirements	25
Non-Functional Requirements	25
Product: Performance Requirements	25
Product: Dependability/Reliability/Security	25
Organizational: Development Requirements	25
Requirements Trace Table	25
<b>Exploratory Studies</b>	26
Relevant Techniques	26
Relevant Packages/Products	28
Broader Impacts	30
<b>System Design</b>	31

Architectural Design	31
Structural Design	32
User Interface Design	33
Behavioral Design	35
Design Alternatives & Decision Rationale	36
UI Alternative Designs	36
Version 1.0	36
Version 2.0	38
Structural Alternative Designs	42
Architectural Alternative Designs	42
Behavioural Alternative Designs	42
<b>System Implementation</b>	43
Programming Languages & Tools	43
Coding Conventions	43
Code Version Control	43
Implementation Alternatives & Decision Rationale	43
Analysis of Key Algorithms	44
<b>System Testing</b>	45
Test Automation Framework	45
Steps for Installing Mocha	45
Steps for Running Test Cases	45
Test Case Design	45
Test Suites	45
Unit Test Cases	45
Integration Test Cases	47
System Test Cases	47
Acceptance Test Cases	47
Test Execution Report	48
Unit Testing Report	48
Integration Testing Report	50
System Testing Report	50
Acceptance Testing Report	50
<b>Challenges &amp; Open Issues</b>	50
Challenges Faced in Requirements Engineering	50
Challenges Faced in System Development	50
Open Issues & Ideas for Solutions	51
<b>System Manuals</b>	52
Instructions for System Development	52
How to setup development environment	52

Notes on system further extension	52
Instructions for System Deployment	52
Platform Requirements	52
System Installation	52
Instructions for System End Users	52
<b>Conclusion</b>	53
<b>References</b>	54

# 1. Abstract

The IBM Watson uses “natural language processing and machine learning” to reveal insights from large amounts of unstructured data<sup>[1]</sup>. 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

### 2.2. Changes in Version 2.0

- Updated progress in Exploratory Studies
- Added in References
- Added in Unit Test Cases
- Added in Unit Test Execution Reports
- Added in UML Class Diagram in Structural Design
- Added in UML State Diagram in Behavioral Design
- Added steps for installing test framework
- Added steps for running test cases
- Added in challenges for system development
- Added alternative UI designs

### 2.3. Changes in Version 2.5

- Modified format of references
- Updated description of UML Diagram
- Added description of requirements trace table
- Modified abstract slightly
- Exploratory Studies expanded to explain purpose
- References mentioned in Exploratory Studies
- Implementation Alternatives & Decision Rationale expanded
- Architectural Design image discussed
- Relevant Packages/Products expanded

### 2.4. Changes in Version 3.0

- Updated Use Cases and Requirements to use appendices
- Updated Test Cases and Test Execution Reports to use appendices

- Added in Test Suite section under System Testing
- Changed order of subsections in System Testing
- Fixed grammar in Problem Statement
- Made a correction in Business Background
- Exploratory studies updated to reflect new strategies in question creation
- Broader Impacts updated to reflect target audience
- Relevant Packages/Products corrected for greater clarity
- State Diagram in Section 6 updated to reflect new web flow
- User Interface pictures updated to reflect current system
- Old user interface images and captions moved to alternative designs
- Update preprocessing algorithm
- Added in a Coding Convention
- Modified System Development Instructions
- Added in further system extension options
- Added in an additional challenge in system development
- Added in an open issue and solution

## 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 advisement 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

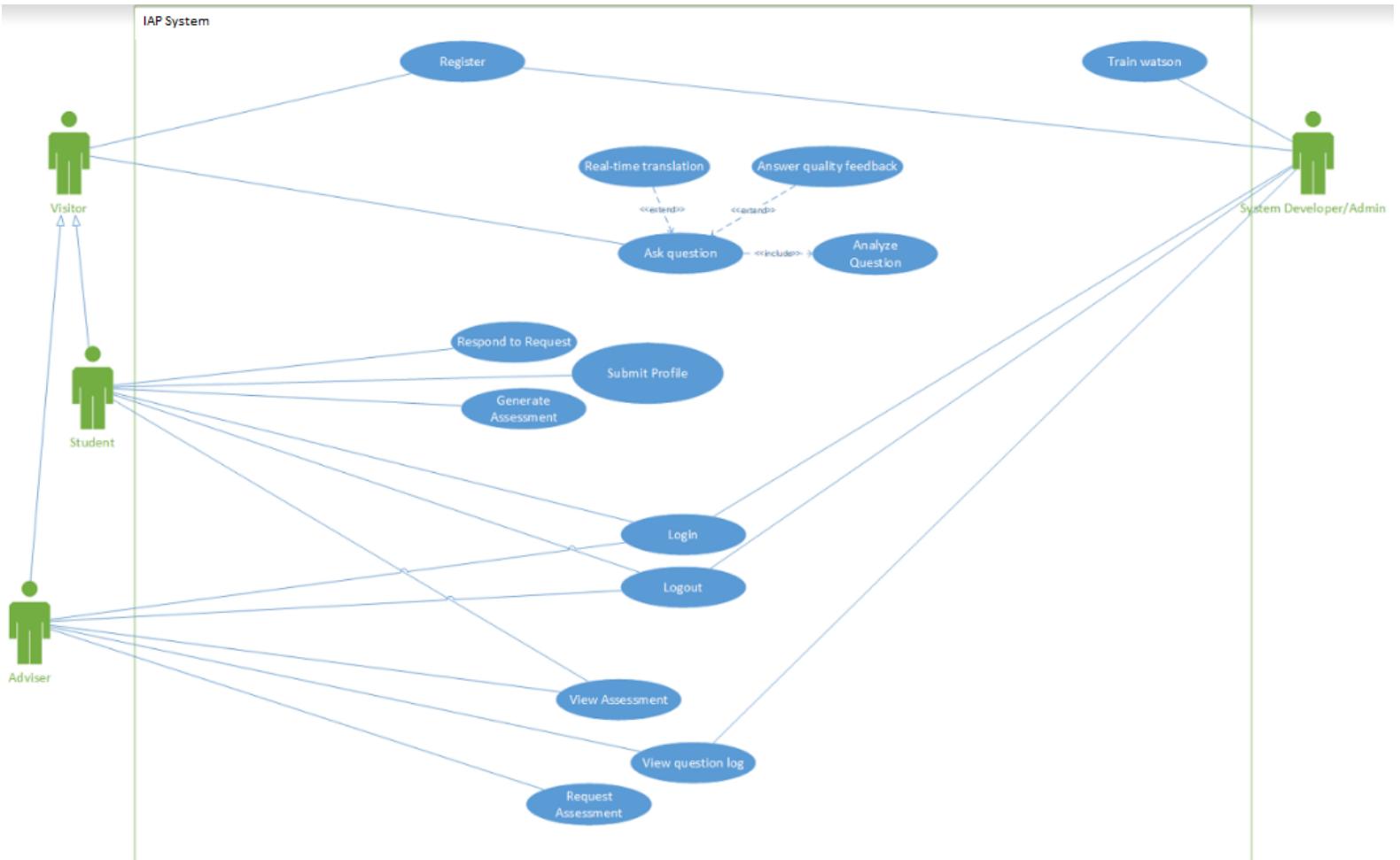
### Glossary of Relevant Domain Technology

- **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 inheritance 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.

Visitors are able to ask a question without having to register. However, since they are not registered and we therefore cannot store information about them, they do not have the ability to create profiles and therefore will not have their personality assessed. These functionalities belong solely to students who have registered. However, asking a question is fully available to anyone that uses the system.

## User Scenarios

**Use Cases can be found in Appendix U**

**Table 4.1: Summary of Use Cases**

**Table 4.2: Use Case 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.

**Table 4.3: Use Case 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.

**Table 4.4: Use Case 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.

**Table 4.5: Use Case 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.

**Table 4.6: Use Case 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.

**Table 4.7: Use Case 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.

**Table 4.8: Use Case 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.

**Table 4.9: Use Case 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.

**Table 4.10: Use Case 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.

**Table 4.11: Use Case 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.

**Table 4.12: Use Case 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.

**Table 4.13: Use Case 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.

**Table 4.14: Use Case 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.

**Table 4.15: Use Case 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

Requirements can be found in Appendix R

**Table 4.16: User Functional Requirements: 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.

**Table 4.17: User Functional Requirements: 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.

**Table 4.18: User Functional Requirements: 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.

**Table 4.19: User Functional Requirements: 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.

**Table 4.20: User Functional Requirements: 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.

**Table 4.21: User Functional Requirements: 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.

**Table 4.22: User Functional Requirements: 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.

**Table 4.23: User Functional Requirements: 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

**Table 4.24: User NonFunctional Requirements: 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

**Table 4.25: User NonFunctional Requirements: 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

**Table 4.26: User NonFunctional Requirements: 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

**Table 4.27: System Functional Requirements: 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.

**Table 4.28: System Functional Requirements: 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.

**Table 4.29: System Functional Requirements: 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.

**Table 4.30: System Functional Requirements: 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.

**Table 4.31: System Functional Requirements: 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.

**Table 4.32: System Functional Requirements: 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.

**Table 4.33: System Functional Requirements: 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 recommend 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.

**Table 4.34: System Functional Requirements: 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.

**Table 4.35: System Functional Requirements: 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.

**Table 4.36: System Functional Requirements: 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.

**Table 4.37: System Functional Requirements: 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.

**Table 4.38: System Functional Requirements: 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.

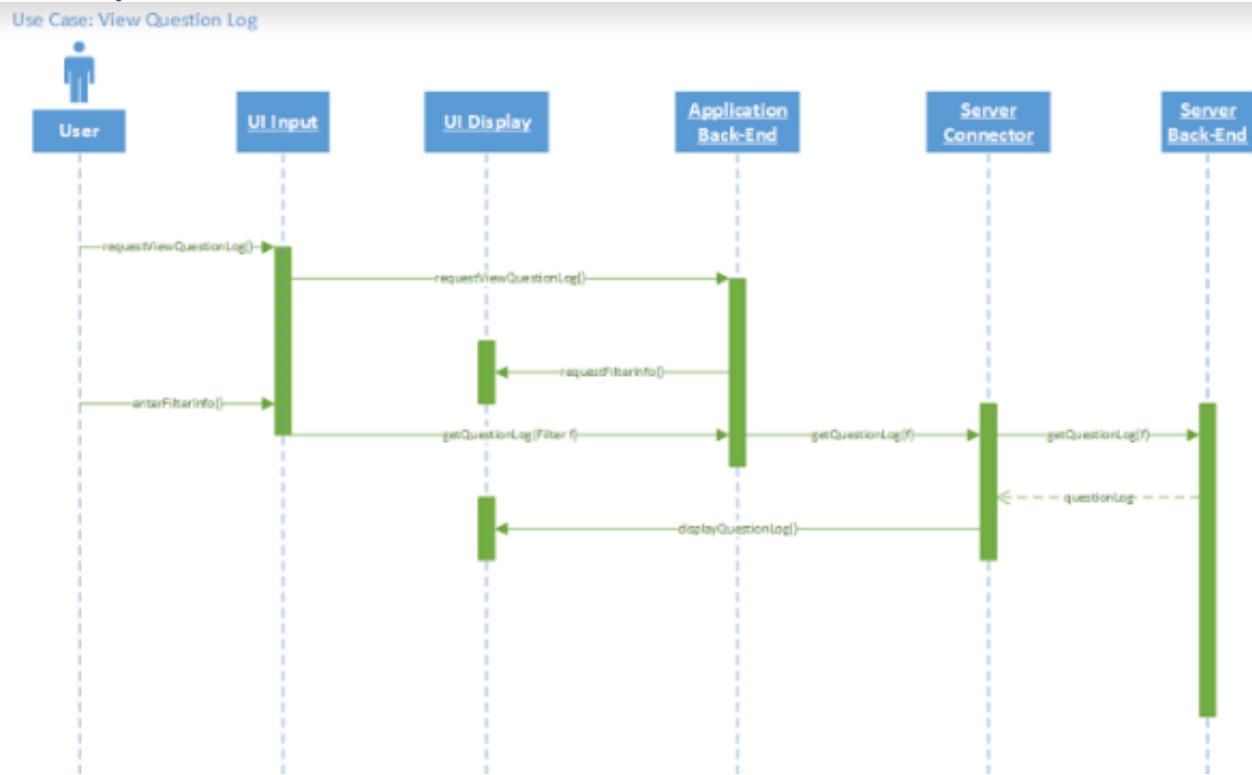
**Table 4.39: System Functional Requirements: SF-E-05:** This system requirement requests that when we create the functionality for creating a profile, the system should summarize data for advisors. Medium priority was given to this requirement because it requires the profile to be fully complete before a summarization can be created.

**Table 4.40: System Functional Requirements: 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.

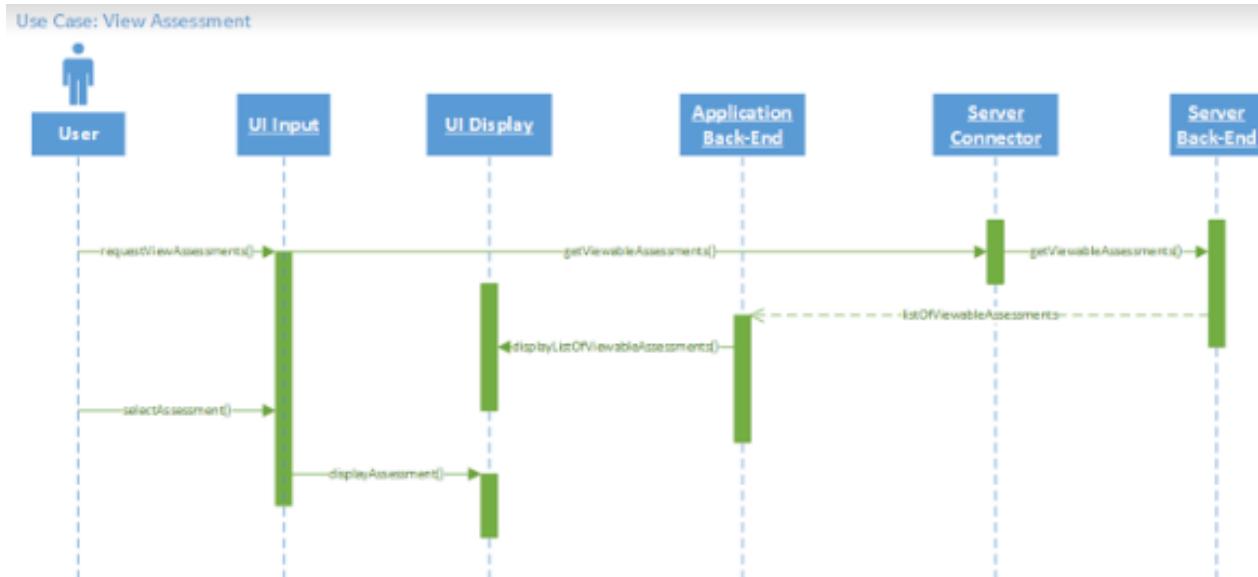
**Table 4.41: System Functional Requirements: SF-G-01:** This system requirement requests that when we create the functionality for viewing the question log, the system should only allow advisors and system developers to view it. Medium priority was given to this requirement because the functionality of the question log is also medium priority.

**Table 4.42: System Functional Requirements: 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.

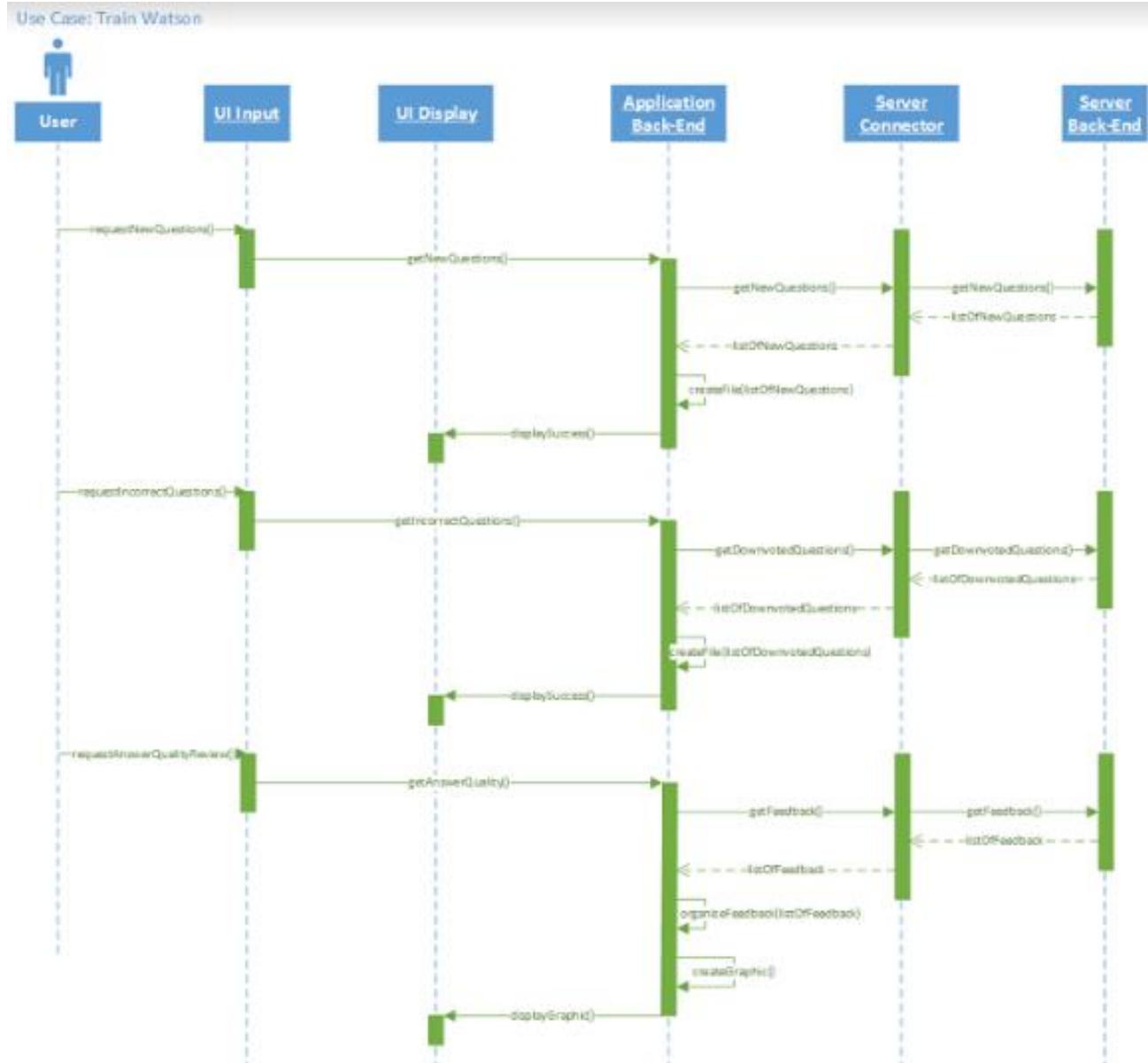
## System Behavior



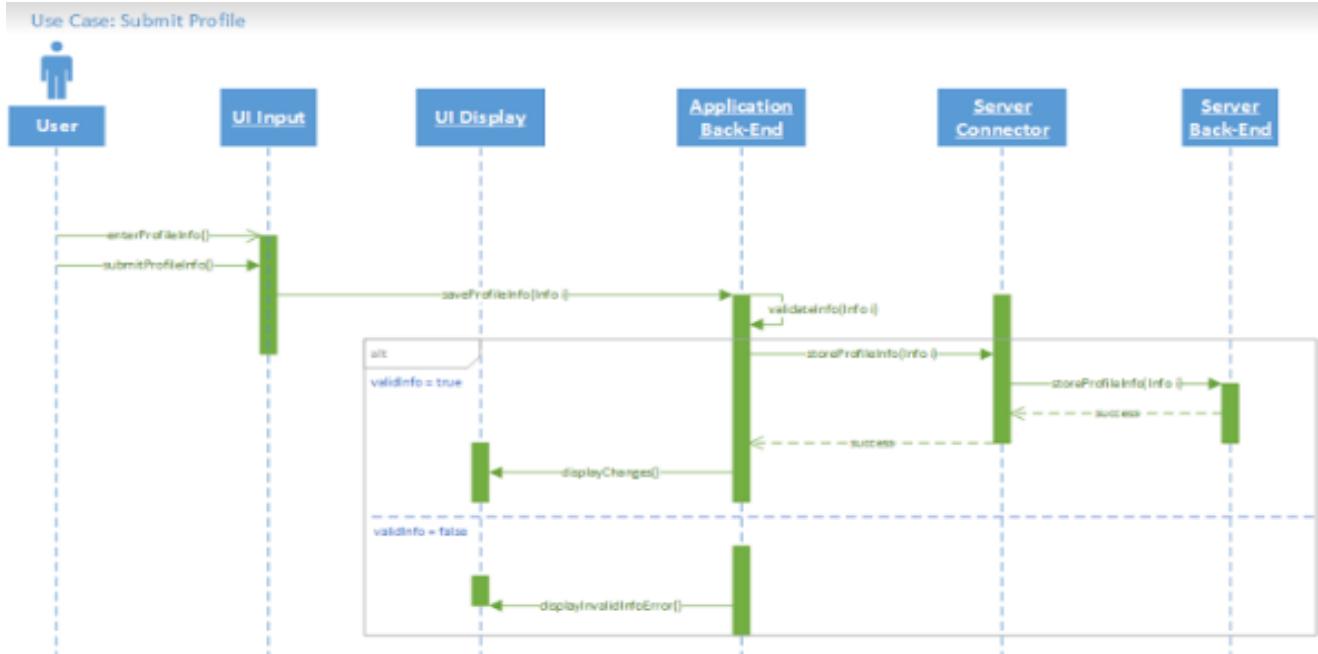
**View Question Log Sequence Diagram:** After requesting to view the question log, the system will ask for filter info and then display the question log for viewing to the user after receiving the questions from the server's back-end.



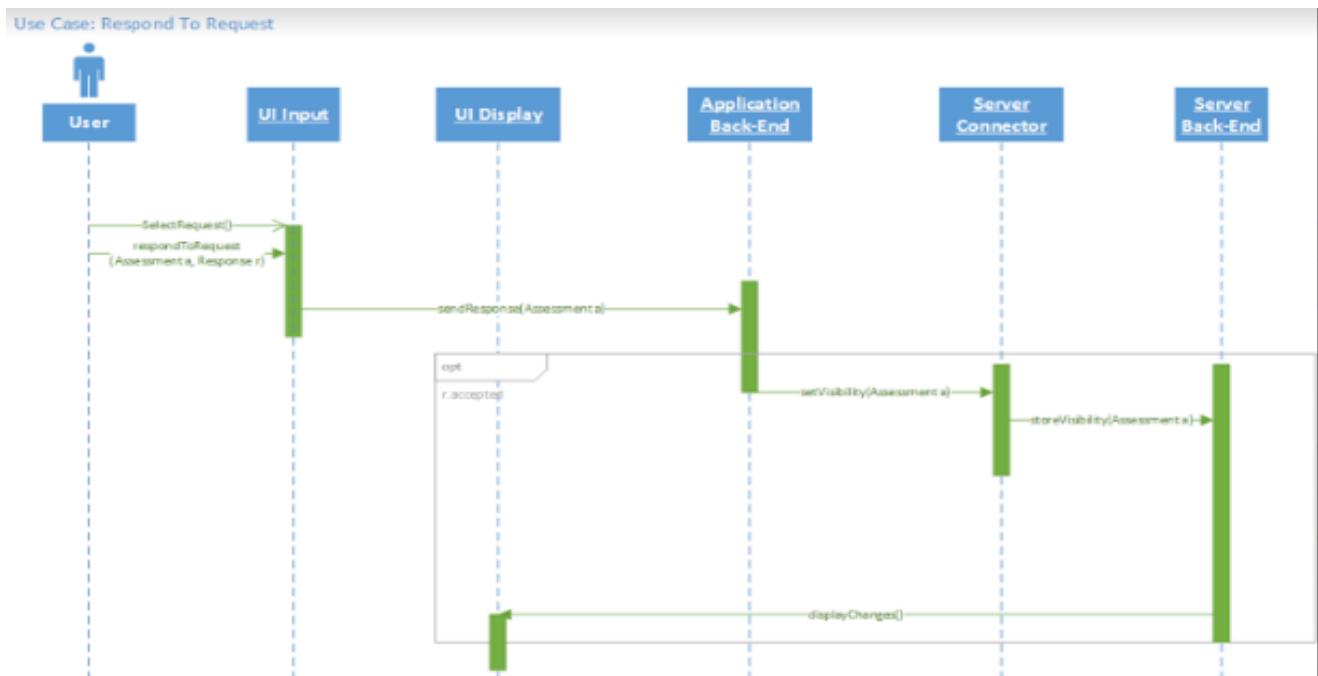
**View Assessment Sequence Diagram:** After requesting to view an assessment, the system will display a list of viewable assessments. Once the user selects which assessment they wish to view, the system will display the assessment information on the screen.



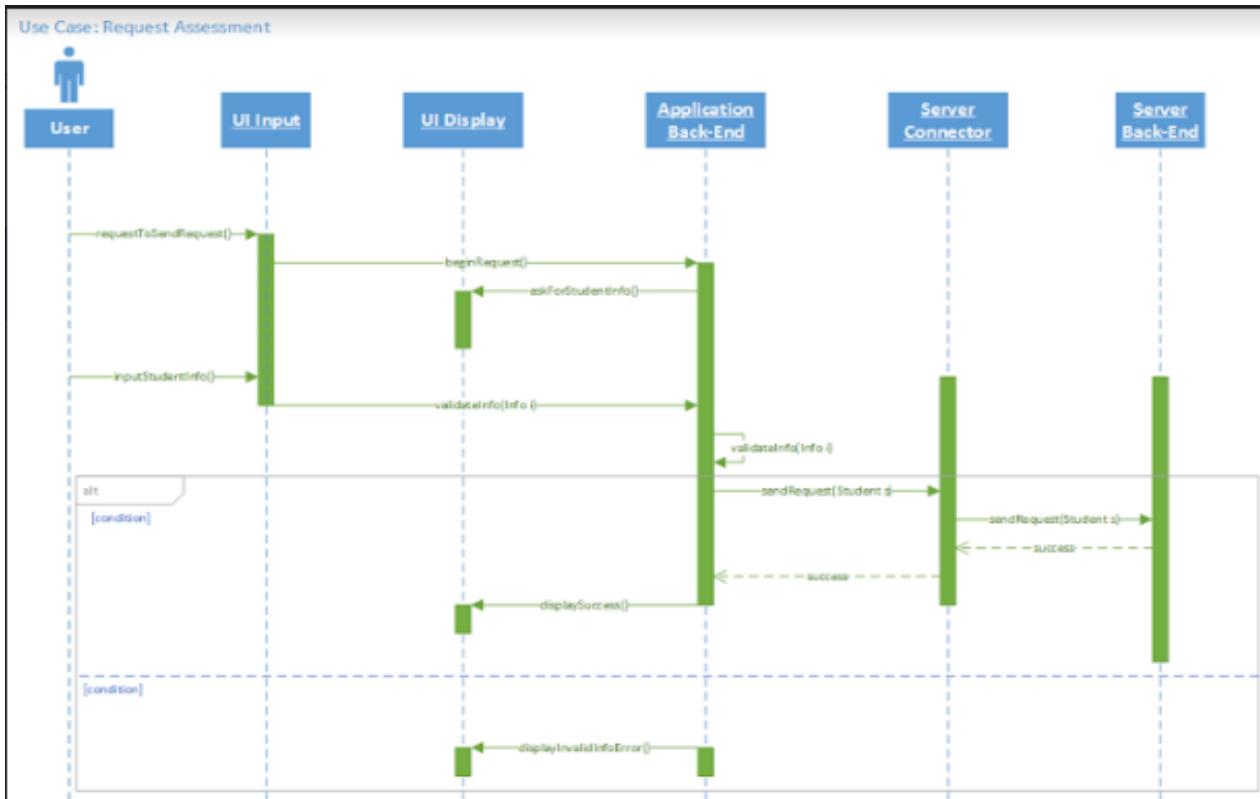
**Train Watson Sequence Diagram:** After the user either requests new questions, incorrect questions, or quality feedback, the system responds by creating a file with a list of the questions matching the filter or a graphic (such as a bar graph) of the quality feedback.



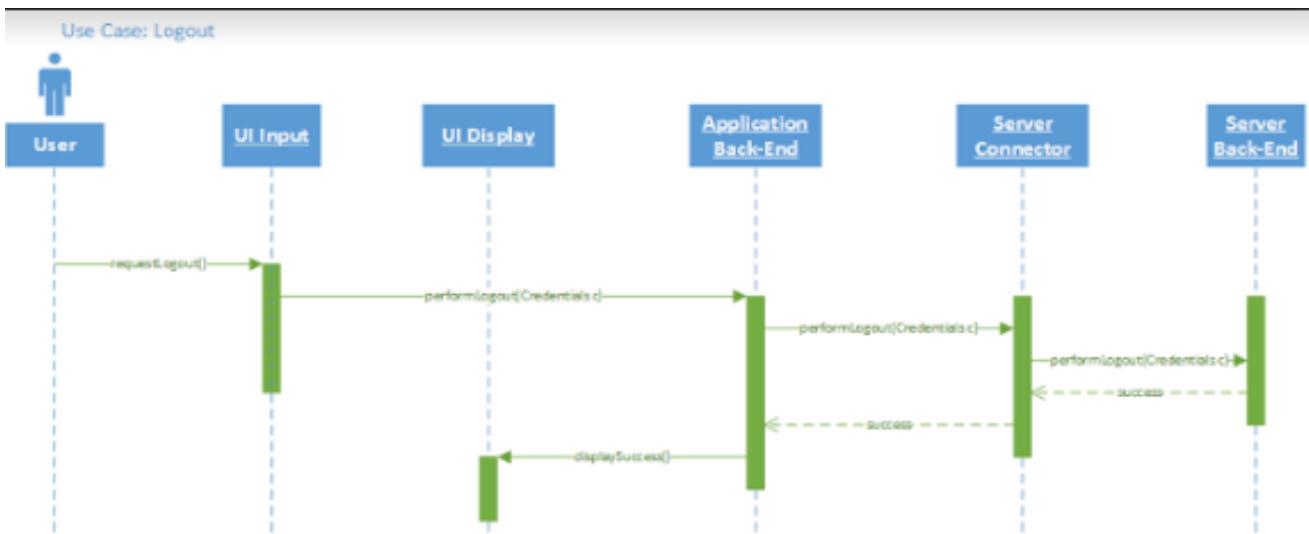
**Submit Profile Sequence Diagram:** After a user inputs their profile information and requests to save, the system validates the information to ensure that there is no invalid or dangerous input. If it passes validation, the information is saved and a message stating that it was saved is displayed.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



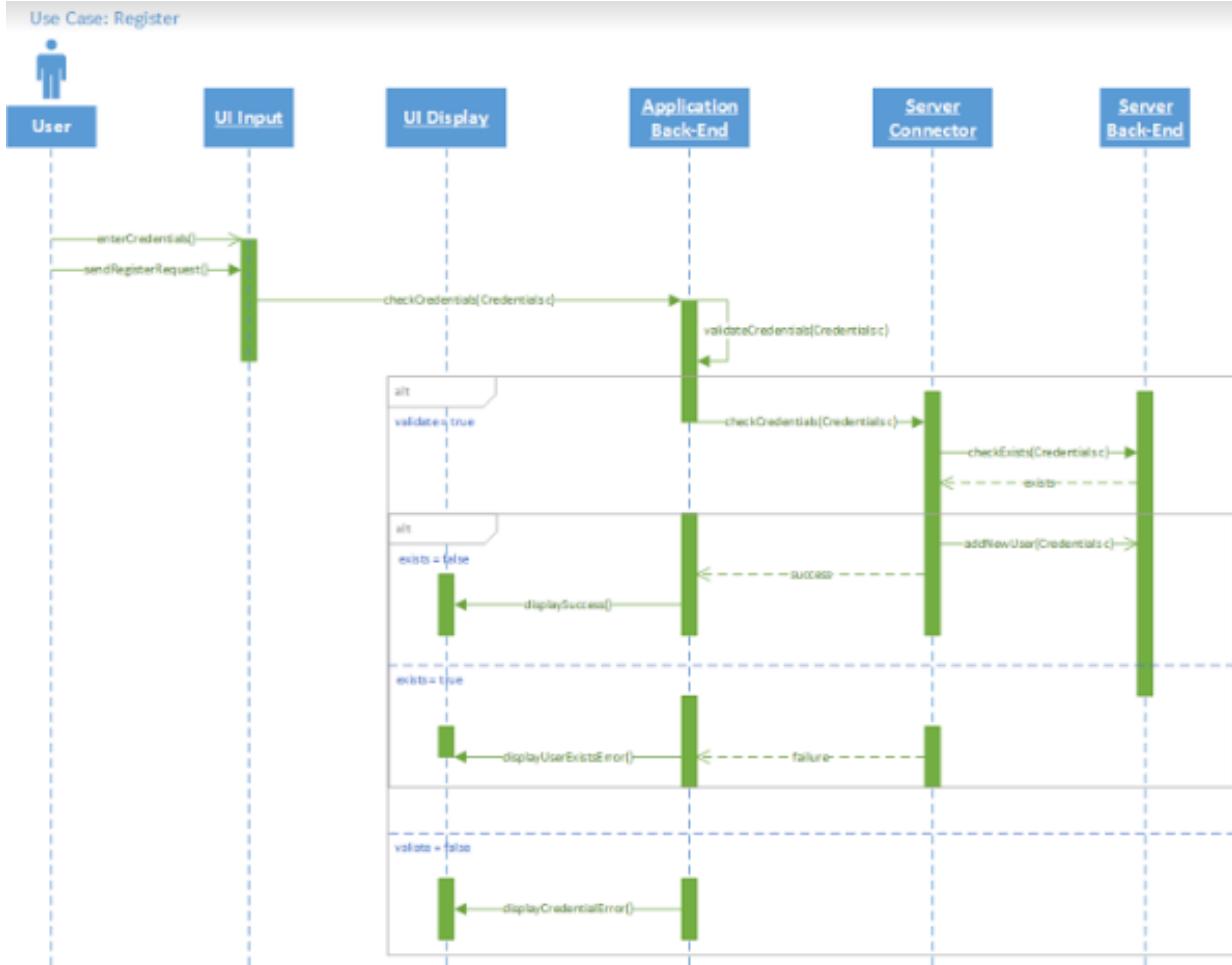
**Respond to Request Sequence Diagram:** After the user selects the request they wish to respond to and submits their response, the system sends a message back to the requester, sets the visibility of the selected requested assessment



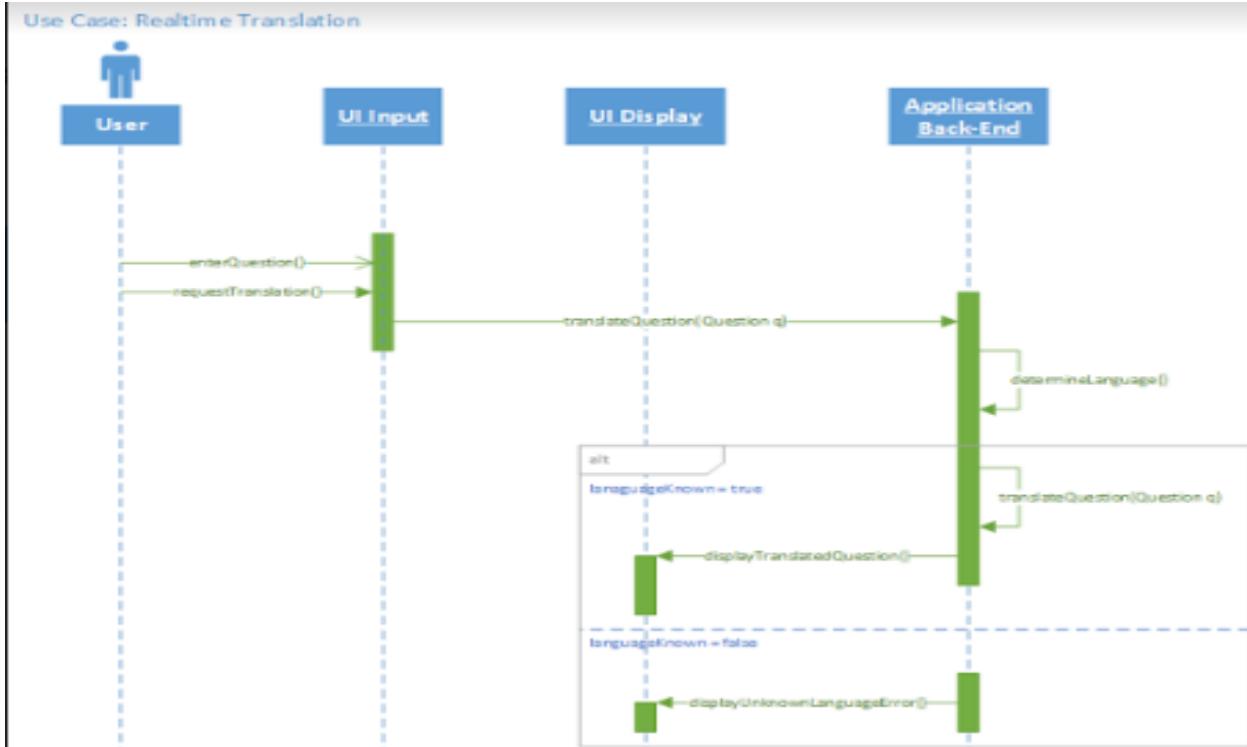
**Request Assessment Sequence Diagram:** After the user requests to send a request, the system will ask for the student information. Once student information is inputted, it is validated to see if the student exists. If the student exists, the request is sent to the student and a success message is displayed.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



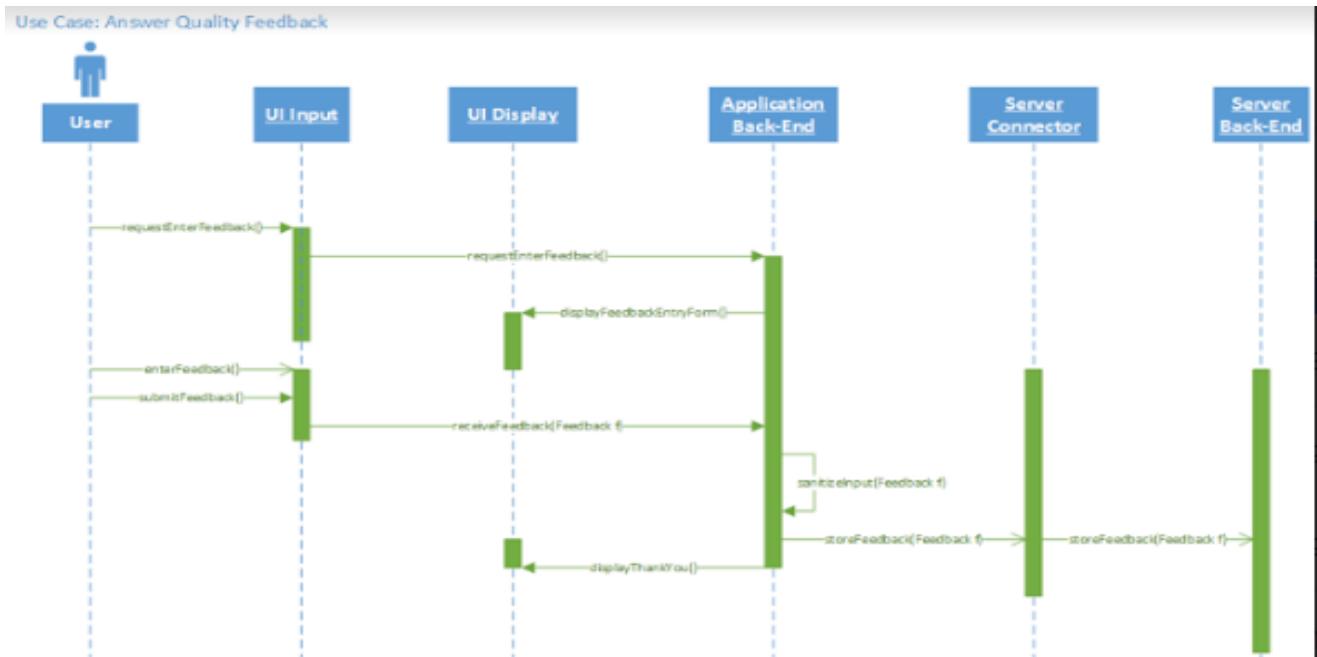
**Logout Sequence Diagram:** After a user requests to logout, the system logs the user out of the server and displays SUCCESS.



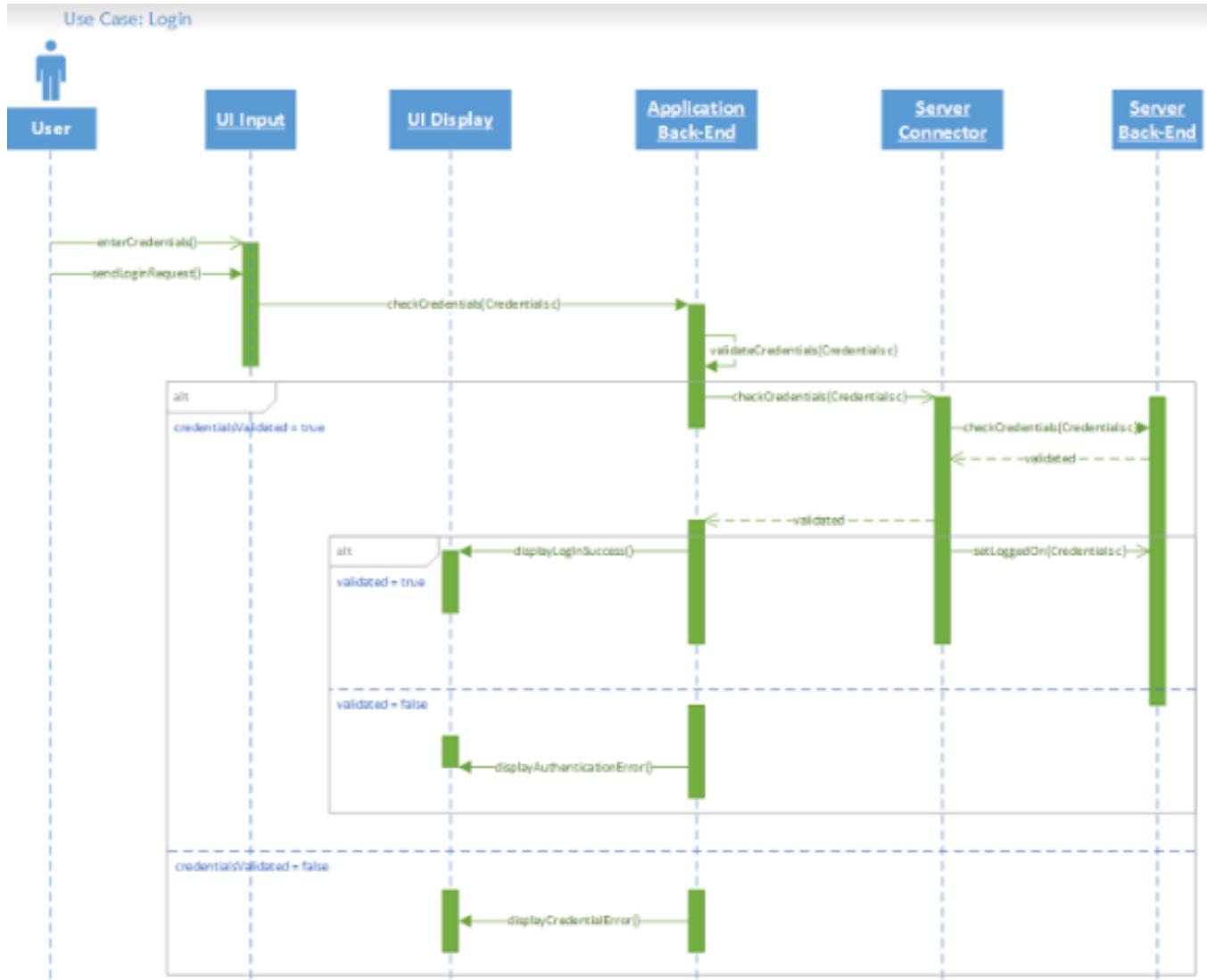
**Register Sequence Diagram:** After the user enters credentials and requests to register, the system checks the credentials to see if they meet the requirements (email format, correct number of characters, etc.). If valid, the system checks if a user with the same email already exists. If a user does 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.



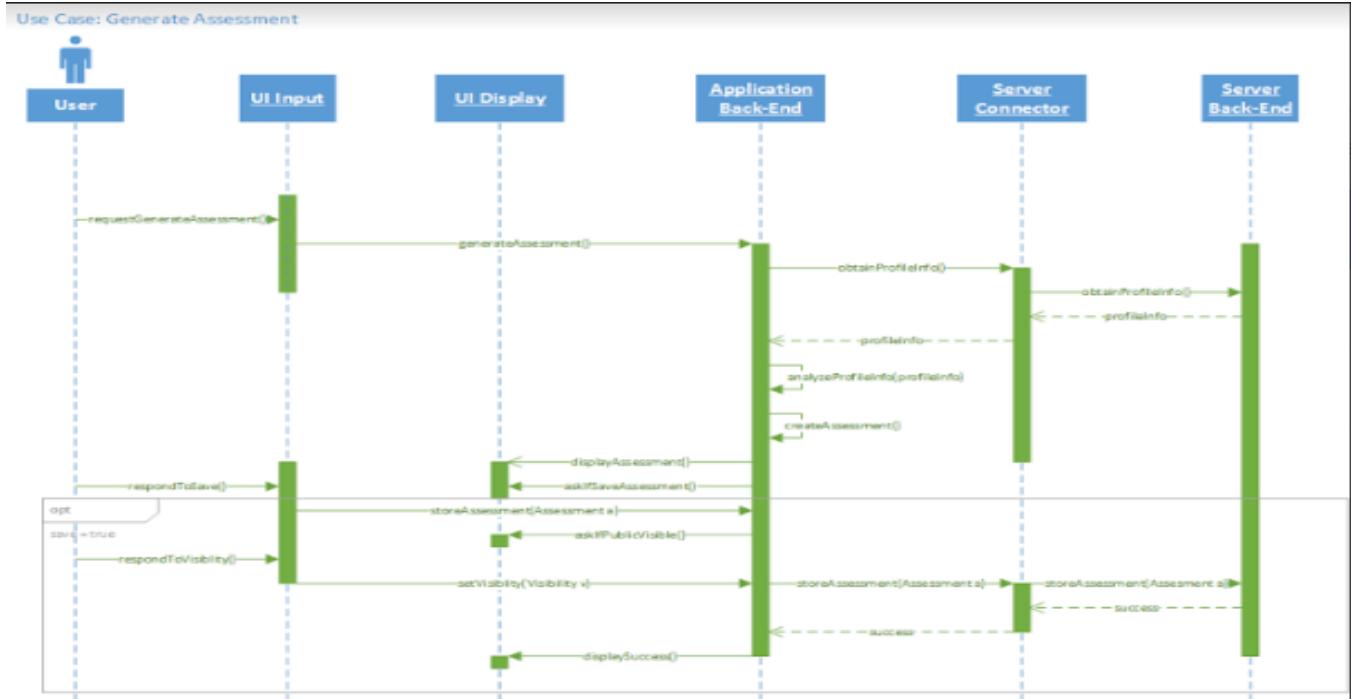
**Real-time Translation Sequence Diagram:** After the user enters the question and requests a translation, the system determines the language of the question. If it can be translated, the translated text is then displayed. Otherwise, an error message is shown.



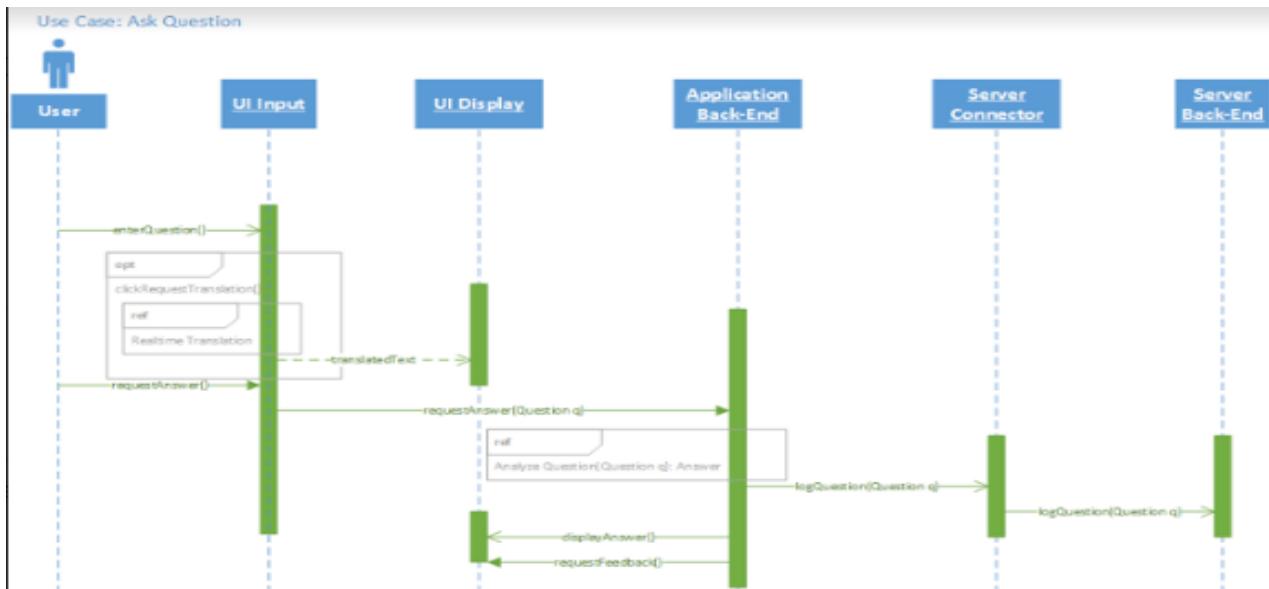
**Answer Quality Feedback Sequence Diagram:** After a user requests to enter feedback, an entry form is displayed for the user to input the feedback. Once submitted, the system checks to make sure the input is not dangerous, and then stores it in the database, and finally displays a thank you message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



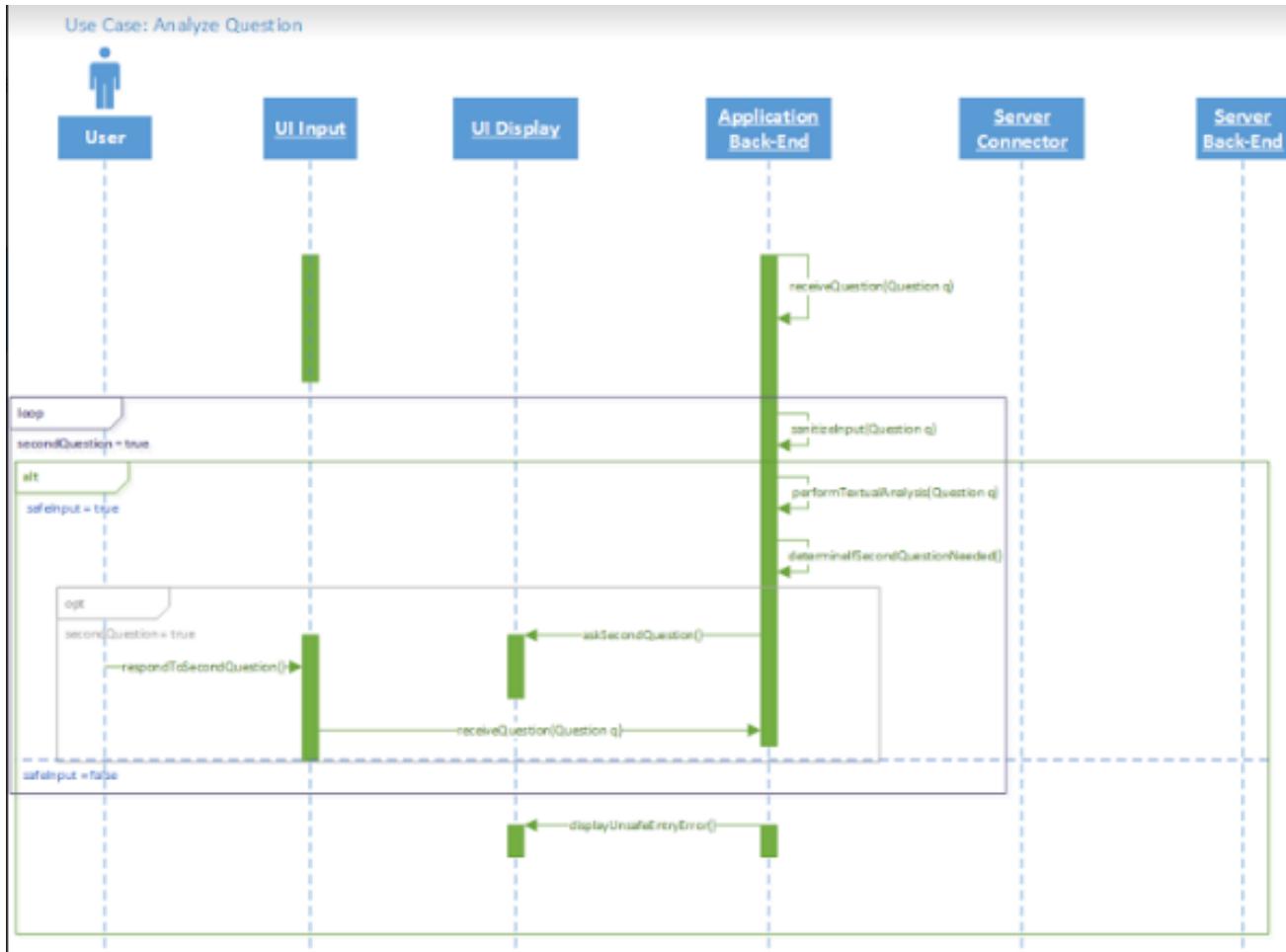
**Login Sequence Diagram:** After the user enters their credentials and sends a login request, the system checks the credentials to see if they meet requirements first, and then checks to see if they are valid. If the credentials meet requirements and are valid, the user is logged on and a success message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

**Table 4.43: System NonFunctional Requirements: 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

**Table 4.44: System NonFunctional Requirements: 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

**Table 4.45: System NonFunctional Requirements: 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

**Table 4.46: Summary of Requirements:** This table shows an overview of all the requirements that were gone over above.

# 5. Exploratory Studies

## 5.1. Relevant Techniques

Our team distributed a survey, containing both qualitative and quantitative questions, to 150 students in an introductory Computer Science course. Our intention was to receive feedback relevant to the thoughts and decision-making processes of a student who has recently completed high school and is considering a degree and career in Computer Science, Software Engineering, or Computer Engineering. We also accepted the input of upper level classmen with the belief that their input could provide important feedback about the information they have gathered and decisions they would have made, in hindsight.

- 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?

While many of our questions were based on the responses of the surveys, some students did not return their survey, some students did not ask five questions, and many students repeated questions asked by their peers.

$$(150 \text{ surveys} \times \frac{5 \text{ questions}}{1 \text{ survey}} - \text{duplicate questions}) \leq 750 \text{ unique questions}$$

*This is a generous calculation of how many questions we were able to create based on the surveys.*

Many additional questions were created through research conducted on the internet. Watson will need to be trained with at least 1000 question-answer pairs to ensure depth and accuracy of system knowledge. Additionally, Watson will be trained with variations of each question, to improve accuracy, should a user of our system phrase a question in a different manner than the question originally asked. References 18-52 were used to generate both questions and answers for our domain.

- The following are examples of question-answer pairs:
  - ◆ Q: Does Penn State Behrend assist students in finding internships and employment opportunities?

A: Penn State Behrend holds a career fair once a semester where a hundred or so different companies come and you can speak with them. In addition, once you post your resume to Nittany Lion Career Network, the career center will start sending your resume out to potential employers,

who will then contact you without you having contacted them yourself. Kinda nifty, in my opinion.

- ◆ Q: Do Computer Engineers get hired by companies like Google, Microsoft, Amazon, and Facebook?

A: Yes. All four of those companies require computer engineers, but they have much more rigorous interviews than most companies. You can get hired, you just need to be very well prepared for the interviews.

Our system will use keyword-concept extraction to update the frequency of each question being asked. This will continue recording into the database, for future analytical use. Our system will use machine learning algorithms to format, structure, and extract data in order to update question-answer tracking and add new useful information, if it is not currently in our knowledgebase. Similarly, by crawling data from user feedback, we can track users' questions to check whether they are related to our knowledge domain, and either update the related information or add new data from the user input.

#### → Data Crawling

- ◆ Question Log: update frequency of each asked question, extract new keyword and concepts to add to our knowledge domain.
- ◆ Suggested Feedback: answers for unclear questions - updating better answer to our knowledge domain.
- ◆ Analysis Data: data gleaned from user's input.

Our system should be able to interpret questions delivered in Natural Language. We will use tags and information extraction to ensure that the user's question is being addressed and understood properly. For example, our system will understand that the word "what" is indicative of a question. Our system should also be able to comprehend "questions" asked in a non-standard manner. For example, an input of "Penn State retention rate" should be interpreted as a question rather than a statement.

#### → Natural Language Processing (NLP) [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

Our documents must be curated to ensure that the information our system is providing is reliable and accurate. This can be done by feeding a variety of informational documents to our system and allowing the system to verify the information's validity through comparison of the sources.

#### → Document Curating [10]

- ◆ Exact match search

- ◆ Wildcard search
- ◆ Proximity search

We plan to use a basic exhaustive search and complete training with fixed training data (our knowledge base). Our system will use a natural language classifier to determine what information is useful. This can be substituted by the Conversation API from Watson.

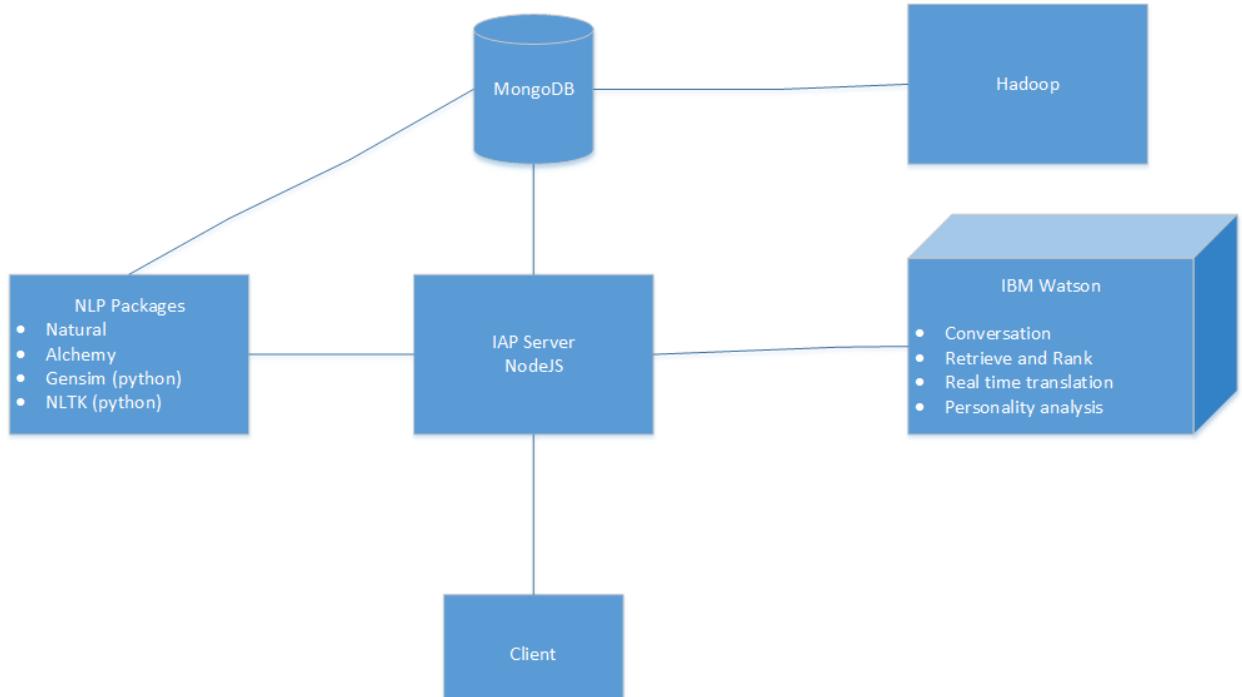
- Machine Learning Algorithms [11]
  - ◆ Supervised Learning
    - Linear Regression
    - Decision Tree (classification)
  - ◆ Unsupervised Learning
    - Gensim (python)  
Use to compare string similarity
    - NLTK (python)  
For data mining purpose
    - MonkeyLearn (Web)  
Use web service to build automated training model

## 5.2. Relevant Packages/Products

Watson services are accessed by APIs usually programmed in the CURL, Java, or NodeJS languages. Since this project is a web application, NodeJS has more native support on the server side, and faster performance in general execution and implementation. A NodeJS server will call BlueMix RESTful Watson APIs on server usage and query MongoDB natively in the format of JavaScript. The application will have NLP handling and analysis functionalities embedded. The development team could also use Hadoop for analysis, since it is especially good for managing Big Data. Since the intention is to maximize usage of Watson services, the best option for the team is to use Alchemy. MongoDB will store the data of three perspectives: user account, user question log, and analysis of all kinds of records that are needed by administration.

- NodeJS<sup>[12]</sup>
- ◆ 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<sup>[5]</sup>
  - Mainly used to classify the question input from user and determine if the question is in our domain or not.
- ◆ Retrieve and rank<sup>[4]</sup>:
  - 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<sup>[15]</sup>
  - Allows instant translation between common languages.

- Displays desired language based on the user's account preference.
  - ◆ Personality analysis<sup>[16]</sup>
    - 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<sup>[14]</sup>
  - ◆ 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<sup>[2]</sup>
    - A mature public accessible API provider to do NLP analysis
  - ◆ Natural<sup>[9]</sup>
    - Equivalent NodeJS library of NLTK (Natural language toolkit) in python. It mainly focuses on customized NLP classification.
  - ◆ Alchemy<sup>[3]</sup>
    - 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.



Relevant packages/products flow diagram

### 5.3. Broader Impacts

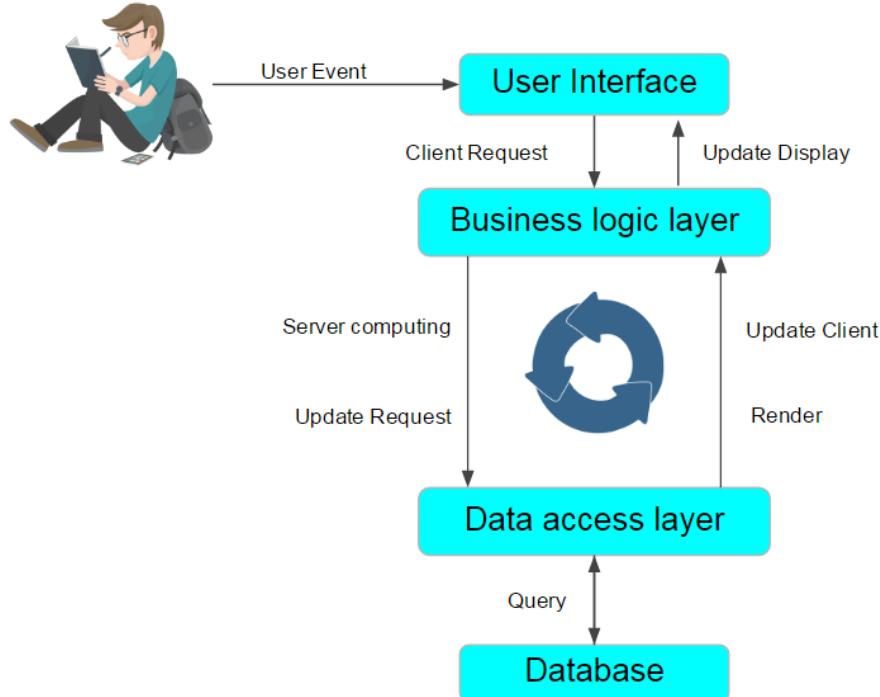
This project can be expanded to include other majors within Behrend or branches of Penn State University. On a grand scale, this type of tool would be beneficial for high school juniors and seniors, as well as college freshman and anyone participating in the role of student advisement.

# 6. System Design

## 6.1. Architectural Design

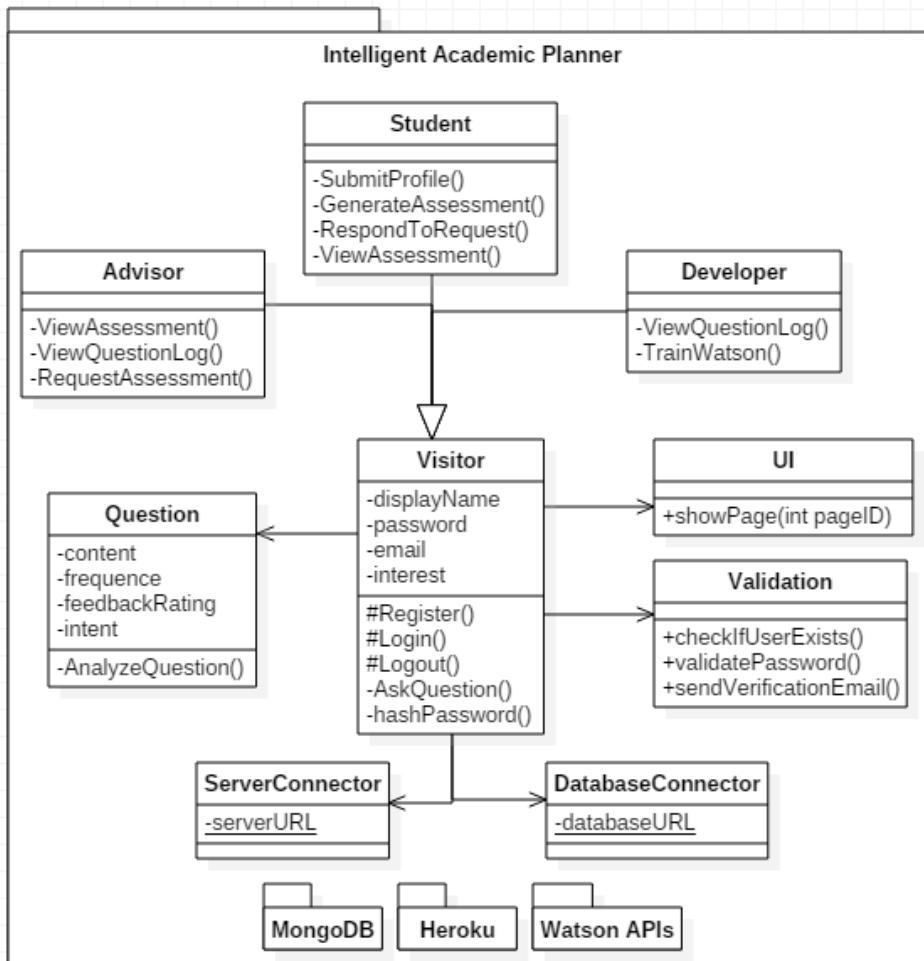
### → Layered Architecture

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



*User initializes a “submit question” event on the web browser, then the server will handle the question through the logic layer. The database can then be queried and, if there are no exceptions, the data will be sent to Watson through BlueMix RESTful APIs. Finally, the response information will be returned in the reverse direction.*

## 6.2. Structural Design



This is the class diagram of our system. We are using Heroku as our server and MongoDB as our database. All visitors have the ability to ask questions, and questions are stored on our server. The Validation class is used to check for safe and correct inputs, improving security of our system.

## 6.3. User Interface Design

The user interface is split into several web pages: Ask Questions, Login & Register, and User Profile. The design is responsive and can be viewed on both desktop and mobile devices. A navigation bar is present on each page to ensure easy navigation.

The screenshot shows the 'Ask Questions' page of the Intelligent Academic Planner. At the top, there's a search bar with the placeholder 'What should I minor in?'. Below it, a blue header bar says 'Answers'. The main content area displays three numbered answers to the question 'What should I minor in?'. Answer 2 includes a link to a website. To the right of the answers is a 'Question Feed' sidebar containing links to various questions like 'Why study computer science?' and 'What is software engineering program?'. A 'Login/Register' button is at the top right.

**Ask Questions:** This is the landing page of our site. A user can ask a question in the search bar, using voice-to-text or textual input. The answers are displayed in list form, with the ability to “Favorite” a preferred answer and give the system feedback through a 5 star rating system. A question feed column is placed along the side of the page to inform users of recently asked questions and to allow them to view their responses.

The screenshot shows the 'Login & Register' page. At the top, there's a search bar with the placeholder 'Eg. What is Computer Science?'. Below it, the page is divided into two main sections: 'Login to our site' on the left and 'Sign up now' on the right. The 'Login to our site' section contains fields for email and password, and a 'Sign in!' button. The 'Sign up now' section contains fields for first name, last name, and email, along with a list of benefits: Personality Assessment, Easy Question Retrieval, and Advising Assistance. A 'Login/Register' button is at the top right.

**Login & Register:** A search bar is at the top of the page to allow a user to ask a question from any page. The user must register with their name, an email, and password. The user must also choose an account type of Student, Advisor, or Developer. A user can register with their Facebook, LinkedIn, or Google account.

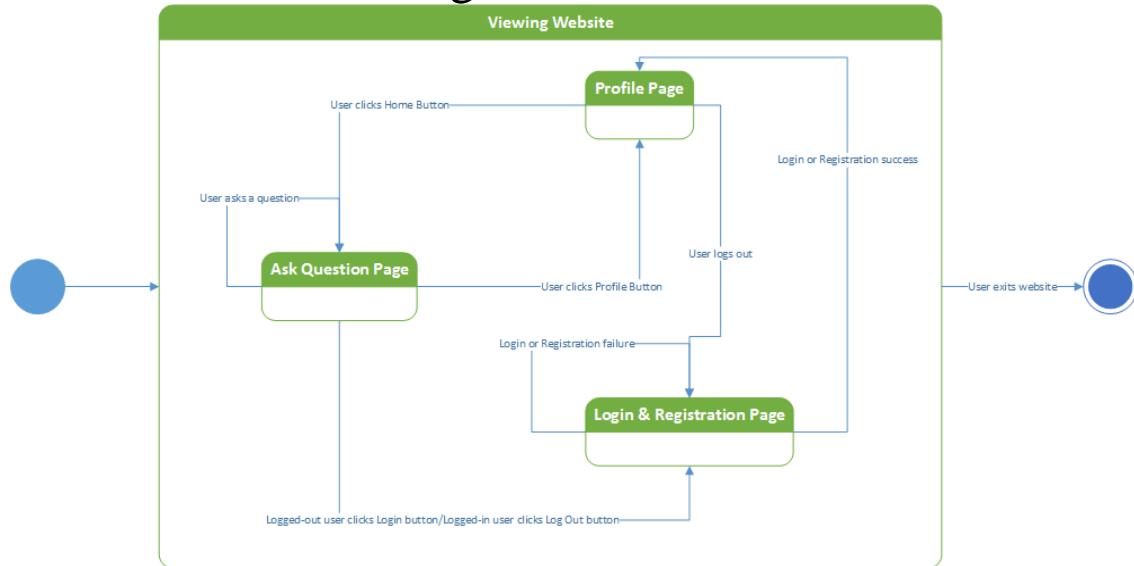
The screenshot shows the 'Profile' page of the Intelligent Academic Planner. At the top, there's a search bar with the placeholder 'Eg. "What is Computer Science?"'. Below it, a sidebar on the left contains sections for 'Your words' (with options to 'Ask more questions', 'Add self introduction', and 'To explore interests!'), 'Introduction' (with a 'Upload self-description' field), and a 'Question Feed' section. On the right, a user profile for 'Krystal Elliott' is displayed, showing her email (kye5098@psu.edu), a placeholder profile picture, and buttons for 'Change Avatar', 'Inbox', and 'Generate Assessment'.

*Profile: Only available after login, display information about the user, including: a conglomeration of words describing the user's interests, a profile picture, their favorited questions, and an introduction written by the user about the user. This page also features the Question Feed and Search Bar.*

The screenshot shows the 'Ask Away!' page of the Intelligent Academic Planner. It features a large search bar at the top with the placeholder 'Tell me about Behrend.' Below it, a main content area has a blue header 'Answers'. Two numbered answers are listed: '1. Tell me about Behrend.' and '2. Well I don't know how many graduate with a job offer because no one tells me that, I'm just a lonely computer. But! I know that 81% of students feel confident that they will find a job in their field after graduation, and that 94% are employed within 2 years!'. The second answer includes a five-star rating icon.

*Responsive Design: This is an example of the view of the Ask Question page on a mobile device.*

## 6.4. Behavioral Design



*This is the state diagram for the web flow of the website where users ask the questions. Users will begin on the Ask Question page. If the user clicks the login/register button, they will be taken to the login/register page. When login or registration completes, they will be taken to their profile page. From any location in the system, if they click on the home button, they will be taken to the Ask Question page and if they click on the logout button, they will be taken to the login page.*

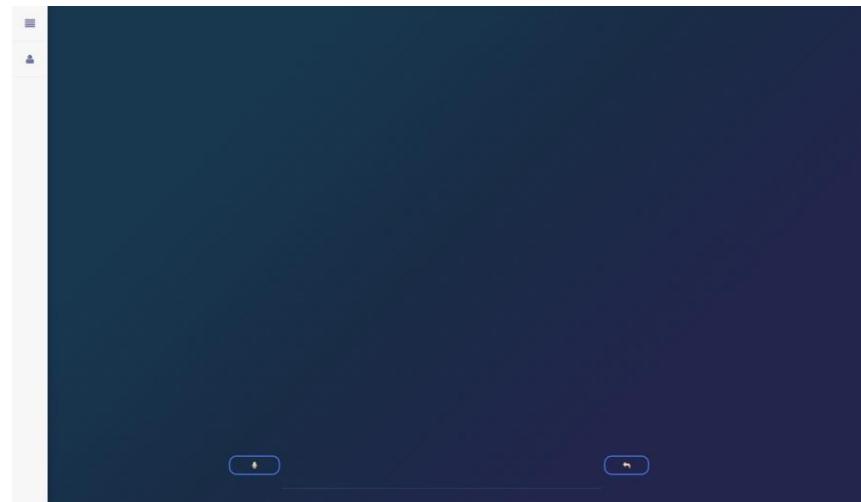
## 6.5. Design Alternatives & Decision Rationale

### UI Alternative Designs

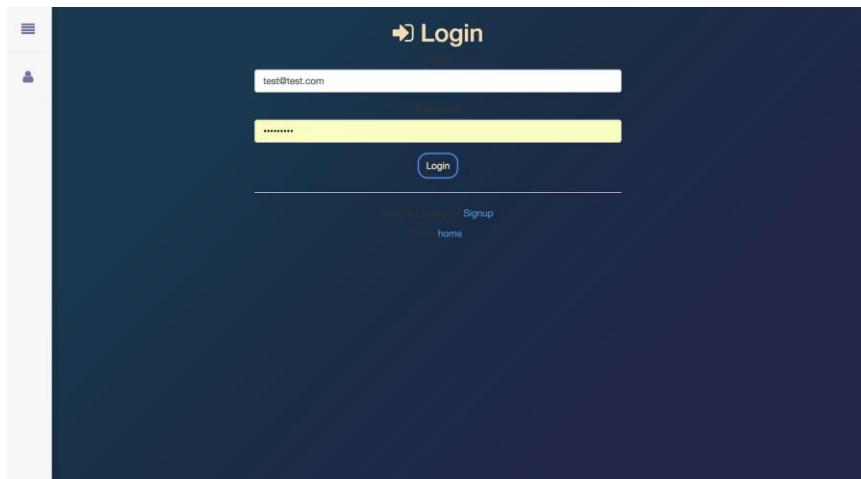
*These are alternative user interfaces.*

#### Version 1.0

*The view will be divided by 2 main sections, left side navigation of functionalities and right section of display/interactions. The user enters the question on the line at the bottom and either hits enter or the arrow button to submit the question. As the question is being answered, a loading icon appears. The user can login or view their profile using the buttons on the left side.*



*Ask Question: use mobile compatible view design template*



*Login: require register email and password*

The screenshot shows a dark-themed 'Signup' page. At the top center is a logo with a person icon and the word 'Signup'. Below it is a form with three input fields: 'Test User Name', 'test@test.com', and '.....'. A 'Signup' button is located at the bottom right of the form. At the very bottom of the page, there is a link to 'Login'.

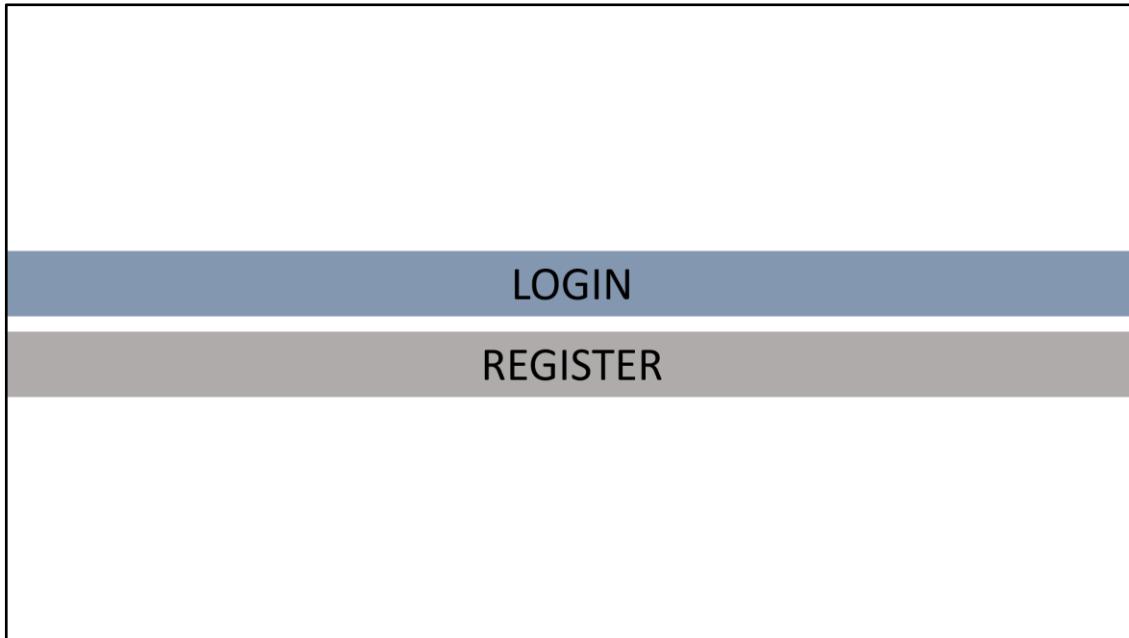
*Register Account: require a display name, register email, and password*

The screenshot shows a dark-themed 'Profile Page' with a blue header bar containing a person icon and the word 'Profile Page'. Below the header is a 'Logout' button. The main content area contains a box labeled 'Local' with account details: 'id: 5796820b61d49170f748fa3', 'email: xiaoyuz2011@gmail.com', and 'name: xiaoyu zhou'.

*Profile: only available after login, display regular account information*

## Version 2.0

The color scheme is much brighter and it includes additional screens, which have not been included in the original UI design. This design is fairly simplistic and would allow for a user to access the project via their mobile device, as well.



*Landing Page:* offers options of logging in and registering

A detailed screenshot of the "LOGIN" screen. The top section is identical to the landing page. Below it is a white form area with the following fields:

- Email: An input field.
- Password: An input field.
- Re-enter Password: An input field.
- First Name: An input field.
- Last Name: An input field.
- Account Type: A dropdown menu with three options: "student" (graduation cap icon), "advisor" (apple icon), and "developer" (computer monitor icon).

At the bottom of the form is a grey "SUBMIT" button. Below the form is a grey horizontal bar containing the word "REGISTER" in white capital letters. At the very bottom of the screen, there is a navigation bar with icons for back, forward, and search.

*Registration:* collects user data to create an account

## LOGIN

Email

Password

[forgot password?](#)

## REGISTER

[Create Account](#)

*Login: allows user to login*

ASK A QUESTION [kye5098@psu.edu](#) logout

## ACCESS

[PROFILE](#) [ASSESSMENT](#) [QUESTIONS](#)



**Krystal Elliott**

**Extracurricular Interests**

**Academic and Professional Goals**

*User Profile: allows user to create a personal profile, complete with picture, interests, and goals*

ASK A QUESTION [kye5098@psu.edu](#) logout

## ACCESS

PROFILE ASSESSMENT QUESTIONS

**Recommendations**

Major	Computer Engineering
Minor	Software Engineering, Business

**Personality Assessment**

Chief among ENFPs' talents is their people skills, a quality that is even more valuable now than ever. Even in traditional Analyst strongholds like engineering, systems analysis and the sciences, ENFPs' ability to network and match the communication styles of their audience means that even as they explore new challenges on their own, they will be able to work with others, explore others' perspectives and glean new insights into their projects. Much of modern progress stems from incorporating other studies into typically disassociated fields, and no one is better equipped to merge broad interests than talented, energetic and future-minded ENFPs.

**Export**

Profile recipient email address  
 Assessment djc5558@psu.edu

**Assessment:** allows user to view their personalized assessment and to send it, partially or in full, to an email address.

ASK A QUESTION [kye5098@psu.edu](#) logout

## ACCESS

PROFILE ASSESSMENT QUESTIONS

**How much does a Software Engineer earn?** X

According to the U.S. Bureau of Labor Statistics, in 2012, the average salary for an application software developer was \$93,000, with only 10% of such developers making more than \$139,000 in salary. Clearly, then, any Google engineer making \$3 million per year is getting most of that in bonuses and/or stock.

**How do I apply for financial aid?** X

The first step in being considered for financial assistance is to complete the Free Application for Federal Student Aid (FAFSA). We highly recommend all students complete the FAFSA on the Web (link is external) to ensure the quickest and most accurate processing. FAFSA opens each year on October 1st. Students are automatically considered for most scholarships, grants and loans upon completing the FAFSA.

**Previous Question Log:** displays questions previously asked by users and their preferred answers

The screenshot shows a web-based application interface for asking and answering questions. At the top, there's a grey header bar with the word "ACCESS" on the left and user information "kye5098@psu.edu logout" on the right. Below this is a blue header bar with the text "ASK A QUESTION". Underneath is a search bar containing the question "How much does a Software Engineer earn?". To the right of the search bar is a magnifying glass icon.

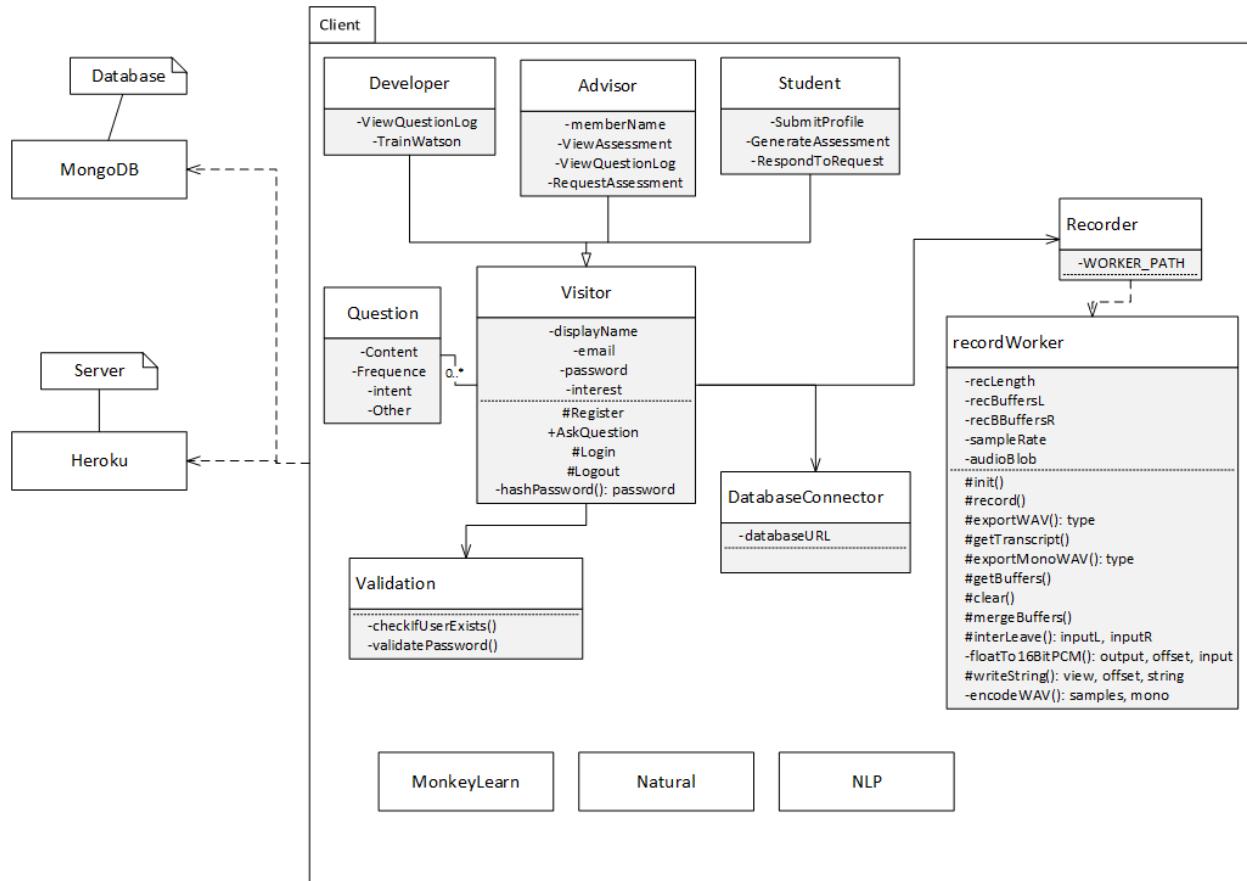
The main content area displays three numbered answers:

- 1 According to the U.S. Bureau of Labor Statistics, in 2012, the average salary for an application software developer was **\$93,000**, with only 10% of such developers making more than \$139,000 in salary. Clearly, then, any Google engineer making \$3 million per year is getting most of that in bonuses and/or stock. ★★★★★
- 2 Software Engineer salaries, Software Engineer benefits packages, Software Engineer bonuses, Software Engineer job descriptions, Software Engineer statistics and Software Engineer job openings. Please select a specific Software Engineer job from the list below for additional information or search Software Engineer salaries. ★★★★★
- 3 The Labor Department reports that software developers made a median salary of \$95,510 in 2014. The highest-paid 10 percent in the profession earned \$149,480 in 2014, while the lowest-paid earned \$56,310. The computer systems design industry and software publishers employ the highest number of software engineers, but the highest-paid positions are in the San Francisco Bay Area cities San Jose and San Francisco, as well as in Seattle. ★★★★★

A vertical scroll bar is visible on the right side of the answer list.

**Ask Question Interface:** allows user to ask questions, rate the received answers, and select one preferred answer

## Structural Alternative Designs



- This design was not chosen because it places a lot of emphasis on the recorder and recordWorker. In addition, they are used for the voice-to-text functionality that we decided to not include.

## Architectural Alternative Designs

- No alternative considered.

## Behavioral Alternative Designs

- No alternative considered.

# 7. System Implementation

## 7.1. Programming Languages & Tools

- BlueMix (conversation, retrieve and rank, AlchemyAPI,...)<sup>[3][4][5]</sup>
  - ◆ Input will be parsed by the server, then go into Conversation service to be guided by default classifier, then, if no exception, the question will be answered by retrieve and rank. In parallel, the question and answer will be analyzed by Alchemy, and placed into the database.
- MongoDB<sup>[14]</sup>
  - ◆ The MongoDB is mainly used to maintain user information, question and answer information, and system analysis.
- Node.js<sup>[12]</sup>
  - ◆ Server application type, programming language in JavaScript style. For production driven development, all libraries will update to the newest version. And programming style will be in industrial standard.

## 7.2. Coding Conventions

- Use ES6 standard for JavaScript

## 7.3. Code Version Control

- GitHub<sup>[13]</sup>

## 7.4. Implementation Alternatives & Decision Rationale

- PHP for Node.JS
  - ◆ NodeJS also has the advantage of an easy setup in a local development environment, which is preferred by the development team. Allen has had some experience developing with NodeJS in the past, so he will be able to guide his teammates, should they have questions or need guidance.
- MySQL for MongoDB
  - ◆ Although MySQL is more familiar to all of us, traditional databases are not efficient and flexible enough to handle unstructured and big data which in our project we will constantly deal with.
- Project Oxford for BlueMix
  - ◆ Since Dr. Su is offering a cognitive system course this semester which Allen is taking, and because Allen already did some projects with BlueMix, it's better for us to select BlueMix as our main tool.
- Mobile platform for web platform
  - ◆ None of the team members have experience in mobile development, so for the sake of quality, we will not attempt to do any mobile native development.

## 7.5. Analysis of Key Algorithms

- Correlation between input question and answer
- Question frequency analysis

```
void updateFreq(questionStr)
    keyConcept <- extractConcept(questionStr)
    foreach question from questionBank
        If question.containsConcept(keyConcept)
            question.ask.frequency ++
```

- Answer quality automate improvement
- Concept extraction from question

```
conceptBank = [...] //all stored keyword from question and answers
Bool containsConcept(inputStr)
    foreach concept in conceptBank
        If string.distance(inputStr, concept) > 0
            return true
        return false
```
- Keyword mapping from whole session

# 8. System Testing

## 8.1. Test Automation Framework

### Steps for Installing Mocha

- Install the new package using the command `npm install --global mocha --save`
- Modify package.json's scripts to include “`test`”: “`mocha`”
- Create test file directory with command `mkdir test`
- Edit test script with command `$EDITOR test/test.js`

### Steps for Running Test Cases

- Type command `npm test`

## 8.2. Test Case Design

Test Cases can be found in Appendix T

### Test Suites

**Table 8.2.1: Test Suite TS-001: Unit Tests:** Summary of all Unit Tests. Unit tests check for functionality of specific parts of the code. In these tests, we make sure that the expected result occurs under the specified conditions. For instance, if an invalid password were to be entered, we expect that a person would not log in. This would pass a test.

### Unit Test Cases

**Table 8.2.2 Test Case TC-001:** Testing if a user can properly log in. We expect that when a correct username and password is inputted, the user should be logged in.

**Table 8.2.3 Test Case TC-002:** Testing if a user can properly log out. We expect that when a user requests to log out, they should be logged out with no inputs needed.

**Table 8.2.4 Test Case TC-003:** Testing if a user can properly register. We expect that when a user inputs an untaken email and password they should be entered into the database properly.

**Table 8.2.5 Test Case TC-004:** Testing if a user can properly ask a question. We expect if a question is asked then the question should be logged to the database for records and a proper answer should be displayed.

**Table 8.2.6 Test Case TC-005:** Testing if a user can properly provide feedback. We expect when a user enters feedback, the feedback should be stored and a thank you message should be displayed to the user.

**Table 8.2.7 Test Case TC-006:** Testing if a user can properly translate text. We expect that if the user enters non-English text, the system should display the text in English.

**Table 8.2.8 Test Case TC-007:** Testing if the system can properly analyze a question. We expect that if a user asks a question that is ambiguous, the system should respond with another question.

**Table 8.2.9 Test Case TC-008:** Testing if a user can properly submit a profile. We expect that when new profile information is entered, the information should be stored to the database.

**Table 8.2.10 Test Case TC-009:** Testing if a user can properly generate an assessment. We expect that if descriptive text is given, the user requests to save their assessment, and they say yes to sharing with advisors then the assessment should be saved to the database with true visibility.

**Table 8.2.11 Test Case TC-010:** Testing if a user can properly respond to a request. We expect that if the user responds yes, the assessment that was requested should be visible by the person that requested it.

**Table 8.2.12 Test Case TC-011:** Testing if a user can properly view an assessment. We expect that the assessment should be displayed when the user requests it to be.

**Table 8.2.13 Test Case TC-012:** Testing if a user can properly train Watson. We expect that when a new question and answer pair is trained, the system should give the proper answer when the question is asked.

**Table 8.2.14 Test Case TC-013:** Testing if a user can properly log in. We expect that when a correct username but an incorrect password is inputted, the user should not be logged in.

**Table 8.2.15 Test Case TC-014:** Testing if a user can properly log in. We expect that when an incorrect username but a correct password is inputted, the user should not be logged in.

**Table 8.2.16 Test Case TC-015:** Testing if a user can properly log in. We expect that when both an incorrect username and an incorrect password are inputted, the user should not be logged in.

**Table 8.2.17 Test Case TC-016:** Testing if a user can properly respond to a request. We expect that if the user responds no, there are no changes to the assessment.

**Table 8.2.18 Test Case TC-017:** Testing if a user can properly register. We expect that when a user inputs an untaken email and an invalid password they should not be entered into the database and an error message is displayed.

**Table 8.2.19 Test Case TC-018:** Testing if a user can properly register. We expect that when a user inputs a taken email and a valid password they should not be entered into the database and an error message is displayed.

**Table 8.2.20 Test Case TC-019:** Testing if a user can properly register. We expect that when a user inputs a taken email and an invalid password they should not be entered into the database and an error message is displayed.

**Table 8.2.21 Test Case TC-020:** Testing if a user can properly generate an assessment. We expect that if descriptive text is given, the user requests to save their assessment, and they say no to sharing with advisors then the assessment should be saved to the database with false visibility.

**Table 8.2.22 Test Case TC-021:** Testing if a user can properly generate an assessment. We expect that if descriptive text is given and the user says no to saving the assessment then the assessment should not be saved to database.

**Table 8.2.23 Test Case TC-022:** Testing if a user can properly view the question log. We expect that when they ask to view the question log, the question log should be displayed.

**Table 8.2.24 Test Case TC-023:** Testing the speed of ask question. We expect that when a user asks a question then the answer should be displayed within 5 seconds.

**Table 8.2.25 Test Case TC-024:** Testing the management of a user's session. We expect that if the user is inactive for an hour or more, they will be logged out automatically by the system.

## Integration Test Cases

Integration tests are being done manually.

## System Test Cases

No test cases have been designed for the system at this time

## Acceptance Test Cases

No test cases have been designed for user acceptance at this time.

## 8.3. Test Execution Report

Test Execution Reports can be found in Appendix TE

### Unit Testing Report

**Table 8.3.1. Execution Report of Test Case TC-001:** Test Execution Report for testing log in. This functionality has been implemented and is passing tests.

**Table 8.3.2. Execution Report of Test Case TC-002:** Test Execution Report for testing log out. This functionality has been implemented and is passing tests.

**Table 8.3.3. Execution Report of Test Case TC-003:** Test Execution Report for testing register. This functionality has been implemented and is passing tests.

**Table 8.3.4. Execution Report of Test Case TC-004:** Test Execution Report for testing asking a question. This functionality has been implemented, but an automated test has yet to be created.

**Table 8.3.5. Execution Report of Test Case TC-005:** Test Execution Report for testing feedback. This functionality has been implemented, but an automated test has yet to be created.

**Table 8.3.6. Execution Report of Test Case TC-006:** Test Execution Report for testing real-time translation. This functionality has been implemented, but an automated test has yet to be created.

**Table 8.3.7. Execution Report of Test Case TC-007:** Test Execution Report for analyzing a question. This functionality has been implemented, but an automated test has yet to be created.

**Table 8.3.8. Execution Report of Test Case TC-008:** Test Execution Report for submitting a profile. This functionality has been implemented, but an automated test has yet to be created.

**Table 8.3.9. Execution Report of Test Case TC-009:** Test Execution Report for generating an assessment and saving it visible to advisors. This functionality has not yet been implemented and is therefore failing tests

**Table 8.3.10. Execution Report of Test Case TC-010:** Test Execution Report for requesting to view an assessment. This functionality has not yet been implemented and is therefore failing tests

**Table 8.3.11. Execution Report of Test Case TC-011:** Test Execution Report for viewing assessment. This functionality has not yet been implemented and is therefore failing tests

**Table 8.3.12. Execution Report of Test Case TC-012:** Test Execution Report for testing training Watson. This functionality has been implemented and is passing tests.

**Table 8.3.13. Execution Report of Test Case TC-013:** Test Execution Report for logging in with an invalid password. This functionality has been implemented and is passing tests.

**Table 8.3.14. Execution Report of Test Case TC-014:** Test Execution Report for logging in with an incorrect username. This functionality has been implemented and is passing tests.

**Table 8.3.15. Execution Report of Test Case TC-015:** Test Execution Report for logging in with an invalid password and incorrect username. This functionality has been implemented and is passing tests.

**Table 8.3.16. Execution Report of Test Case TC-016:** Test Execution Report for requesting to view an assessment and it is denied. This functionality has not yet been implemented and is therefore failing tests

**Table 8.3.17. Execution Report of Test Case TC-017:** Test Execution Report for registering with an invalid password. This functionality has been implemented and is passing tests.

**Table 8.3.18. Execution Report of Test Case TC-018:** Test Execution Report for logging in with a taken email. This functionality has been implemented and is passing tests.

**Table 8.3.19. Execution Report of Test Case TC-019:** Test Execution Report for logging in with an invalid password and taken email. This functionality has been implemented and is passing tests.

**Table 8.3.20. Execution Report of Test Case TC-020:** Test Execution Report for generating an assessment that is requested to be saved but not shared. This functionality has not yet been implemented and is therefore failing tests

**Table 8.3.21. Execution Report of Test Case TC-021:** Test Execution Report for generating an assessment but not saving it. This functionality has not yet been implemented and is therefore failing tests

**Table 8.3.22. Execution Report of Test Case TC-022:** Test Execution Report for viewing question log. This functionality has not yet been implemented and is therefore failing tests

**Table 8.3.23. Execution Report of Test Case TC-023:** Test Execution Report for speed of asking a question. This functionality has been implemented, but an automated test has yet to be created

**Table 8.3.24. Execution Report of Test Case TC-024:** Test Execution Report for logging a user out after 1 hour of inactivity. This functionality has been implemented, and it is being tested manually rather than with an automated test. However, the test is currently failing, so a bug exists.

## Integration Testing Report

Integration Tests are being done manually.

## System Testing Report

No test cases have been designed for the system at this time

## Acceptance Testing Report

No test cases have been designed for user acceptance at this time

# 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, so 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.
- Getting enough questions to be able to train Watson properly.
- Utilize AlchemyAPI to preprocess user input in order to increase system accuracy
- Optimize user feedback/system self-learning functionality

## 9.2. Challenges Faced in System Development

- Build and automate the backend system for training with retrieve and rank
- Build workable model to analyze questions and answer accuracy
- Try to utilize IBM Watson API to maximize the performance
- Security

### 9.3. Open Issues & Ideas for Solutions

- How to automate handle incorrect answer for system to learn the correct one?  
Use cognitive technology compare big data then use probability analysis

# 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 server-implementation branch will always be automatically deployed and run on Heroku in development phase. There is no restriction on what IDE will be used by each group member. The ideal team tasks arrangement is 3 team members each work on frontend, backend, testing with domain expert working on training Watson.

### Notes on system further extension

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

## 10.2. Instructions for System Deployment

### Platform Requirements

NodeJS: main server platform, version require v6.0 and up

MongoDB: version requires 3.0.x and up

Modern Web Browser: Firefox, Chrome, Edge, Safari, Opera, Iceweasel

BlueMix

Heroku-GitHub

### System Installation

NodeJS: local install by installer or any online NodeJS IDE

MongoDB: no installation required

BlueMix: no installation required

Client: user defined browser installation

## 10.3. Instructions for System End Users

## 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](http://www.cs.columbia.edu/~mcollins/papers/tutorial_colt.pdf) (accessed Oct 21, 2016).
- 10) IBM Knowledge Center  
<https://www.ibm.com/support/knowledgecenter/SSSR99/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).
- 18) Pennsylvania State University - Erie  
<http://www.ratemyprofessors.com/campusRatings.jsp?sid=1291> (accessed November 12, 2016).
- 19) Pennsylvania State University -- Erie, The Behrend College  
<http://colleges.usnews.rankingsandreviews.com/best-colleges/penn-state-erie-3333/academics> (accessed November 12, 2016).
- 20) The Pennsylvania State University <http://engineering-schools.startclass.com/l/223/The-Pennsylvania-State-University> (accessed November 12, 2016).

- 21) Penn State Erie - The Behrend College <https://colleges.niche.com/penn-state-erie---the-behrend-college/> (accessed November 12, 2016).
- 22) Frequently Asked Questions - Honors Program  
<https://psbehrend.psu.edu/Academics/academic-programs/honors/frequently-asked-questions-1> (accessed November 12, 2016).
- 23) What does a Computer Software Engineer do? Could you give me a description of the field? <http://tryengineering.org/ask-expert/what-does-computer-software-engineer-do-could-you-give-me-description-field> (accessed November 11, 2016).
- 24) Undergraduate Majors and Minors  
<https://psbehrend.psu.edu/Academics/academic-programs/majors-minors> (accessed November 11, 2016).
- 25) Computer Engineering, B.S. Curriculum <https://psbehrend.psu.edu/school-of-engineering/academic-programs/computer-engineering/curriculum> (accessed November 11, 2016).
- 26) Computer Engineering <http://psbehrend.psu.edu/school-of-engineering/academic-programs/computer-engineering> (accessed November 11, 2016).
- 27) Penn State Erie, The Behrend College  
[https://en.wikipedia.org/wiki/Penn\\_State\\_Erie,\\_The\\_Behrend\\_College](https://en.wikipedia.org/wiki/Penn_State_Erie,_The_Behrend_College) (accessed November 11, 2016).
- 28) Computer Science & Engineering  
[http://www.cs.washington.edu/prospective\\_students/undergrad/faq](http://www.cs.washington.edu/prospective_students/undergrad/faq) (accessed November 10, 2016).
- 29) Science and Engineering Indicators 2012  
<https://www.nsf.gov/statistics/seind12/c2/c2s2.htm> (accessed November 10, 2016).
- 30) Annual Security Reports <http://police.psu.edu/annual-security-reports> (accessed November 10, 2016).
- 31) Adult Learner Support Services <https://psbehrend.psu.edu/Academics/academic-services/adult/support-services> (accessed November 10, 2016).
- 32) Electrical & Computer Engineering  
<http://www.ee.uh.edu/undergraduate/computer-engineering-faq> (accessed November 10, 2016).
- 33) Culture and engineering in the USA and Japan  
<http://link.springer.com/article/10.1007/s00146-003-0280-z> (accessed November 9, 2016).
- 34) Traveling opportunities as a software engineer?  
<http://www.flyertalk.com/forum/travelbuzz/920489-traveling-opportunities-software-engineer.html> (accessed November 9, 2016).
- 35) A job in computers that lets me travel a lot? <http://ask.metafilter.com/88578/A-job-in-computers-that-lets-me-travel-a-lot> (accessed November 9, 2016).
- 36) Traveling computer jobs <http://www.indeed.com/q-Traveling-Computer-jobs.html> (accessed November 9, 2016).
- 37) Average weather for Erie, Pennsylvania, USA  
<https://weatherspark.com/averages/30194/Erie-Pennsylvania-United-States> (accessed November 8, 2016).
- 38) Campus Dining <http://behrendcampusliving.psu.edu/campus-dining> (accessed November 8, 2016).
- 39) Tuition Calculator <http://tuition.psu.edu/costestimate.aspx> (accessed November 8, 2016).

- 40) Find 2016 Engineering Internships <http://www.internships.com/engineering> (accessed November 8, 2016).
- 41) What is the percentage of junior year computer science students in the top 30 US universities that are likely to be accepted to internships in 2016?  
<https://www.quora.com/What-is-the-percentage-of-junior-year-computer-science-students-in-the-top-30-US-universities-that-are-likely-to-be-accepted-to-internships-in-2016> (accessed November 8, 2016).
- 42) Software Engineering / Computer Science difference  
<https://users.csc.calpoly.edu/~djanzen/secsdiff.html> (accessed November 7, 2016).
- 43) Pennsylvania State University Erie Behrend College  
[http://www.studentsreview.com/PA/PSUE\\_comments.html](http://www.studentsreview.com/PA/PSUE_comments.html) (accessed November 10, 2016).
- 44) Academic Calendars  
[http://registrar.psu.edu/academic\\_calendar/calendar\\_index.cfm](http://registrar.psu.edu/academic_calendar/calendar_index.cfm) (accessed November 7, 2016).
- 45) Top 10 Benefits of Becoming an Engineer  
<http://old.ece.ufl.edu/admission/undergraduate/benefits.htm> (accessed November 10, 2016).
- 46) How Much Does a Computer Engineer get Paid Per Hour?  
<http://work.chron.com/much-computer-engineer-paid-per-hour-21954.html> (accessed November 8, 2016).
- 47) Tau Beta Pi <https://www.tbp.org/home.cfm> (accessed November 8, 2016).
- 48) Cost of Living in Erie <https://www.numbeo.com/cost-of-living/in/Erie> (accessed November 10, 2016).
- 49) Computer Hardware Engineer Career  
<https://www.mymajors.com/career/computer-engineers/education/> (accessed November 12, 2016).
- 50) What do Engineers have in common? <http://engineeringrevision.com/367/what-do-engineers-have-in-common/> (accessed November 12, 2016).
- 51) Hobbies that engineers have?  
[https://www.reddit.com/r/engineering/comments/2ed8sj/hobbies\\_that\\_engineers\\_have/](https://www.reddit.com/r/engineering/comments/2ed8sj/hobbies_that_engineers_have/) (accessed November 12, 2016).
- 52) Computer Engineering, Software Engineering, or Computer Science?  
<https://engiegirlsatuwaterloo.wordpress.com/2013/08/29/computer-engineering-software-engineering-or-computer-science/> (accessed November 12, 2016).

**Table 4.1. Use Case Index Table**

<b>Project Name: Intelligent Academic Planner</b>				
<b>Use Case ID</b>	<b>Use Case Name</b>	<b>Level</b>	<b>Author</b>	<b>Version</b>
UC-001	Respond to Request	Primary task	Daria Cook	0.2
UC-004	Register	Primary task	Daria Cook	0.7
UC-005	View assessment	Subfunction	Xiaoyu Zhou	0.8
UC-006	View question log	Primary task	Xiaoyu Zhou	0.3
UC-007	Train Watson	Primary task	Xiaoyu Zhou	0.6
UC-009	Real-time translation	Subfunction	Xiaoyu Zhou	0.6
UC-010	Answer quality feedback	Primary task	Xiaoyu Zhou	0.7
UC-011	Ask question	Primary task	Xiaoyu Zhou	0.6
UC-012	Analyze Question	Primary task	Xiaoyu Zhou	0.6
UC-013	Submit Profile	Primary task	Xiaoyu Zhou	0.3
UC-015	Generate Assessment	Primary task	Xiaoyu Zhou	0.3
UC-016	Login	Primary task	Xiaoyu Zhou	0.5
UC-018	Logout	Primary task	Xiaoyu Zhou	0.3
UC-019	Request Assessment	Primary task	Xiaoyu Zhou	0.4

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 4.2. Use Case UC-001**

<b>Project Name:</b>	
<b>Use Case ID:</b>	UC-001
<b>Use Case Name:</b>	Respond to Request
<b>User Goal:</b>	User approves of advisor viewing assessment
<b>Scope:</b>	IAP System
<b>Level:</b>	Primary task
<b>Relevant User Reqs:</b>	UF-E
<b>Relevant System Reqs:</b>	SF-E-05
<b>Primary Actor:</b>	Student
<b>Precondition:</b>	User has a request awaiting approval
<b>Minimal Guarantee:</b>	User's assessment are non-viewable
<b>Success Guarantee:</b>	User's assessment becomes viewable for advisor that requested
<b>Trigger:</b>	User requests to respond to request
<b>Success Scenario:</b>	<p><b>Step Actions</b></p> <p>1 The user requests to respond to request      2 The system asks for user's response      3 The user responds</p>
<b>Extensions:</b>	<b>Branching Scenarios</b>
3A	<p><b>Condition:</b> If request is declined</p> <p><b>Step Actions</b></p> <p>1 The system sends notification to advisor      2 Exit out of functionality</p>
3B	<p><b>Condition:</b> If request is accepted</p> <p><b>Step Actions</b></p> <p>1 The system sends notification to advisor      2 The system allows advisor to view assessment of student</p>
<b>Acknowledgment:</b> Generated from the CapStone process management system ©2015	

**Table 4.3. Use Case UC-004**

<b>Project Name:</b>	
<b>Use Case ID:</b>	UC-004
<b>Use Case Name:</b>	Register
<b>User Goal:</b>	To be recognized by the system.
<b>Scope:</b>	IAP System
<b>Level:</b>	Primary task
<b>Relevant User Reqs:</b>	UF-F
<b>Relevant System Reqs:</b>	SF-F-01
<b>Primary Actor:</b>	Visitor, System Developer
<b>Precondition:</b>	User is viewing program
<b>Minimal Guarantee:</b>	User is not recognized by system
<b>Success Guarantee:</b>	User is recognized by system
<b>Trigger:</b>	User requests to register
<b>Success Scenario:</b>	<p><b>Step Actions</b></p> <p>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</p>
<b>Extensions:</b>	<b>Branching Scenarios</b>
4A	<p><b>Condition:</b> Information is invalid</p> <p><b>Step Actions</b></p> <p>1 The system notifies user of invalid information      2 Return to step 2</p>

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 4.4. Use Case UC-005**

<b>Project Name:</b>				
<b>Use Case ID:</b>	UC-005			
<b>Use Case Name:</b>	View assessment			
<b>User Goal:</b>	User is able to view assessment			
<b>Scope:</b>	IAP System			
<b>Level:</b>	Subfunction			
<b>Relevant User Reqs:</b>	UF-E			
<b>Relevant System Reqs:</b>	SF-E-03			
<b>Primary Actor:</b>	student, advisor			
<b>Precondition:</b>	The user is logged in			
<b>Minimal Guarantee:</b>	System does not display student's assessment			
<b>Success Guarantee:</b>	system display student's assessment			
<b>Trigger:</b>	User requests to view assessments			
<b>Success Scenario:</b>	<table border="1"> <thead> <tr> <th style="background-color: #ADD8E6;">Step Actions</th> </tr> </thead> <tbody> <tr> <td>1 The user requests to view assessments</td> </tr> <tr> <td>2 The system displays assessments that can be viewed</td> </tr> </tbody> </table>	Step Actions	1 The user requests to view assessments	2 The system displays assessments that can be viewed
Step Actions				
1 The user requests to view assessments				
2 The system displays assessments that can be viewed				
<b>Extensions:</b>	Branching Scenarios			
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>				

**Table 4.5. Use Case UC-006**

<b>Project Name:</b>	
<b>Use Case ID:</b>	UC-006
<b>Use Case Name:</b>	View question log
<b>User Goal:</b>	User is able to view the question log
<b>Scope:</b>	IAP System
<b>Level:</b>	Primary task
<b>Relevant User Reqs:</b>	UF-G
<b>Relevant System Reqs:</b>	SF-G-01
<b>Primary Actor:</b>	Adviser, System Developer
<b>Precondition:</b>	User is logged in
<b>Minimal Guarantee:</b>	System does not display question log
<b>Success Guarantee:</b>	System displays question log
<b>Trigger:</b>	User requests to view question log
<b>Success Scenario:</b>	<p><b>Step Actions</b></p> <p>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</p>
<b>Extensions:</b>	Branching Scenarios

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 4.6. Use Case UC-007**

<b>Project Name:</b>	
<b>Use Case ID:</b>	UC-007
<b>Use Case Name:</b>	Train Watson
<b>User Goal:</b>	User be able to train Watson with new questions that user asked but hasn't been answered
<b>Scope:</b>	IAP System
<b>Level:</b>	Primary task
<b>Relevant User Reqs:</b>	UF-H
<b>Relevant System Reqs:</b>	SF-A-01
<b>Primary Actor:</b>	System Developer
<b>Precondition:</b>	User is logged in
<b>Minimal Guarantee:</b>	Watson is not further trained
<b>Success Guarantee:</b>	Watson is further trained
<b>Trigger:</b>	User requests to train Watson
<b>Success Scenario:</b>	<p><b>Step Actions</b></p> <p>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</p>
<b>Extensions:</b>	<b>Branching Scenarios</b>
1A	<b>Condition:</b> User request system export all new questions from last time export
	<p><b>Step Actions</b></p> <p>1 The system export question list that is new from last time</p>
1B	<b>Condition:</b> User request system export all new questions by major
	<p><b>Step Actions</b></p>
1C	<b>Condition:</b> User request system export all new questions other condition
	<p><b>Step Actions</b></p> <p>1 The system export top 10 unanswered FAQ      2 The system export top 10% unanswered FAQ</p>

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 4.7. Use Case UC-009**

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

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 4.8. Use Case UC-010**

<b>Project Name:</b>						
<b>Use Case ID:</b>	UC-010					
<b>Use Case Name:</b>	Answer quality feedback					
<b>User Goal:</b>	User is able to provide feedback on question response.					
<b>Scope:</b>	IAP System					
<b>Level:</b>	Primary task					
<b>Relevant User Reqs:</b>	UF-H					
<b>Relevant System Reqs:</b>	SF-H-01					
<b>Primary Actor:</b>	Visitor, Student, Advisor					
<b>Precondition:</b>	User is asking a question					
<b>Minimal Guarantee:</b>	Feedback is not stored					
<b>Success Guarantee:</b>	Feedback is stored					
<b>Trigger:</b>	User requests to submit feedback					
<b>Success Scenario:</b>	<table border="1"> <thead> <tr> <th style="background-color: #ADD8E6;">Step Actions</th> </tr> </thead> <tbody> <tr> <td>1 The user requests to submit feedback</td> </tr> <tr> <td>2 The system allows user to enter feedback</td> </tr> <tr> <td>3 The user submits feedback</td> </tr> <tr> <td>4 The system stores feedback</td> </tr> </tbody> </table>	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
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						
<b>Extensions:</b>	Branching Scenarios					

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 4.9. Use Case UC-011**

<b>Project Name:</b>	
<b>Use Case ID:</b>	UC-011
<b>Use Case Name:</b>	Ask question
<b>User Goal:</b>	User is able to ask questions
<b>Scope:</b>	IAP System
<b>Level:</b>	Primary task
<b>Relevant User Reqs:</b>	UF-C
<b>Relevant System Reqs:</b>	SF-C-01,SF-C-02,SF-C-03,SF-C-04,SF-C-05
<b>Primary Actor:</b>	Visitor, Student, Advisor
<b>Precondition:</b>	User is viewing program
<b>Minimal Guarantee:</b>	Question is not logged
<b>Success Guarantee:</b>	Question is logged
<b>Trigger:</b>	The user requests to ask a question
<b>Success Scenario:</b>	<p><b>Step Actions</b></p> <p>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 &lt;&lt;Analyze Question&gt;&gt;      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</p>
<b>Extensions:</b>	<b>Branching Scenarios</b>
3A	<b>Condition:</b> If user requests translation
	<p><b>Step Actions</b></p> <p>1 The system translates text &lt;&lt;Realtime Translation&gt;&gt;</p>
7A	<b>Condition:</b> If user requests to provide feedback
	<p><b>Step Actions</b></p> <p>1 The system requests feedback information &lt;&lt;Answer Quality Feedback&gt;&gt;</p>

**Acknowledgment:** Generated from the CapStone process management system ©2015

**Table 4.10. Use Case UC-012**

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

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 4.11. Use Case UC-013**

<b>Project Name:</b>	
<b>Use Case ID:</b>	UC-013
<b>Use Case Name:</b>	Submit Profile
<b>User Goal:</b>	User can create profile
<b>Scope:</b>	IAP System
<b>Level:</b>	Primary task
<b>Relevant User Reqs:</b>	UF-E
<b>Relevant System Reqs:</b>	SF-E-01,SF-E-02,SF-E-03,SF-E-04,SF-E-05
<b>Primary Actor:</b>	student
<b>Precondition:</b>	User is logged in
<b>Minimal Guarantee:</b>	Profile is not stored
<b>Success Guarantee:</b>	Profile is stored
<b>Trigger:</b>	User requests to update profile
<b>Success Scenario:</b>	<p><b>Step Actions</b></p> <p>1 The user requests to update profile      2 The system verify the profile      3 The system upload the profile to server</p>
<b>Extensions:</b>	<b>Branching Scenarios</b>
2A	<b>Condition:</b> If system detect invalid format of the profile <p><b>Step Actions</b></p> <p>1 The system notify user of problem      2 Return to step 1</p>
<b>Acknowledgment:</b> Generated from the CapStone process management system ©2015	

**Table 4.12. Use Case UC-015**

<b>Project Name:</b>	
<b>Use Case ID:</b>	UC-015
<b>Use Case Name:</b>	Generate Assessment
<b>User Goal:</b>	User receives assessment of profile information
<b>Scope:</b>	IAP System
<b>Level:</b>	Primary task
<b>Relevant User Reqs:</b>	UF-E
<b>Relevant System Reqs:</b>	SF-E-04,SF-E-05
<b>Primary Actor:</b>	Student
<b>Precondition:</b>	User has submitted information to profile
<b>Minimal Guarantee:</b>	No assessment generated
<b>Success Guarantee:</b>	System generate assessment and display to user
<b>Trigger:</b>	User requests to generate assessment
<b>Success Scenario:</b>	<p><b>Step Actions</b></p> <p>1 The user requests to generate assessment      2 The system analyzes profile and generates assessment      3 The system displays assessment</p>
<b>Extensions:</b>	<b>Branching Scenarios</b>
3A	<b>Condition:</b> keep assessment on user's record <p><b>Step Actions</b></p> <p>1 The user request system save the assessment      2 The system upload assessment to server and relate to user account</p>
3B	<b>Condition:</b> The user request assessment to be viewable by advisers <p><b>Step Actions</b></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>
<b>Acknowledgment:</b> Generated from the CapStone process management system ©2015	

**Table 4.13. Use Case UC-016**

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

**Table 4.14. Use Case UC-018**

<b>Project Name:</b>	
<b>Use Case ID:</b>	UC-018
<b>Use Case Name:</b>	Logout
<b>User Goal:</b>	User is able to log out
<b>Scope:</b>	IAP System
<b>Level:</b>	Primary task
<b>Relevant User Reqs:</b>	UF-B
<b>Relevant System Reqs:</b>	SF-B-01
<b>Primary Actor:</b>	Student, Advisor, System Devel
<b>Precondition:</b>	User is logged in
<b>Minimal Guarantee:</b>	User is not logged out
<b>Success Guarantee:</b>	User is logged out
<b>Trigger:</b>	User requests to log out
<b>Success Scenario:</b>	<p><b>Step Actions</b></p> <p>1 The user requests to logout      2 The system verifies all information is saved      3 The system logs users out</p>
<b>Extensions:</b>	<b>Branching Scenarios</b>
2A	<b>Condition:</b> If some information is unsaved <p><b>Step Actions</b></p> <p>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.</p>
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>	

**Table 4.15. Use Case UC-019**

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

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 4.16. User Functional Requirements: UF-A**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	UF-A	Type	Functional	Non-Functional	
<b>Creation:</b>	Sep 20 2016 01:25 PM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	Oct 05 2016 01:50 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	A user can log in.				
<b>Priority:</b>	Highest	✓ High	Medium	Low	Lowest
<b>This Req. is Refined Into:</b>	SF-A-01				
<b>Justify why UF-A can be completely covered by SF-A-01</b>	If a user can login within 5 seconds, they can log in properly.				
<b>Traceability:</b>	Use cases cf.	UC-016			
	Test cases cf.	TC-001, TC-013, TC-014, TC-015			
<b>Acknowledgment</b>	Generated from the CapStone Process Management System ©2015				

**Table 4.17. User Functional Requirements: UF-B**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	UF-B	Type	Functional	Non-Functional	
<b>Creation:</b>	Sep 20 2016 04:17 PM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	Oct 05 2016 01:50 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	A user can log out.				
<b>Priority:</b>	Highest	✓ High	Medium	Low	Lowest
<b>This Req. is Refined Into:</b>	SF-B-01				
<b>Justify why UF-B can be completely covered by SF-B-01</b>	If a user can log out within 5 seconds, they can log out properly				
<b>Traceability:</b>	Use cases cf.	UC-018			
	Test cases cf.	TC-002			
<b>Acknowledgment</b>	Generated from the CapStone Process Management System ©2015				

Table 4.18. User Functional Requirements: UF-C

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<a href="#">UF-C</a>	Type	Functional	Non-Functional	
<b>Creation:</b>	<a href="#">Sep 20 2016 04:18 PM</a>	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<a href="#">Sep 20 2016 04:58 PM</a>	System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<b>Description:</b>	A user can ask the system questions.				
<b>Priority:</b>	<input checked="" type="checkbox"/> Highest	High	Medium	Low	Lowest
<b>This Req. is Refined Into:</b>	<a href="#">SF-C-01, SF-C-02, SF-C-03, SF-C-04, SF-C-05</a>				
<b>Justify why UF-C can be completely covered by SF-C-01, SF-C-02, SF-C-03, SF-C-04, SF-C-05</b>	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.				
<b>Traceability:</b>	Use cases cf.	<a href="#">UC-009, UC-011, UC-012</a>			
	Test cases cf.	<a href="#">TC-004, TC-006, TC-007</a>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.19. User Functional Requirements: UF-D

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<a href="#">UF-D</a>	Type	Functional	Non-Functional	
<b>Creation:</b>	<a href="#">Sep 23 2016 12:52 PM</a>	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<a href="#">Oct 05 2016 01:51 AM</a>	System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<b>Description:</b>	A user should receive multiple responses to a question.				
<b>Priority:</b>	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
<b>This Req. is Refined Into:</b>	<a href="#">SF-D-01</a>				
<b>Justify why UF-D can be completely covered by SF-D-01</b>	By requiring a minimum of 1 response to be given, it is given that multiple responses are given to a question.				
<b>Traceability:</b>	Use cases cf.	<a href="#">UC-012</a>			
	Test cases cf.	<a href="#">TC-007</a>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.20. User Functional Requirements: UF-E

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<b>UF-E</b>	Type	Functional	Non-Functional	
<b>Creation:</b>	<b>Oct 04 2016 11:51 PM</b>	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<b>Oct 05 2016 12:06 AM</b>	System	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	A user can create a profile				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Refined Into:</b>	<b>SF-E-01, SF-E-02, SF-E-03, SF-E-04, SF-E-05</b>				
<b>Justify why UF-E can be completely covered by SF-E-01, SF-E-02, SF-E-03, SF-E-04, SF-E-05</b>	All of the system requirements associated with this cover what can be put into their profile.				
<b>Traceability:</b>	Use cases cf.	<b>UC-001, UC-005, UC-013, UC-015, UC-019</b>			
	Test cases cf.	<b>TC-008, TC-009, TC-010, TC-011, TC-016, TC-020, TC-021</b>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.21. User Functional Requirements: UF-F

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<b>UF-F</b>	Type	Functional	Non-Functional	
<b>Creation:</b>	<b>Oct 05 2016 12:15 AM</b>	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<b>Oct 05 2016 01:51 AM</b>	System	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	A user can register				
<b>Priority:</b>	Highest	<input checked="" type="checkbox"/> High	Medium	Low	Lowest
<b>This Req. is Refined Into:</b>	<b>SF-F-01</b>				
<b>Justify why UF-F can be completely covered by SF-F-01</b>	If a user can register within 5 seconds, they can register properly.				
<b>Traceability:</b>	Use cases cf.	<b>UC-004</b>			
	Test cases cf.	<b>TC-003, TC-017, TC-018, TC-019</b>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.22. User Functional Requirements: UF-G**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	<a href="#">UF-G</a>	Type	Functional	Non-Functional	
<b>Creation:</b>	<a href="#">Oct 05 2016 12:36 AM</a>	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<a href="#">Oct 05 2016 01:52 AM</a>	System	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	A user can view a log of asked questions.				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Refined Into:</b>	<a href="#">SF-G-01</a>				
<b>Justify why UF-G can be completely covered by SF-G-01</b>	If advisors and system developers can view the question log, then users can view the question log.				
<b>Traceability:</b>	Use cases cf.	<a href="#">UC-006</a>			
	Test cases cf.	<a href="#">TC-022</a>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.23. User Functional Requirements: UF-H**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	<a href="#">UF-H</a>	Type	Functional	Non-Functional	
<b>Creation:</b>	<a href="#">Oct 05 2016 02:12 AM</a>	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<a href="#">Oct 05 2016 02:16 AM</a>	System	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	A user can provide information to improve accuracy of the system.				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Refined Into:</b>	<a href="#">SF-H-01</a>				
<b>Justify why UF-H can be completely covered by SF-H-01</b>	If answer quality feedback is submitted, accuracy of responses can be increased.				
<b>Traceability:</b>	Use cases cf.	<a href="#">UC-007, UC-010</a>			
	Test cases cf.	<a href="#">TC-005, TC-012</a>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.24. User NonFunctional Requirements: UP-03

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<b>UP-03</b>	Type	Functional	Non-Functional	
<b>Creation:</b>	<b>Oct 05 2016 01:36 AM</b>	User	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<b>Modification:</b>	<b>Oct 05 2016 01:54 AM</b>	System	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	A user should receive a quick response after asking a question				
<b>Priority:</b>	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
<b>This Req. is Refined Into:</b>	<b>SP-03-01</b>				
<b>Justify why UP-03 can be completely covered by SP-03-01</b>	By specifying performance requirements, it is ensured that the question will be answered in a quick manner.				
<b>Traceability:</b>	Use cases cf.	<b>N/A</b>			
	Test cases cf.	<b>TC-023</b>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.25. User NonFunctional Requirements: UP-01

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<b>UP-01</b>	Type	Functional	Non-Functional	
<b>Creation:</b>	<b>Oct 05 2016 12:10 AM</b>	User	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<b>Modification:</b>	<b>Oct 05 2016 01:53 AM</b>	System	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	A user's profile should be secure.				
<b>Priority:</b>	Highest	<input checked="" type="checkbox"/> High	Medium	Low	Lowest
<b>This Req. is Refined Into:</b>	<b>SP-01-01</b>				
<b>Justify why UP-01 can be completely covered by SP-01-01</b>	Ensures only specific people can view a user's profile, making it secure.				
<b>Traceability:</b>	Use cases cf.	<b>N/A</b>			
	Test cases cf.	<b>TC-009</b>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.26. User NonFunctional Requirements: UO-01

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<b>UO-01</b>	Type	Functional	Non-Functional	
<b>Creation:</b>	<b>Sep 23 2016 12:42 PM</b>	User	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<b>Modification:</b>	<b>Oct 05 2016 01:53 AM</b>	System	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	A user's session should be managed.				
	<b>Organizational (sub-type below)</b>				
	<b>Development Requirements</b>				
<b>Priority:</b>	Highest	<input checked="" type="checkbox"/> High	Medium	Low	Lowest
<b>This Req. is Refined Into:</b>	<b>SO-01-01</b>				
<b>Justify why UO-01 can be completely covered by SO-01-01</b>	Ensures that a user can only be logged in for 1 hour, managing their session.				
<b>Traceability:</b>	Use cases cf.	<b>N/A</b>			
	Test cases cf.	<b>TC-024</b>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.27. System Functional Requirements: SF-A-01

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<b>SF-A-01</b>	Type	Functional	Non-Functional	
<b>Creation:</b>	<b>Sep 23 2016 12:22 PM</b>	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<b>Oct 05 2016 01:41 AM</b>	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	The system should log-in a user within 5 seconds.				
<b>Priority:</b>	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
<b>This Req. is Engineered From:</b>	<b>UF-A</b>				
<b>Justify why meeting SF-A-01 can contribute to the fulfilment of UF-A</b>	Provides performance requirement for logging in.				
<b>Traceability:</b>	Use cases cf.	<b>UC-007, UC-016</b>			
	Test cases cf.	<b>TC-001, TC-013, TC-014, TC-015</b>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.28. System Functional Requirements: SF-B-01**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	SF-B-01	Type	Functional	Non-Functional	
<b>Creation:</b>	Sep 23 2016 12:32 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	Oct 05 2016 01:41 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	The system should log-out a user within 5 seconds.				
<b>Priority:</b>	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
<b>This Req. is Engineered From:</b>	UF-B				
<b>Justify why meeting SF-B-01 can contribute to the fulfilment of UF-B</b>	Explains performance requirement.				
<b>Traceability:</b>	Use cases cf.	UC-018			
	Test cases cf.	TC-002			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.29. System Functional Requirements: SF-C-01**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	SF-C-01	Type	Functional	Non-Functional	
<b>Creation:</b>	Sep 20 2016 04:22 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	Oct 05 2016 01:42 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	The system should conduct textual analyses.				
<b>Priority:</b>	Highest	<input checked="" type="checkbox"/> High	Medium	Low	Lowest
<b>This Req. is Engineered From:</b>	UF-C				
<b>Justify why meeting SF-C-01 can contribute to the fulfilment of UF-C</b>	In order to provide an answer to questions asked, system must be able to perform this function.				
<b>Traceability:</b>	Use cases cf.	UC-011, UC-012			
	Test cases cf.	TC-004, TC-007			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.30. System Functional Requirements: SF-C-02

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<b>SF-C-02</b>	Type	Functional	Non-Functional	
<b>Creation:</b>	<b>Oct 05 2016 01:59 AM</b>	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<b>Oct 05 2016 02:00 AM</b>	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	The system should be able to handle input from multiple well-known languages.				
<b>Priority:</b>	Highest	High	Medium	Low	<input checked="" type="checkbox"/> Lowest
<b>This Req. is Engineered From:</b>	<b>UF-C</b>				
<b>Justify why meeting SF-C-02 can contribute to the fulfilment of UF-C</b>	Allows users to ask a question in a variety of ways.				
<b>Traceability:</b>	Use cases cf.	<b>UC-009, UC-011</b>			
	Test cases cf.	<b>TC-004, TC-006</b>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.31. System Functional Requirements: SF-C-03

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<b>SF-C-03</b>	Type	Functional	Non-Functional	
<b>Creation:</b>	<b>Sep 20 2016 04:31 PM</b>	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<b>Oct 05 2016 01:44 AM</b>	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	The system should recommend majors suitable for the user based on the personality assessment.				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Engineered From:</b>	<b>UF-C</b>				
<b>Justify why meeting SF-C-03 can contribute to the fulfilment of UF-C</b>	This allows questions to be answered more accurately.				
<b>Traceability:</b>	Use cases cf.	<b>UC-011, UC-012</b>			
	Test cases cf.	<b>TC-004, TC-007</b>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.32. System Functional Requirements: SF-C-04**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	SF-C-04	Type	Functional	Non-Functional	
<b>Creation:</b>	Sep 20 2016 04:34 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	Oct 05 2016 01:44 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	The system should gather data unique to each user.				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Engineered From:</b>	UF-C				
<b>Justify why meeting SF-C-04 can contribute to the fulfilment of UF-C</b>	This allows questions to be answered more accurately.				
<b>Traceability:</b>	Use cases cf.	UC-011, UC-012			
	Test cases cf.	TC-004, TC-007			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.33. System Functional Requirements: SF-C-05**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	SF-C-05	Type	Functional	Non-Functional	
<b>Creation:</b>	Sep 20 2016 04:35 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	Oct 05 2016 01:44 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	The system should recommend courses based on the recommended majors.				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Engineered From:</b>	UF-C				
<b>Justify why meeting SF-C-05 can contribute to the fulfilment of UF-C</b>	This allows questions to be answered more accurately.				
<b>Traceability:</b>	Use cases cf.	UC-011, UC-012			
	Test cases cf.	TC-004, TC-007			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.34. System Functional Requirements: SF-D-01

Intelligent Academic Planner					
Requirement #:	SF-D-01			Type	Functional
Creation:	Sep 23 2016 12:54 PM			User	<input type="checkbox"/>
Modification:	Oct 05 2016 01:46 AM			System	<input checked="" type="checkbox"/>
Description:	The system should show a minimum of 1 related search/question.				
Priority:	Highest	High	Medium	<input checked="" type="checkbox"/> 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.	TC-007			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

Table 4.35. System Functional Requirements: SF-E-01

Intelligent Academic Planner					
Requirement #:	SF-E-01			Type	Functional
Creation:	Oct 04 2016 11:56 PM			User	<input type="checkbox"/>
Modification:	Oct 05 2016 01:46 AM			System	<input checked="" 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.	TC-008			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

**Table 4.36. System Functional Requirements: SF-E-02**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Requirement #:</b>	SF-E-02	Type	Functional	Non-Functional		
<b>Creation:</b>	Oct 04 2016 11:56 PM	User	<input type="checkbox"/>	<input type="checkbox"/>		
<b>Modification:</b>	Oct 05 2016 01:47 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<b>Description:</b>	The system should allow a user to submit 100 words of self-description about their personality.					
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest	
<b>This Req. is Engineered From:</b>	UF-E					
<b>Justify why meeting SF-E-02 can contribute to the fulfilment of UF-E</b>	Allows user to enter personality information on their profile					
<b>Traceability:</b>	Use cases cf.	UC-013				
	Test cases cf.	TC-008				
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>					

**Table 4.37. System Functional Requirements: SF-E-03**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Requirement #:</b>	SF-E-03	Type	Functional	Non-Functional		
<b>Creation:</b>	Oct 05 2016 12:00 AM	User	<input type="checkbox"/>	<input type="checkbox"/>		
<b>Modification:</b>	Oct 05 2016 01:47 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<b>Description:</b>	The system should allow a user to view their personality assessments.					
<b>Priority:</b>	Highest	<input checked="" type="checkbox"/> High	Medium	Low	Lowest	
<b>This Req. is Engineered From:</b>	UF-E					
<b>Justify why meeting SF-E-03 can contribute to the fulfilment of UF-E</b>	Allows a user to learn about themselves based on profile information.					
<b>Traceability:</b>	Use cases cf.	UC-005, UC-013				
	Test cases cf.	TC-011				
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>					

**Table 4.38. System Functional Requirements: SF-E-04**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	<a href="#">SF-E-04</a>				Type
<b>Creation:</b>	<a href="#">Oct 05 2016 01:43 AM</a>				User
<b>Modification:</b>	<a href="#">Oct 05 2016 01:48 AM</a>				System
<b>Description:</b>	The system should create a personality assessment unique to each user based on the data gathered.				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Engineered From:</b>	<a href="#">UF-E</a>				
<b>Justify why meeting SF-E-04 can contribute to the fulfilment of UF-E</b>	Allows user to view information about themselves on their profile that they did not input.				
<b>Traceability:</b>	Use cases cf.	<a href="#">UC-013, UC-015</a>			
	Test cases cf.	<a href="#">TC-009, TC-020, TC-021</a>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.39. System Functional Requirements: SF-E-05**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	<a href="#">SF-E-05</a>				Type
<b>Creation:</b>	<a href="#">Oct 05 2016 01:45 AM</a>				User
<b>Modification:</b>	<a href="#">Oct 05 2016 01:45 AM</a>				System
<b>Description:</b>	The system should summarize this data to be used by an advisor directing the student.				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Engineered From:</b>	<a href="#">UF-E</a>				
<b>Justify why meeting SF-E-05 can contribute to the fulfilment of UF-E</b>	Allows user to get assistance from advisors based on their profile.				
<b>Traceability:</b>	Use cases cf.	<a href="#">UC-001, UC-013, UC-015, UC-019</a>			
	Test cases cf.	<a href="#">TC-009, TC-010, TC-016, TC-020, TC-021</a>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.40. System Functional Requirements: SF-F-01**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	SF-F-01	Type	Functional	Non-Functional	
<b>Creation:</b>	Oct 05 2016 12:16 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	Oct 05 2016 01:49 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	The system should register the user within 5 seconds.				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Engineered From:</b>	UF-F				
<b>Justify why meeting SF-F-01 can contribute to the fulfilment of UF-F</b>	Places a performance requirement on registration.				
<b>Traceability:</b>	Use cases cf.	UC-004			
	Test cases cf.	TC-003, TC-017, TC-018, TC-019			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.41. System Functional Requirements: SF-G-01**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	SF-G-01	Type	Functional	Non-Functional	
<b>Creation:</b>	Oct 05 2016 12:37 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	Oct 05 2016 12:37 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	The system should only allow Advisors and System Developers to view the question log.				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Engineered From:</b>	UF-G				
<b>Justify why meeting SF-G-01 can contribute to the fulfilment of UF-G</b>	Adds security to the question log.				
<b>Traceability:</b>	Use cases cf.	UC-006			
	Test cases cf.	TC-022			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.42. System Functional Requirements: SF-H-01

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<b>SF-H-01</b>	Type	Functional	Non-Functional	
<b>Creation:</b>	<b>Oct 05 2016 02:15 AM</b>	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<b>Oct 05 2016 02:15 AM</b>	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>Description:</b>	A user can provide answer quality feedback after asking a question.				
<b>Priority:</b>	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
<b>This Req. is Engineered From:</b>	<b>UF-H</b>				
<b>Justify why meeting SF-H-01 can contribute to the fulfilment of UF-H</b>	Allows feedback to be submitted.				
<b>Traceability:</b>	Use cases cf.	<b>UC-010</b>			
	Test cases cf.	<b>TC-005</b>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.43. System NonFunctional Requirements: SP-03-01

<b>Project Name:</b>	<b>Intelligent Academic Planner</b>				
<b>Requirement #:</b>	<b>SP-03-01</b>	Type	Functional	Non-Functional	
<b>Creation:</b>	<b>Oct 20 2016 01:53 AM</b>	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<b>Oct 20 2016 01:54 AM</b>	System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<b>Description:</b>	The system should provide an answer to a question within 5 seconds.		<b>Product (sub-type below)</b>		
			<b>Performance Requirements</b>		
<b>Priority:</b>	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
<b>This Req. is Engineered From:</b>	<b>UP-03</b>				
<b>Justify why meeting SP-03-01 can contribute to the fulfilment of UP-03</b>	Ensures an answer is quickly given by adding a performance requirement.				
<b>Traceability:</b>	Use cases cf.	<b>N/A</b>			
	Test cases cf.	<b>TC-023</b>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.44. System NonFunctional Requirements: SP-01-01**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	<a href="#">SP-01-01</a>	Type	Functional	Non-Functional	
<b>Creation:</b>	<a href="#">Oct 05 2016 12:12 AM</a>	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<a href="#">Oct 05 2016 12:14 AM</a>	System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<b>Description:</b>	The system should only display information that the user knows is being displayed.				<a href="#">Product (sub-type below)</a>
					<a href="#">Dependability/Reliability/Security</a>
<b>Priority:</b>	Highest	<input checked="" type="checkbox"/> High	Medium	Low	Lowest
<b>This Req. is Engineered From:</b>	<a href="#">UP-01</a>				
<b>Justify why meeting SP-01-01 can contribute to the fulfilment of UP-01</b>	Makes the user's profile more secure by allowing the user to specify which information is visible.				
<b>Traceability:</b>	Use cases cf.	<a href="#">N/A</a>			
	Test cases cf.	<a href="#">TC-009</a>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.45. System NonFunctional Requirements: SO-01-01**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Requirement #:</b>	<a href="#">SO-01-01</a>	Type	Functional	Non-Functional	
<b>Creation:</b>	<a href="#">Sep 23 2016 12:42 PM</a>	User	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Modification:</b>	<a href="#">Oct 20 2016 01:59 AM</a>	System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<b>Description:</b>	The system should log the user out after 1 hour of inactivity.				<a href="#">Organizational (sub-type below)</a>
					<a href="#">Development Requirements</a>
<b>Priority:</b>	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
<b>This Req. is Engineered From:</b>	<a href="#">UO-01</a>				
<b>Justify why meeting SO-01-01 can contribute to the fulfilment of UO-01</b>	Ensures a user is not logged on for too long, managing their session.				
<b>Traceability:</b>	Use cases cf.	<a href="#">N/A</a>			
	Test cases cf.	<a href="#">TC-024</a>			
<b>Acknowledgment</b>	<i>Generated from the CapStone Process Management System ©2015</i>				

**Table 4.46. Mapping from user requirements to system requirements**

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*

**Table 8.2.1. Test Suite TS-001: Unit Tests**

TS-001: Unit Tests		
Test Case ID	Test Stage	Test Case Description
TC-001	Unit	Login with valid user and password
TC-002	Unit	logout
TC-003	Unit	Register with untaken email and valid password
TC-004	Unit	Ask question
TC-005	Unit	answer quality feedback
TC-006	Unit	Real time translation
TC-007	Unit	Analyze question
TC-008	Unit	submit profile
TC-009	Unit	generate assessment and save with visibility
TC-010	Unit	Respond yes to request
TC-011	Unit	view assessment
TC-012	Unit	train watson
TC-013	Unit	Login with incorrect password
TC-014	Unit	Login with incorrect username
TC-015	Unit	Login with both incorrect username and password
TC-016	Unit	Respond no to Request
TC-017	Unit	Register with invalid password
TC-018	Unit	Register with taken email
TC-019	Unit	Register with both taken email and invalid password
TC-020	Unit	Generate assessment and save with false visibility
TC-021	Unit	Generate Assessment and do not save
TC-022	Unit	View Question Log
TC-023	Unit	Ask Question performance
TC-024	Unit	Log out after 1 hour of inactivity

**Table 8.2.2. Test Case TC-001**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-001 (Unit Test)	
<b>What To Test</b>	Login with valid user and password	
<b>Test Data Input</b>	valid username, valid password	
<b>Expected Result</b>	User is logged in	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-A
	<b>Relevant System Req.(s)</b>	SF-A-01
	<b>Relevant Use Case(s)</b>	UC-016

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.3. Test Case TC-002**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-002 (Unit Test)	
<b>What To Test</b>	logout	
<b>Test Data Input</b>		
<b>Expected Result</b>	User is logged out.	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-B
	<b>Relevant System Req.(s)</b>	SF-B-01
	<b>Relevant Use Case(s)</b>	UC-018

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.4. Test Case TC-003**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-003 (Unit Test)	
<b>What To Test</b>	Register with untaken email and valid password	
<b>Test Data Input</b>	untaken email, valid password	
<b>Expected Result</b>	User is entered into database.	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-F
	<b>Relevant System Req.(s)</b>	SF-F-01
	<b>Relevant Use Case(s)</b>	UC-004
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

**Table 8.2.5. Test Case TC-004**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-004 (Unit Test)	
<b>What To Test</b>	Ask question	
<b>Test Data Input</b>	question string	
<b>Expected Result</b>	Question is logged to database and proper response is displayed	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-C
	<b>Relevant System Req.(s)</b>	SF-C-01,SF-C-02,SF-C-03,SF-C-04,SF-C-05
	<b>Relevant Use Case(s)</b>	UC-011

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.6. Test Case TC-005**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-005 (Unit Test)	
<b>What To Test</b>	answer quality feedback	
<b>Test Data Input</b>	feedback	
<b>Expected Result</b>	feedback is stored in database, thank you message is displayed	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-H
	<b>Relevant System Req.(s)</b>	SF-H-01
	<b>Relevant Use Case(s)</b>	UC-010

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.7. Test Case TC-006**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-006 (Unit Test)	
<b>What To Test</b>	Real time translation	
<b>Test Data Input</b>	non-english text	
<b>Expected Result</b>	english text	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-C
	<b>Relevant System Req.(s)</b>	SF-C-02
	<b>Relevant Use Case(s)</b>	UC-009

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.8. Test Case TC-007**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-007 (Unit Test)	
<b>What To Test</b>	Analyze question	
<b>Test Data Input</b>	question with ambiguous response	
<b>Expected Result</b>	another question to make answer less ambiguous	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-C,UF-D
	<b>Relevant System Req.(s)</b>	SF-C-01,SF-C-03,SF-C-04,SF-C-05,SF-D-01
	<b>Relevant Use Case(s)</b>	UC-012
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

**Table 8.2.9. Test Case TC-008**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-008 (Unit Test)	
<b>What To Test</b>	submit profile	
<b>Test Data Input</b>	new profile information	
<b>Expected Result</b>	profile information is stored to database	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-E
	<b>Relevant System Req.(s)</b>	SF-E-01,SF-E-02
	<b>Relevant Use Case(s)</b>	UC-013

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.10. Test Case TC-009**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-009 (Unit Test)	
<b>What To Test</b>	generate assessment and save with visibility	
<b>Test Data Input</b>	descriptive text, yes to saving assessment, yes to visible to advisors	
<b>Expected Result</b>	assessment is stored to database with true visibility	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-E, UP-01
	<b>Relevant System Req.(s)</b>	SF-E-04, SF-E-05, SP-01-01
	<b>Relevant Use Case(s)</b>	UC-015

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.11. Test Case TC-010**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-010 (Unit Test)	
<b>What To Test</b>	Respond yes to request	
<b>Test Data Input</b>	yes to request	
<b>Expected Result</b>	assessment is visible to person that requested it	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-E
	<b>Relevant System Req.(s)</b>	SF-E-05
	<b>Relevant Use Case(s)</b>	UC-001

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.12. Test Case TC-011**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-011 (Unit Test)	
<b>What To Test</b>	view assessment	
<b>Test Data Input</b>		
<b>Expected Result</b>	assessment is displayed	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-E
	<b>Relevant System Req.(s)</b>	SF-E-03
	<b>Relevant Use Case(s)</b>	UC-005
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

**Table 8.2.13. Test Case TC-012**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-012 (Unit Test)	
<b>What To Test</b>	train watson	
<b>Test Data Input</b>	answer question pair	
<b>Expected Result</b>	question is properly answered when asked	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-H
	<b>Relevant System Req.(s)</b>	
	<b>Relevant Use Case(s)</b>	UC-007
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

**Table 8.2.14. Test Case TC-013**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-013 (Unit Test)	
<b>What To Test</b>	Login with incorrect password	
<b>Test Data Input</b>	correct username, incorrect password	
<b>Expected Result</b>	User is not logged in. Error message is displayed	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-A
	<b>Relevant System Req.(s)</b>	SF-A-01
	<b>Relevant Use Case(s)</b>	UC-016

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.15. Test Case TC-014**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-014 (Unit Test)	
<b>What To Test</b>	Login with incorrect username	
<b>Test Data Input</b>	incorrect username, correct password	
<b>Expected Result</b>	User is not logged in. Error message is displayed.	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-A
	<b>Relevant System Req.(s)</b>	SF-A-01
	<b>Relevant Use Case(s)</b>	UC-016

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.16. Test Case TC-015**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-015 (Unit Test)	
<b>What To Test</b>	Login with both incorrect username and password	
<b>Test Data Input</b>	incorrect username, incorrect password	
<b>Expected Result</b>	User is not logged in. Error message is displayed.	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-A
	<b>Relevant System Req.(s)</b>	SF-A-01
	<b>Relevant Use Case(s)</b>	UC-016
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

**Table 8.2.17. Test Case TC-016**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-016 (Unit Test)	
<b>What To Test</b>	Respond no to Request	
<b>Test Data Input</b>	no to request	
<b>Expected Result</b>	no changes made to visibility	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-E
	<b>Relevant System Req.(s)</b>	SF-E-05
	<b>Relevant Use Case(s)</b>	UC-001

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.18. Test Case TC-017**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-017 (Unit Test)	
<b>What To Test</b>	Register with invalid password	
<b>Test Data Input</b>	untaken email, invalid password	
<b>Expected Result</b>	User is not stored in database. Error message indicating invalid password displayed.	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-F
	<b>Relevant System Req.(s)</b>	SF-F-01
	<b>Relevant Use Case(s)</b>	UC-004

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.19. Test Case TC-018**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-018 (Unit Test)	
<b>What To Test</b>	Register with taken email	
<b>Test Data Input</b>	taken email, valid password	
<b>Expected Result</b>	User is not entered into database. Error is displayed indicating email is taken.	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-F
	<b>Relevant System Req.(s)</b>	SF-F-01
	<b>Relevant Use Case(s)</b>	UC-004

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.20. Test Case TC-019**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-019 (Unit Test)	
<b>What To Test</b>	Register with both taken email and invalid password	
<b>Test Data Input</b>	taken email, invalid password	
<b>Expected Result</b>	User is not entered into database. Error is displayed indicating taken email.	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-F
	<b>Relevant System Req.(s)</b>	SF-F-01
	<b>Relevant Use Case(s)</b>	UC-004
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

**Table 8.2.21. Test Case TC-020**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-020 (Unit Test)	
<b>What To Test</b>	Generate assessment and save with false visibility	
<b>Test Data Input</b>	descriptive text, yes to saving assessment, no to visible to advisors	
<b>Expected Result</b>	Assessment is stored to database with false visibility	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-E
	<b>Relevant System Req.(s)</b>	SF-E-04,SF-E-05
	<b>Relevant Use Case(s)</b>	UC-015

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.22. Test Case TC-021**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-021 (Unit Test)	
<b>What To Test</b>	Generate Assessment and do not save	
<b>Test Data Input</b>	descriptive text, no to saving assessment	
<b>Expected Result</b>	Assessment is not stored to database.	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-E
	<b>Relevant System Req.(s)</b>	SF-E-04,SF-E-05
	<b>Relevant Use Case(s)</b>	UC-015

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.23. Test Case TC-022**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-022 (Unit Test)	
<b>What To Test</b>	View Question Log	
<b>Test Data Input</b>		
<b>Expected Result</b>	Question Log is displayed	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UF-G
	<b>Relevant System Req.(s)</b>	SF-G-01
	<b>Relevant Use Case(s)</b>	UC-006

*Acknowledgment: Generated from the CapStone process management system ©2015*

**Table 8.2.24. Test Case TC-023**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-023 (Unit Test)	
<b>What To Test</b>	Ask Question performance	
<b>Test Data Input</b>	Question with known answer	
<b>Expected Result</b>	Answer is received within 5 seconds	
<b>Traceability</b>	<b>Relevant User Req.(s)</b>	UP-03
	<b>Relevant System Req.(s)</b>	SP-03-01
	<b>Relevant Use Case(s)</b>	UC-011
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

**Table 8.2.25. Test Case TC-024**

<b>Project Name:</b>	Intelligent Academic Planner	
<b>Test Suite</b>	TS-001: Unit Tests	
<b>Test Case ID</b>	TC-024 (Unit Test)	
<b>What To Test</b>	Log out after 1 hour of inactivity	
<b>Test Data Input</b>		
<b>Expected Result</b>	User is logged out after 1 hour of inactivity	
<b>Traceability</b>	Relevant User Req.(s)	UO-01
	Relevant System Req.(s)	SO-01-01
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

**Table 8.3.1. Execution Report of Test Case TC-001**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-001					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 click left side bar 2 click user icon 3 click perfered login method 4 input valid credential					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	10/15/2016	User is not logged in	Fail	Not implemented	10/30/2016 by Allen
2	Daria	11/15/2016	User is logged in	Pass		
3	Daria	2/1/2017	User is logged in	Pass		
<b>Execution Summary:</b> Success						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.2. Execution Report of Test Case TC-002**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-002					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 click left sidebar 2 click user icon 3 click logout button 4 back to main page					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	10/15/2016	User is not logged out	Fail	Not yet implemented	10/30/2016 by Allen
2	Daria	11/15/2016	User is logged out	Pass		
3	Daria	2/1/2017	User is logged out	Pass		
<b>Execution Summary:</b> Success						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.3. Execution Report of Test Case TC-003**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-003					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 click left side bar 2 click user icon 3 click sign in icon 4 click sign up 5 input username, email, password 6 go to profile page					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	10/15/2016	User is not added to database	Fail	Not yet implemented	10/30/2016 by Allen
2	Daria	11/15/2016	User is added to database	Pass		
3	Daria	2/1/2017	User is added to database	Pass		
<b>Execution Summary:</b> Success						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.4. Execution Report of Test Case TC-004**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-004					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 input question 2 view answer					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/15/2016	Question is not answered	Fail	Not yet implemented	
<b>Execution Summary:</b> Not yet implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.5. Execution Report of Test Case TC-005**

<b>Project Name:</b>	Intelligent Academic Planner				
<b>Test Case ID:</b>	TC-005				
<b>Testing Tools Used:</b>	Mocha				
<b>Testing Type:</b>	Agile (automated) testing				
<b>Execution Steps:</b>	1 User asks question 2 System displays response and requests feedback 3 User enters feedback 4 See that feedback is stored in database				
<b>Test Execution Records:</b>					
#	Tester	Test Date	Actual Result	Status	Defect
1	Daria	11/15/2016	Feedback is not added to database and thank you message not displayed	Fail	Not yet implemented
<b>Execution Summary:</b> Not yet implemented					
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>					

**Table 8.3.6. Execution Report of Test Case TC-006**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-006					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 User enters untranslated text 2 User requests translation 3 See that text becomes translated					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/15/2016	No text outputted	Fail	Not yet implemented	
<b>Execution Summary:</b> Not yet implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.7. Execution Report of Test Case TC-007**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-007					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 User enters ambiguous question 2 See that system responds with another question relevant to the question asked					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/15/2016	No output	Fail	Not yet implemented	
<b>Execution Summary:</b> Not yet implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.8. Execution Report of Test Case TC-008**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-008					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 User enters profile information 2 User submits profile information 3 See that profile information is stored in database					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/15/2016	Profile information not stored to database	Fail	Not yet implemented	
<b>Execution Summary:</b> Not yet implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.9. Execution Report of Test Case TC-009**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-009					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 User requests assessment 2 User requests to save assessment 3 User requests that assessment is made visible to advisers 4 See that assessment is stored in database with true visibility					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/15/2016	No assessment shown	Fail	Not yet implemented	
<b>Execution Summary:</b>		Not yet implemented				
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.10. Execution Report of Test Case TC-010**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-010					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 User receives a request to view assessment 2 User agrees to allow access to assessment 3 See that visibility of assessment is change to true in database					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/15/2016	Visibility not changed	Fail	Not yet implemented	
<b>Execution Summary:</b> Not yet implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.11. Execution Report of Test Case TC-011**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-011					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 User requests to view assessment 2 See that assessment is displayed					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/15/2016	Assessment not shown	Fail	Not yet implemented	
<b>Execution Summary:</b> Not yet implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.12. Execution Report of Test Case TC-012**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-012					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 go to retrieve and rank 2 go to corpus 3 input question and answer 4 select answer					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	10/15/2016	No change in answer quality	Fail	Not yet implemented	10/30/2016 by Allen
2	Daria	11/15/2016	Question successfully answered	Pass		
<b>Execution Summary:</b> Success						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.13. Execution Report of Test Case TC-013**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-013					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Parameter value coverage					
<b>Execution Steps:</b>	1 User attempts to login with invalid password 2 See that user is not logged in and you stay on current page.					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	10/18/2016	User is not logged in but no error is displayed	Fail	Not implemented	10/30/2016 by Allen
2	Daria	2/1/2017	User is not logged in and you stay on current page	Pass		
<b>Execution Summary:</b> Passing as expected						
<b>Acknowledgment:</b> Generated from the CapStone process management system ©2015						

**Table 8.3.14. Execution Report of Test Case TC-014**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-014					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Parameter value coverage					
<b>Execution Steps:</b>	1 User attempts to login with incorrect username 2 See that user is not logged in and you stay on current page					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	10/18/2016	User is not logged in but error is not displayed	Fail	Not implemented	10/30/2016 by Allen
2	Daria	2/1/2017	User is not logged in and you stay on current page	Pass		
<b>Execution Summary:</b> Passing as expected						
<b>Acknowledgment:</b> Generated from the CapStone process management system ©2015						

**Table 8.3.15. Execution Report of Test Case TC-015**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-015					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Parameter value coverage					
<b>Execution Steps:</b>	1 User attempts to login with both incorrect username and password 2 See that error message is displayed and you stay on current page					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	10/18/2016	User is not logged in but no error is displayed	Fail	Not implemented	10/30/2016 by Allen
2	Daria	2/1/2017	User is not logged in and stays on current page	Pass		
<b>Execution Summary:</b> Passing as expected						
<b>Acknowledgment:</b> Generated from the CapStone process management system ©2015						

**Table 8.3.16. Execution Report of Test Case TC-016**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-016					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Parameter value coverage					
<b>Execution Steps:</b>	1 User receives request to view assessment 2 User denies request 3 See that request's visibility stays false in database					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/18/2016	Can't Test	Fail	Not implemented	
<b>Execution Summary:</b> Not implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.17. Execution Report of Test Case TC-017**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-017					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Parameter value coverage					
<b>Execution Steps:</b>	1 User attempts to register with invalid password 2 See that a user is not added to database and you stay on current page					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	10/18/2016	User is not added to database but no error is displayed	Fail	Not implemented	10/30/2016 by Allen
2	Daria	2/1/2017	User is not added to database and you stay on current page	Pass		
<b>Execution Summary:</b> Passing as expected						
<b>Acknowledgment:</b> Generated from the CapStone process management system ©2015						

**Table 8.3.18. Execution Report of Test Case TC-018**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-018					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Parameter value coverage					
<b>Execution Steps:</b>	1 User attempts to register with a taken email 2 See that the user is not added to database and you stay on current page					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	10/18/2016	User is not added to database but no error is displayed	Fail	Not implemented	10/30/2016 by Allen
2	Daria	2/1/2017	User is not added to database and you stay on current page	Pass		
<b>Execution Summary:</b> Passing as expected						
<b>Acknowledgment:</b> Generated from the CapStone process management system ©2015						

**Table 8.3.19. Execution Report of Test Case TC-019**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-019					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Parameter value coverage					
<b>Execution Steps:</b>	1 User attempts to login with both taken email and invalid password 2 See that the user is not added to database and you stay on current page					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	10/18/2016	User is not added to database but no error is displayed	Fail	Not implemented	10/30/2016 by Allen
2	Daria	2/1/2017	User is not added to database and you stay on current page	Pass		
<b>Execution Summary:</b> Passing as expected						
<b>Acknowledgment:</b> Generated from the CapStone process management system ©2015						

**Table 8.3.20. Execution Report of Test Case TC-020**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-020					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Parameter value coverage					
<b>Execution Steps:</b>	1 User requests to generate assessment 2 User requests to save assessment 3 User requests that advisers cannot view their assessment 4 See that the assessment is stored in the database with false visibility					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/18/2016	Can't Test	Fail	Not Implemented	
<b>Execution Summary:</b> Not Implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.21. Execution Report of Test Case TC-021**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-021					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Parameter value coverage					
<b>Execution Steps:</b>	1 User requests to generate assessment 2 User requests to not save assessment 3 See that assessment is not stored in database					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/18/2016	Can't Test	Fail	Not Implemented	
<b>Execution Summary:</b> Not Implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.22. Execution Report of Test Case TC-022**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-022					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 Developer requests to view question log 2 See that question log is displayed					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/17/2016	Can't Test	Fail	Not Implemented	
<b>Execution Summary:</b> Not Implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.23. Execution Report of Test Case TC-023**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-023					
<b>Testing Tools Used:</b>	Mocha					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 User asks question 2 See that system responds within 5 seconds					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/18/2016	Can't Test	Fail	Not Implemented	
<b>Execution Summary:</b> Not Implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

**Table 8.3.24. Execution Report of Test Case TC-024**

<b>Project Name:</b>	Intelligent Academic Planner					
<b>Test Case ID:</b>	TC-024					
<b>Testing Tools Used:</b>	Manual					
<b>Testing Type:</b>	Agile (automated) testing					
<b>Execution Steps:</b>	1 User is inactive for 1 hour 2 See that system logs user out					
<b>Test Execution Records:</b>						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria	11/18/2016	User stays logged in	Fail	Not implemented	
2	Daria	2/1/2017	User stays logged in	Fail	Not Implemented	
<b>Execution Summary:</b> Not Implemented						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						