



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

**April 2017
(Version 5.0)**

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



Table of Contents

Table of Contents	1
Abstract	4
Report Revision History	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
Changes in Version 3.5	6
Changes in Version 4.0	6
Changes in Version 5.0	6
Problem Statement	8
Business Background	8
Needs	8
Objectives	8
Requirements	9
User Requirements	9
Glossary of Relevant Domain Technology	9
User Groups	9
Functional Requirements	10
Project Scope	10
User Scenarios	11
List of User Functional Requirements	11
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
Exploratory Studies	26
Relevant Techniques	26
Relevant Packages/Products	28
Broader Impacts	30
System Design	31
Architectural Design	31
Structural Design	32
User Interface Design	33
Behavioral Design	37
Design Alternatives & Decision Rationale	38
UI Alternative Designs	38
Version 1.0	38
Version 2.0	40
Structural Alternative Designs	44
Architectural Alternative Designs	44
Behavioural Alternative Designs	44
System Implementation	45
Programming Languages & Tools	45
Coding Conventions	45
Code Version Control	45
Implementation Alternatives & Decision Rationale	45
Analysis of Key Algorithms	45
System Testing	47
Test Automation Framework	47
Steps for Installing Mocha	47
Steps for Running Test Cases	47
Test Case Design	47
Test Suites	47
Unit Test Cases	47
Integration Test Cases	47
System Test Cases	47
Acceptance Test Cases	48
Test Execution Report	48
Unit Testing Report	48
Integration Testing Report	48

System Testing Report	48
Acceptance Testing Report	48
Challenges & Open Issues	48
Challenges Faced in Requirements Engineering	48
Challenges Faced in System Development	48
Open Issues & Ideas for Solutions	48
System Manuals	49
Instructions for System Development	49
How to setup development environment	49
Notes on system further extension	49
Instructions for System Deployment	49
Platform Requirements	49
System Installation	49
Instructions for System End Users	50
Conclusion	54
References	56
Appendix R	
Appendix U	
Appendix T	
Appendix TE	

1. Abstract

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

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

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

2. Report Revision History

2.1. Changes in Version 1.5

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

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 webflow
- 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

2.5. Changes in Version 3.5

- Modified table descriptions for use cases, requirements, test cases, and test execution reports to be more user friendly and less boring.
- Modified Objectives to include local, institutional, and global impacts
- Added in End User Manual section
- Modified design patterns

2.6. Changes in Version 4.0

- Removed Real Time Language Translation, Recommend Courses and Majors based on Personality from requirements
 - ◆ Real time translation was removed because it would reduce the accuracy of our system
 - ◆ Courses and majors were removed because it would have uncertain results at this point in time, so we are focusing on making Watson accurate.
- Modified Question Log Requirement
- Modified Use Case Diagram to reflect changes in requirements
- Modified Use Cases to reflect changes in requirements
- Added in new Test Cases
- Added in new Test Execution Reports
- Updated *Relevant packages/products flow diagram*
- Changed AlchemyAPI to Natural Language Understanding
- Updated screenshots for use case
- Updated Sequence Diagram
- Updated Relevant Techniques
- Updated Behavioural Design
- Updated User Interface Design

2.7. Changes in Version 5.0

- Moved information from Objectives to Broader Impact
- Fixed minor spelling mistakes
- Fixed some words based on recommendations
- Added in Conclusion section

- Added in figure numbers for images in the base report
- Added Achievement in Conclusion section
- Added Lesson Learned in Conclusion section
- Updated references
- Updated Broader Impact

3. Problem Statement

3.1. Business Background

IBM Watson's services, provided on the BlueMix platform, and other services from 3rd parties or developers can be utilized to conduct textual analysis and output a numerical scale of performance factor. Watson 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 corpus 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 use the IBM Watson cognitive services to create a tool that can provide consultation to students, who are interested in computing majors for their academic queries, while factoring this in assessment of their personality. The tool should be able to create an assessment based upon the student's self description and other documents they submit to the system (for example, a transcript), and the types of questions the user asks.

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

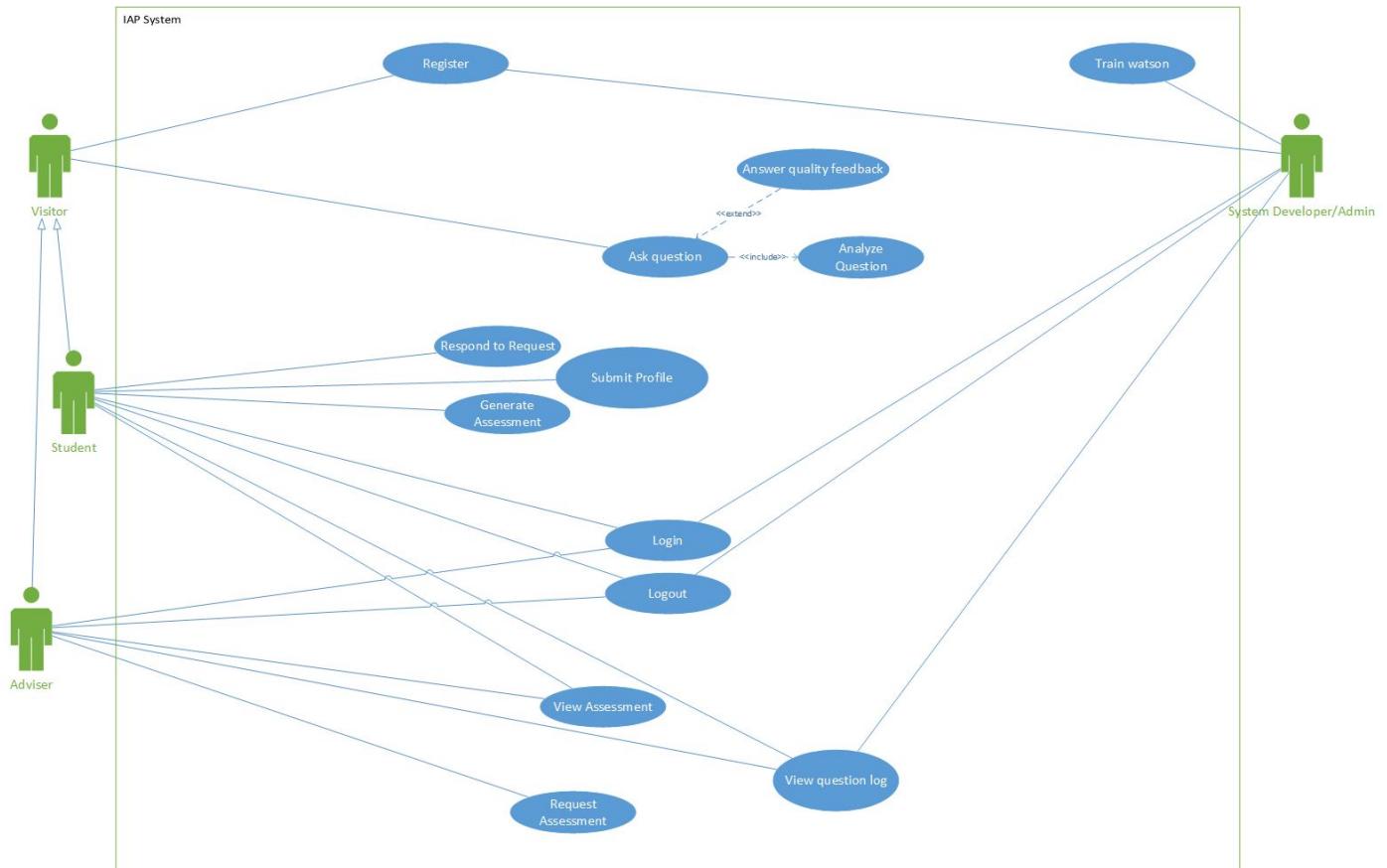


Figure 1: This is the Use Case Diagram of our system (Intelligent Academic Planner (IAP) System). Student and Advisor share the options that Visitor has (represented by the inheritance arrows). Each blue bubble represents a functionality that is present in the system, and will be gone into detail in each use case. The functionality of Answer quality feedback are sometimes used in the Ask question functionality, and Analyze question is always used. This is represented by the “extend” arrows for situational functionality, and “include” arrows for full-time functionality done by the system.

User Scenarios

For a detailed description of how each Use Case performs its tasks, see Appendix U Tables 4.1 - 4.15

A **visitor** has two major use cases: Register and Ask Question. As we aim to make this system usable by anyone, users are not required to log in to ask a question. Thus, we require that the ask question functionality be available to Visitors, who are users that have yet to register.

A **student** has the ability to login, logout, respond to a request, submit a profile, and generate an assessment, as well as the same use cases a visitor has access to (ask question and register). This enables a registered student to access personality assessments and have questions catered to their personality. In addition, they can create their own personal profile.

An **advisor** has the ability to login, logout, view an assessment, view question log, and request an assessment. In addition, they have access to the same use cases as visitors (ask question and register). This allows an advisor to advise students based on questions they've asked and assessments they've received.

A **developer** has special access to tools for training watson, displayed as the “Train Watson” use case. In addition, they have the ability to view a log of asked questions just like the adviser.

List of User Functional Requirements

For a detailed description of each requirement, see Appendix R Tables 4.16 - 4.42

- A user can log in
 - log-in should occur within 5 seconds
- A user can logout
 - log out should occur within 5 seconds
- A user can ask the system questions
 - System should conduct a textual analysis
 - System should gather data unique to the user
- The system should respond with multiple answers to a question
 - System should show a minimum of 1 related answer
- A user can create a profile page
 - System should allow 100-600 words of academic and professional interests
 - System should allow 100 words of self-description
 - System should create a personality assessment based on profile information
 - System should summarize profile and personality data to be used by advisors on request
- A user can register
 - Register should occur within 5 second
- A user can view a log of asked questions

- The system should allow users to send a list of asked questions to advisors
- A user can provide information to improve the accuracy of the system
 - A user can provide answer quality feedback
- A user's session should be managed
 - A user should log out after 1 hour of inactivity
- A user's profile should be secure
 - The system should only display information the user knows is being displayed
- A user should receive a quick response to a question
 - System should provide an answer within 5 seconds of asking the question.

Note: If you are not interested in details about the individual requirements, skip to the next section. The following contains in-depth information regarding each requirement for clarity.

Table 4.16: Log In Requirement: 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: Log Out Requirement: 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: Ask Question Requirement: 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: Multiple Answer Requirement: This user requirement requests that we include functionality in the system for users to receive multiple responses to a question. Low priority was given to this requirement since it requires we first implement asking a question, which was given highest priority.

Table 4.20: Create Profile Requirement: 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: Register Requirement: 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: View Question Log Requirement: 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: System Feedback Requirement: 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: Quick Answer Requirement: 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: Secure Profile Requirement: 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: Manage Session Requirement: 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: Log In within 5 Seconds Requirement: 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: Log Out within 5 Seconds Requirement: 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: Textual Analysis Requirement: 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 4.32: Unique Data Requirement: 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 4.34: At Least One Answer Requirement: 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: Academic Profile Requirement: 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: Self-Description Requirement: 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: View Personality Assessment Requirement: 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: Generate Personality Assessment Requirement: 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: Summarize Data Requirement: 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: Register Within 5 Seconds Requirement: 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: Question Log to Advisor Requirement: This system requirement requests that when we allow students to send questions they have asked to their advisor. Medium priority was given to this requirement because the functionality of the question log is also medium priority.

Table 4.42: Ask For Feedback Requirement: 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

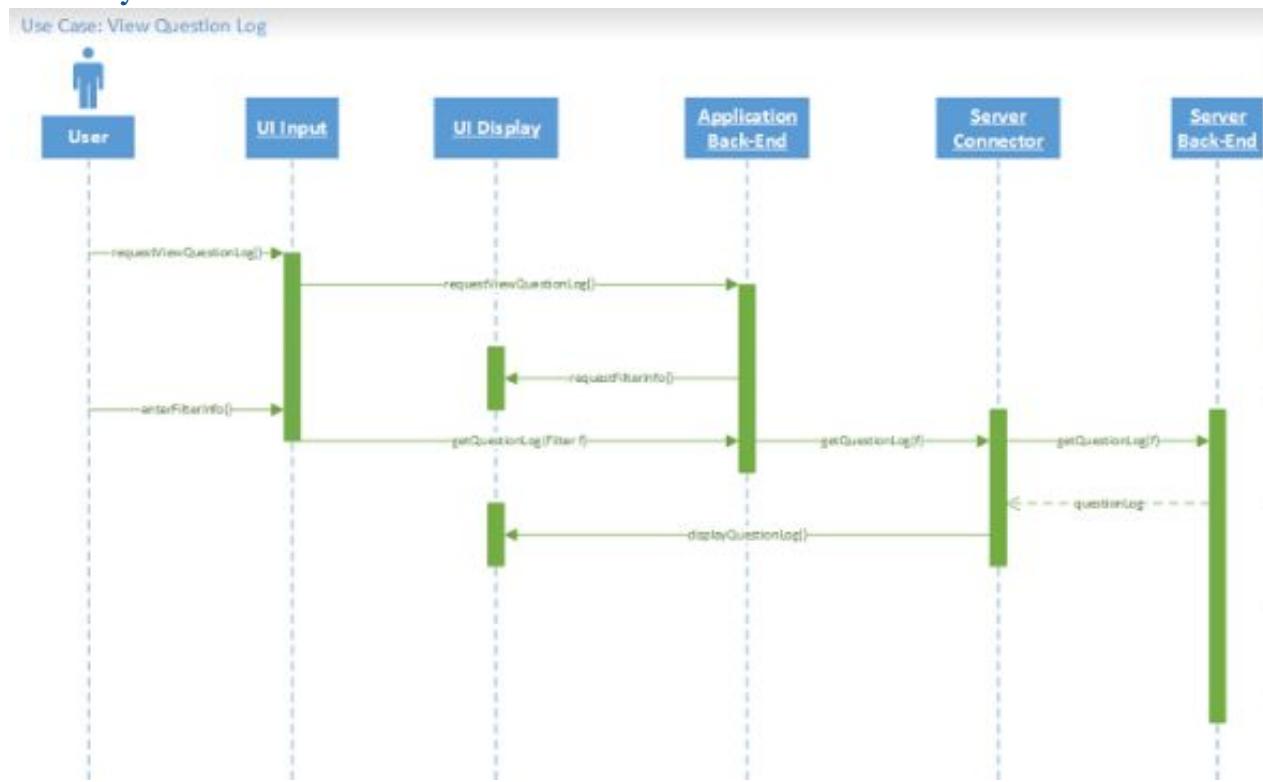


Figure 2: View Question Log Sequence Diagram: After requesting to view the question log, the system will ask for filter info and then display the question log for viewing to the user after receiving the questions from the server's back-end.

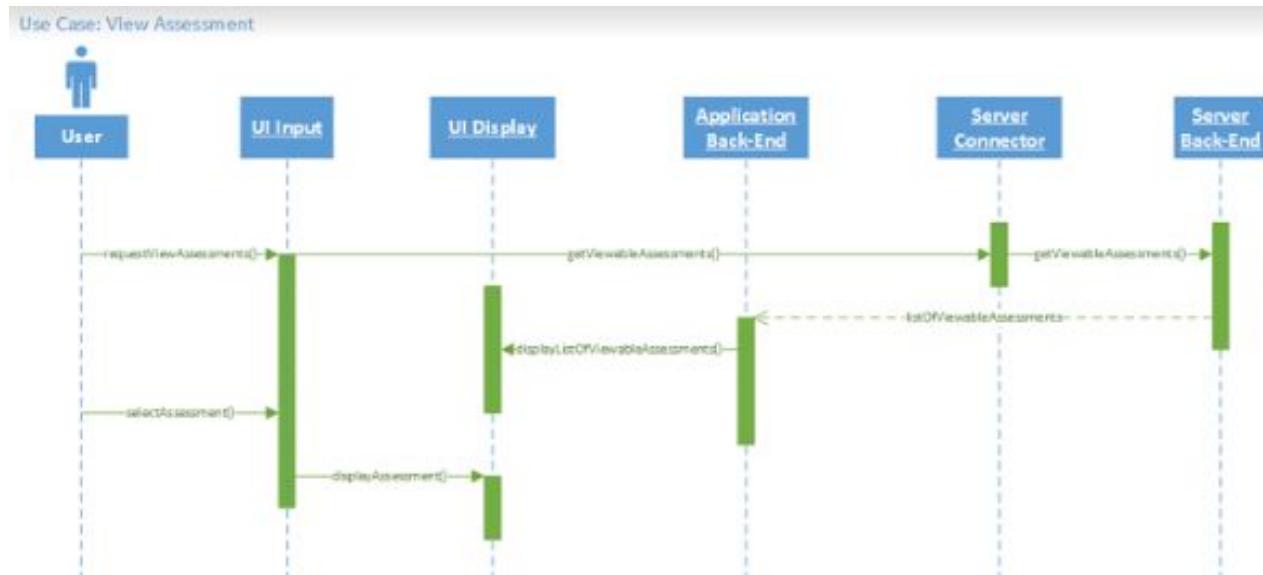


Figure 3: View Assessment Sequence Diagram: After requesting to view an assessment, the system will display a list of viewable assessments. Once the user selects which assessment they wish to view, the system will display the assessment information on the screen.

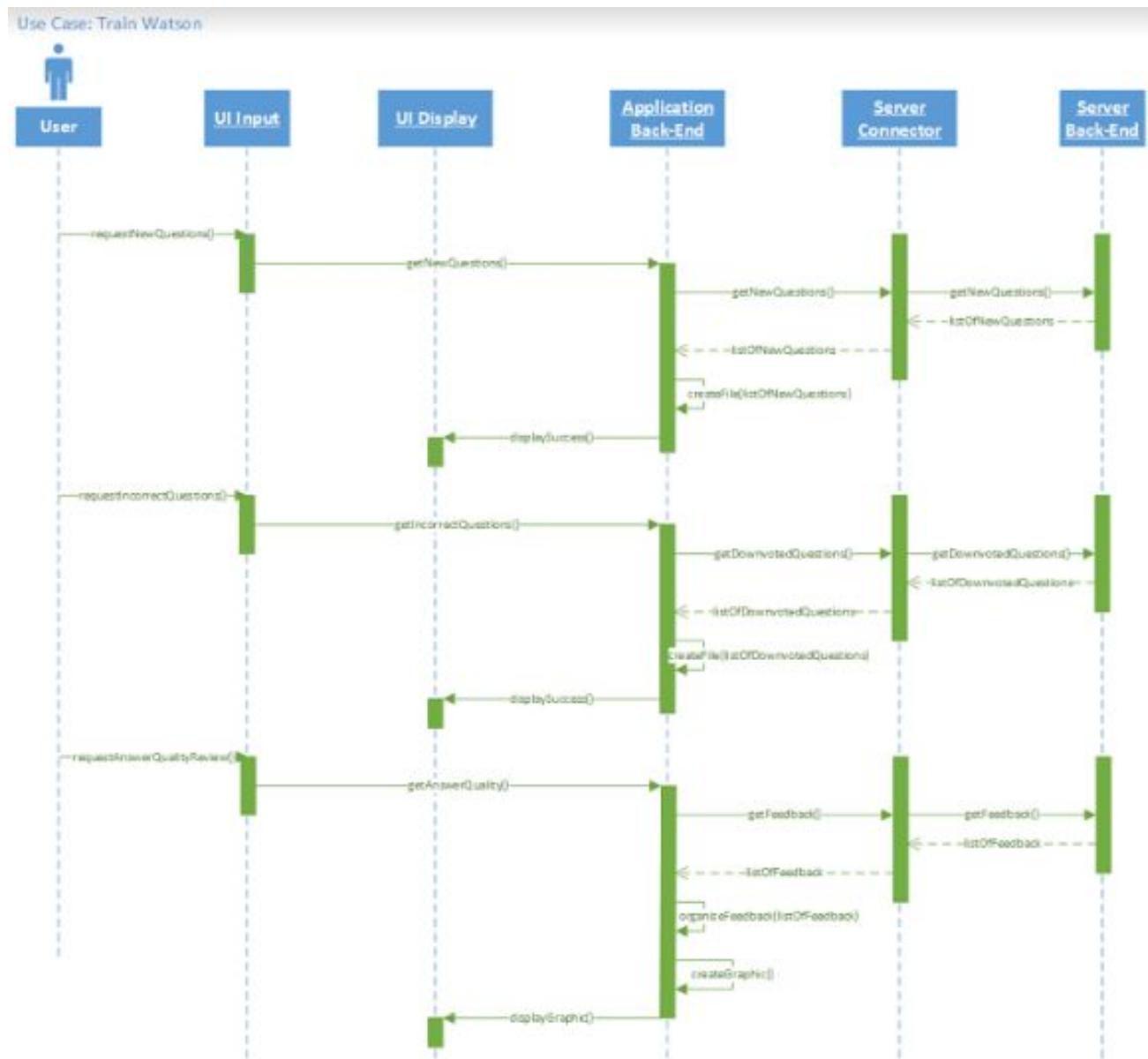


Figure 4: Train Watson Sequence Diagram: After the user either requests new questions, incorrect questions, or quality feedback, the system responds by creating a file with a list of the questions matching the filter or a graphic (such as a bar graph) of the quality feedback.

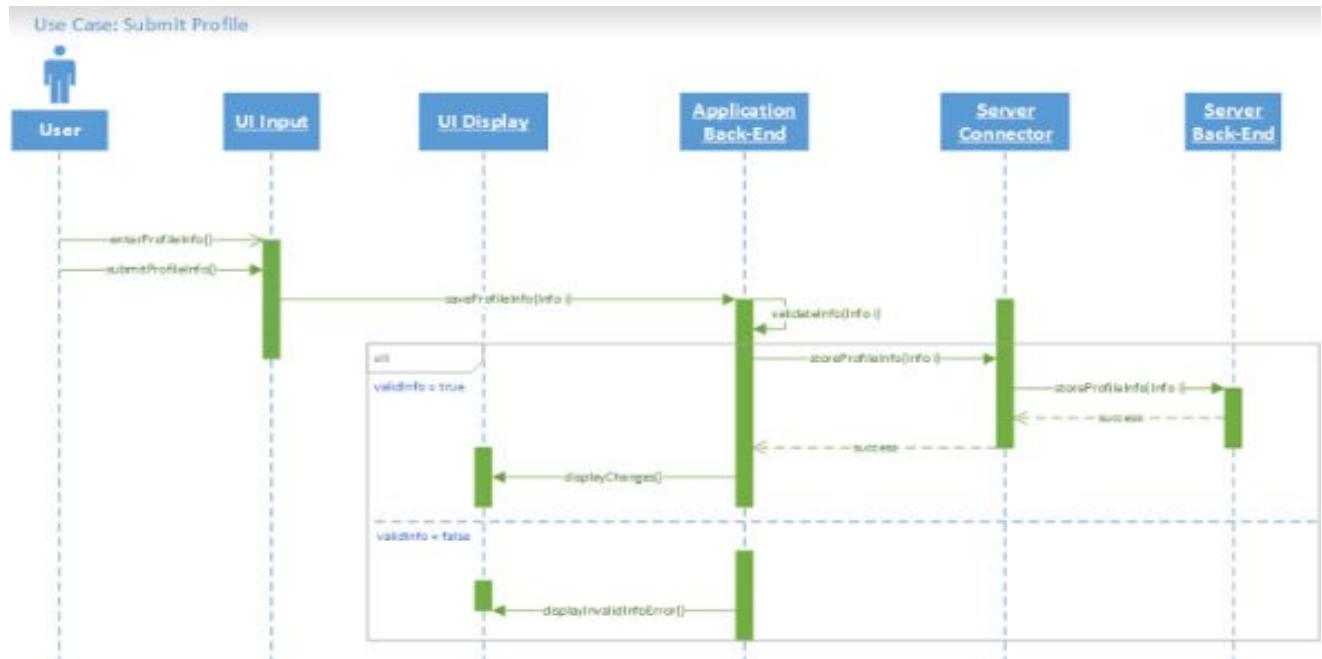


Figure 5: Submit Profile Sequence Diagram: After a user inputs their profile information and requests to save, the system validates the information to ensure that there is no invalid or dangerous input. If it passes validation, the information is saved and a message stating that it was saved is displayed. Otherwise, a message displaying that there was a problem is displayed.

Use Case: Respond To Request

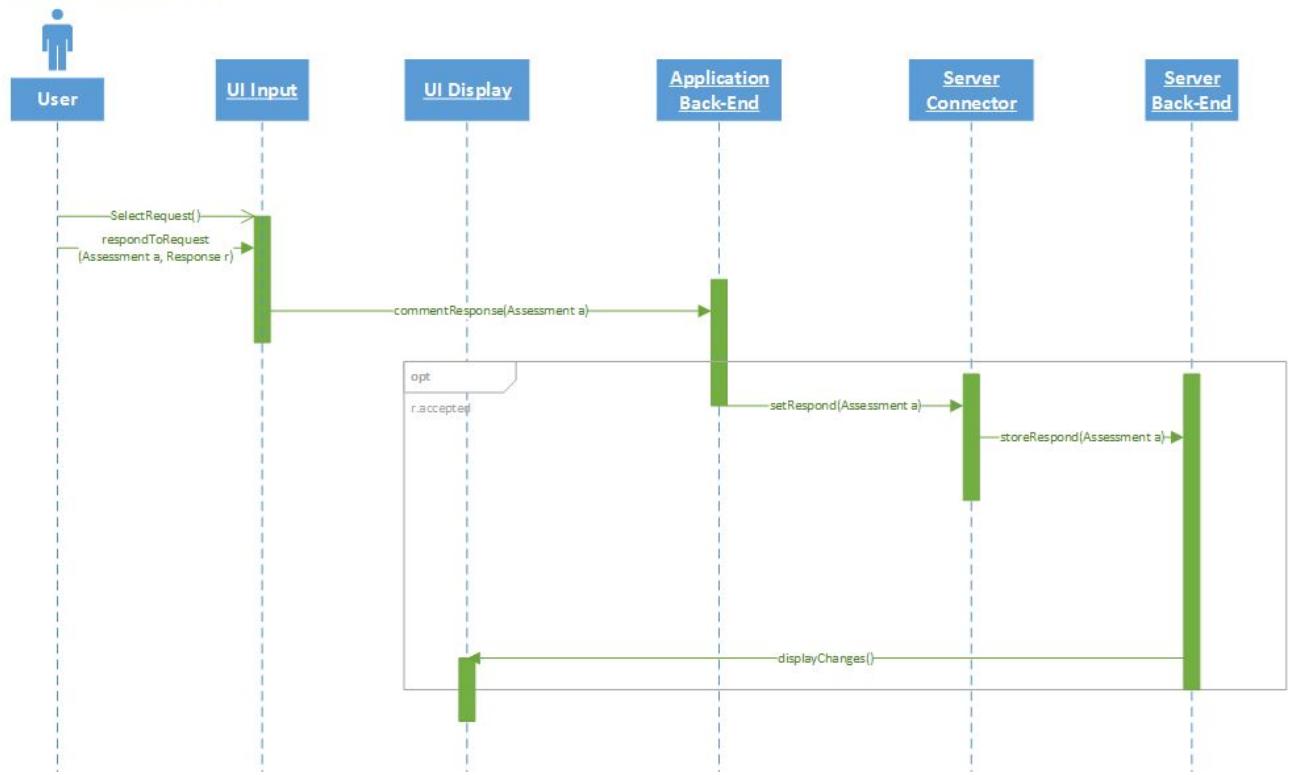


Figure 6: Respond to Request Sequence Diagram: After the user selects the request they wish to respond to and submits their response, the system sends a message back to the requester, sets the visibility of the selected requested assessment

Use Case: Logout

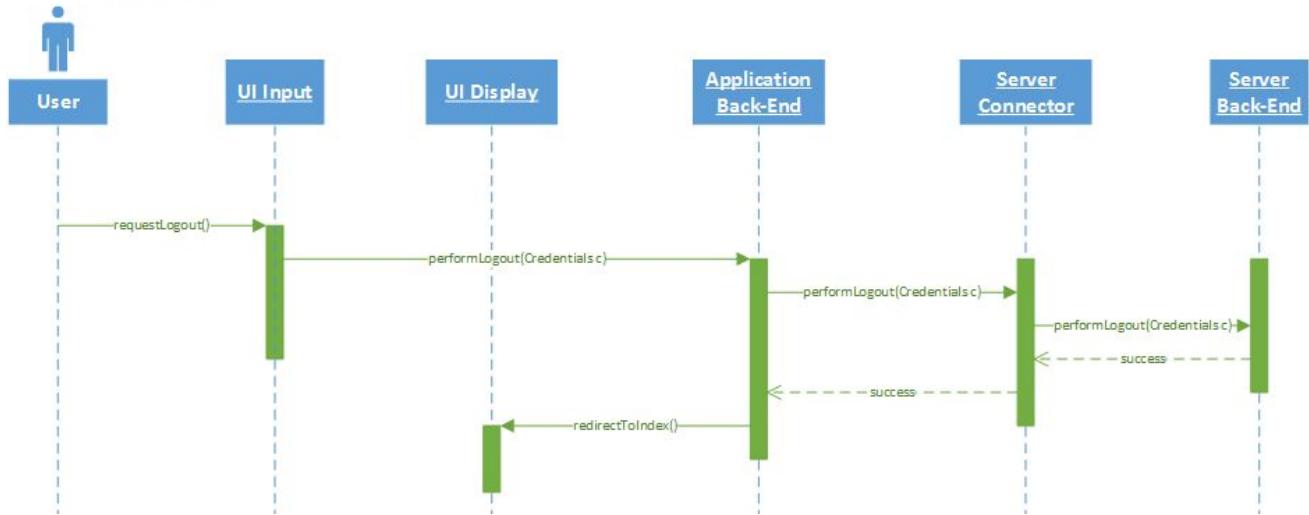


Figure 7: Logout Sequence Diagram: After a user requests to logout, the system logs the user out of the server and redirect user to index page.

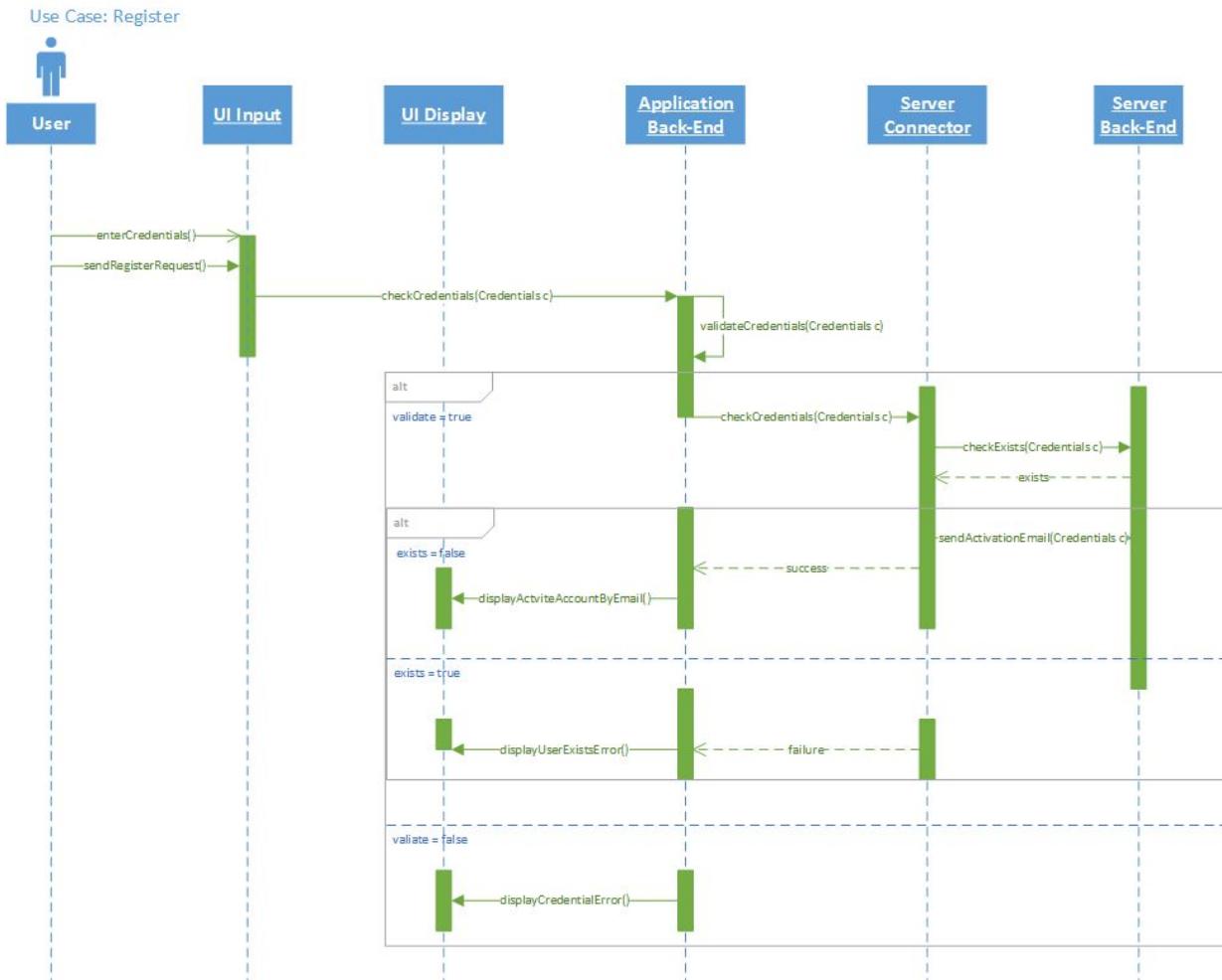


Figure 8: Register Sequence Diagram: After the user enters credentials and requests to register, the system checks the credentials to see if they meet the requirements (email format, correct number of characters, etc). If valid, the system checks if a user with the same email already exists. If a user does not exist, the system adds a new user (Inactive) to the database and displays success with message guide user active account by email that used in registration. If a user with the same email already exists or the credentials do not meet requirements, an error message is displayed.

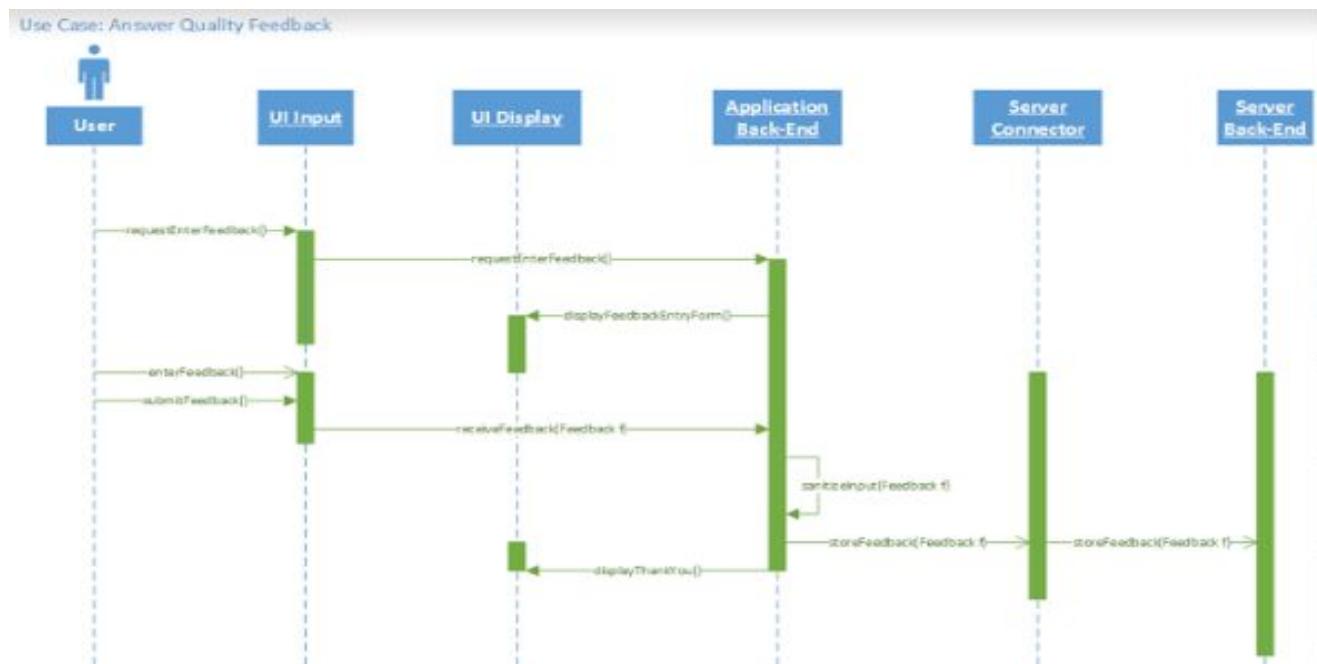


Figure 9: Answer Quality Feedback Sequence Diagram: After a user requests to enter feedback, an entry form is displayed for the user to input the feedback. Once submitted, the system checks to make sure the input is not dangerous, then stores it in the database, and finally displays a thank you message to the user.

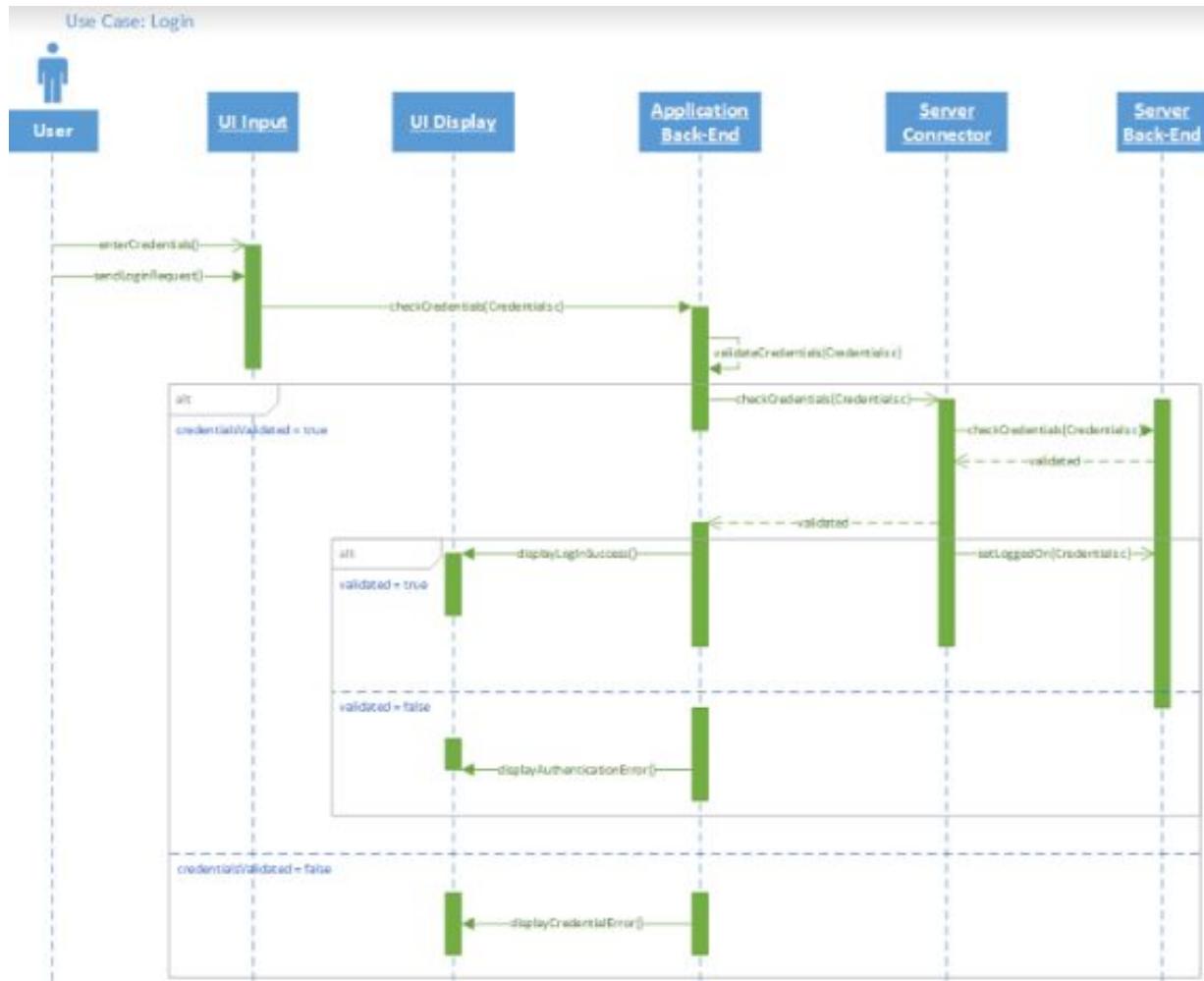


Figure 10: Login Sequence Diagram: After the user enters their credentials and sends a login request, the system checks the credentials to see if they meet requirements first, and then checks to see if they are valid. If the credentials meet requirements and are valid, the user is logged on and a success message is displayed. Otherwise, an error message is displayed.

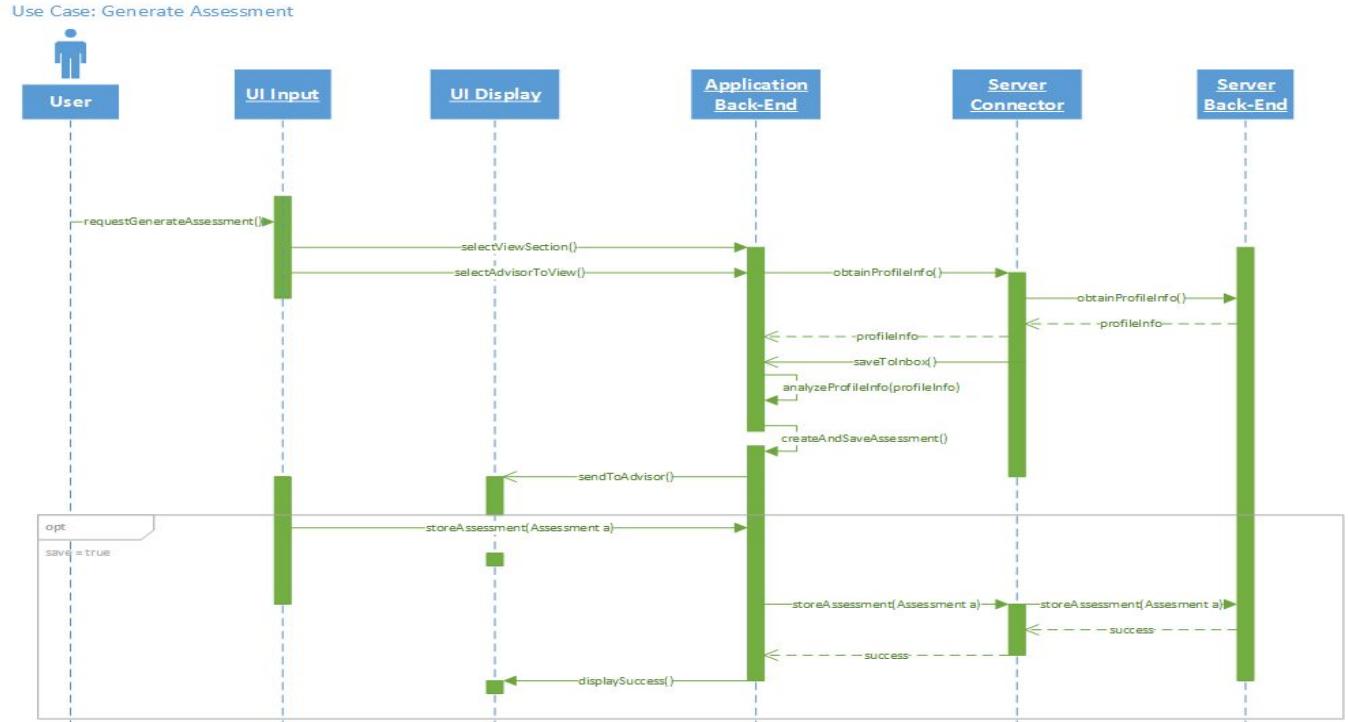


Figure 11: Generate Assessment Sequence Diagram: When the user requests to generate an assessment, the system ask what section should be included in the assessment, then obtains corresponding profile information and analyzes it. Once analyzed, an assessment is created and send to advisor. The system then automatically save the assessment to the inbox.

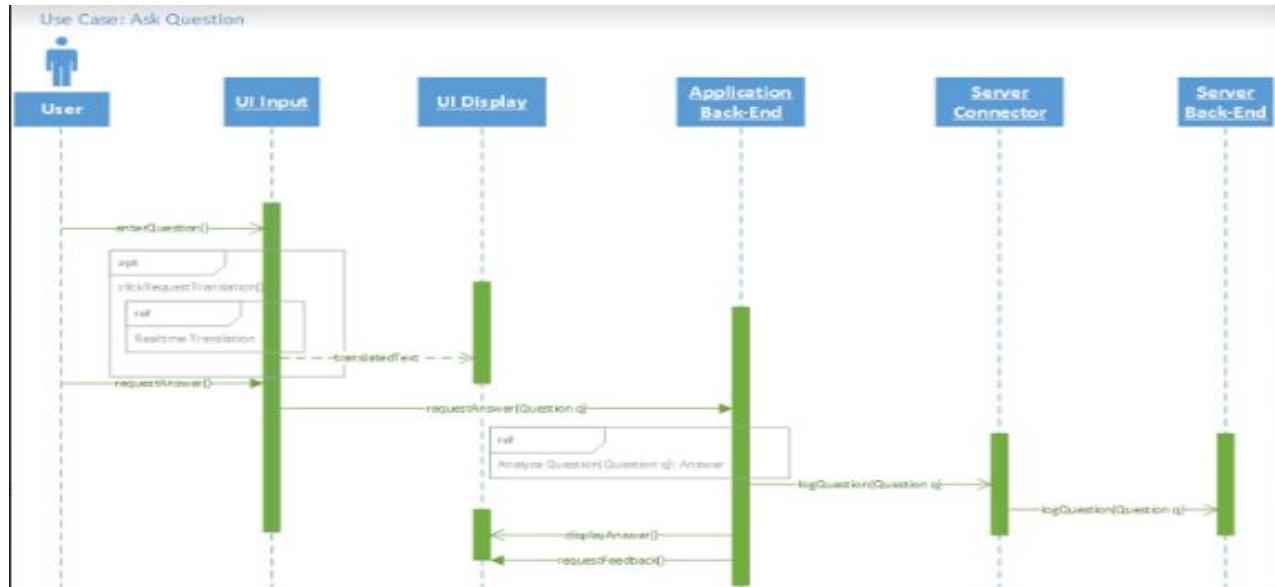


Figure 12: Ask Question Sequence Diagram: When the user enters a question, they can request translation before requesting their answer. Once an answer is requested, the system calls to Analyze Questions and performs the functionality and returns the answer. The question is then logged and the answer is displayed. After the answer is displayed, the system asks for feedback.



Figure 13: Analyze Question Sequence Diagram: The system receives a question from the Ask Question functionality. First, the system checks to see if the question is dangerous. If it is not dangerous, a textual analysis is performed. After the textual analysis, the system determines if it needs to ask a second question. If a second question is needed, it displays it to the user and then awaits input. The process above is repeated until another question no longer needs to be asked.

Data Requirements

- **UC001** - Input: User whose assessment is being requested.
- **UC004** - Input: User email address and real name for registration and login.
- **UC004** - Input: User password for registration and login.
- **UC006** - Input: Filter information for useful question log (eg. major, minor, courses).
- **UC007** - Output: List of newly generated questions exported to a text file.
- **UC007** - Output: List of unanswered frequently asked questions.
- **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: Answer Question Within 5 Seconds Requirement: 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: Secure Profile Requirement: 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: Manage Session Requirement: 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

See Table 4.46: Summary of Requirements in Appendix R.

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 was initially trained with 1000 question-answer pairs to ensure depth and accuracy of system knowledge.

An additional 4000 questions were later added to the system; the questions were based on the original 1000. For example, if the statement “Tell me about Erie.” was already in the system, then the questions “What is Erie like?” and “Can you tell me more about Erie?” would have been added. Our intention was to ensure that the system had many variations of question. We believed that this would increase the Watson’s accuracy and understanding. To our surprise, with the additional questions and training, Watson began to answer with less accuracy. Although much time was lost in retraining, knowledge of Watson’s learning process was gained. 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 Natural Language Understanding. 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^[12]
 - ◆ Server application type

- ◆ We decided to use this because most of the Bluemix APIs are designed for Node.JS handling and Node.JS is an advanced, efficient server application.
- Watson Platform
 - ◆ Conversation^[5]
 - Mainly used to classify the question input from user and determine if the question is in our domain or not.
 - ◆ Retrieve and rank^[4]:
 - Core functionality of the project
 - Uses machine training and NLP(Natural language process) to study the training dataset and provide feedback on the answer with an accuracy measurement.
 - ◆ Personality analysis^[16]
 - Assessment functionality to generate personality analysis for user. This will allow us to give more specific answers to a user based on their personality.
- MongoDB^[14]
 - ◆ Main non-relational DB for the application, since we will handle a lot of unstructured data.
 - ◆ MongoDB has good features to store and sort.
- Natural language processing
 - ◆ Monkey learn^[2]
 - A mature public accessible API provider to do NLP analysis
 - ◆ Natural^[9]
 - Equivalent NodeJS library of NLTK(Natural language toolkit) in python. It mainly focuses on customized NLP classification.
 - ◆ Natural Language Understanding^[3]
 - Similar to Monkey learn, but is managed by IBM. It will provide feedback on the detailed analysis of a sentence, so we can analyze parts of sentences to determine the intent of the question.
- Hadoop
 - ◆ The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce. Hadoop splits files into large blocks and distributes them across nodes in a cluster.

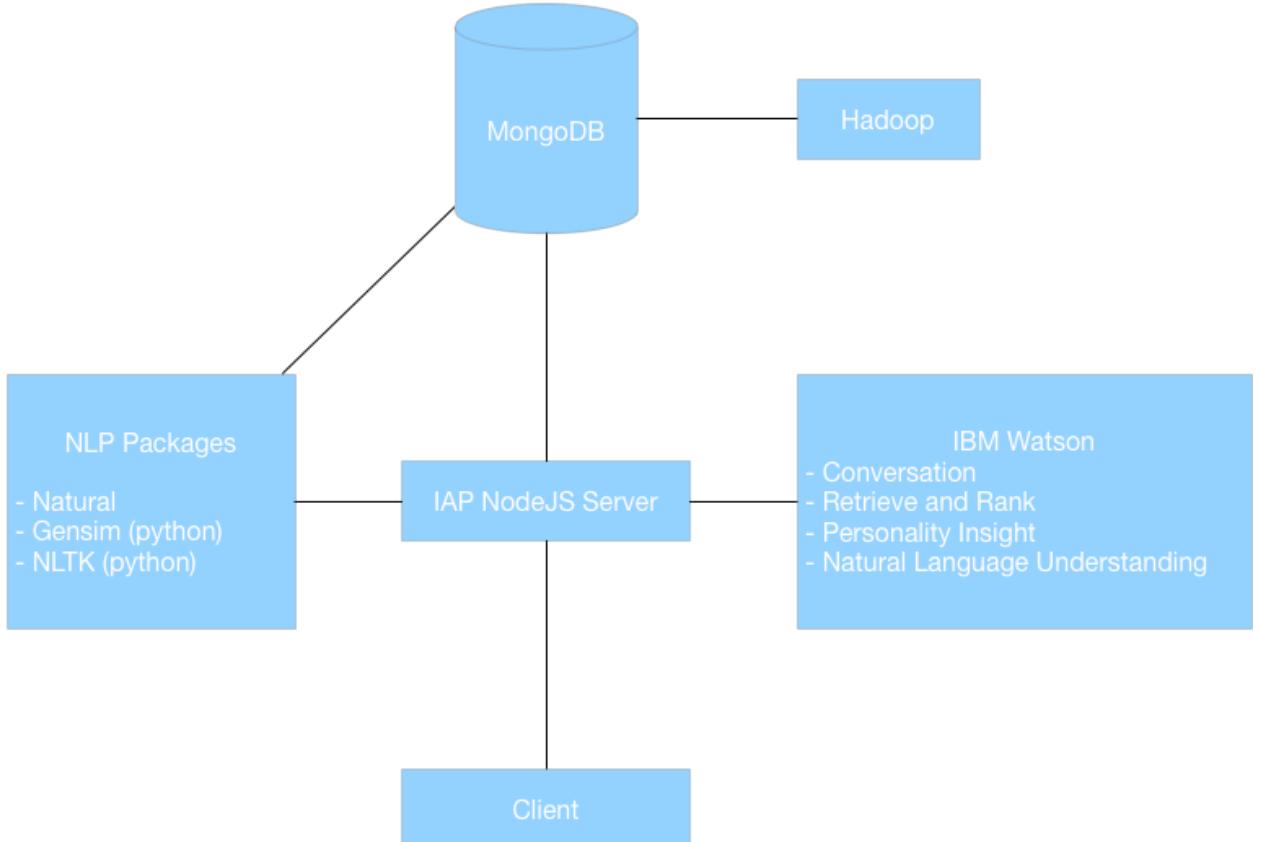


Figure 14: Relevant packages/products flow diagram

5.3.Broader Impacts

Locally, we hope to impact those interested in Penn State Behrend's CSSE programs, by offering guidance concerning the decisions pertaining to their future career. The target demographic would be incoming students, especially those just graduating from high school or transferring from another college. This would impact all involved in the advisement process, from parents to teachers.

On an organizational level, we believe that our tool could be scaled to assist other programs in universities or institutes providing counsel, such as a juvenile detention center, where perhaps a user is not yet aware of the opportunities that exist for them. More specifically, our project could be used for Penn State's World Campus, as an advisement tool that is available, for World Campus students, 24 hours a day and 7 days a week.

Globally, we hope to further the field of knowledge that combines inquiries with a user's personal information for a custom tailored answer retrieval system. Watson and its capabilities have not been fully explored. We hope to contribute to the knowledge base and understanding of Watson and its many services.

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

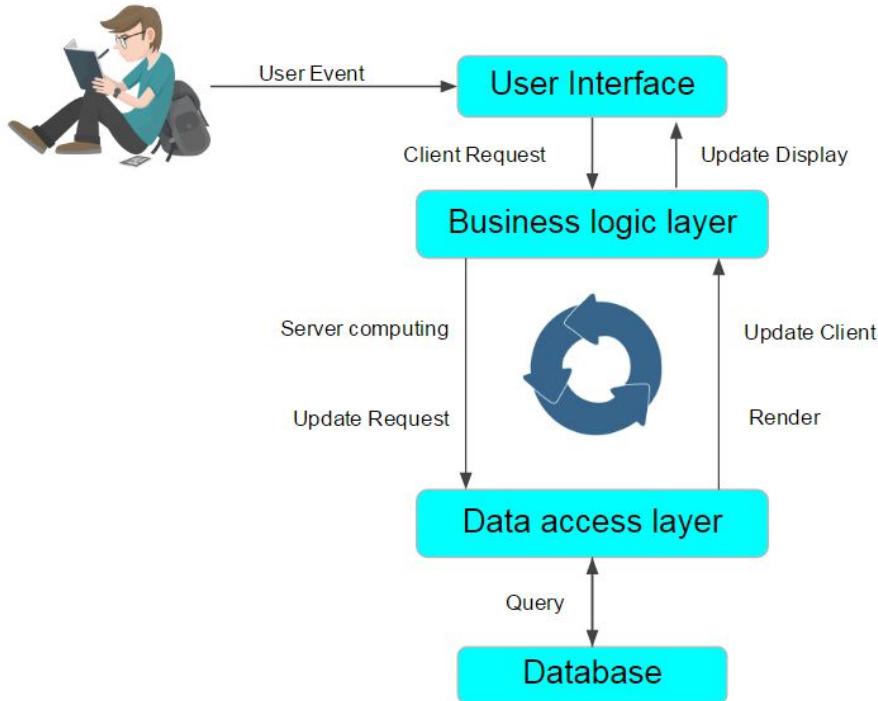


Figure 15: User initializes a “submit question” event on the web browser, then the server will handle the question through the logic layer. The database can then be queried and, if there are no exceptions, the data will be sent to Watson through Bluemix RESTful APIs. Finally, the response information will be returned in the reverse direction.

6.2. Structural Design

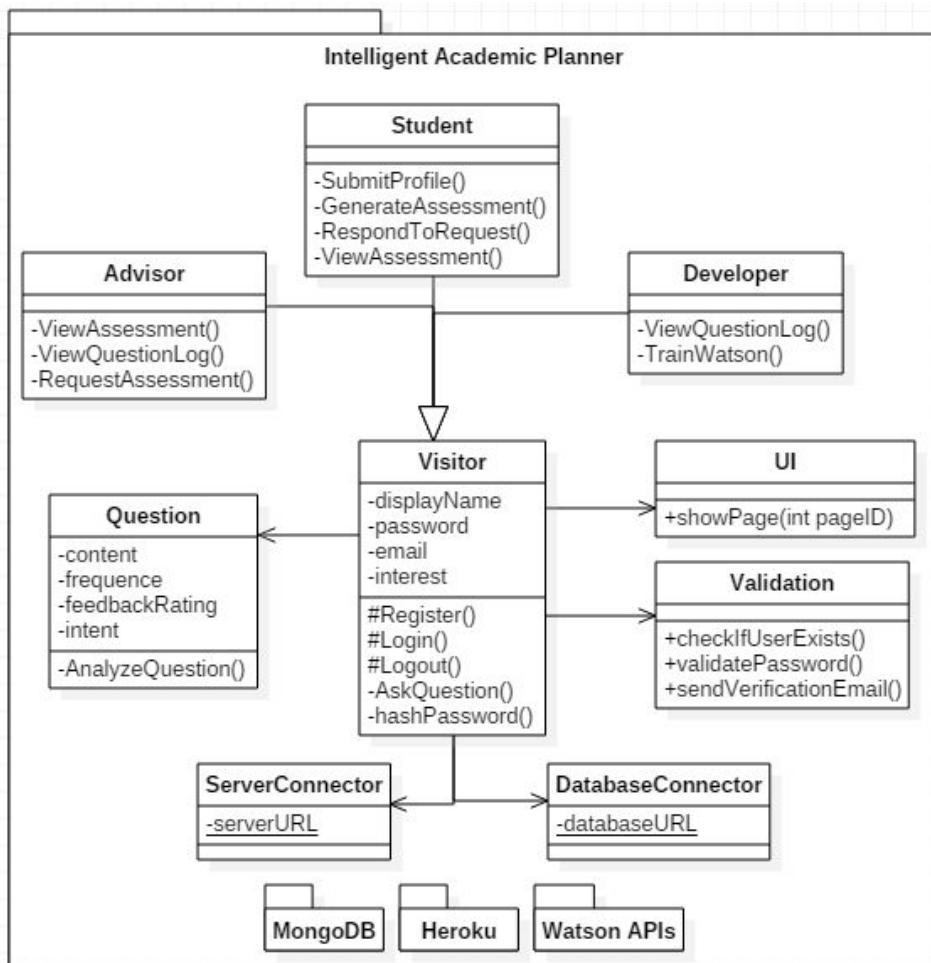


Figure 16: This is the class diagram of our system. We are using Heroku as our server and MongoDB as our database. All visitors have the ability to ask questions, and questions are stored on our server. The Validation class is used to check for safe and correct inputs, improving security of our system.

6.3. User Interface Design

The user interface is split into several web pages: Ask Questions, Login & Register, User Profile, Inbox, and Advising (for advisor accounts). 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.

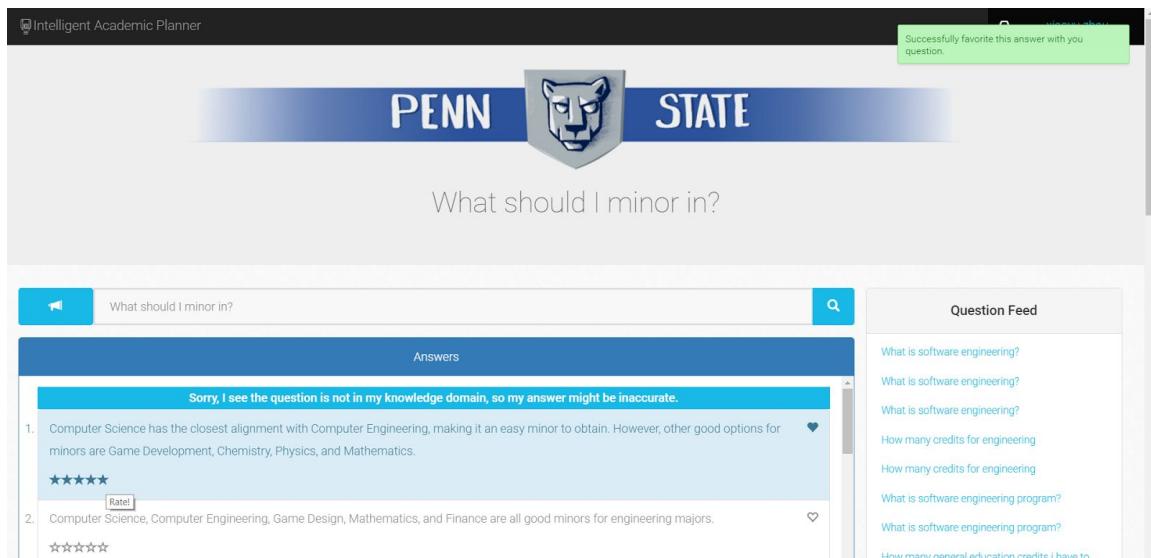


Figure 17: Ask Questions: This is the landing page of our site. A user can ask a question in the search bar, using voice-to-text or textual input. The answers are displayed in list form, with the ability to “Favorite” a preferred answer and give the system feedback through a 5 star rating system. A question 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.

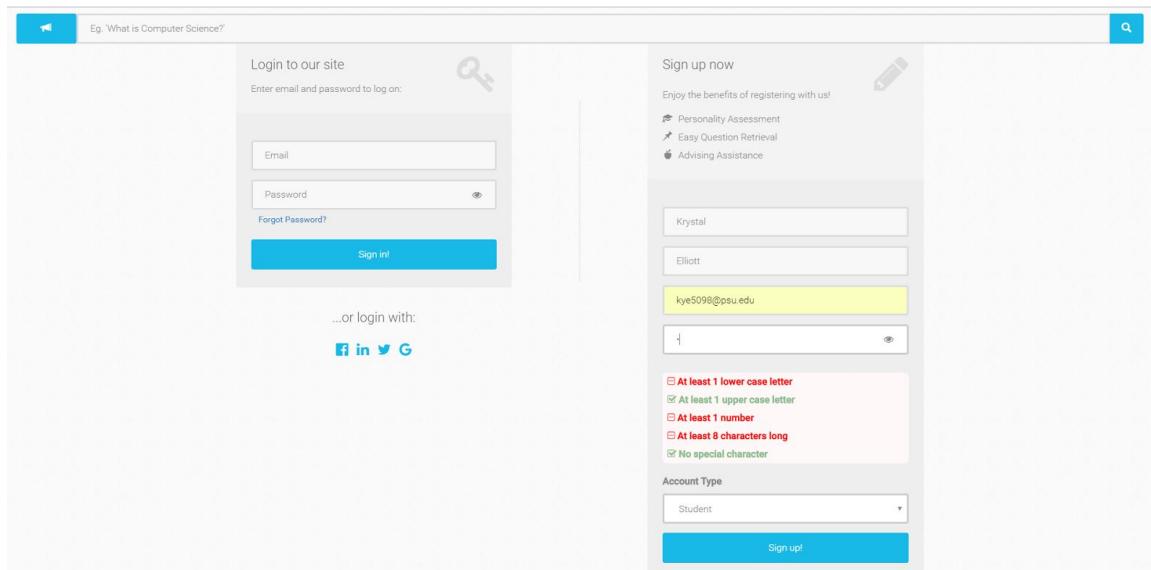


Figure 18: Login & Register: A search bar is at the top of the page to allow a user to ask a question from any page. The user must register with their name, an email, and password. The user must also choose an account type of Student, Advisor, or Developer. There are password requirements as well as the ability to retrieve a forgotten password. A user can login with their Facebook, LinkedIn, or Google account. The account must be activated before it will become active.

The screenshot shows the Behrend Profile page. At the top left is a word cloud titled "Your words" centered around the word "Computer". Below it is an "Introduction" section containing a text box with a user's personal story and an "Upload Document" section with a file input field. To the right is a "Question Feed" sidebar listing various user questions.

Figure 19: Profile: Only available after login, display information about the user, including: a conglomeration of words describing the user's interests, a profile picture, an introduction written by the user about the user, and interests. This page also features the Question Feed and Search Bar.

The screenshot shows the Behrend Profile page further down. It features a "Personality" section with a donut chart illustrating the Big 5 Personality Traits: Openness (55.2%), Conscientiousness (55.1%), Extraversion (3.0%), Agreeableness (2.0%), and Emotional range (75.9%). Below this is a "Favorite Question and Answer" section displaying a pair of Q&A entries from the user's history.

Figure 20: Profile: Further down the Profile page, there is a personality assessment, displayed as a graph depicting the Big 5 Personality Traits^[53], as well as the user's favorite question-answer pairs.

Eg. What is Computer Science?
and Alex Wouden.
Plastics Engineering Technology:
Jonathan Meckley, Greg Dillon, Anne Gohn, Bradley Johnson, Lucy Lenhardt, Alicyn Rhoades, Gary Smith, Jason Williams, and Brian Young.

Other Question and History

Q: Who is meng su?
A: The following professors currently teach both Software Engineering and Computer Science: Meng Su, Adriano Cavalcanti, Ron DelPorto, George Dudas, Xiaocong Fan, Melanie Ford, Jalaa Hoblos, Naseem Ibrahim, Wen Li Wang, Zhifeng Xiao, Guang Zhang, and Richard Zhao.

Q: What is software engineering program?
A: Computer applications software engineers analyze user needs and design, construct, and maintain general computer applications software or specialized utility programs. These engineers use different programming languages, depending on the purpose of the program. The programming languages most often used are C, C++, and Java, with Fortran and COBOL used less commonly. Some software engineers develop both packaged systems and systems software, or create customized applications.

Q: What should i minor in?
A: undefined

Copyright © 2017 | Contact

Figure 21: Profile: The bottom of the profile page will display all of the user's questions and the top answers associated with each.

Intelligent Academic Planner xiaoyu zhou ▾

PENN STATE

Inbox

Inbox	1
Outbox	
Trash	

Inbox

Sender Name
March 19, 2017
Assessment Request: Interests, Questions

Copyright © 2017 | Contact

Figure 22: Inbox: The Inbox will host the generated assessments.

The screenshot shows a web-based application for student advising. It features a sidebar on the left and a main content area. The main content area contains four sections: 'Interests' (with a text input field and two blue buttons), 'Personality' (with a text input field and two blue buttons), 'Favorite' (with a text input field and two blue buttons), and 'Question History' (with a text input field and two blue buttons). Below these sections is a 'Summary' section with a larger text input field and two blue buttons. At the bottom of the page is a footer with the text 'Copyright © 2017 | Contact'.

Figure 23: Advising: The Advising page will consist of the same fields contained within a student profile. Each section will be populated with the information from the student who sent the assessment. There is an opportunity for the advisor to leave a comment below the section. At the bottom, there is a place for the advisor to leave a summary of what they would recommend for the student.

The screenshot shows a mobile application for asking questions. At the top, it displays the title 'Intelligent Academic Planner' and a user profile for 'Krystal Elliott'. Below this is a large heading 'Ask Away!'. A subtext states, 'Our system is designed to answer all of your questions about Penn State Behrend's CSSE Department.' A search bar at the top includes a microphone icon, the text 'Tell me about Behrend.', and a magnifying glass icon. Below the search bar is a blue header labeled 'Answers'. The first two 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!'. This second answer is highlighted with a light blue background. At the bottom of this section is a five-star rating icon.

Figure 24: Responsive Design: This is an example of the view of the Ask Question page on a mobile device.

6.4.Behavioral Design

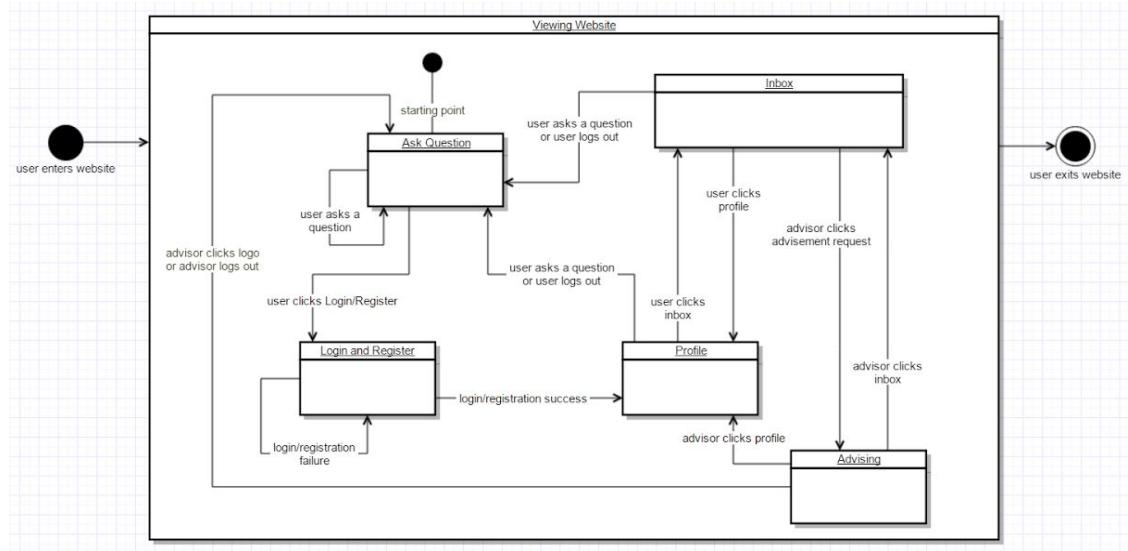


Figure 25: This is the state diagram for the webflow of the website where users ask the questions. Users will begin on the Ask Question page. If the user clicks the login/register button, they will be taken to the login/register page. When login or registration completes, they will be taken to their 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. The user can navigate to the inbox via the inbox button. An advisor can navigate to the advising page via a requested assessment in their inbox.

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.

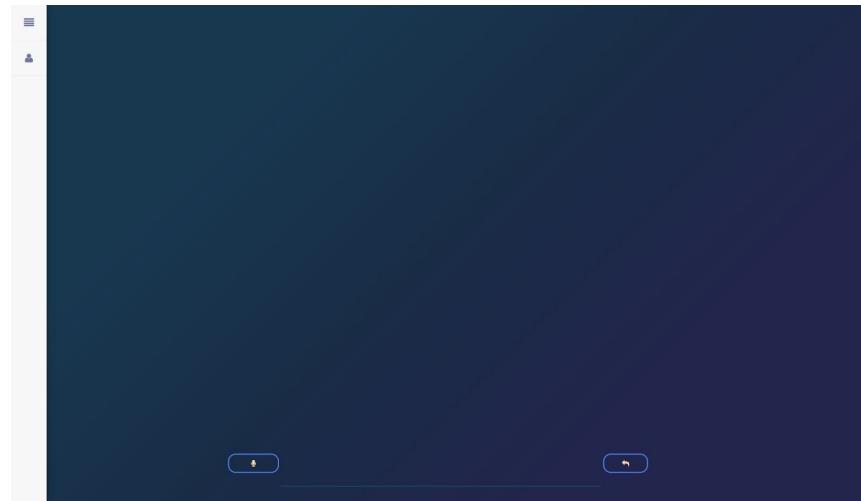


Figure 26: Ask Question: use mobile compatible view design template

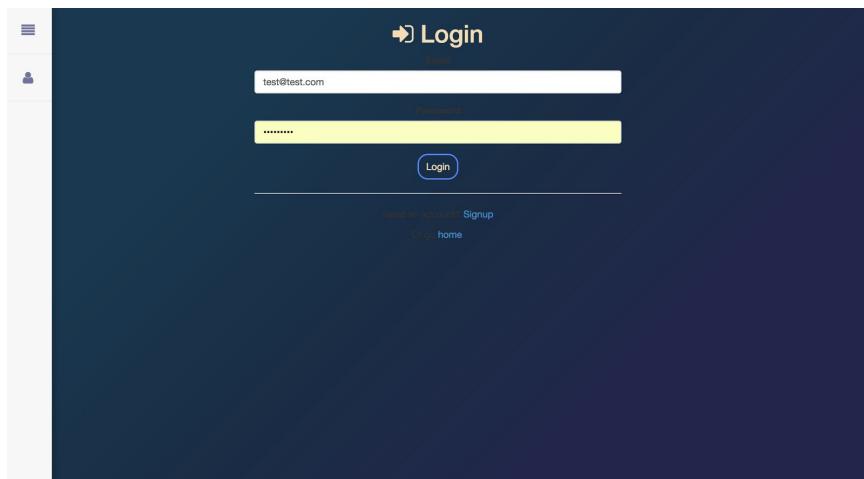


Figure 27: Login: require register email and password

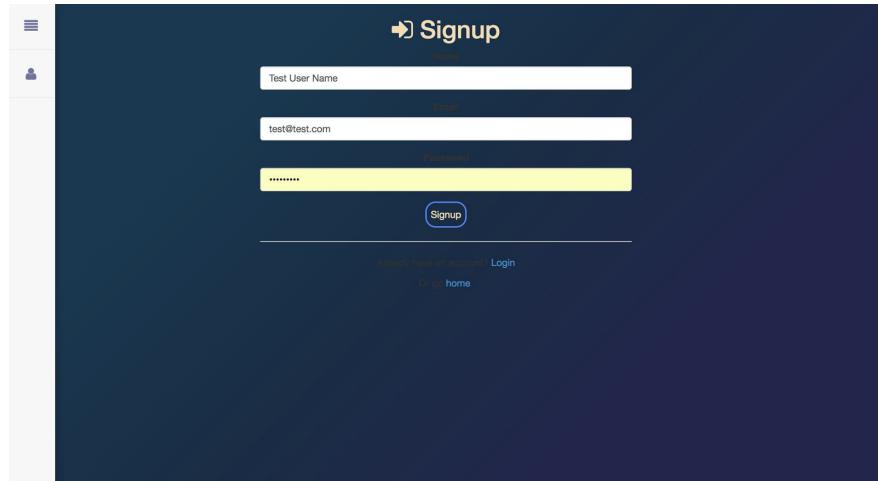


Figure 28: Register Account: require a display name, register email, and password

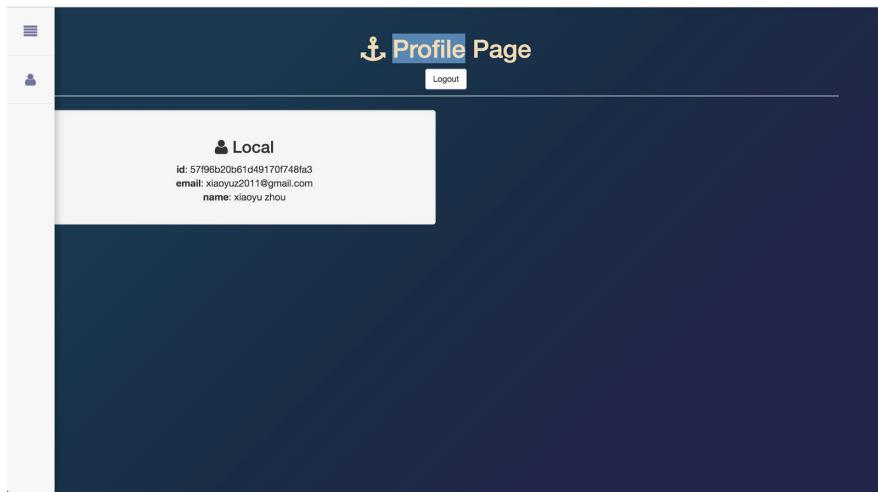


Figure 29: Profile: only available after login, display regular account information

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.

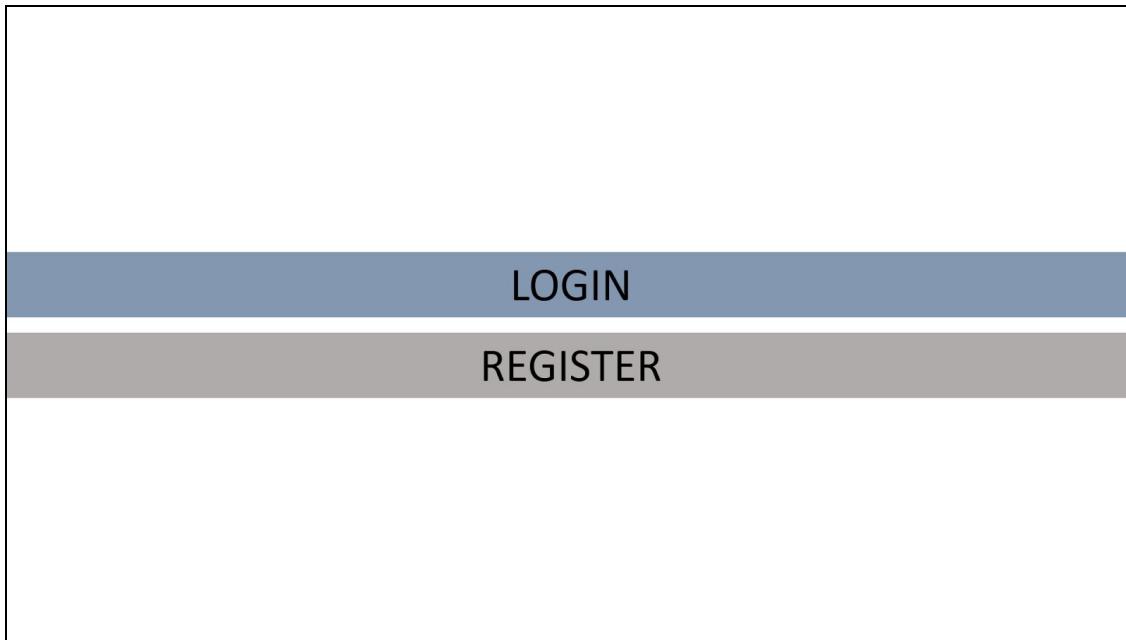


Figure 30: Landing Page: offers options of logging in and registering

A screenshot of a registration form. At the top is a blue header bar with the word "LOGIN" in white capital letters. Below it is a grey header bar with the word "REGISTER" in white capital letters. The main area contains several input fields: "Email" (text input), "Password" (text input), "Re-enter Password" (text input), "First Name" (text input), "Last Name" (text input), and "Account Type" (dropdown menu with three options: student, advisor, developer). Below the dropdown is a "SUBMIT" button. At the bottom left are navigation icons.

Figure 31: Registration: collects user data to create an account

The login screen features a dark blue header with the word "LOGIN" in white capital letters. Below the header is a light gray content area containing two input fields: "Email" with the value "kye5098@psu.edu" and "Password" with the value "*****". To the right of the password field is a "forgot password?" link. At the bottom of the content area is a dark gray "ENTER" button. A "REGISTER" button is located at the bottom of the page.

Figure 32: Login: allows user to login

The user profile screen has a dark blue header with the word "ACCESS" in white capital letters. In the top left corner is a "ASK A QUESTION" button, and in the top right corner are links for "kye5098@psu.edu" and "logout". Below the header is a navigation bar with three tabs: "PROFILE" (underlined), "ASSESSMENT", and "QUESTIONS". The main content area includes a profile picture of Krystal Elliott, her name "Krystal Elliott", and an "edit" icon. It also contains an "Email" field with the value "kye5098@psu.edu" and a "SAVE" button. There are two text input fields: one for "Extracurricular Interests" containing "art, video games, music, cooking, robots, myers briggs" and another for "Academic and Professional Goals" containing "I hope to become a Technical Lead or a Front End Developer with a strong focus on User Experience (UX)."

Figure 33: User Profile: allows user to create a personal profile, complete with picture, interests, and goals

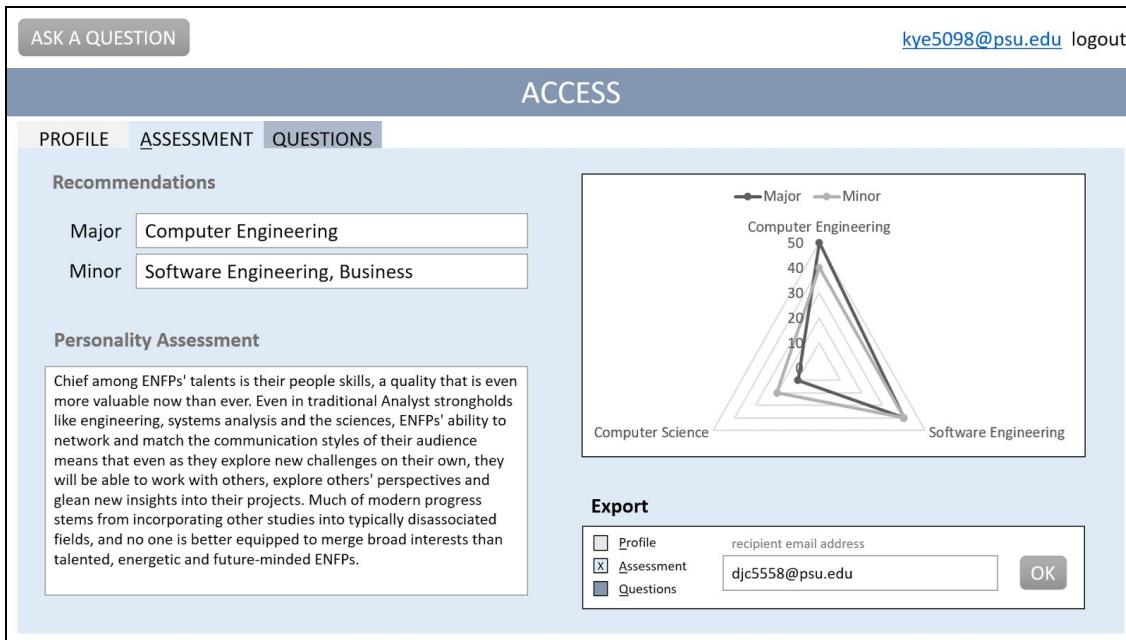


Figure 34: Assessment: allows user to view their personalized assessment and to send it, partially or in full, to an email address.

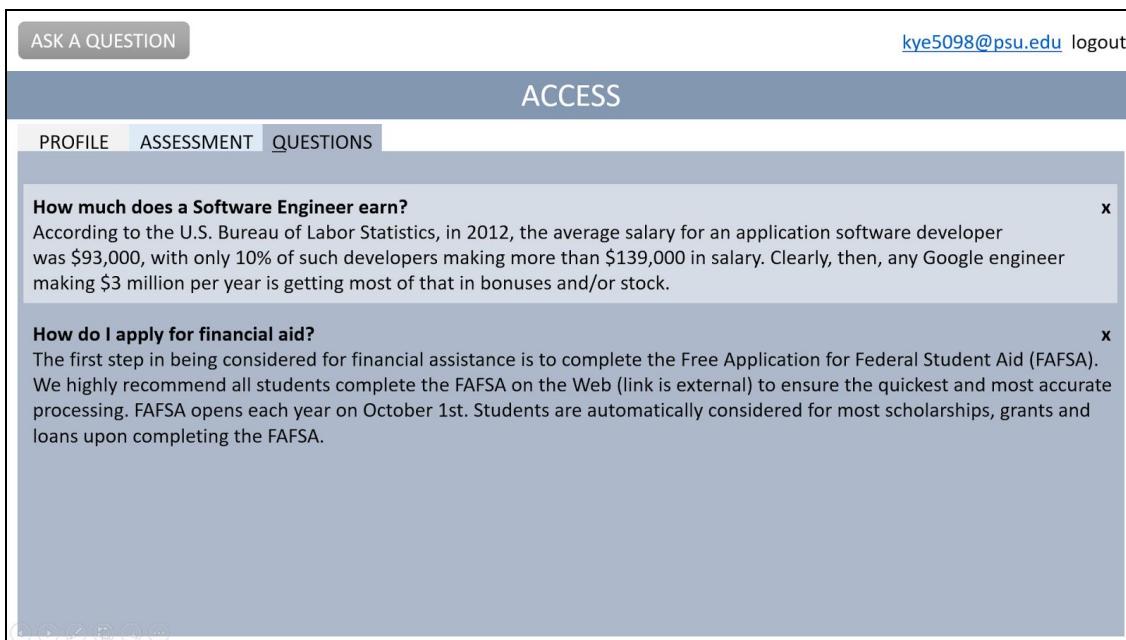


Figure 35: Previous Question Log: displays questions previously asked by users and their preferred answers

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 a user email "kye5098@psu.edu" with a "logout" link 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 answers listed as numbered items:

- 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. ★★★★★

Each answer item includes a small blue heart icon at the end. The interface has a clean, modern design with a white background and light blue header bars.

Figure 36: Ask Question Interface: allows user to ask questions, rate the received answers, and select one preferred answer

Structural Alternative Designs

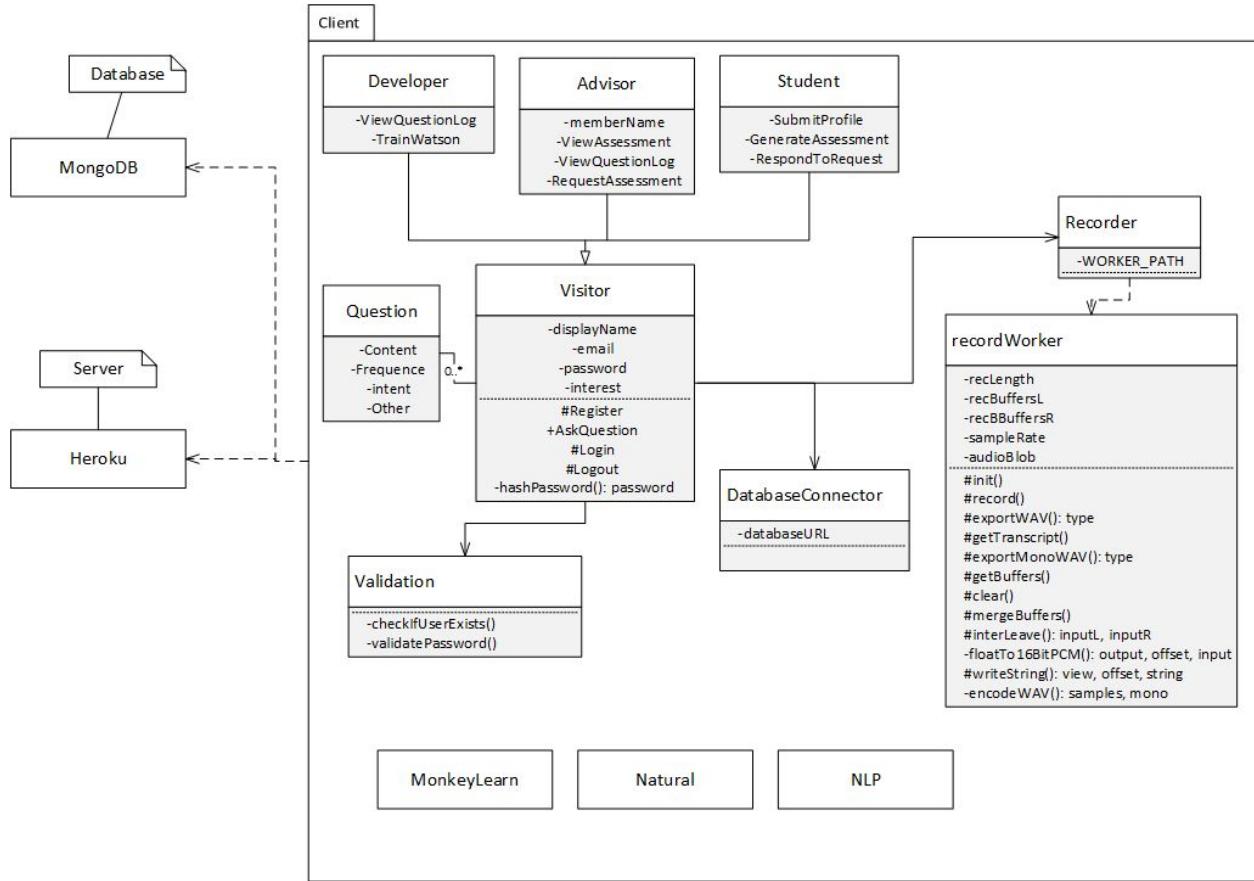


Figure 37: alternative structural design

- 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.

Behavioural Alternative Designs

- No alternative considered.

7. System Implementation

7.1. Programming Languages & Tools

- Bluemix (conversation, retrieve and rank, Natural Language Understanding,...)^{[3][4][5]}
 - ◆ 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 Natural Language Understanding, and placed into the database.
- MongoDB^[14]
 - ◆ The MongoDB is mainly used to maintain user information, question and answer information, and system analysis.
- Node.js^[12]
 - ◆ 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^[13]

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

- ask_question(input){

```
    humanizeString(input);
    getInterests(input);
    formatAIReadable(input);
    updateUserInterest(input);
}

→ formatAIReadable(input){
    Foreach interest in input do:
        interestWeight <- getConfidenceOfInterest(input);
        input.append(interest * interestWeight)
    Return input;
}
```

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

For a detailed description of each test case, see Appendix T Tables 8.2.1 - 8.2.25

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 login. This would pass a test.

Table 8.2.2: Test Suite TS-002: Acceptance Tests: Summary of all automated Acceptance Tests. In these tests, we make sure that the requirements we have stated in the documentation are successfully being met.

Table 8.2.3: Test Suite TS-003: Manual Tests: Summary of all manual Acceptance Tests. These perform the same duties as the automated ones, but are performed manually by our QA Lead (Daria Cook).

Unit Test Cases

Please see Appendix T for information about individual test cases.

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

Please see Appendix T for information about individual test cases.

8.3. Test Execution Report

To see a report of each test case being executed, see Appendix TE Tables 8.3.1 - 8.3.24

Unit Testing Report

All unit tests are passing or failing as expected, depending if a specific feature has been implemented at this time.

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

All acceptance tests are passing or failing as expected, depending if a specific feature has been implemented 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 Natural Language Understanding 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
- Migrate AlchemyAPI to Natural Language Understanding due to the deprecation of AlchemyAPI at March 2017.

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 member 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^[9]
- Analytical library
- Visual recognition on building^[7]
- Attitude analysis on question
- Campus direction utility
- Course info helper (location, material, etc...)
- LionPath
- Schedule
- API
- News/feeds utility^[17]

10.2. Instructions for System Deployment

Platform Requirements

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

MongoDB: version require 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

1. Navigate to intelligent-student-advisor.herokuapp.com.
2. Ask a question on the index page by typing the question on the search bar in the middle.

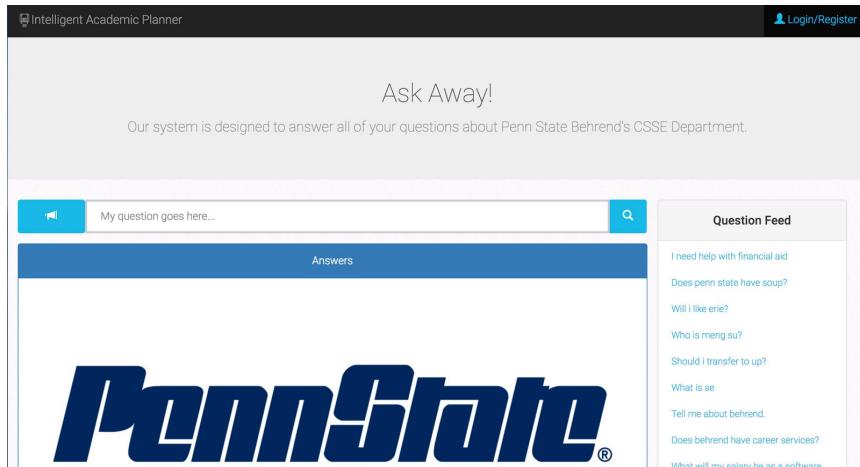


Figure 38: Index page

3. You can press the magnifier button or press the [enter] button to ask the question. You should see the answers show up in the answer section as a group of 6. The top answer is on the top with a blue background color.

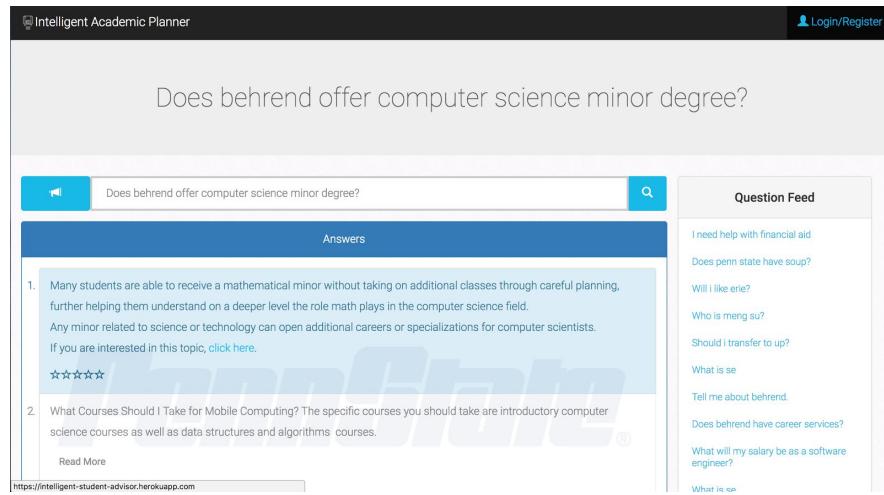


Figure 39: Index page with question asked

- To login or register for access to personality assessments, click on the Login/Register button in the upper right corner. Fill in your email and password to login. Or, fill in first name, last name, email, password, and account role to register.

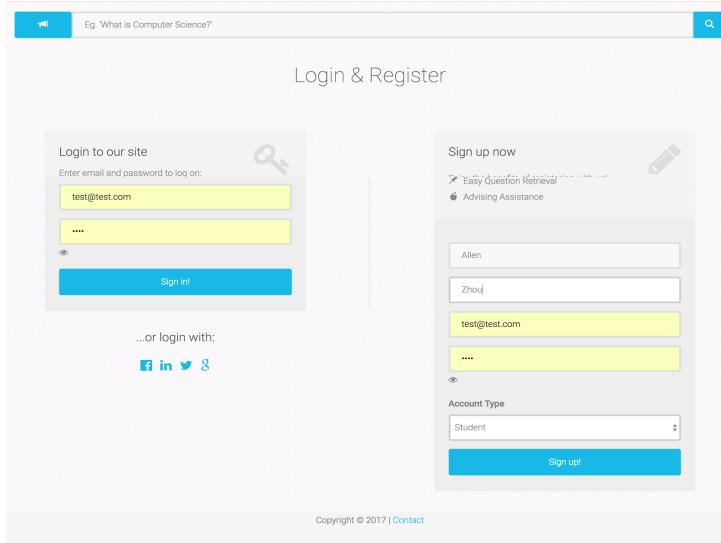


Figure 40: Login / Register Page

- After logging in, you can favorite and rate answers based on your satisfaction. By favoriting an answer ,the system remembers the answer with the question pair for you, you can go to your profile page to view it again in the future.

An electrical engineer is someone who designs and develops new electrical equipment, solves problems and tests equipment. They work with all kinds of electronic devices, from the smallest pocket devices to large supercomputers.

Electrical engineering deals with electricity, electro-magnetism and electronics.

Computer engineering is the branch of engineering that integrates electronic engineering with computer sciences.

Computer engineers design and develop computer systems and other technological devices

Electrical Engineering encompasses Computer Engineering. An Electrical Engineer can do a Computer Engineers work, but in some cases Computer Engineer can't work with core electrical stuff. There are even available courses with can help Electrical engineers to get jobs in IT industry but you hardly find any courses which helps Computer Engineers to get jobs in the electrical field.

★★★★★

Figure 41: Favoriting and Rating of Questions

- Click your name and avatar on the top right corner, then click the profile button to view your personal information.

7. Once you are on the profile page, you can view following section about yourself:
 - a. InterestCloud, a key term map indicates your interests from you introduction and questions.

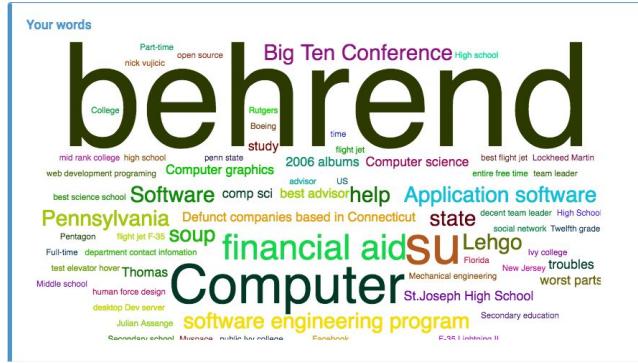


Figure 42: InterestCloud

- b. Introduction: where you write a self descriptive paragraph or submit a text/word file about you, if you want to get an analysis about your introduction, you need to input 100 words minimum.

Introduction

It's been almost one year since I left from college. At the beginning of self challenging then became desperate deal with suspicions of my capability, the only thing on top of my mind now is: go back to college and continue to explore. My story started from St.Joseph High School, where I met Thomas, my classmate and a close friend, also a decent team leader. We hanged out every weekend. At first the conversation was

Save Undo Redo Cancel

Upload Document

Drop files here to upload

Figure 43: Introduction and document uploading

- c. Question history: shows your favorite question and answer pairs.

Figure 44: Favorited questions on profile page

8. You can click any question that someone already asked through the question feeds panel, which will take you to the index page again and automatically input the question for you.



Figure 45: Question Feed

9. To change your profile's avatar, click the change avatar button to change the avatar on the profile page.

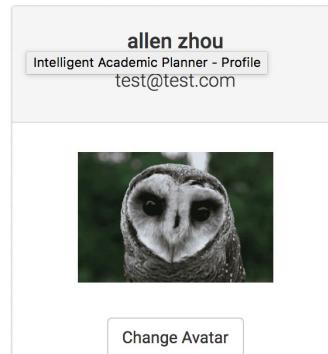


Figure 46: Change avatar

10. To generate an assessment, click the generate assessment to get an assessment of available information from you and send to an advisor.
11. You can view your personal messages from advisors by clicking the inbox button in the upper right corner.

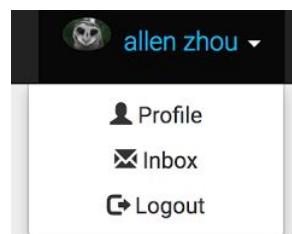


Figure 47: Drop-down menu

12. Clicking the logout button will end your session.

11. Conclusion

11.1. Achievement

- Production ready implementation, the project is able to be accessed remotely and publicly. The team also finished building a small scale mature knowledge domain corpus and used it to train Watson. In general, the Natural Language Understanding functionality is able to recognize 80% of a user's question and respond with at least 60% accuracy for any question within the knowledge domain. The team also conducted unit tests to cover at least 90% of frequently used functionalities. Finally, the project passed the 1st phase of the Nittany Lion IBM Watson Challenge^[54] securing a reward of seed funding.

11.2. Lessons Learned

- Daria
 - ◆ A good portion of my time for this project was spent working with Watson's Retrieve and Rank in order to improve the accuracy of the system. It was a lot of work to gather all of the documents we needed at the beginning of the semester, totalling 1,000 questions and approximately 700 answers. Later on to try to improve the accuracy, I did a retraining by rewording the questions in different ways and ended up with around 4,000 questions after about a month and a half of work. This did not improve Watson's accuracy at all, which significantly hurt our progress, but it gave us significant insight into how Watson learns.
 - ◆ During this project, I learned several new things (NodeJS, JavaScript, some HTML while looking at Krystal's Frontend code, Mocha Testing). I also learned more things about GitHub, and how to use it with Project Management.
- Allen
 - ◆ Most of the time I have reviewed and updated my knowledge of web development, including both frontend and backend: server system(Node.JS), algorithm(NLP) and data structure, such as ES2017 and JSX.
 - ◆ During the same time, I had lot of experience with the IBM Bluemix Watson platform and its services, such as Retrieve and Rank, Natural Language Understanding (previously known as AlchemyAPI), and conversation services. I gained the most experience in using API calls, SDKs, and some of the most advanced AI technology in Natural Language Processing.
 - ◆ Also, I have touched API and SDK for the potential extension of the project, such as React Native for mobile development, Twilio for programmable VOIP, and ngrok for proxy local server to internet.
- Krystal
 - ◆ My main focus was the front end of the application, thus I improved my HTML, CSS, and JS coding abilities. I learned more about creating a mock up of a system based on requirements and desired features. I

learned about responsive designs to accommodate desktop and mobile users. I gained experience with Bootstrap's framework.

- ◆ I learned a lot about teamwork and the importance of hearing everyone's voice in the creation of a design. It's important to make sure that the front and back end developers have the same vision to work towards. It's important to stay in constant communication and to regularly integrate front and back end code to ensure that all of the pieces are connecting properly. If the pieces aren't connecting then it's important to be able to work through it together to figure out a solution to continue the advancement of the project.
- ◆ I learned about Watson's services and about the process of training an AI, which in our case meant using Retrieve and Rank. I learned how much information must be gathered and fed to the AI to create a smart system. Through our meetings with our advisor, I learned about how an AI makes connections to determine the answers from clusters of information.
- ◆ Finally, I learned a lot about collaboration of code and version control. Git, though frustrating at times, is ultimately a life saver. Commit, commit, commit!

11.3. Acknowledgement

- This project was financially supported by Penn State Behrend and in part through the seed funding acquired in the 1st phase of the Nittany Lion IBM Watson Challenge .
- This project was sponsored by Dr. Adriano Cavalcanti of Penn State Behrend.
- This project was advised by Dr. Meng Su and Dr. Ron DelPorto of Penn State Behrend.
- Dr. Xiaocong Fan of Penn State Behrend taught the SWENG480 and SWENG481 courses that we worked on this project in.
- A special thank you to our sponsors and advisors for bringing their enthusiasm and well thought out suggestions to every meeting and presentation.
- A special thank you to Caitlin Maloney for the amazing banner artwork.
- A special thank you to Todd Say for allowing our team to hold meetings during work as long as all other tasks were completed. Many Friday shifts were dedicated to team integrations. Because of you, we were able to sleep in a little later on Saturday mornings.

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) Natural Language Understanding
<https://www.ibm.com/watson/developercloud/doc/natural-language-understanding/index.html> (accessed Apr 5, 2017).
- 4) Watson Developer Cloud
<https://www.ibm.com/watson/developercloud/retrieve-rank.html> (accessed Oct 5, 2016).
- 5) Watson Developer Cloud
<https://www.ibm.com/watson/developercloud/conversation.html> (accessed Oct 5, 2016).
- 6) Watson Developer Cloud
<https://www.ibm.com/watson/developercloud/speech-to-text.html> (accessed Oct 5, 2016).
- 7) Watson Developer Cloud
<https://www.ibm.com/watson/developercloud/visual-recognition.html> (accessed Oct 5, 2016).
- 8) 3 Types of Survey Research, When to Use Them, and How they Can Benefit Your Organization!
<https://fluidsurveys.com/university/3-types-survey-research-use-can-benefit-organization/> (accessed Oct 21, 2016).
- 9) Machine Learning Methods in Natural Language Processing
http://www.cs.columbia.edu/~mcollins/papers/tutorial_colt.pdf (accessed Oct 21, 2016).
- 10) IBM Knowledge Center
<https://www.ibm.com/support/knowledgecenter/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
<https://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 (accessed November 11, 2016).
- 28) Computer Science & Engineering
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/secdiff.html> (accessed November 7, 2016).
- 43) Pennsylvania State University Erie Behrend College
http://www.studentsreview.com/PA/PSUE_comments.html (accessed November 10, 2016).
- 44) Academic Calendars
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/ (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).
- 53) The Big Five Personality Traits
<https://www.verywell.com/the-big-five-personality-dimensions-2795422>
(accessed April 16, 2017).
- 54) Nittany Watson Challenge
<http://edtechnetwork.psu.edu/watson> (accessed April 27, 2017).

Table 4.15. User Functional Requirements: UF-A

Project Name:	Intelligent Academic Planner				
Requirement #:	UF-A	Type	Functional	Non-Functional	
Creation:	Sep 20 2016 01:25 PM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:50 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user can log in.				
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Refined Into:	SF-A-01				
Justify why UF-A can be completely covered by SF-A-01	If a user can login within 5 seconds, they can log in properly.				
Traceability:	Use cases cf.	UC-016			
	Test cases cf.	TC-001, TC-018			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

Table 4.16. User Functional Requirements: UF-B

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

Table 4.17. User Functional Requirements: UF-C

Project Name:	Intelligent Academic Planner				
Requirement #:	UF-C	Type	Functional	Non-Functional	
Creation:	Sep 20 2016 04:18 PM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Modification:	Sep 20 2016 04:58 PM	System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Description:	A user can ask the system questions.				
Priority:	<input checked="" type="checkbox"/> Highest	High	Medium	Low	Lowest
This Req. is Refined Into:	SF-C-01, SF-C-04				
Justify why UF-C can be completely covered by SF-C-01, SF-C-04	If the system is able to perform the functionality in SF-C-01 through SF-C-06 then a user will undoubtedly have been able to ask the system questions. The system would not be able to perform these functions without a user first asking it a question.				
Traceability:	Use cases cf.	UC-011, UC-012			
	Test cases cf.	TC-004, TC-007			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.18. User Functional Requirements: UF-D

Project Name:	Intelligent Academic Planner				
Requirement #:	UF-D	Type	Functional	Non-Functional	
Creation:	Sep 23 2016 12:52 PM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:51 AM	System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Description:	A user should receive multiple responses to a question.				
Priority:	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
This Req. is Refined Into:	SF-D-01				
Justify why UF-D can be completely covered by SF-D-01	By requiring a minimum of 1 response to be given, it is given that multiple responses are given to a question.				
Traceability:	Use cases cf.	UC-012			
	Test cases cf.	TC-007, TC-014			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.19. User Functional Requirements: UF-E

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

Table 4.20. User Functional Requirements: UF-F

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

Table 4.21. User Functional Requirements: UF-G

Project Name:	Intelligent Academic Planner				
Requirement #:	UF-G	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 12:36 AM	User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Modification:	Apr 05 2017 08:11 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user can view a log of asked questions.				
Priority:	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
This Req. is Refined Into:	SF-G-01				
Justify why UF-G can be completely covered by SF-G-01	If a student can send their log of asked questions to an advisor, then they can also view the questions they asked.				
Traceability:	Use cases cf.	UC-006			
	Test cases cf.	TC-022			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.22. User Functional Requirements: UF-H

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

Table 4.23. User NonFunctional Requirements: UP-03

Project Name:	Intelligent Academic Planner				
Requirement #:	UP-03	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 01:36 AM	User	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Modification:	Oct 05 2016 01:54 AM	System	<input type="checkbox"/>	<input type="checkbox"/>	
Description:	A user should receive a quick response after asking a question				
Priority:	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
This Req. is Refined Into:	SP-03-01				
Justify why UP-03 can be completely covered by SP-03-01	By specifying performance requirements, it is ensured that the question will be answered in a quick manner.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	TC-017			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.24. User NonFunctional Requirements: UP-01

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

Table 4.25. User NonFunctional Requirements: UO-01

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

Table 4.26. System Functional Requirements: SF-A-01

Intelligent Academic Planner					
Requirement #:	SF-A-01			Type	Functional
Creation:	Sep 23 2016 12:22 PM			User	<input type="checkbox"/>
Modification:	Oct 05 2016 01:41 AM			System	<input checked="" type="checkbox"/>
Description:	The system should log-in a user within 5 seconds.				
Priority:	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
This Req. is Engineered From:	UF-A				
Justify why meeting SF-A-01 can contribute to the fulfilment of UF-A	Provides performance requirement for logging in.				
Traceability:	Use cases cf.	UC-007, UC-016			
	Test cases cf.	TC-001, TC-018			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

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

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

Table 4.28. System Functional Requirements: SF-C-01

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

Table 4.29. System Functional Requirements: SF-C-04

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

Table 4.30. 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, TC-014			
Acknowledgment	Generated from the CapStone Process Management System ©2015				

Table 4.31. System Functional Requirements: SF-E-01

Project Name:	Intelligent Academic Planner				
Requirement #:	SF-E-01	Type	Functional	Non-Functional	
Creation:	Oct 04 2016 11:56 PM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 05 2016 01:46 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should allow between 100 and 600 words to describe a user's academic and professional interests.				
Priority:	<input checked="" type="checkbox"/> Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	UF-E				
Justify why meeting SF-E-01 can contribute to the fulfilment of UF-E	Allows user to enter information about themselves to their profile				
Traceability:	Use cases cf.	UC-013			
	Test cases cf.	TC-008			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.32. System Functional Requirements: SF-E-02

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

Table 4.33. System Functional Requirements: SF-E-03

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

Table 4.34. System Functional Requirements: SF-E-04

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

Table 4.35. System Functional Requirements: SF-E-05

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

Table 4.36. System Functional Requirements: SF-F-01

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

Table 4.37. System Functional Requirements: SF-G-01

Project Name:	Intelligent Academic Planner				
Requirement #:	SF-G-01	Type	Functional	Non-Functional	
Creation:	Oct 05 2016 12:37 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Apr 05 2017 08:10 AM	System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Description:	The system should allow users to send a list of asked questions to advisors				
Priority:	Highest	High	<input checked="" type="checkbox"/> Medium	Low	Lowest
This Req. is Engineered From:	UF-G				
Justify why meeting SF-G-01 can contribute to the fulfilment of UF-G	Adds security				
Traceability:	Use cases cf.	UC-006			
	Test cases cf.	TC-022			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.38. System Functional Requirements: SF-H-01

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

Table 4.39. System NonFunctional Requirements: SP-03-01

Project Name:	Intelligent Academic Planner				
Requirement #:	SP-03-01	Type	Functional	Non-Functional	
Creation:	Oct 20 2016 01:53 AM	User	<input type="checkbox"/>	<input type="checkbox"/>	
Modification:	Oct 20 2016 01:54 AM	System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Description:	The system should provide an answer to a question within 5 seconds.				Product (sub-type below)
					Performance Requirements
Priority:	Highest	High	Medium	<input checked="" type="checkbox"/> Low	Lowest
This Req. is Engineered From:	UP-03				
Justify why meeting SP-03-01 can contribute to the fulfilment of UP-03	Ensures an answer is quickly given by adding a performance requirement.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	TC-017			
Acknowledgment	<i>Generated from the CapStone Process Management System ©2015</i>				

Table 4.40. System NonFunctional Requirements: SP-01-01

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

Table 4.41. System NonFunctional Requirements: SO-01-01

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

Table 4.42. 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-04	The system should gather data unique to each user.
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 allow users to send a list of asked questions to advisors
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 4.1. Use Case Index Table

Project Name: Intelligent Academic Planner				
Use Case ID	Use Case Name	Level	Author	Version
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-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

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

Table 4.3. Use Case UC-004

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

Project Name:				
Use Case ID:	UC-005			
Use Case Name:	View assessment			
User Goal:	User is able to view assessment			
Scope:	IAP System			
Level:	Subfunction			
Relevant User Reqs:	UF-E			
Relevant System Reqs:	SF-E-03			
Primary Actor:	student, advisor			
Precondition:	The user is logged in			
Minimal Guarantee:	System does not display student's assessment			
Success Guarantee:	system display student's assessment			
Trigger:	User requests to view assessments			
Success Scenario:	<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				
Extensions:	Branching Scenarios			
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>				

Table 4.5. Use Case UC-006

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

Acknowledgment: Generated from the CapStone process management system ©2015

Table 4.6. Use Case UC-007

Project Name:	
Use Case ID:	UC-007
Use Case Name:	Train Watson
User Goal:	User be able to train Watson with new questions that user asked but hasn't been answered
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-H
Relevant System Reqs:	SF-A-01
Primary Actor:	System Developer
Precondition:	User is logged in
Minimal Guarantee:	Watson is not further trained
Success Guarantee:	Watson is further trained
Trigger:	User requests to train Watson
Success Scenario:	<p>Step Actions</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>
Extensions:	Branching Scenarios
1A	Condition: User request system export all new questions from last time export
	<p>Step Actions</p> <p>1 The system export question list that is new from last time</p>
1B	Condition: User request system export all new questions by major
	<p>Step Actions</p>
1C	Condition: User request system export all new questions other condition
	<p>Step Actions</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-010

Project Name:						
Use Case ID:	UC-010					
Use Case Name:	Answer quality feedback					
User Goal:	User is able to provide feedback on question response.					
Scope:	IAP System					
Level:	Primary task					
Relevant User Reqs:	UF-H					
Relevant System Reqs:	SF-H-01					
Primary Actor:	Visitor, Student, Advisor					
Precondition:	User is asking a question					
Minimal Guarantee:	Feedback is not stored					
Success Guarantee:	Feedback is stored					
Trigger:	User requests to submit feedback					
Success Scenario:	<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						
Extensions:	Branching Scenarios					
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 4.8. Use Case UC-011

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

Acknowledgment: Generated from the CapStone process management system ©2015

Table 4.9. Use Case UC-012

Project Name:	
Use Case ID:	UC-012
Use Case Name:	Analyze Question
User Goal:	System determines answer for user
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-C,UF-D
Relevant System Reqs:	SF-C-01,SF-C-04,SF-D-01
Primary Actor:	N/A
Precondition:	User asks a question
Minimal Guarantee:	Question is not analyzed
Success Guarantee:	Question is analyzed
Trigger:	User submits a question
Success Scenario:	<p>Step Actions</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>
Extensions:	Branching Scenarios
1A	<p>Condition: If another question could be asked</p> <p>Step Actions</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 4.10. Use Case UC-013

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

Table 4.11. Use Case UC-015

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

Table 4.12. Use Case UC-016

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

Table 4.13. Use Case UC-018

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

Acknowledgment: Generated from the CapStone process management system ©2015

Table 4.14. Use Case UC-019

Project Name:	
Use Case ID:	UC-019
Use Case Name:	Request Assessment
User Goal:	System notify student that an adviser wants to see his assessment
Scope:	IAP System
Level:	Primary task
Relevant User Reqs:	UF-E
Relevant System Reqs:	SF-E-05
Primary Actor:	Advisor
Precondition:	User is logged in
Minimal Guarantee:	No request is sent
Success Guarantee:	Request is sent
Trigger:	User requests to send request to student
Success Scenario:	<p>Step Actions</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>
Extensions:	Branching Scenarios
4A	Condition: If invalid student information
	<p>Step Actions</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 8.2.1. Test Suite TS-001: Unit Tests

TS-001: Unit Tests		
Test Case ID	Test Stage	Test Case Description
TC-002	Unit	logout
TC-003	Acceptance	Register with untaken email and valid password
TC-004	Acceptance	Ask question
TC-005	Acceptance	answer quality feedback
TC-006	Unit	Connection to localhost
TC-007	Acceptance	Analyze question
TC-012	Unit	Connection to user database
TC-013	Unit	connection to question database
TC-025	Unit	Should access page / with no problems
TC-026	Unit	Should redirect to profile page if authenticated
TC-027	Unit	Should stay on login page if not authenticated
TC-028	Unit	Should not login user if there is no email provided or email is empty
TC-029	Unit	Should not login user if there is no password or password is empty
TC-030	Unit	Should not log in user if it is not a valid email
TC-031	Unit	Should not log in user if email doesn't exist
TC-032	Unit	Should not log in user if password is incorrect
TC-033	Unit	Should properly log in user if no problems
TC-034	Unit	User cannot register with empty first name
TC-035	Unit	user cannot register with empty last name
TC-036	Unit	user cannot register with empty email
TC-037	Unit	user cannot register with empty password
TC-038	Unit	user cannot register with empty account role
TC-039	Unit	user cannot register as an admin with no admin token
TC-040	Unit	user cannot register as an advisor with no advisor token
TC-041	Unit	User cannot register if invalid email
TC-042	Unit	a user cannot register with a taken email
TC-043	Unit	user should be saved as student in database if registered as student
TC-044	Unit	user should be saved as advisor in database if registered as advisor
TC-045	Unit	user should be saved as admin in database if registered as admin
TC-046	Unit	Should store user in database properly
TC-047	Unit	user should be redirected to login page if not logged in and accessing profile
TC-048	Unit	/profile should successfully navigate to profile if logged in

TC-049	Unit	profile should return error for invalid type
TC-050	Unit	it should connect to facebook for auth
TC-051	Unit	it should connect to twitter for auth
TC-052	Unit	should connect to google for auth
TC-053	Unit	should connect to linkedin for auth
TC-054	Unit	user should be redirected to login page if not logged in and accessing /inbox
TC-055	Unit	should access page /inbox with no problems if user is logged in
TC-056	Unit	user should be redirected to login page if not logged in and accessing /advising
TC-057	Unit	should not access /advising page if user is not an advisor
TC-058	Unit	should access page /advising with no problems if user is logged in as an advisor
TC-059	Unit	should logout with no problems
TC-060	Unit	user should be redirected to login page if not logged in and accessing /admin
TC-061	Unit	should not access page /admin if user is not an admin
TC-062	Unit	should access page /admin with no problems if user is logged in as an admin
TC-063	Unit	user should be redirected to login page if not logged in and accessing /QuestionAnswerManagement
TC-064	Unit	should not access /QuestionAnswerManagement page if user is not an admin
TC-065	Unit	Should access page /QuestionAnswerManagement with no problems if user is logged in as admin
TC-066	Unit	should redirect user to login page if not logged in as admin and accessing /postQuestionAnswer
TC-067	Unit	should return error if question is empty
TC-068	Unit	should find question in database
TC-069	Unit	user should be redirected to login page if not logged in and accessing /SystemManagement
TC-070	Unit	Should not access SystemManagement page if user is not an admin
TC-071	Unit	Should access page /SystemManagement with no problems if user is logged in as an admin

Table 8.2.2. Test Suite TS-002: User Acceptance Tests

TS-002: User Acceptance Tests		
Test Case ID	Test Stage	Test Case Description
TC-001	Acceptance	Login with valid user and password
TC-014	Acceptance	The system should provide multiple responses to an answer
TC-015	Acceptance	The system should store data unique to the user
TC-017	Acceptance	Should answer within 5 seconds
TC-018	Acceptance	A user can login within 5 seconds
TC-019	Acceptance	A user should register within 5 seconds
TC-022	Acceptance	View Question Log

Table 8.2.3. Test Suite TS-003: Manual Tests

TS-003: Manual Tests		
Test Case ID	Test Stage	Test Case Description
TC-008	Acceptance	submit profile
TC-009	Acceptance	generate assessment
TC-011	Acceptance	view assessment
TC-024	Acceptance	Log out after 1 hour of inactivity

Table 8.2.4. Test Case TC-002

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

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.5. Test Case TC-006

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-006 (Unit Test)	
What To Test	Connection to localhost	
Test Data Input	localhost URL	
Expected Result	status code of 200 and no errors	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.6. Test Case TC-012

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-012 (Unit Test)	
What To Test	Connection to user database	
Test Data Input	user database URL	
Expected Result	a valid database variable that is not null	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.7. Test Case TC-013

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-013 (Unit Test)	
What To Test	connection to question database	
Test Data Input	question database url	
Expected Result	database variable that is not null	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.8. Test Case TC-025

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-025 (Unit Test)	
What To Test	Should access page / with no problems	
Test Data Input		
Expected Result	status code of 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.9. Test Case TC-026

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-026 (Unit Test)	
What To Test	Should redirect to profile page if authenticated	
Test Data Input	login information	
Expected Result	browser redirects, status code of 302, header location to be /profile	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.10. Test Case TC-027

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-027 (Unit Test)	
What To Test	Should stay on login page if not authenticated	
Test Data Input		
Expected Result	browser does not redirect, status code of 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.11. Test Case TC-028

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-028 (Unit Test)	
What To Test	Should not login user if there is no email provided or email is empty	
Test Data Input	login information with an empty email	
Expected Result	status code of 200, response text includes an error that states the email can't be empty	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.12. Test Case TC-029

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-029 (Unit Test)	
What To Test	Should not login user if there is no password or password is empty	
Test Data Input	login information with empty password	
Expected Result	status code of 200, response text is an error that states the password can't be empty	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.13. Test Case TC-030

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-030 (Unit Test)	
What To Test	Should not log in user if it is not a valid email	
Test Data Input	login information with an invalid email	
Expected Result	status code of 200, response text includes an error that states invalid email	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.14. Test Case TC-031

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-031 (Unit Test)	
What To Test	Should not log in user if email doesn't exist	
Test Data Input	login information with a email that doesn't exist	
Expected Result	status code of 200, response text is an error that includes can't find any user with that email	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.15. Test Case TC-032

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-032 (Unit Test)	
What To Test	Should not log in user if password is incorrect	
Test Data Input	login information with incorrect password	
Expected Result	status code of 200, response text that is an error with a password incorrect message	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.16. Test Case TC-033

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-033 (Unit Test)	
What To Test	Should properly log in user if no problems	
Test Data Input	login information	
Expected Result	response code of 200, response text stating login success	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.17. Test Case TC-034

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-034 (Unit Test)	
What To Test	User cannot register with empty first name	
Test Data Input	register information with no first name	
Expected Result	status code of 200, response text that is an error which includes name can't be empty	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.18. Test Case TC-035

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-035 (Unit Test)	
What To Test	user cannot register with empty last name	
Test Data Input	register information with empty last name	
Expected Result	status code of 200, response text that is an error with message name can't be empty	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.19. Test Case TC-036

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-036 (Unit Test)	
What To Test	user cannot register with empty email	
Test Data Input	register information with empty email	
Expected Result	status code of 200, response text that is an error with a message email can't be empty	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.20. Test Case TC-037

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-037 (Unit Test)	
What To Test	user cannot register with empty password	
Test Data Input	register information with empty password	
Expected Result	status code of 200 with response text that is an error stating that the password can't be empty	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.21. Test Case TC-038

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-038 (Unit Test)	
What To Test	user cannot register with empty account role	
Test Data Input	register information with empty account role	
Expected Result	status code of 200, response text that is an error with message invalid account role	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.22. Test Case TC-039

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-039 (Unit Test)	
What To Test	user cannot register as an admin with no admin token	
Test Data Input	register information, setting account role to admin, without admin token	
Expected Result	status code of 200, response text that is an error with message missing admin token	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.23. Test Case TC-040

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-040 (Unit Test)	
What To Test	user cannot register as an advisor with no advisor token	
Test Data Input	register information, setting account role to advisor, with no advisor token	
Expected Result	status code of 200, response text that is an error with message invalid advisor token	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.24. Test Case TC-041

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-041 (Unit Test)	
What To Test	User cannot register if invalid email	
Test Data Input	register information with invalid email	
Expected Result	status code of 200, response text that is an error with message email not valid	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.25. Test Case TC-042

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-042 (Unit Test)	
What To Test	a user cannot register with a taken email	
Test Data Input	register information with taken email	
Expected Result	status code of 200, response text that is an error with message email already registered	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.26. Test Case TC-043

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-043 (Unit Test)	
What To Test	user should be saved as student in database if registered as student	
Test Data Input	register information, account role set to student	
Expected Result	user in database has role set to student	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.27. Test Case TC-044

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-044 (Unit Test)	
What To Test	user should be saved as advisor in database if registered as advisor	
Test Data Input	register information, account role set to advisor, valid token inputted	
Expected Result	user in database has role set to advisor	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.28. Test Case TC-045

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-045 (Unit Test)	
What To Test	user should be saved as admin in database if registered as admin	
Test Data Input	register information, account role set to admin, valid admin token	
Expected Result	user in database has admin as role	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.29. Test Case TC-046

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-046 (Unit Test)	
What To Test	Should store user in database properly	
Test Data Input	register information	
Expected Result	User's type, email, password, first name, last name, display name, and role are set to what was entered. status code of 200. response text that includes a message stating successfully registered	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.30. Test Case TC-047

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-047 (Unit Test)	
What To Test	user should be redirected to login page if not logged in and accessing profile	
Test Data Input		
Expected Result	browser redirects, status code of 302, header location of /login	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.31. Test Case TC-048

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-048 (Unit Test)	
What To Test	/profile should successfully navigate to profile if logged in	
Test Data Input	login information	
Expected Result	status code of 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.32. Test Case TC-049

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-049 (Unit Test)	
What To Test	profile should return error for invalid type	
Test Data Input	change user database to have a type that is not valid	
Expected Result	status code of 200, response text of request user type is unexcepted	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.33. Test Case TC-050

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-050 (Unit Test)	
What To Test	it should connect to facebook for auth	
Test Data Input		
Expected Result	status code of 302, header location that includes facebook	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.34. Test Case TC-051

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-051 (Unit Test)	
What To Test	it should connect to twitter for auth	
Test Data Input		
Expected Result	status code of 302, header location of twitter	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.35. Test Case TC-052

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-052 (Unit Test)	
What To Test	should connect to google for auth	
Test Data Input		
Expected Result	status code of 302, header location includes googles	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.36. Test Case TC-053

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-053 (Unit Test)	
What To Test	should connect to linkedin for auth	
Test Data Input		
Expected Result	status code of 302, header location includes linkedin	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.37. Test Case TC-054

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-054 (Unit Test)	
What To Test	user should be redirected to login page if not logged in and accessing /inbox	
Test Data Input		
Expected Result	browser redirects, status code of 302, header location of /login	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.38. Test Case TC-055

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-055 (Unit Test)	
What To Test	should access page /inbox with no problems if user is logged in	
Test Data Input		
Expected Result	response code of 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.39. Test Case TC-056

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-056 (Unit Test)	
What To Test	user should be redirected to login page if not logged in and accessing /advising	
Test Data Input		
Expected Result	browser redirects, status code of 302, header location of /login	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.40. Test Case TC-057

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-057 (Unit Test)	
What To Test	should not access /advising page if user is not and advisor	
Test Data Input		
Expected Result	status code is not equal to 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.41. Test Case TC-058

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-058 (Unit Test)	
What To Test	should access page /advising with no problems if user is logged in as an advisor	
Test Data Input	logged in as advisor	
Expected Result	status code of 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.42. Test Case TC-059

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-059 (Unit Test)	
What To Test	should logout with no problems	
Test Data Input		
Expected Result	status code of 302, header location is /	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.43. Test Case TC-060

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-060 (Unit Test)	
What To Test	user should be redirected to login page if not logged in and accessing /admin	
Test Data Input		
Expected Result	browser redirects, status code of 302, header location of /login	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.44. Test Case TC-061

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-061 (Unit Test)	
What To Test	should not access page /admin if user is not an admin	
Test Data Input		
Expected Result	status code not equal to 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.45. Test Case TC-062

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-062 (Unit Test)	
What To Test	should access page /admin with no problems if user is logged in as an admin	
Test Data Input	logged in as an admin	
Expected Result	status code of 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.46. Test Case TC-063

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-063 (Unit Test)	
What To Test	user should be redirected to login page if not logged in and accessing /QuestionAnswerManagement	
Test Data Input		
Expected Result	browser redirects, status code 302, header location of /login	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.47. Test Case TC-064

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-064 (Unit Test)	
What To Test	should not access /QuestionAnswerManagement page if user is not an admin	
Test Data Input	logged in as a non-admin (advisor or student) or not logged in	
Expected Result	status code not equal to 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.48. Test Case TC-065

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-065 (Unit Test)	
What To Test	Should access page /QuestionAnswerManagement with no problems if user is logged in as admin	
Test Data Input	logged in as admin	
Expected Result	status code of 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.49. Test Case TC-066

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-066 (Unit Test)	
What To Test	should redirect user to login page if not logged in as admin and accessing /postQuestionAnswer	
Test Data Input		
Expected Result	browser redirects, status code of 302, header location of /login	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.50. Test Case TC-067

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-067 (Unit Test)	
What To Test	should return error if question is empty	
Test Data Input	post an empty question	
Expected Result	status code of 200, response text includes question cannot be empty	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.51. Test Case TC-068

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-068 (Unit Test)	
What To Test	should find question in database	
Test Data Input	logged in as admin	
Expected Result	expect question asked to be in database and response text includes success	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.52. Test Case TC-069

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-069 (Unit Test)	
What To Test	user should be redirected to login page if not logged in and accessing /SystemManagement	
Test Data Input		
Expected Result	browser redirects, status code of 302, header location of /login	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.53. Test Case TC-070

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-070 (Unit Test)	
What To Test	Should not access SystemManagement page if user is not an admin	
Test Data Input	logged in user that is not an admin	
Expected Result	response code that is not 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.54. Test Case TC-071

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-071 (Unit Test)	
What To Test	Should access page /SystemManagement with no problems if user is logged in as an admin	
Test Data Input	logged in as an admin	
Expected Result	browser does not redirect, status code of 200	
Traceability	Relevant User Req.(s)	
	Relevant System Req.(s)	
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.55. Test Case TC-001

Project Name:	Intelligent Academic Planner	
Test Suite	TS-002: User Acceptance Tests	
Test Case ID	TC-001 (Acceptance Test)	
What To Test	Login with valid user and password	
Test Data Input	valid username, valid password	
Expected Result	User is logged in	
Traceability	Relevant User Req.(s)	UF-A
	Relevant System Req.(s)	SF-A-01
	Relevant Use Case(s)	UC-016
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.56. Test Case TC-003

Project Name:	Intelligent Academic Planner	
Test Suite	TS-001: Unit Tests	
Test Case ID	TC-003 (Acceptance Test)	
What To Test	Register with untaken email and valid password	
Test Data Input	untaken email, valid password	
Expected Result	User is entered into database.	
Traceability	Relevant User Req.(s)	UF-F
	Relevant System Req.(s)	SF-F-01
	Relevant Use Case(s)	UC-004

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.57. Test Case TC-004

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

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.58. Test Case TC-005

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

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.59. Test Case TC-007

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

Table 8.2.60. Test Case TC-008

Project Name:	Intelligent Academic Planner	
Test Suite	TS-003: Manual Tests	
Test Case ID	TC-008 (Acceptance Test)	
What To Test	submit profile	
Test Data Input	new profile information	
Expected Result	profile information is stored to database	
Traceability	Relevant User Req.(s)	UF-E
	Relevant System Req.(s)	SF-E-01,SF-E-02
	Relevant Use Case(s)	UC-013

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.61. Test Case TC-009

Project Name:	Intelligent Academic Planner	
Test Suite	TS-003: Manual Tests	
Test Case ID	TC-009 (Acceptance Test)	
What To Test	generate assessment	
Test Data Input	descriptive text, yes to saving assessment, yes to visible to advisors	
Expected Result	assessment is stored to database with true visibility	
Traceability	Relevant User Req.(s)	UF-E, UP-01
	Relevant System Req.(s)	SF-E-04, SF-E-05, SP-01-01
	Relevant Use Case(s)	UC-015

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.62. Test Case TC-011

Project Name:	Intelligent Academic Planner	
Test Suite	TS-003: Manual Tests	
Test Case ID	TC-011 (Acceptance Test)	
What To Test	view assessment	
Test Data Input		
Expected Result	assessment is displayed	
Traceability	Relevant User Req.(s)	UF-E
	Relevant System Req.(s)	SF-E-03
	Relevant Use Case(s)	UC-005

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.63. Test Case TC-014

Project Name:	Intelligent Academic Planner	
Test Suite	TS-002: User Acceptance Tests	
Test Case ID	TC-014 (Acceptance Test)	
What To Test	The system should provide multiple responses to an answer	
Test Data Input	question	
Expected Result	more than 1 answer	
Traceability	Relevant User Req.(s)	UF-D
	Relevant System Req.(s)	SF-D-01
	Relevant Use Case(s)	
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.64. Test Case TC-015

Project Name:	Intelligent Academic Planner	
Test Suite	TS-002: User Acceptance Tests	
Test Case ID	TC-015 (Acceptance Test)	
What To Test	The system should store data unique to the user	
Test Data Input	question	
Expected Result	User in database will have unique information stored	
Traceability	Relevant User Req.(s)	UF-E
	Relevant System Req.(s)	SF-E-04
	Relevant Use Case(s)	UC-013

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.65. Test Case TC-017

Project Name:	Intelligent Academic Planner	
Test Suite	TS-002: User Acceptance Tests	
Test Case ID	TC-017 (Acceptance Test)	
What To Test	Should answer within 5 seconds	
Test Data Input	question	
Expected Result	answer within 5 seconds	
Traceability	Relevant User Req.(s)	UP-03
	Relevant System Req.(s)	SP-03-01
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.66. Test Case TC-018

Project Name:	Intelligent Academic Planner	
Test Suite	TS-002: User Acceptance Tests	
Test Case ID	TC-018 (Acceptance Test)	
What To Test	A user can login within 5 seconds	
Test Data Input	login information	
Expected Result	user logs in within 5 seconds	
Traceability	Relevant User Req.(s)	UF-A
	Relevant System Req.(s)	SF-A-01
	Relevant Use Case(s)	UC-016

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.67. Test Case TC-019

Project Name:	Intelligent Academic Planner	
Test Suite	TS-002: User Acceptance Tests	
Test Case ID	TC-019 (Acceptance Test)	
What To Test	A user should register within 5 seconds	
Test Data Input	register information	
Expected Result	user is registered within 5 seconds	
Traceability	Relevant User Req.(s)	UF-F
	Relevant System Req.(s)	SF-F-01
	Relevant Use Case(s)	UC-004
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>		

Table 8.2.68. Test Case TC-022

Project Name:	Intelligent Academic Planner	
Test Suite	TS-002: User Acceptance Tests	
Test Case ID	TC-022 (Acceptance Test)	
What To Test	View Question Log	
Test Data Input		
Expected Result	Question Log is displayed	
Traceability	Relevant User Req.(s)	UF-G
	Relevant System Req.(s)	SF-G-01
	Relevant Use Case(s)	UC-006

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.2.69. Test Case TC-024

Project Name:	Intelligent Academic Planner	
Test Suite	TS-003: Manual Tests	
Test Case ID	TC-024 (Acceptance Test)	
What To Test	Log out after 1 hour of inactivity	
Test Data Input		
Expected Result	User is logged out after 1 hour of inactivity	
Traceability	Relevant User Req.(s)	UO-01
	Relevant System Req.(s)	SO-01-01
	Relevant Use Case(s)	

Acknowledgment: Generated from the CapStone process management system ©2015

Table 8.3.1. Execution Report of Test Case TC-002

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-002					
Testing Tools Used:	Mocha					
Testing Type:	Functional testing					
Execution Steps:	1 click left sidebar 2 click user icon 3 click logout button 4 back to main page 5 Run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User is not logged out	Fail	Not implemented	08/10/2016 by Xiaoyu Zhou
2	Daria	10/15/2016	User is not logged out	Fail	Not yet implemented	10/30/2016 by Allen
3	Daria	11/15/2016	User is logged out	Pass		
4	Daria Cook	01/01/2017	User is logged out	Pass		
5	Daria	2/1/2017	User is logged out	Pass		
6	Daria Cook	04/05/2017	User is logged out	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.2. Execution Report of Test Case TC-006

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-006					
Testing Tools Used:	Mocha					
Testing Type:	Agile (automated) testing					
Execution Steps:	1 Run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	Connection to localhost fails	Fail	Not implemented	08/05/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	Can connect to localhost	Pass		
3	Daria Cook	04/05/2017	Can connect to localhost	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.3. Execution Report of Test Case TC-012

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-012					
Testing Tools Used:	Mocha					
Testing Type:	Agile (automated) testing					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	Cannot connect to user database	Fail	Not Implemented	
2	Daria Cook	01/01/2017	Can connect to user database	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.4. Execution Report of Test Case TC-013

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-013					
Testing Tools Used:	Mocha					
Testing Type:	Agile (automated) testing					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	Cannot connect to question database	Fail	Not Implemented	08/05/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	Can connect to question database	Pass		
3	Daria Cook	04/05/2017	Can connect to question database	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.5. Execution Report of Test Case TC-025

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-025					
Testing Tools Used:	Mocha					
Testing Type:	Function coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	There is no / page	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	Accesses / page with no problems	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.6. Execution Report of Test Case TC-026

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-026					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot log in	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is redirected to profile page after logging in	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.7. Execution Report of Test Case TC-027

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-027					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot login	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User stays on login page if login fails	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.8. Execution Report of Test Case TC-028

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-028					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	Cannot Login	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is not logged in if email is empty	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.9. Execution Report of Test Case TC-029

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-029					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot login	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User does not get logged in when password is empty	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.10. Execution Report of Test Case TC-030

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-030					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot login	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is not logged in if email is not valid	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.11. Execution Report of Test Case TC-031

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-031					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot login	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is not logged in if email doesn't exist	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.12. Execution Report of Test Case TC-032

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-032					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot login	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User does not log in if password is incorrect	Pass		
3	Daria Cook	04/05/2017	User does not log in if password is incorrect	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.13. Execution Report of Test Case TC-033

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-033					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot log in	Fail	Not Implemented	08/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User logs in with no problems if valid credentials	Pass		
3	Daria Cok	04/05/2017	User logs in with no problems if valid credentials	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.14. Execution Report of Test Case TC-034

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-034					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User cannot register with empty first name	Pass		
3	Daria Cook	04/05/2017	User cannot register with empty first name	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.15. Execution Report of Test Case TC-035

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-035					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User cannot register with empty last name	Pass		
3	Daria Cook	04/05/2017	User cannot register with empty last name	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.16. Execution Report of Test Case TC-036

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-036					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User does not register with empty email	Pass		
3	Daria Cook	04/05/2017	User does not register with empty email	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.17. Execution Report of Test Case TC-037

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-037					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User does not register with empty password	Pass		
3	Daria Cook	04/05/2017	User does not register with empty password	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.18. Execution Report of Test Case TC-038

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-038					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User does not register with empty account role	Pass		
3	Daria Cook	04/05/2017	User does not register with empty account role	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.19. Execution Report of Test Case TC-039

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-039					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User does not register with no admin token	Pass		
3	Daria Cook	04/05/2017	User does not register with no admin token	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.20. Execution Report of Test Case TC-040

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-040					
Testing Tools Used:	Mocha					
Testing Type:	Agile (automated) testing					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User does not register with no advisor token	Pass		
3	Daria Cook	04/05/2017	User does not register with no advisor token	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.21. Execution Report of Test Case TC-041

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-041					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User does not register if invalid email	Pass		
3	Daria Cook	04/05/2017	User does not register if invalid email	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.22. Execution Report of Test Case TC-042

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-042					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User does not register with taken email	Pass		
3	Daria Cook	04/05/2017	User does not register with taken email			
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.23. Execution Report of Test Case TC-043

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-043					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User is saved as student in database	Pass		
3	Daria Cook	04/05/2017	User is saved as student in database			
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.24. Execution Report of Test Case TC-044

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-044					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User is saved as advisor in database	Pass		
3	Daria Cook	04/05/2017	User is saved as advisor in database			
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.25. Execution Report of Test Case TC-045

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-045					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User is saved as admin in database	Pass		
3	Daria Cook	04/05/2017	User is saved as admin in database	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.26. Execution Report of Test Case TC-046

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-046					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User is stored in database properly	Pass		
3	Daria Cook	04/05/2017	User is stored in database properly	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.27. Execution Report of Test Case TC-047

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-047					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No Profile Page to access	Fail	Not Implemented	02/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is redirected to login page when accessing /profile not logged in	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.28. Execution Report of Test Case TC-048

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-048					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No profile page	Fail	Not Implemented	02/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User can navigate to /profile when logged in	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.29. Execution Report of Test Case TC-049

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-049					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No profile page	Fail	Not Implemented	02/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	Profile page returns error on invalid type	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.30. Execution Report of Test Case TC-050

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-050					
Testing Tools Used:	Mocha					
Testing Type:	Function coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No facebook login	Fail	Not Implemented	09/20/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	System connects to facebook for login	Pass		
3	Daria Cook	04/05/2017	System connects to facebook for login			
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.31. Execution Report of Test Case TC-051

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-051					
Testing Tools Used:	Mocha					
Testing Type:	Function coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No Twitter login	Fail	Not Implemented	09/20/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	System connects to twitter to login	Pass		
3	Daria Cook	04/05/2017	System connects to twitter to login	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.32. Execution Report of Test Case TC-052

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-052					
Testing Tools Used:	Mocha					
Testing Type:	Function coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	Cannot login using Google	Fail	Not Implemented	09/20/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	System connects to Google for login	Pass		
3	Daria Cook	04/05/2017	System connects to Google for login	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.33. Execution Report of Test Case TC-053

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-053					
Testing Tools Used:	Mocha					
Testing Type:	Function coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	Cannot login to LinkedIn	Fail	Not Implemented	09/20/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	System connects to LinkedIn for login	Pass		
3	Daria Cook	04/05/2017	System connects to LinkedIn for login	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.34. Execution Report of Test Case TC-054

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-054					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No inbox page	Fail	Not Implemented	03/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is redirected to login page when not logged in and accessing /inbox	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.35. Execution Report of Test Case TC-055

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-055					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No inbox page	Fail	Not Implemented	03/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	Can access /inbox when logged in	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.36. Execution Report of Test Case TC-056

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-056					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No advising page	Fail	Not Implemented	03/20/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is redirected to /login if accessing /advising and not logged in	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.37. Execution Report of Test Case TC-057

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-057					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No advising page	Fail	Not Implemented	03/20/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	Cannot access /advising unless an advisor	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.38. Execution Report of Test Case TC-058

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-058					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No advising page	Fail	Not Implemented	03/20/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	Can access /advising with no problems as admin	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.39. Execution Report of Test Case TC-059

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-059					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot logout	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User logs out with no problems	Pass		
3	Daria Cook	04/05/2017	User logs out with no problems	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.40. Execution Report of Test Case TC-060

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-060					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No admin page	Fail	Not Implemented	03/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is redirected to login page when not logged in and accessing /admin	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.41. Execution Report of Test Case TC-061

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-061					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No admin page	Fail	Not Implemented	03/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User does not access /admin when not admin	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.42. Execution Report of Test Case TC-062

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-062					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No admin page	Fail	Not Implemented	03/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User can access /admin when logged in as admin	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.43. Execution Report of Test Case TC-063

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-063					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No question answer management page	Fail	Not Implemented	03/20/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is redirected to login page when not logged in and accessing /QAM	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.44. Execution Report of Test Case TC-064

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-064					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No question answer management page	Fail	Not Implemented	03/20/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User does not access /QAM when not an admin	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.45. Execution Report of Test Case TC-065

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-065					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No question answer management page	Fail	Not Implemented	03/20/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User can access /QAM as admin	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.46. Execution Report of Test Case TC-066

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-066					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No post question answer page	Fail	Not Implemented	03/20/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is redirected to login page if not an admin and accessing /postQA	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.47. Execution Report of Test Case TC-067

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-067					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No post question answer page	Fail	Not Implemented	03/20/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	Returns error if sent an empty question	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.48. Execution Report of Test Case TC-068

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-068					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No post question answer page	Fail	Not Implemented	03/20/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	After sending a question, the question can be found in the database	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.49. Execution Report of Test Case TC-069

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-069					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No system management page	Fail	Not Implemented	03/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is redirected to login page if not logged in and accessing /systemmanagement	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.50. Execution Report of Test Case TC-070

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-070					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No system management page	Fail	Not Implemented	03/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User cannot access /SystemManagement if not an admin	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.51. Execution Report of Test Case TC-071

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-071					
Testing Tools Used:	Mocha					
Testing Type:	Condition coverage					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	No system management page	Fail	Not Implemented	03/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User can access /SystemManagement as an admin	pass		
Execution Summary: test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.52. Execution Report of Test Case TC-001

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-001					
Testing Tools Used:	Mocha					
Testing Type:	Functional testing					
Execution Steps:	1 click left side bar 2 click user icon 3 click perfered login method 4 input valid credential 5 Run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User does not get logged in	Fail	Not implemented	08/10/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User is logged in	Pass		
3	Daria Cook	04/05/2017	User is logged in	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.53. Execution Report of Test Case TC-003

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-003					
Testing Tools Used:	Mocha					
Testing Type:	Functional testing					
Execution Steps:	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 7 Run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User is not registered	Fail	Not implemented	08/10/2016 by Xiaoyu Zhou
2	Daria	10/15/2016	User is not added to database	Fail	Not yet implemented	10/30/2016 by Allen
3	Daria	11/15/2016	User is added to database	Pass		
4	Daria Cook	01/01/2017	User is registered	Pass		
5	Daria	2/1/2017	User is added to database	Pass		
6	Daria Cook	04/05/2017	User is registered	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.54. Execution Report of Test Case TC-004

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-004					
Testing Tools Used:	Mocha					
Testing Type:	Functional testing					
Execution Steps:	1 input question 2 view answer 3 Run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot ask a question	Fail	Not Implemented	10/1/2016 by Xiaoyu Zhou
2	Daria	11/15/2016	Question is not answered	Fail	Not yet implemented	
3	Daria Cook	01/01/2017	User can ask a question	Pass		
4	Daria Cook	04/05/2017	User can ask a question	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.55. Execution Report of Test Case TC-005

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-005					
Testing Tools Used:	Mocha					
Testing Type:	Functional testing					
Execution Steps:	1 User asks question 2 System displays response and requests feedback 3 User enters feedback 4 See that feedback is stored in database 5 Run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot provide feedback	Fail	Not Implemented	
2	Daria	11/15/2016	Feedback is not added to database and thank you message not displayed	Fail	Not yet implemented	
3	Daria Cook	01/01/2017	User cannot provide feedback	Fail	Not implemented	
4	Daria Cook	04/05/2017	User can provide feedback, but it is not stored in database	Fail	Frontend is implemented, backend is not storing to database	
Execution Summary: Test is failing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.56. Execution Report of Test Case TC-007

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-007					
Testing Tools Used:	Mocha					
Testing Type:	Functional testing					
Execution Steps:	1 User enters ambiguous question 2 See that system responds with another question relevant to the question asked 3 Run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	System doesn't analyze a question	Fail	Not implemented	10/20/2016 by Xiaoyu Zhou
2	Daria	11/15/2016	No output	Fail	Not yet implemented	
3	Daria Cook	01/01/2017	System analyzes a question	Pass		
4	Daria Cook	04/05/2017	System analyzes a question	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.57. Execution Report of Test Case TC-008

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-008					
Testing Tools Used:	Manual					
Testing Type:	Operational readiness testing					
Execution Steps:	1 User enters profile information 2 User submits profile information 3 See that profile information is stored in database 4 Create an account or login to existing account 5 You should appear on Profile page.					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	Cannot access profile page	Fail	Not implemented	03/01/2017 by Xiaoyu Zhou
2	Daria	11/15/2016	Profile information not stored to database	Fail	Not yet implemented	
3	Daria Cook	04/03/2017	User can access profile page	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.58. Execution Report of Test Case TC-009

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-009					
Testing Tools Used:	Manual					
Testing Type:	Operational readiness testing					
Execution Steps:	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 5 Register a new account or login to existing account 6 Click Generate Assessment on Profile Page 7 Select what you want in your assessment 8 Select an advisor to send it to 9 See that assessment is generated and it is sent to advisor specified					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	Assessment cannot be generated	Fail	Not Implemented	
2	Daria	11/15/2016	No assessment shown	Fail	Not yet implemented	
3	Daria Cook	01/01/2017	Assessment cannot be generated	Fail	Not Implemented	
4	Daria Cook	04/05/2017	You can ask to generate an assessment but it is not displayed	Fail	Frontend implemented, Backend not	04/20/2017 by Xiaoyu Zhou
5	Daria Cook	04/26/2017	Assessment is sent to advisor	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.59. Execution Report of Test Case TC-011

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-011					
Testing Tools Used:	Manual					
Testing Type:	Operational readiness testing					
Execution Steps:	1 User requests to view assessment 2 See that assessment is displayed					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	Cannot view assessment	Fail	Not Implemented	
2	Daria	11/15/2016	Assessment not shown	Fail	Not yet implemented	
3	Daria Cook	01/01/2017	Cannot view assessment	Fail	Not Implemented	
4	Daria Cook	04/05/2017	Cannot view assessment	Fail	Not Implemented	04/20/2017 by Xiaoyu Zhou
5	Daria Cook	04/26/2017	Can view an assessment	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.60. Execution Report of Test Case TC-014

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-014					
Testing Tools Used:	Mocha					
Testing Type:	Non-Functional testing					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	System does not provide response	Fail	Not Implemented	10/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	System provides only one response	Fail	Implemented incorrectly.	02/01/2017 by Xiaoyu Zhou
3	Daria Cook	04/05/2017	System provides more than one response	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.61. Execution Report of Test Case TC-015

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-015					
Testing Tools Used:	Mocha					
Testing Type:	Non-Functional testing					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	System does not store data	Fail	Not Implemented	10/01/2017 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	System does not store data unique to the user	Fail	Unique data not stored	01/20/2017 by Xiaoyu Zhou
3	Daria Cook	04/05/2017	System stores data unique to user	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.62. Execution Report of Test Case TC-017

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-017					
Testing Tools Used:	Mocha					
Testing Type:	Non-Functional testing					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	System does not answer	Fail	Not Implemented	10/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	System answers within 5 seconds	Pass		
3	Daria Cook	04/05/2017	System answers within 5 seconds	Pass		
Execution Summary: Tests are passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.63. Execution Report of Test Case TC-018

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-018					
Testing Tools Used:	Mocha					
Testing Type:	Non-Functional testing					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User can't login	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User logs in within 5 seconds	Pass		
3	Daria Cook	04/05/2017	User logs in within 5 seconds	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.64. Execution Report of Test Case TC-019

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-019					
Testing Tools Used:	Mocha					
Testing Type:	Non-Functional testing					
Execution Steps:	1 run npm test					
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User can't register	Fail	Not Implemented	09/01/2016 by Xiaoyu Zhou
2	Daria Cook	01/01/2017	User registers within 5 seconds	Pass		
3	Daria Cook	04/05/2017	User registers within 5 seconds	Pass		
Execution Summary: Tests are passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						

Table 8.3.65. Execution Report of Test Case TC-022

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-022					
Testing Tools Used:	Mocha					
Testing Type:	Functional testing					
Execution Steps:						
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User cannot view questions asked	Fail	Not Implemented	02/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User can view questions asked	Pass		
Execution Summary: Test is passing						
Acknowledgment: Generated from the CapStone process management system ©2015						

Table 8.3.66. Execution Report of Test Case TC-024

Project Name:	Intelligent Academic Planner					
Test Case ID:	TC-024					
Testing Tools Used:	Manual					
Testing Type:	Operational readiness testing					
Execution Steps:						
Test Execution Records:						
#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Daria Cook	08/01/2016	User is never logged out	Fail	Not Implemented	02/01/2017 by Xiaoyu Zhou
2	Daria Cook	04/05/2017	User is logged out after 1 hour	Pass		
Execution Summary: Test is passing						
<i>Acknowledgment: Generated from the CapStone process management system ©2015</i>						