Table of Contents

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

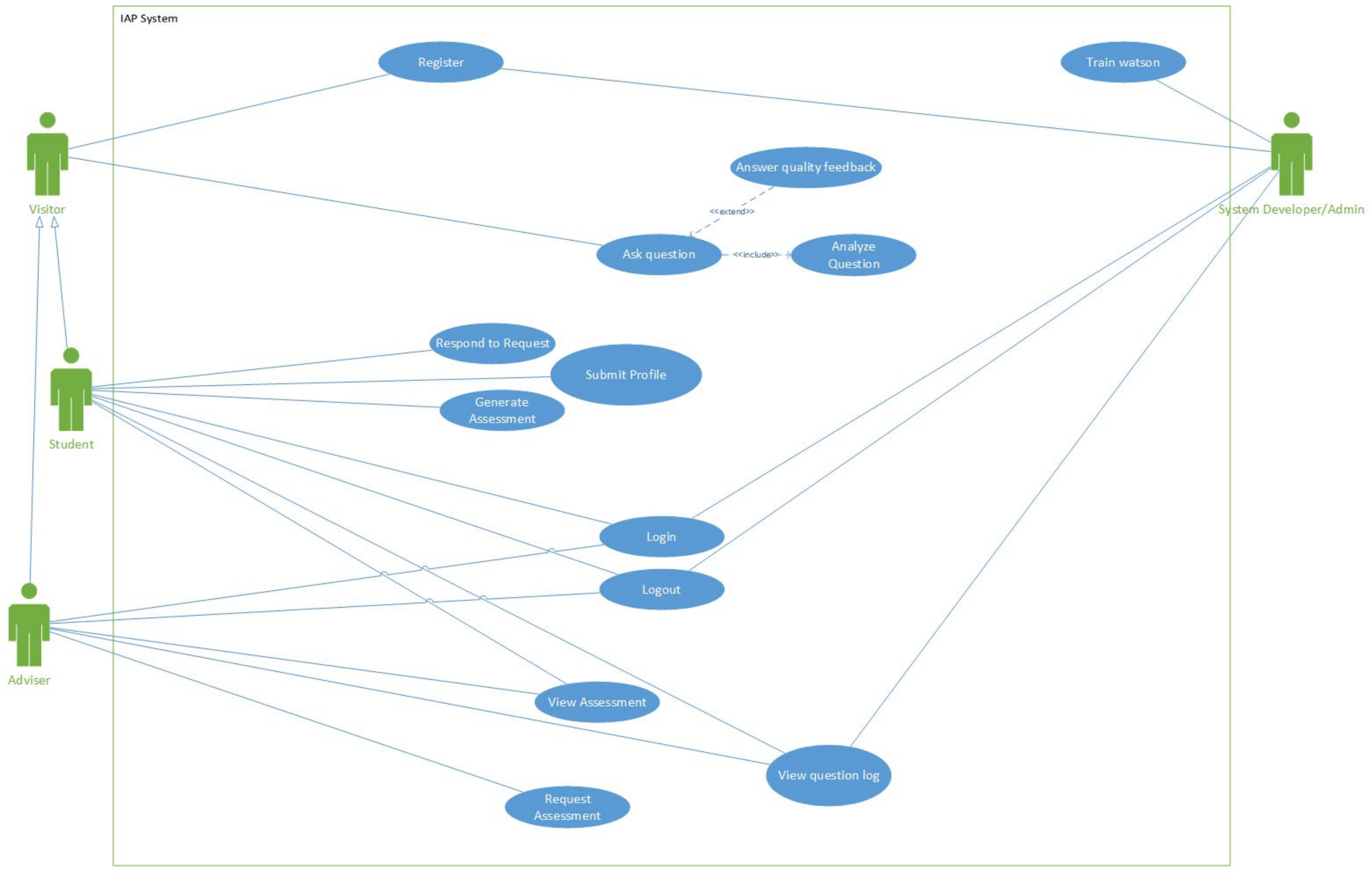


Figure 1: This is the Use Case Diagram of our system (Intelligent Academic Planner (IAP) System). Student and Advisor share the options that Visitor has (represented by the inheritance arrows). Each blue bubble represents a functionality that is present in the system, and will be gone into detail in 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

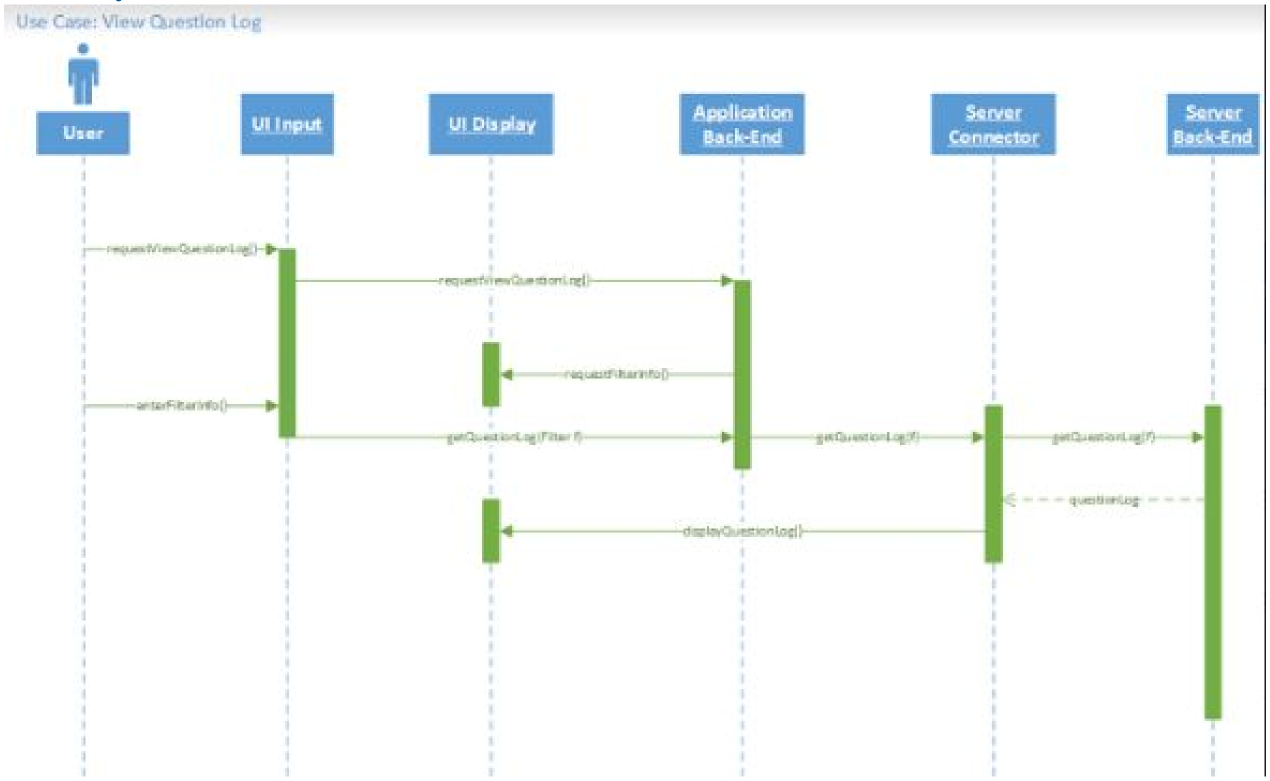
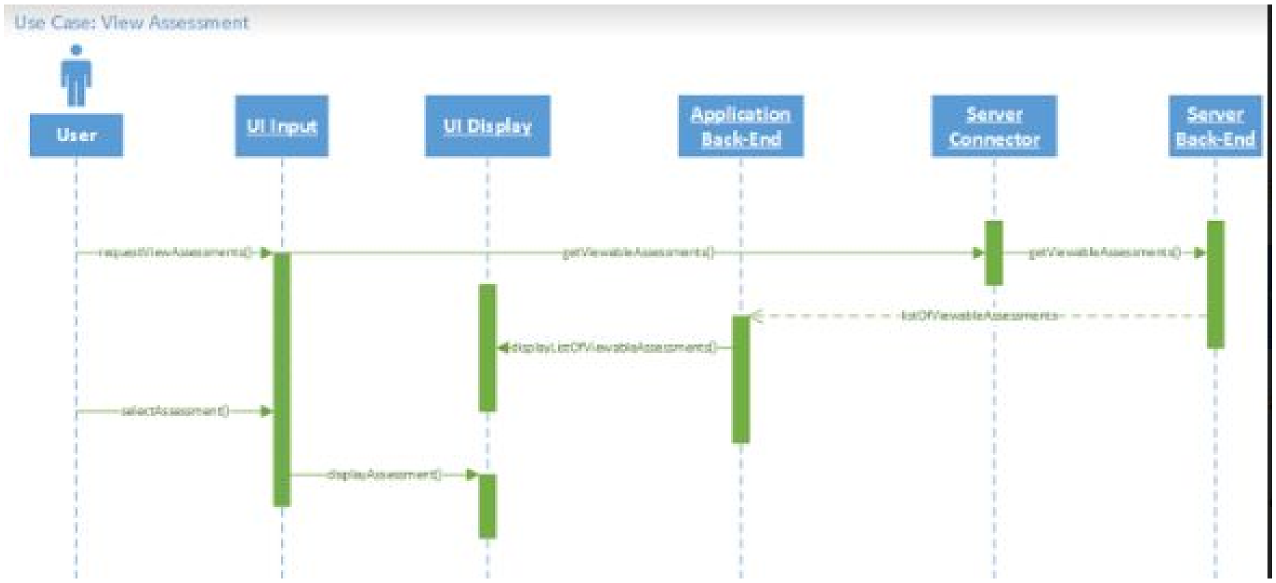
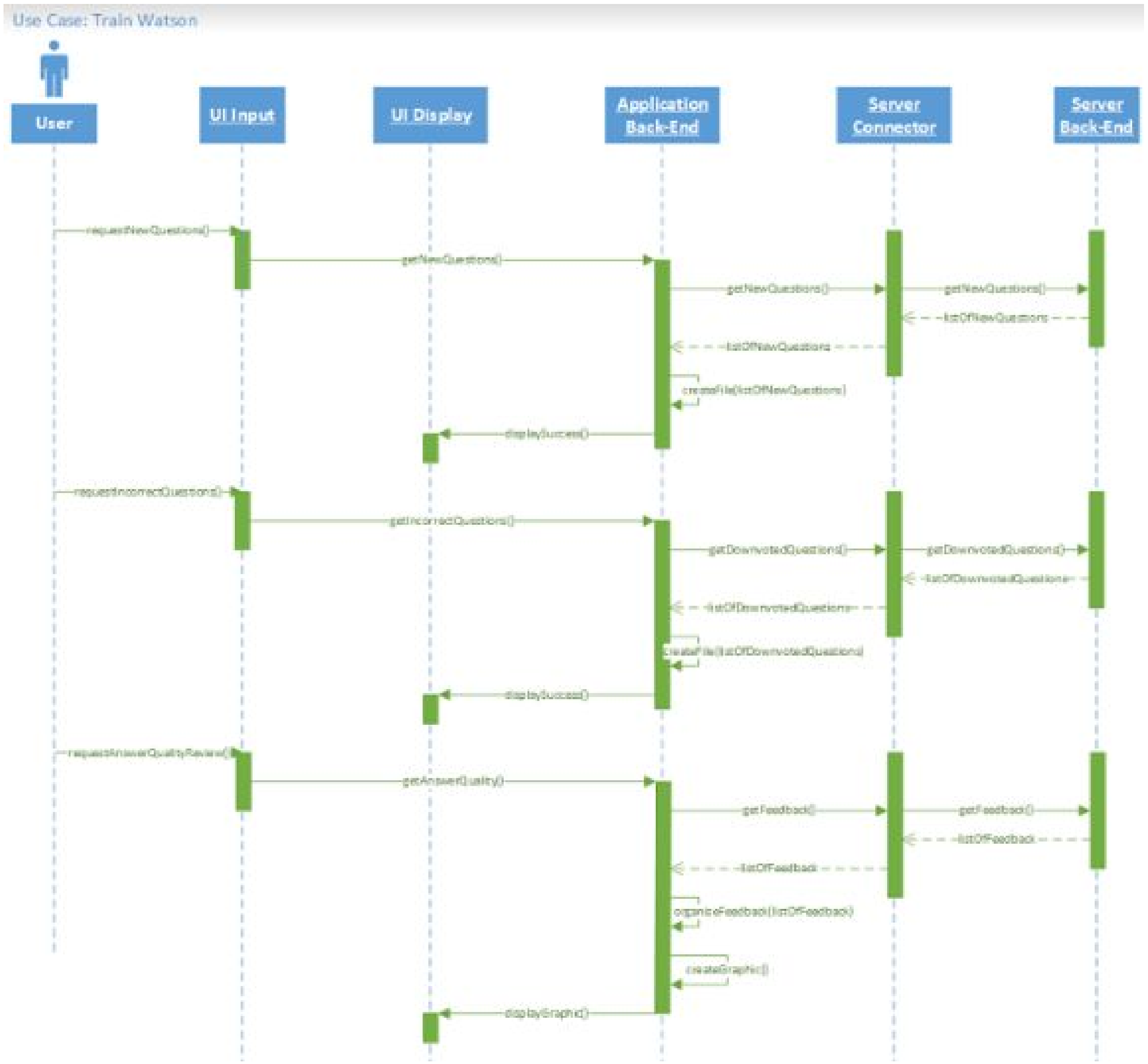


Figure 2: View Question Log Sequence Diagram: *After requesting to view the question log, the system will ask for filter info and then display the question log for viewing to the user after receiving the questions from the server's back-end.*



<u>Figure 3: View Assessment Sequence Diagram:</u> *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.*



<u>Figure 4: Train Watson Sequence Diagram:</u> *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.*

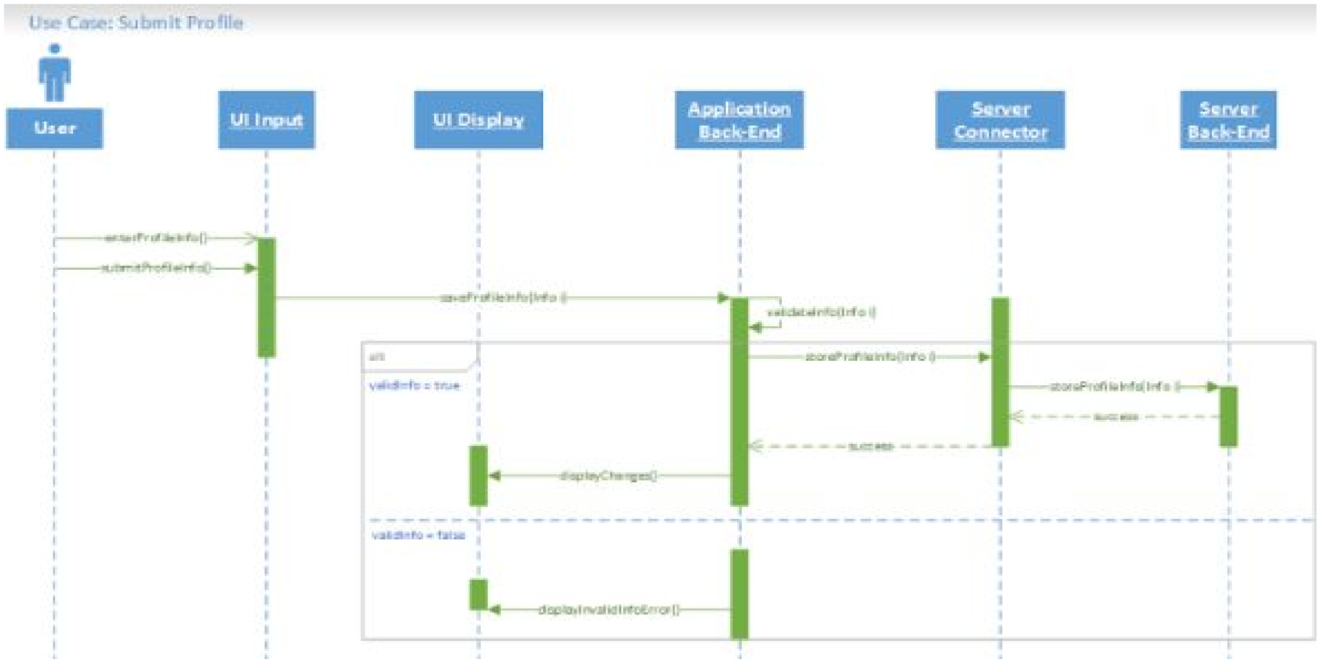


Figure 5: Submit Profile Sequence Diagram: *After a user inputs their profile information and requests to save, the system validates the information to ensure that there is no invalid or dangerous input. If it passes validation, the information is saved and a message stating that it was saved is displayed.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

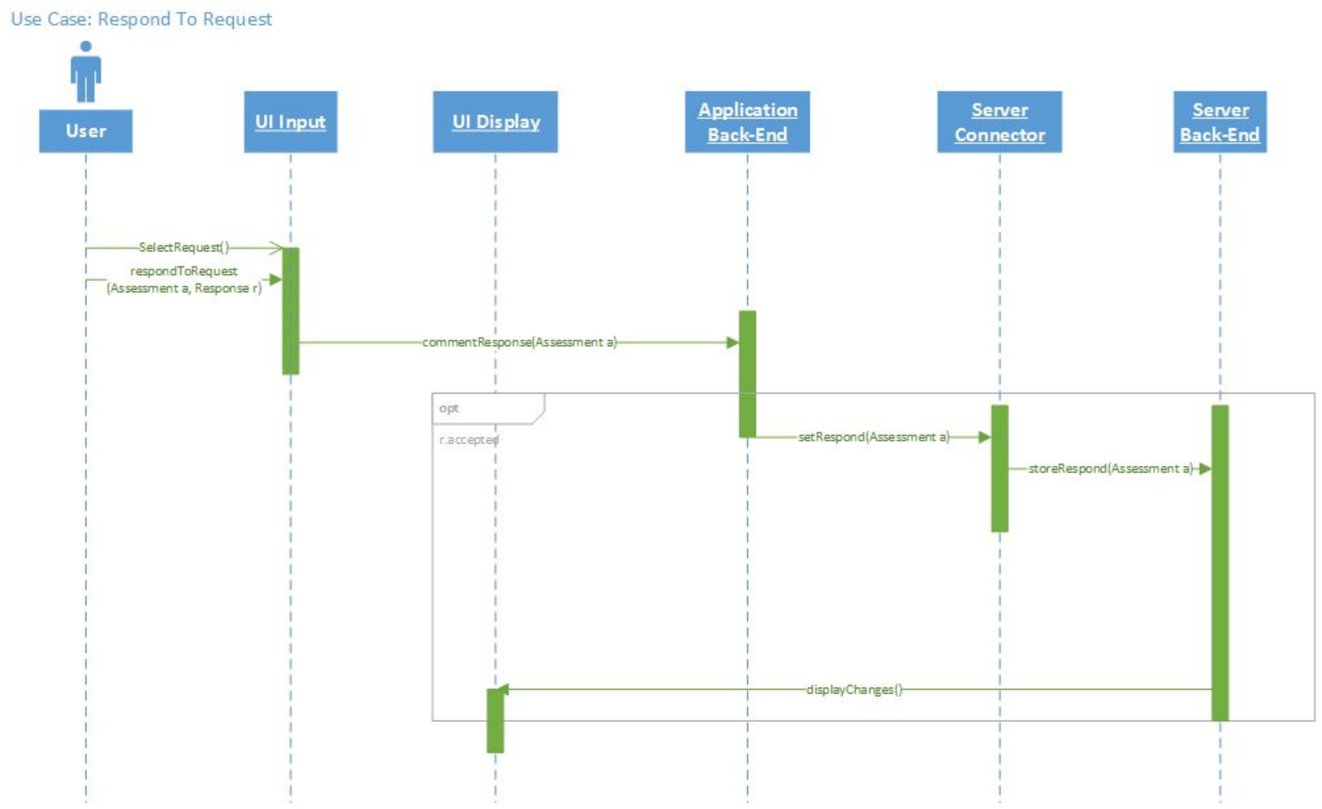


Figure 6: Respond to Request Sequence Diagram: *After the user selects the request they with to respond to and submits their response, the system sends a message back to the requester, sets the visibility of the selected requested assessment*

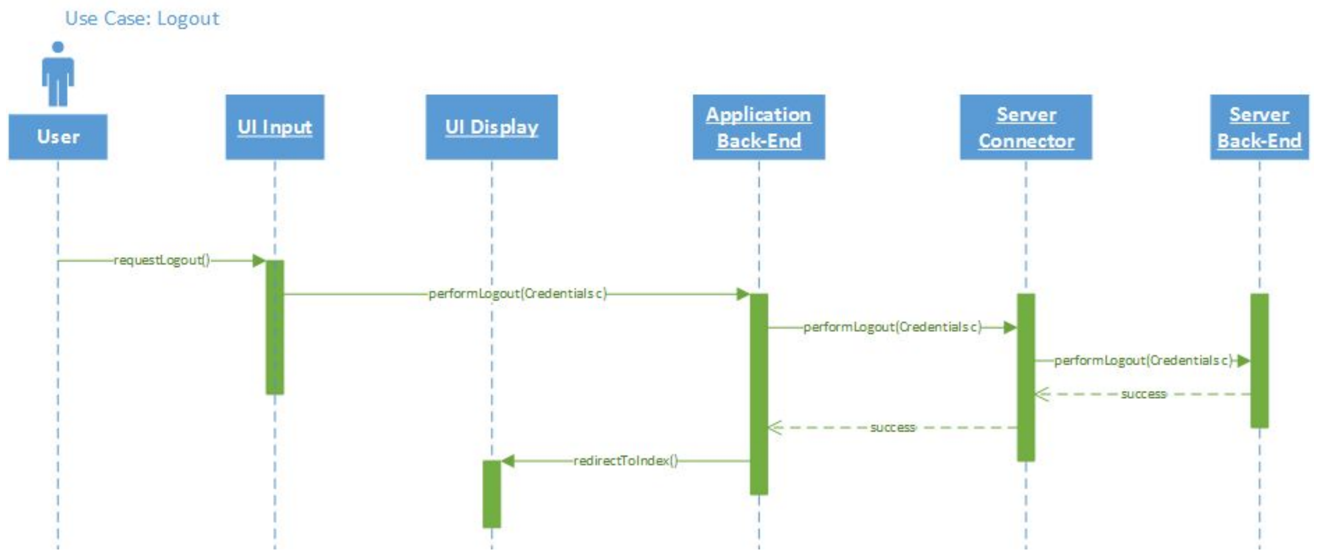


Figure 7: Logout Sequence Diagram: *After a user requests to logout, the system logs the user out of the server and redirect user to index page.*

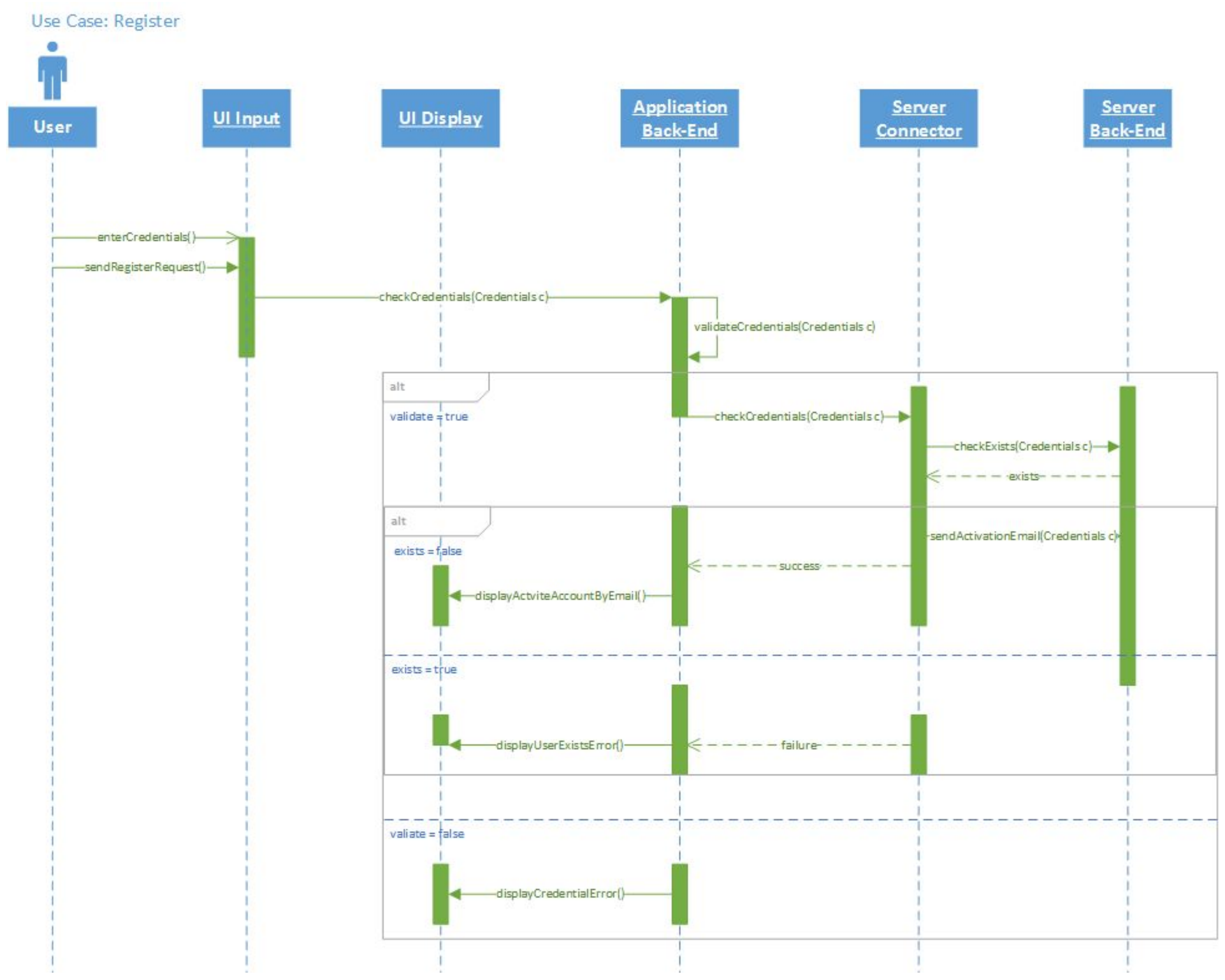


Figure 8: Register Sequence Diagram: *After the user enters credentials and requests to register, the system checks the credentials to see if they meet the requirements (email format, correct number of characters, etc). If valid, the system checks if a user with the same email already exists. If a user 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.*

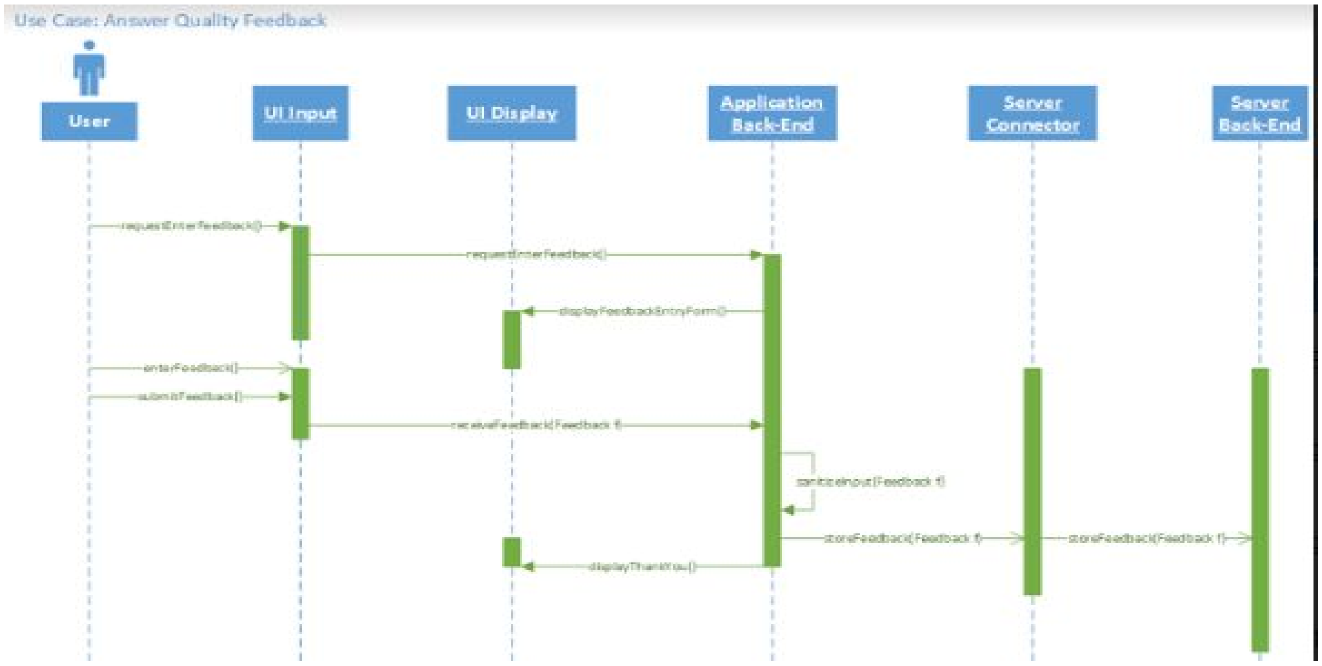


Figure 9: Answer Quality Feedback Sequence Diagram: *After a user requests to enter feedback, an entry form is displayed for the user to input the feedback. Once submitted, the system checks to make sure the input is not dangerous, then stores it in the database, and finally displays a thank you message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

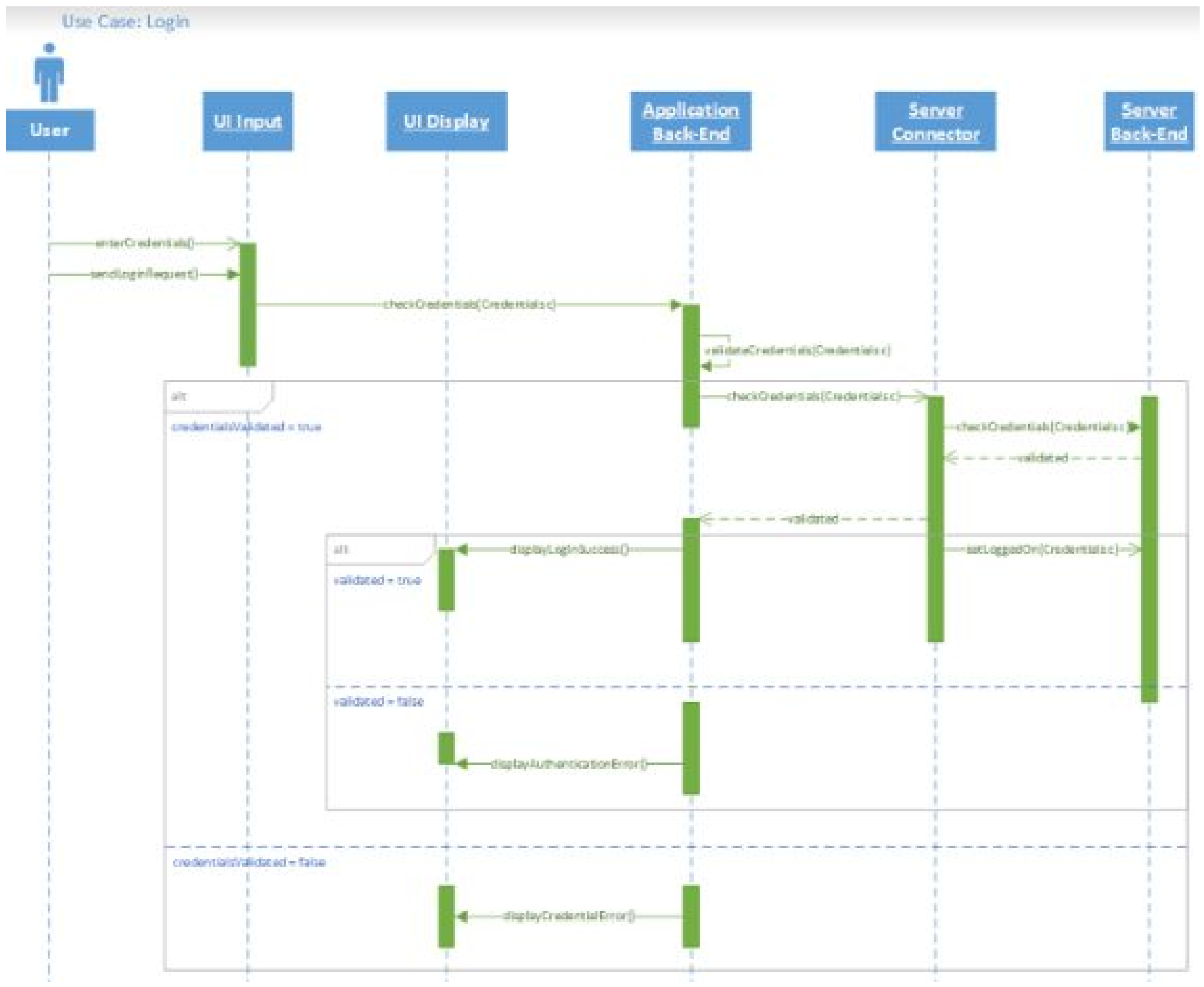
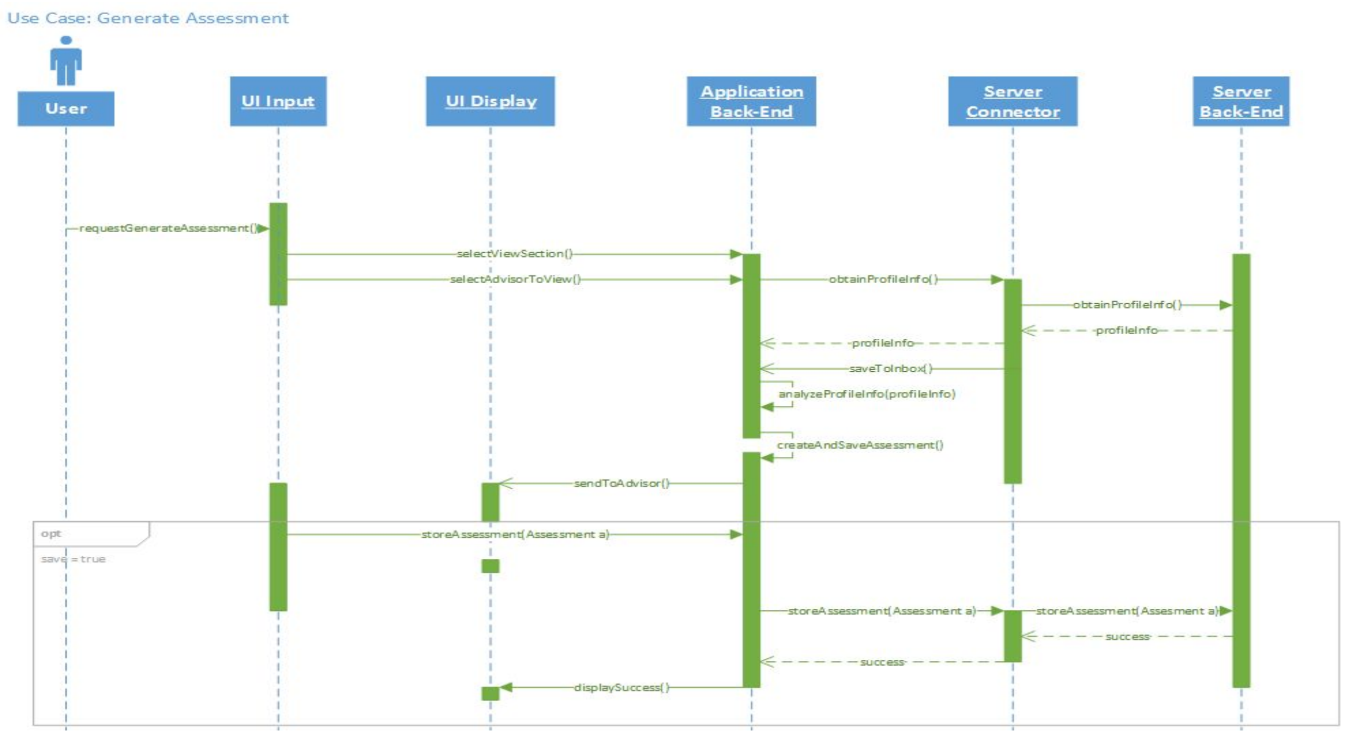

Use Case: Login
User
UI Input
UI Display
Application Back-End
Server Connector
Server Back-End
enterCredentials()
sendLoginRequest()
checkCredentials(Credentials c)
validateCredentials(Credentials c)
alt
credentialsValidated = true
checkCredentials(Credentials c)
checkCredentials(Credentials c)
validated
validated
alt
displayLoginSuccess()
setLoggedOn(Credentials c)
validated = true
validated = false
displayAuthenticationError()
credentialsValidated = false
displayCredentialError()

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.*

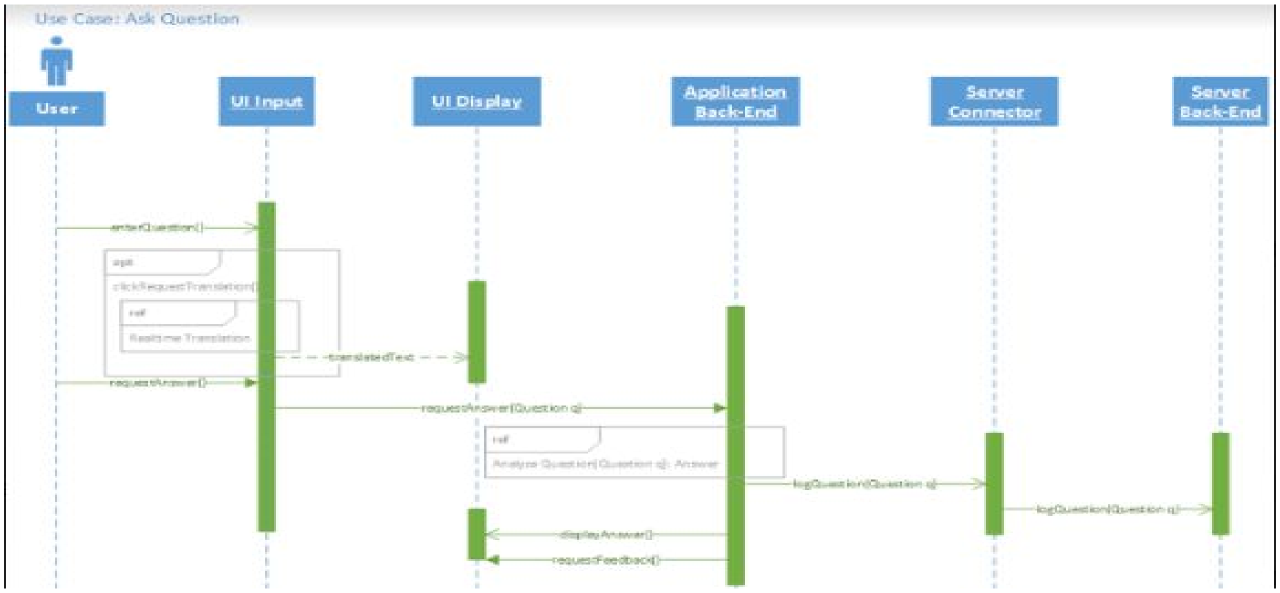


Figure 12: Ask Question Sequence Diagram: *When the user enters a question, they can request translation before requesting their answer. Once an answer is requested, the system calls to Analyze Questions and performs the functionality and returns the answer. The question is then logged and the answer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

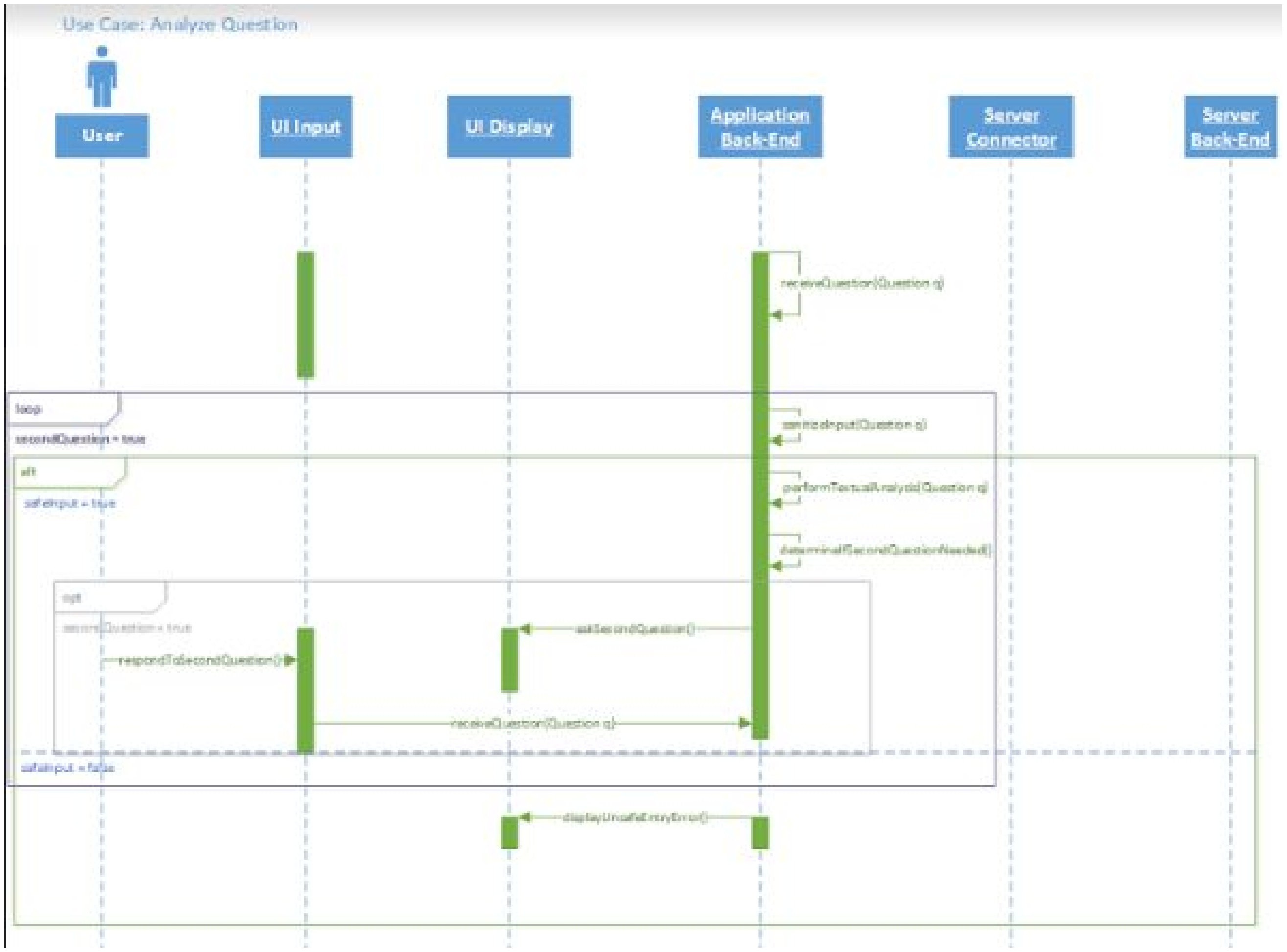

Use Case: Analyze Question
User
UI Input
UI Display
Application Back-End
Server Connector
Server Back-End
receiveQuestion(Question q)
loop
secondQuestion = true
sanitizeInput(Question q)
alt
safeInput = true
performTextualAnalysis(Question q)
determineIfSecondQuestionNeeded()
opt
secondQuestion = true
askSecondQuestion()
respondToSecondQuestion()
receiveQuestion(Question q)
safeInput = false
displayUnsafeEntryError()

__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.

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

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 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 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.

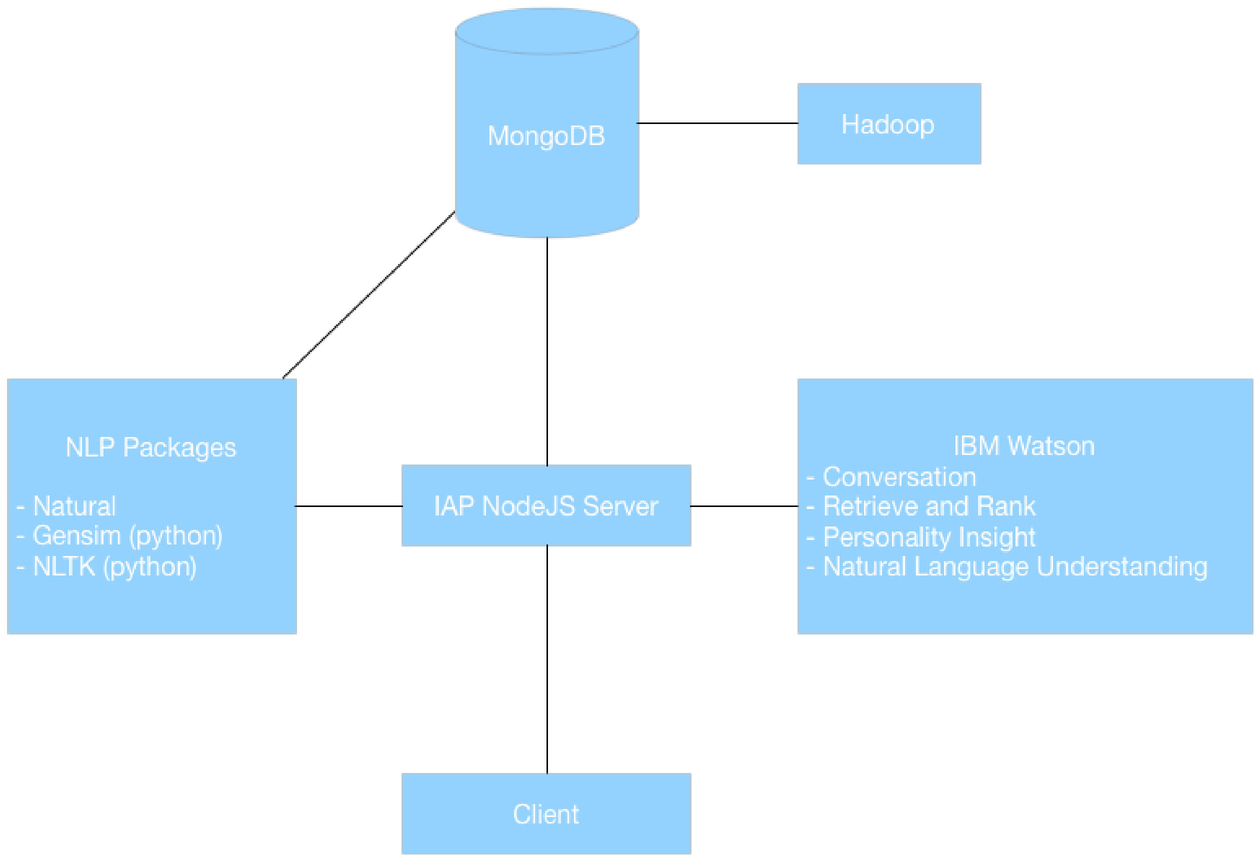


Figure 14: Relevant packages/products flow diagram

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

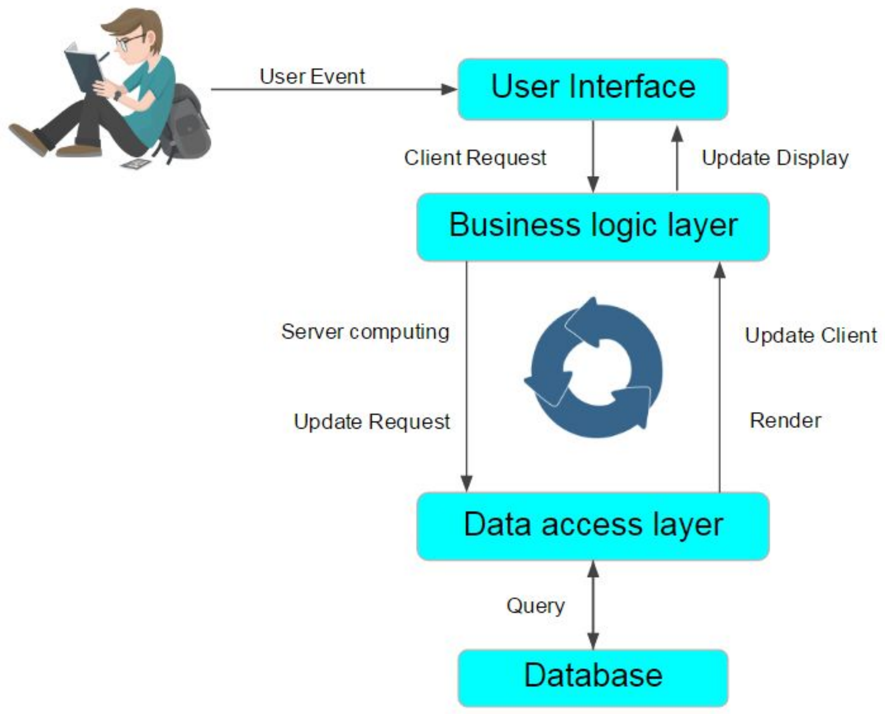


Figure 15: User initializes a "submit question" event on the web browser, then the server will handle the question through the logic layer. The database can then be queried and, if there are no exceptions, the data will be sent to Watson through Bluemix RESTful APIs. Finally, the response information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

6.2. Structural Design

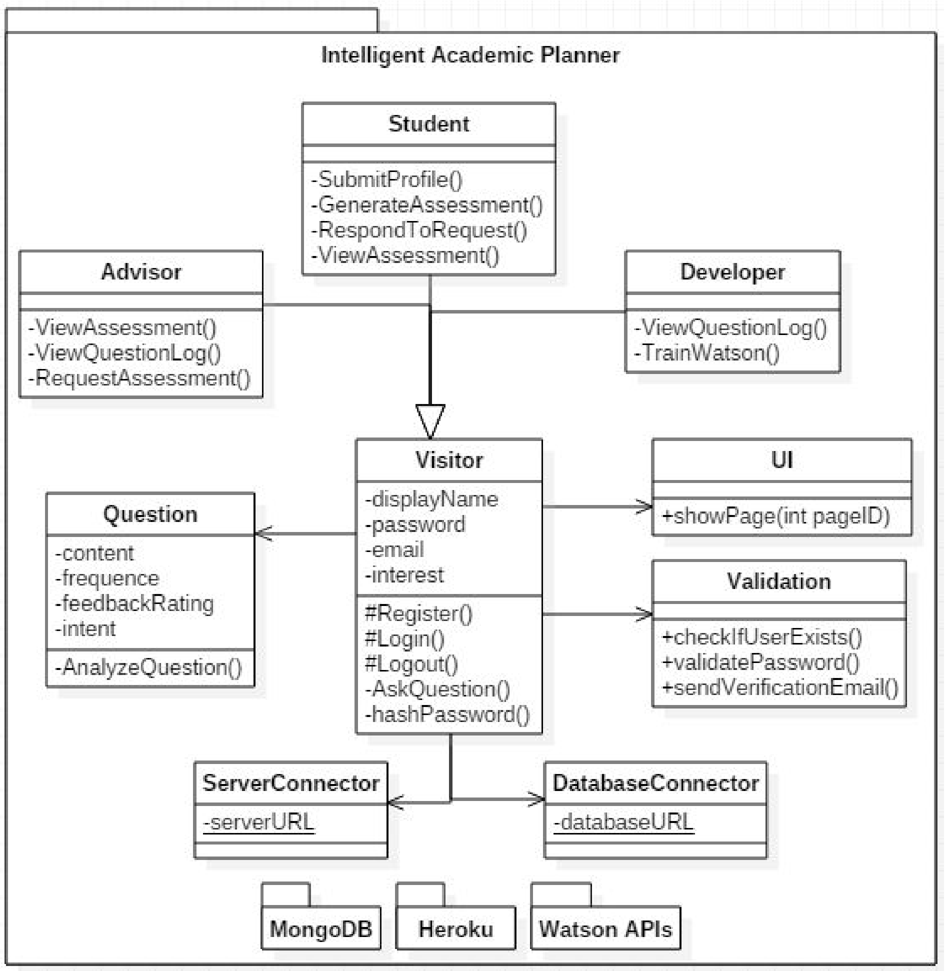
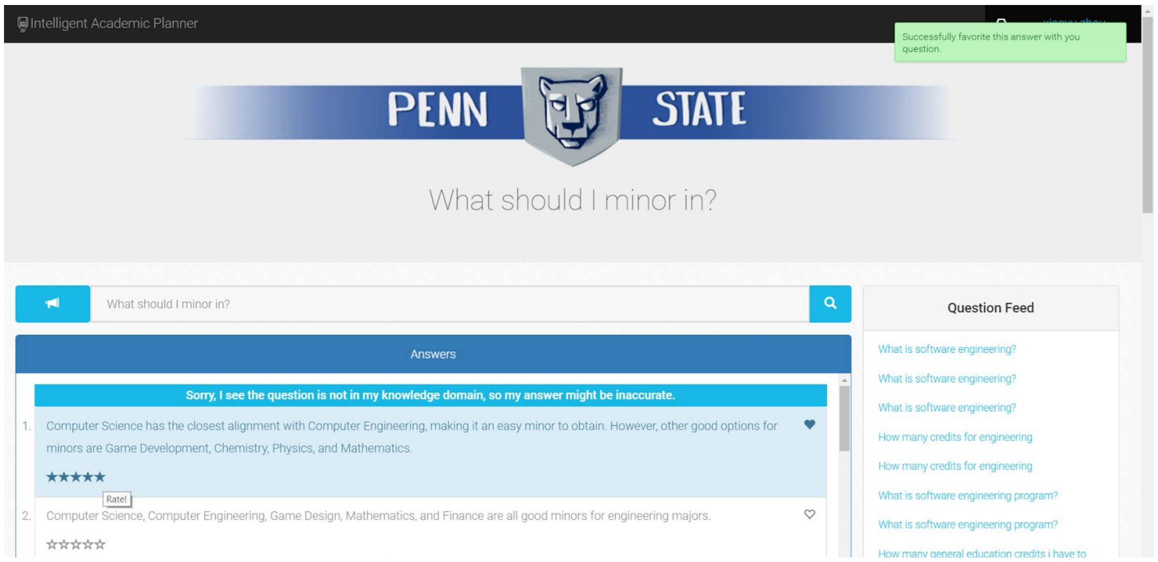


Figure 16: This is the class diagram of our system. We are using Heroku as our server and MongoDB as our database. All visitors have the ability to ask questions, and questions are stored on our server. The Validation class is used to check for safe and correct inputs, improving security of our system.

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.


Intelligent Academic Planner

Successfully favorite this answer with you question.

What should I minor in?

Answers

Sorry, I see the question is not in my knowledge domain, so my answer might be inaccurate.

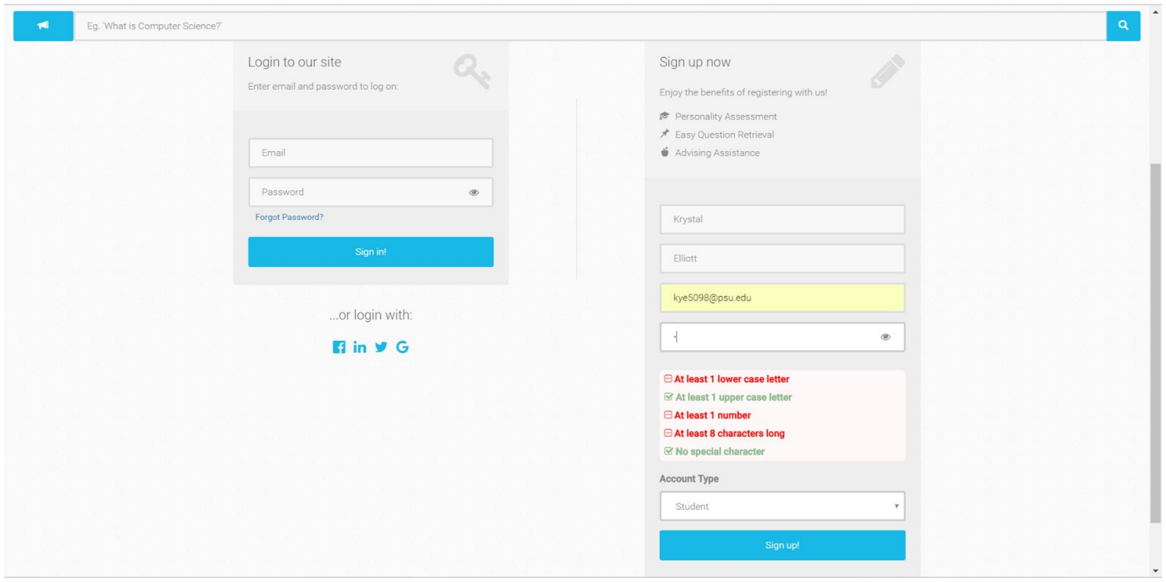
1. Computer Science has the closest alignment with Computer Engineering, making it an easy minor to obtain. However, other good options for minors are Game Development, Chemistry, Physics, and Mathematics. ★★★★★

2. Computer Science, Computer Engineering, Game Design, Mathematics, and Finance are all good minors for engineering majors. ☆☆☆☆☆

Question Feed

- What is software engineering?
- What is software engineering?
- What is software engineering?
- How many credits for engineering
- How many credits for engineering
- What is software engineering program?
- What is software engineering program?
- How many general education credits i have to

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.


Eg. 'What is Computer Science?'

Login to our site

Enter email and password to log on:

Email

Password

Forgot Password?

Sign in!

...or login with:

Sign up now

Enjoy the benefits of registering with us!

- Personality Assessment
- Easy Question Retrieval
- Advising Assistance

Krystal

Elliott

kye5098@psu.edu

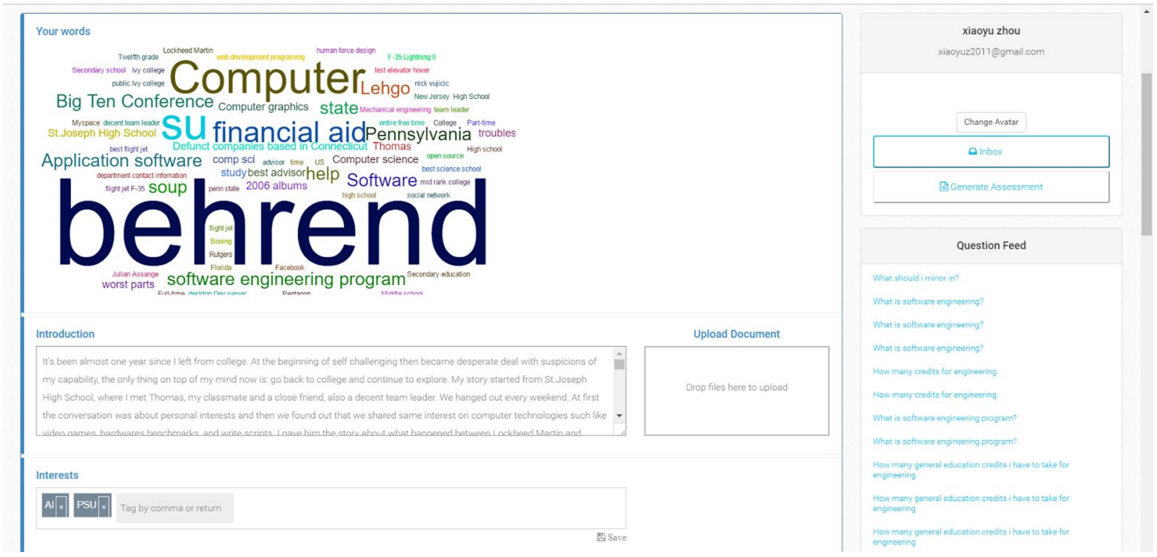
- At least 1 lower case letter
- ✓ At least 1 upper case letter
- At least 1 number
- At least 8 characters long
- ✓ No special character

Account Type

Student

Sign up!

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.



xiaoyuz2011@gmail.com

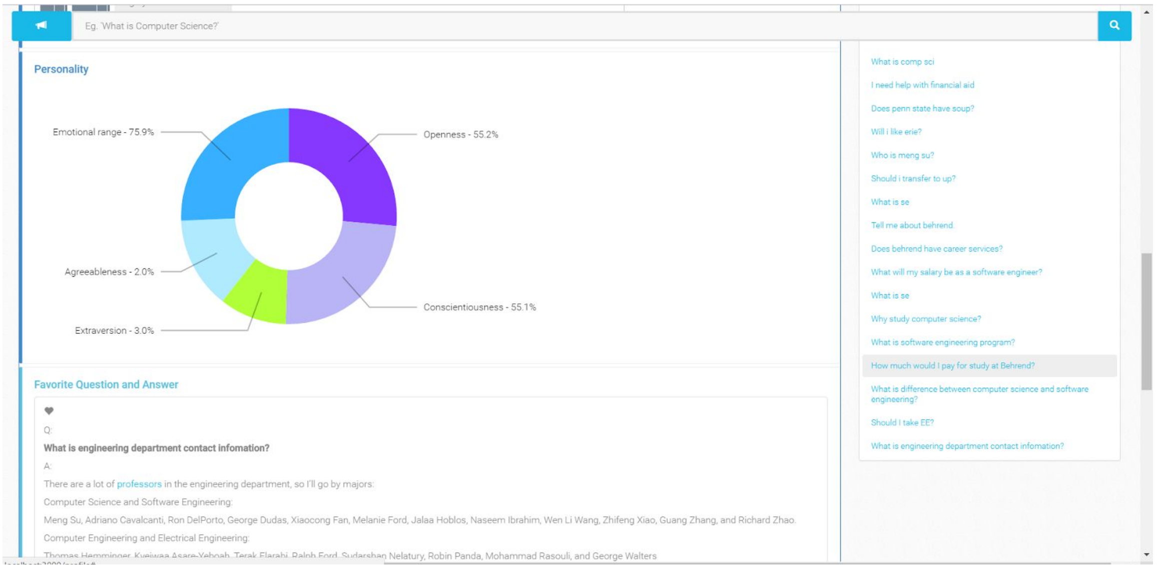
It's been almost one year since I left from college. At the beginning of self challenging then became desperate deal with suspicions of my capability, the only thing on top of my mind now is: go back to college and continue to explore. My story started from St.Joseph High School, where I met Thomas,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

Drop files here to upload

AI PSU Tag by comma or return

What should i minor in?
What is software engineering?
What is software engineering?
What is software engineering?
How many credits for engineering
How many credits for engineering
What is software engineering program?
What is software engineering program?
How many general education credits i have to take for engineering
How many general education credits i have to take for engineering
How many general education credits i have to take for engineering

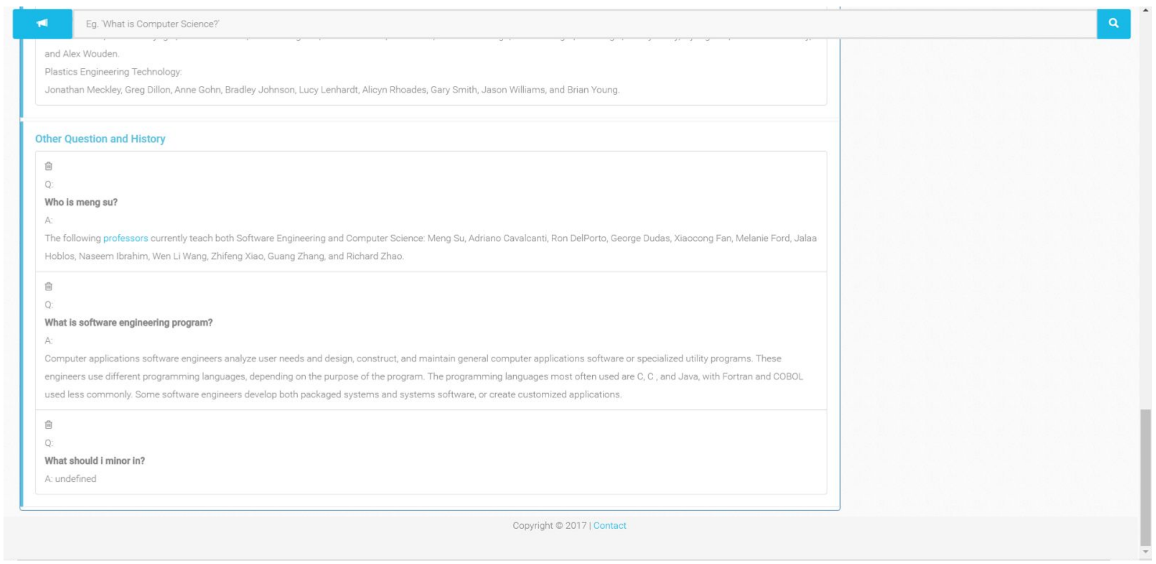
Figure 19: Profile: Only available after login, display information about the user, including: a conglomeration of words describing the user's interests, a profile picture, an introduction written by the user about the user, and interests. This page also features the Question Feed and Search Bar.



What is comp sci
I need help with financial aid
Does penn state have soup?
Will i like erie?
Who is meng su?
Should i transfer to up?
What is se
Tell me about behrend.
Does behrend have career services?
What will my salary be as a software engineer?
What is se
Why study computer science?
What is software engineering program?
How much would I pay for study at Behrend?
What is difference between computer science and software engineering?
Should I take EE?
What is engineering department contact infomation?

Q:
What is engineering department contact infomation?
A:
There are a lot of professors in the engineering department, so I'll go by majors:
Computer Science and Software Engineering:
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.
Computer Engineering and Electrical Engineering:

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.

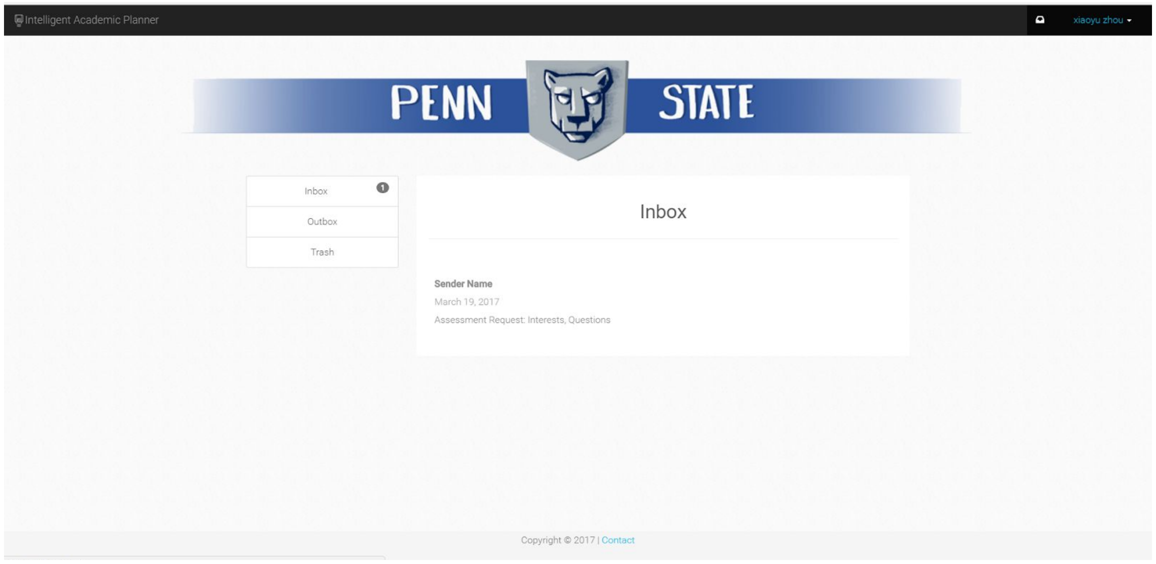
Q:

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.



Inbox

Sender Name

March 19, 2017

Assessment Request: Interests, Questions

Copyright © 2017 | Contact

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

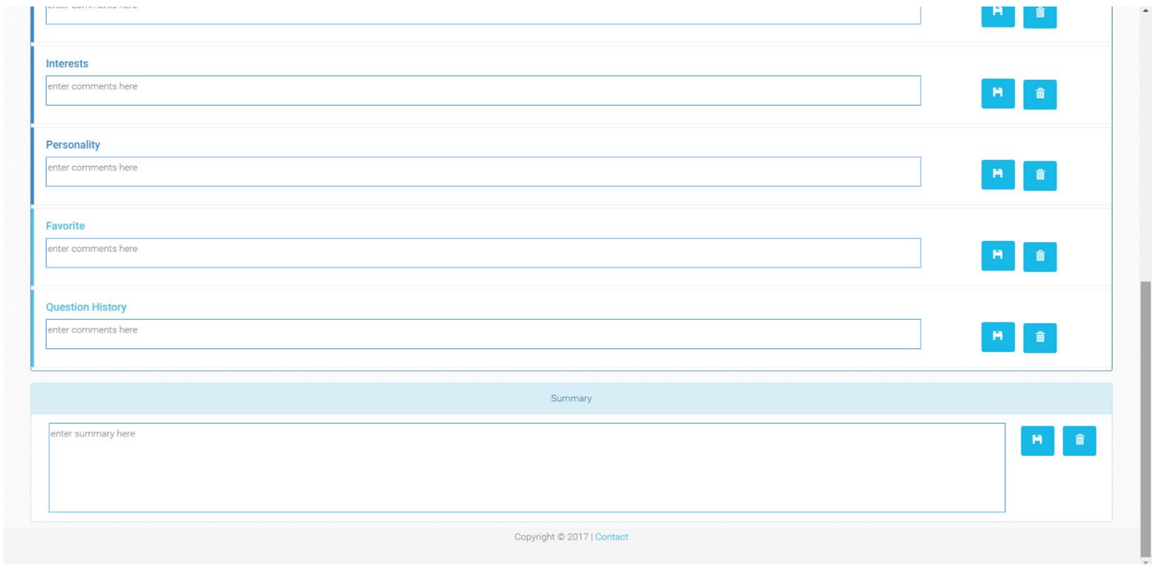
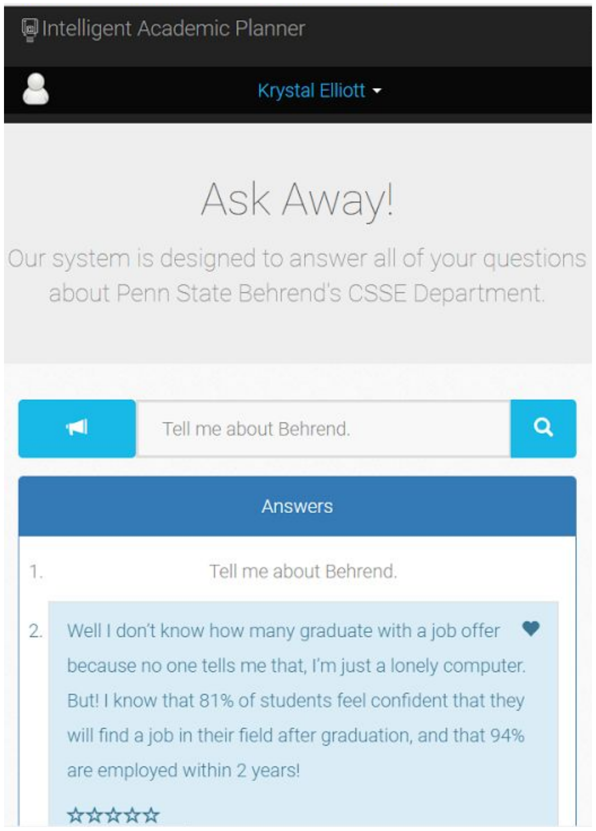


Figure 23: Advising: The Advising page will consist of the same fields contained within a student profile. Each section will be populated with the information from the student who sent the assessment. There is an opportunity for the advisor to leave a comment below the section. At the bottom, there is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


Intelligent Academic Planner

Krystal Elliott

Ask Away!

Our system is designed to answer all of your questions about Penn State Behrend's CSSE Department.

Tell me about Behrend.

Answers

1. Tell me about Behrend.
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!

☆☆☆☆☆

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

6.4.Behavioral Design

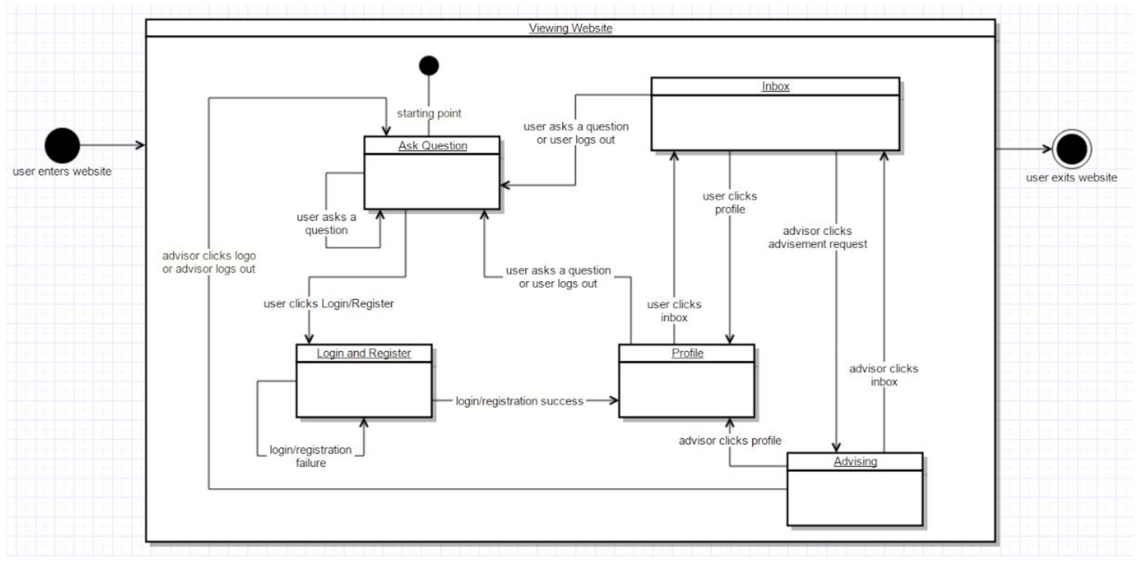

Viewing Website
user enters website
user exits website
starting point
Ask Question
Inbox
user asks a question or user logs out
user asks a question
advisor clicks logo or advisor logs out
user clicks profile
advisor clicks advisement request
user asks a question or user logs out
user clicks Login/Register
user clicks inbox
Login and Register
Profile
advisor clicks inbox
login/registration success
login/registration failure
advisor clicks profile
Advising

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.

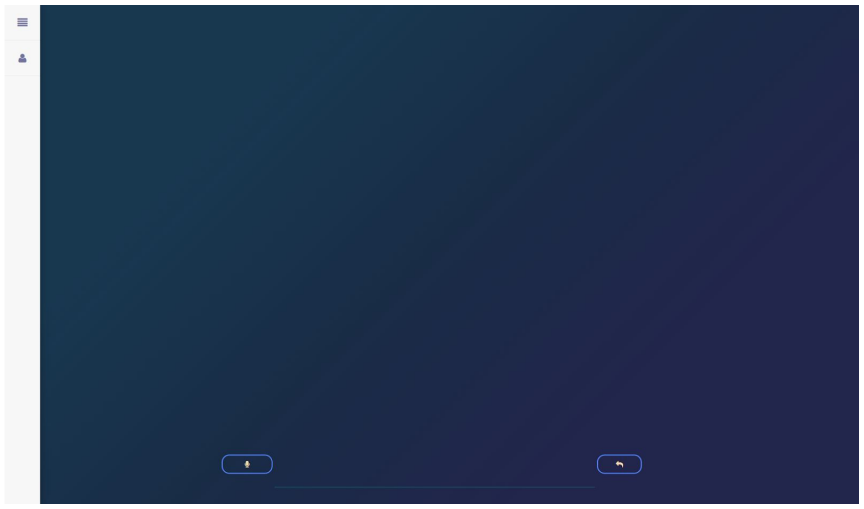


***Figure 26: Ask Question**: use mobile compatible view design template*

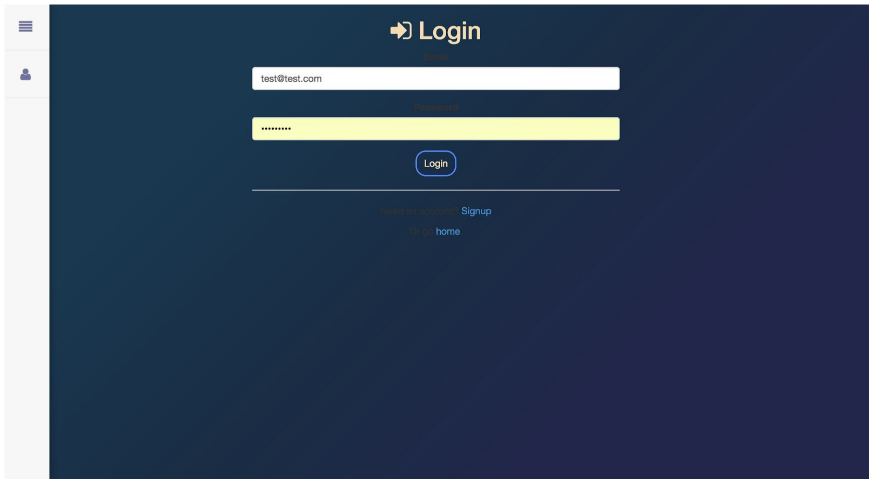
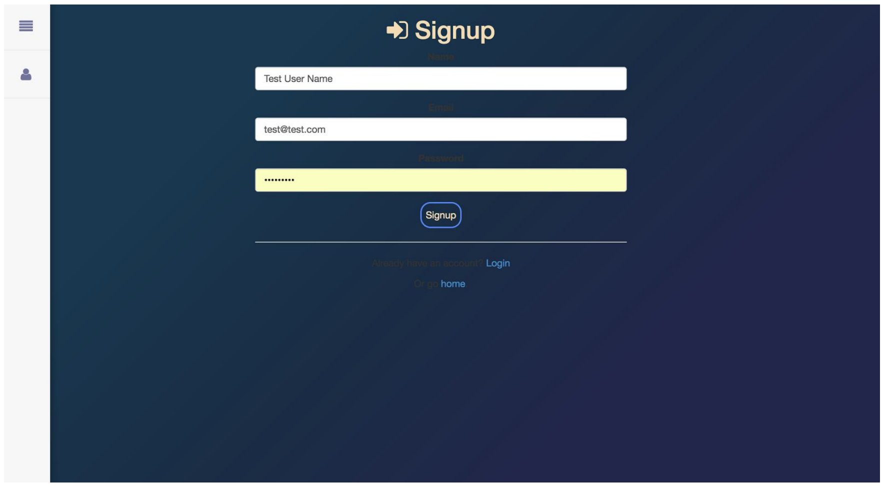


***Figure 27: Login**: require register email and password*

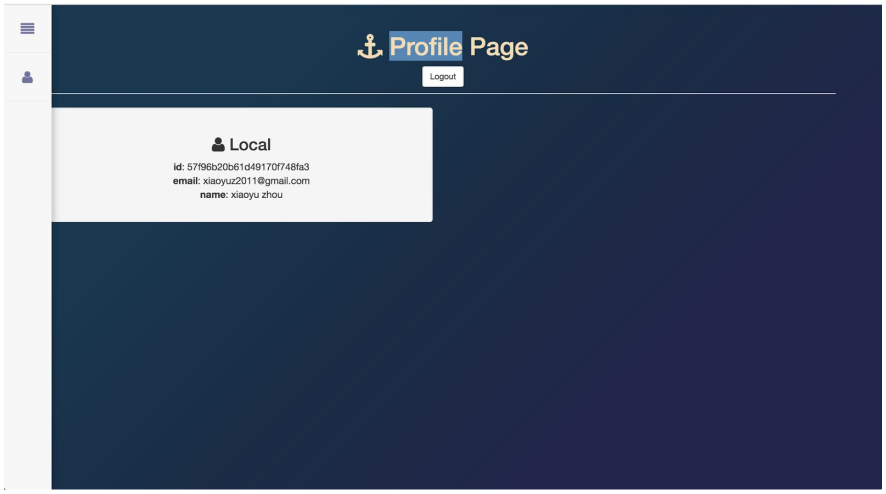


Test User Name

test@test.com

••••••••••

__Figure 28: Register Account__: require a display name, register email, and password



id: 57f96b20b61d49170f748fa3
email: xiaoyuz2011@gmail.com
name: xiaoyu zhou

__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.

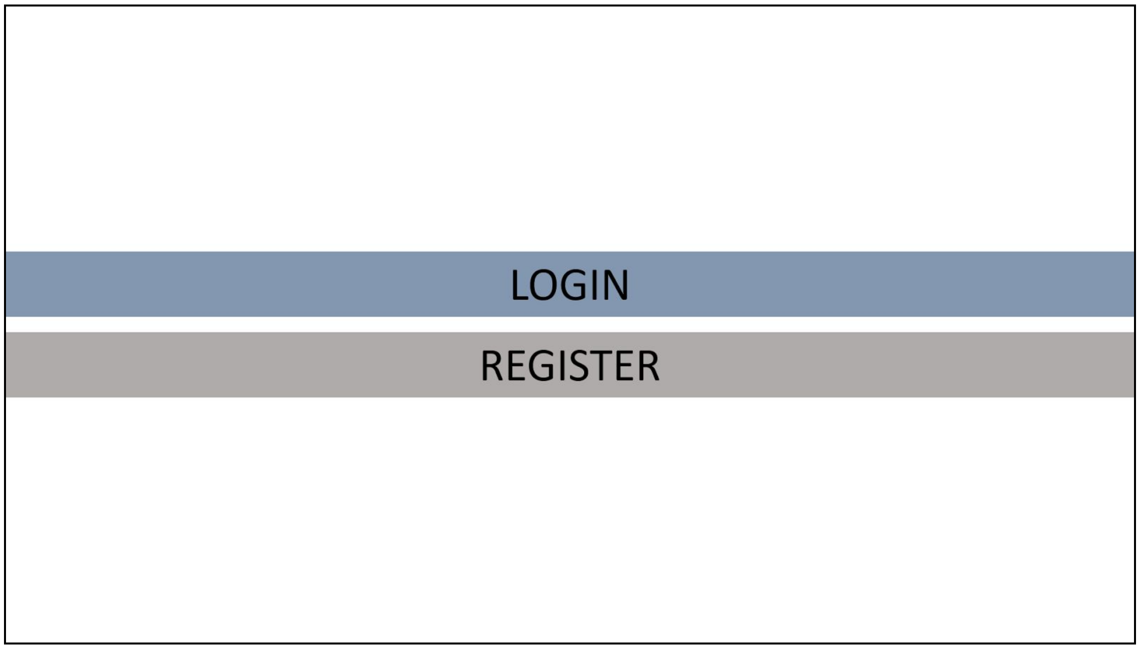
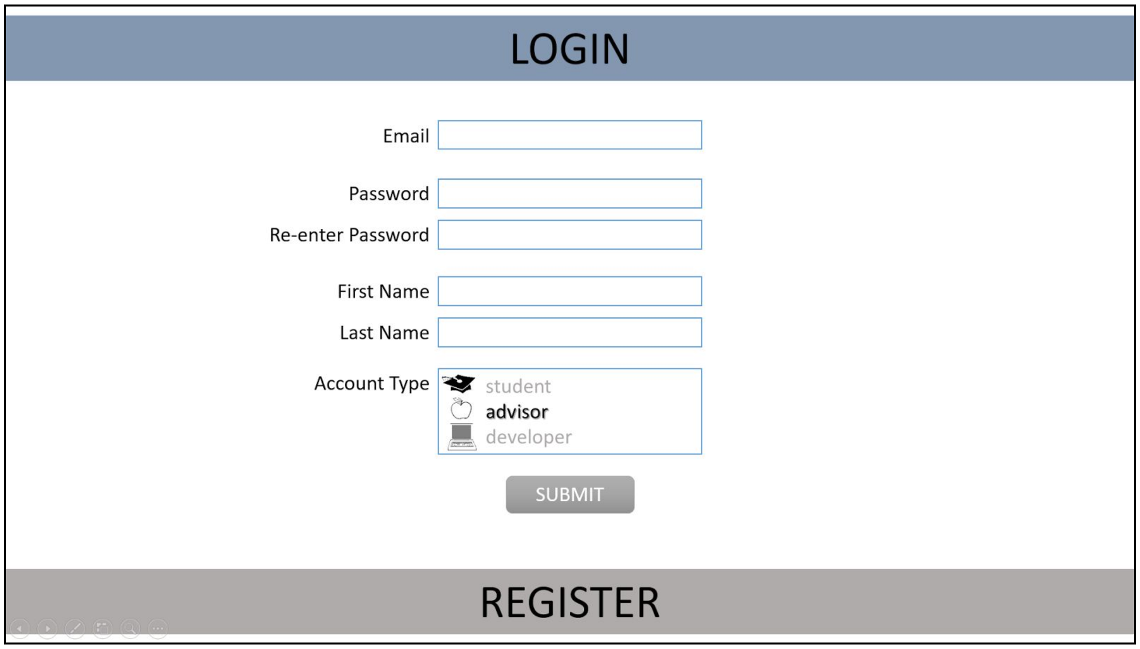
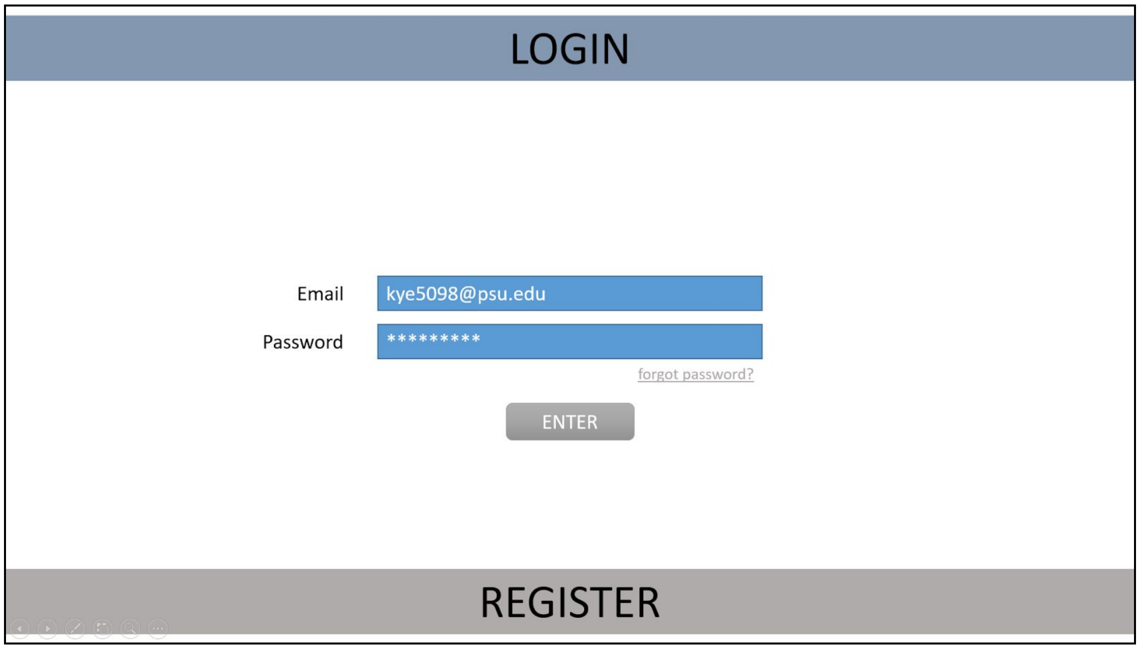


Figure 30: Landing Page: *offers options of logging in and registering*



Email	
Password	
Re-enter Password	
First Name	
Last Name	
Account Type	student advisor developer

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


LOGIN

Email kye5098@psu.edu

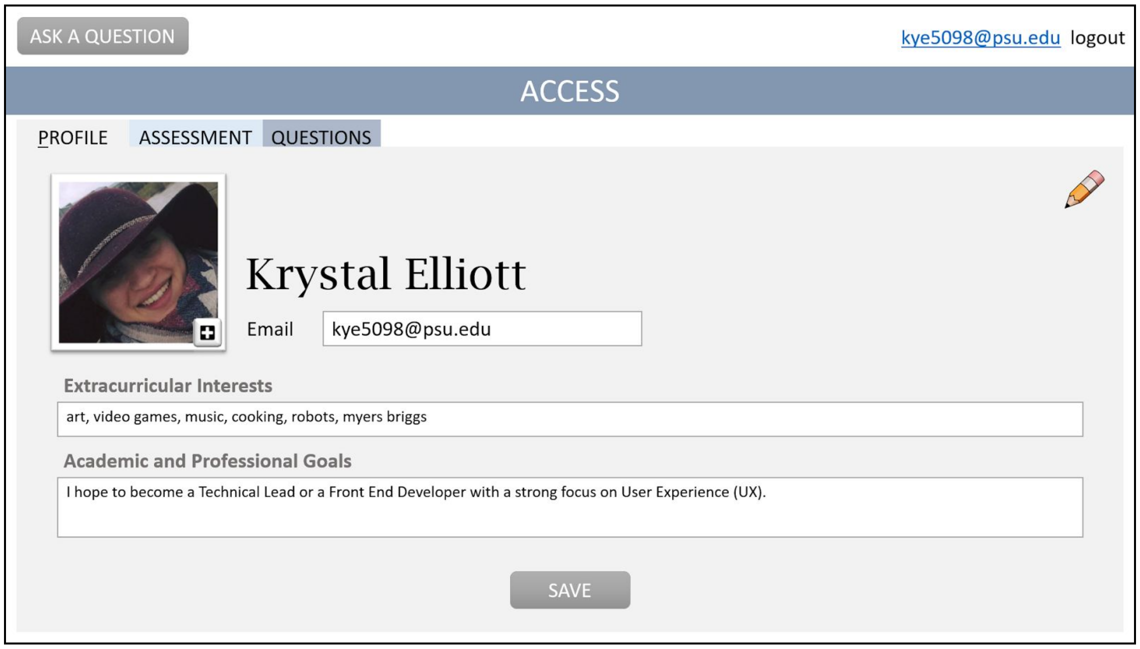
Password *********

forgot password?

ENTER

REGISTER

Figure 32: Login: allows user to login


ASK A QUESTION

kye5098@psu.edu logout

ACCESS

PROFILE ASSESSMENT QUESTIONS

Krystal Elliott

Email kye5098@psu.edu

Extracurricular Interests

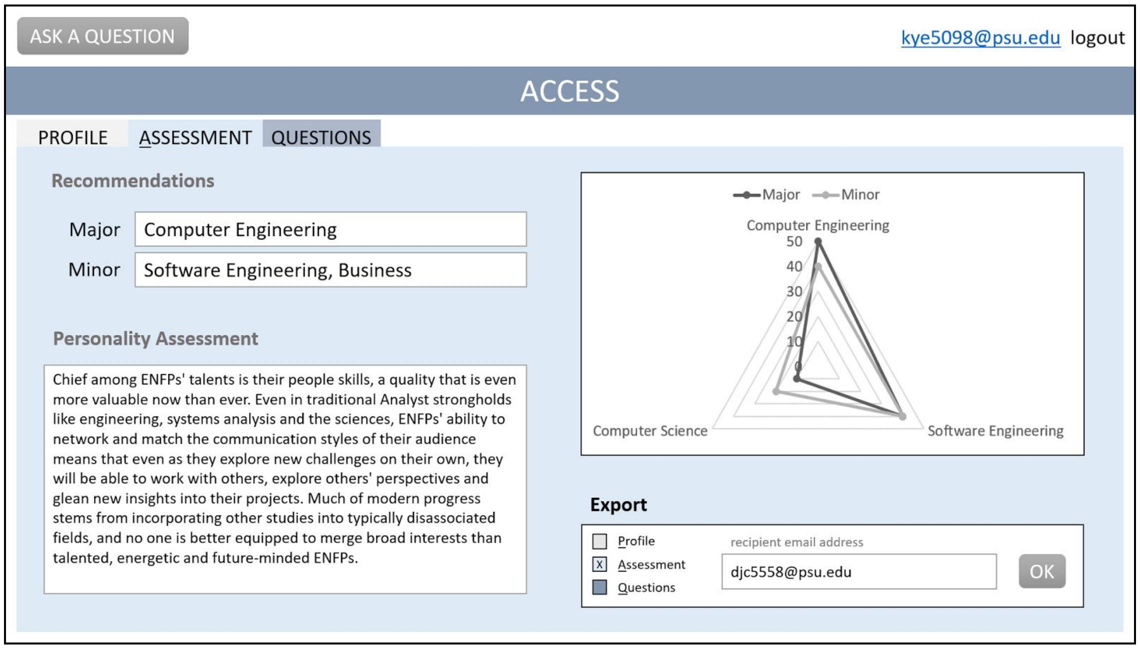
art, video games, music, cooking, robots, myers briggs

Academic and Professional Goals

I hope to become a Technical Lead or a Front End Developer with a strong focus on User Experience (UX).

SAVE

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



Recommendations

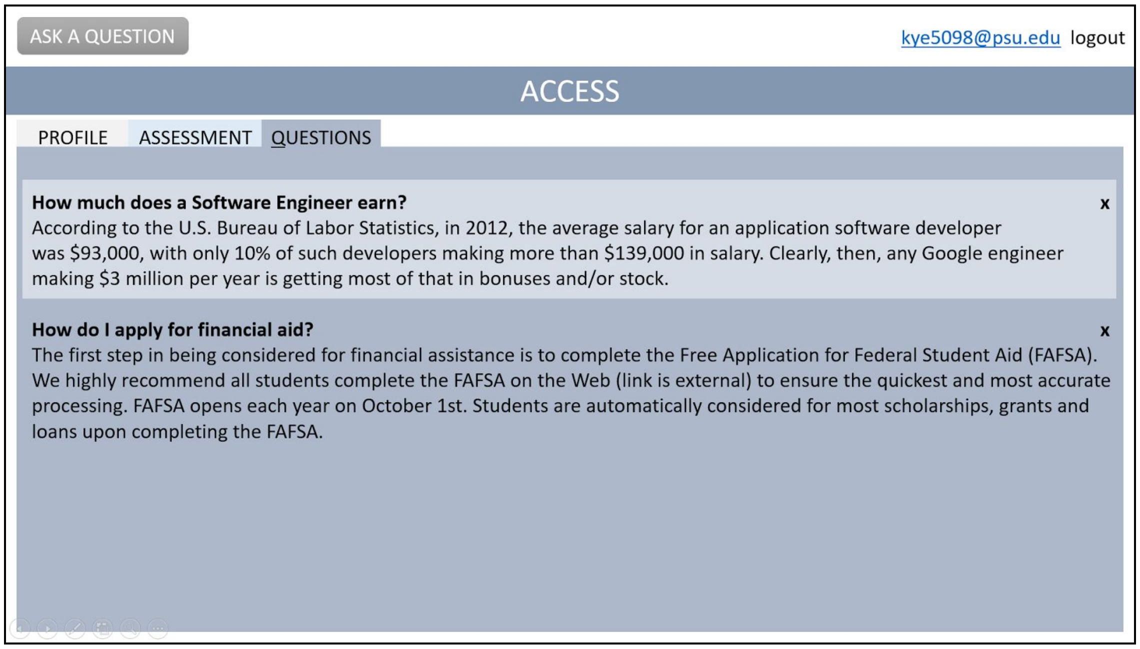
Major	Computer Engineering
Minor	Software Engineering, Business

Personality Assessment

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

Export: ☐ Profile, ☒ Assessment, ▪ Questions — recipient email address: djc5558@psu.edu

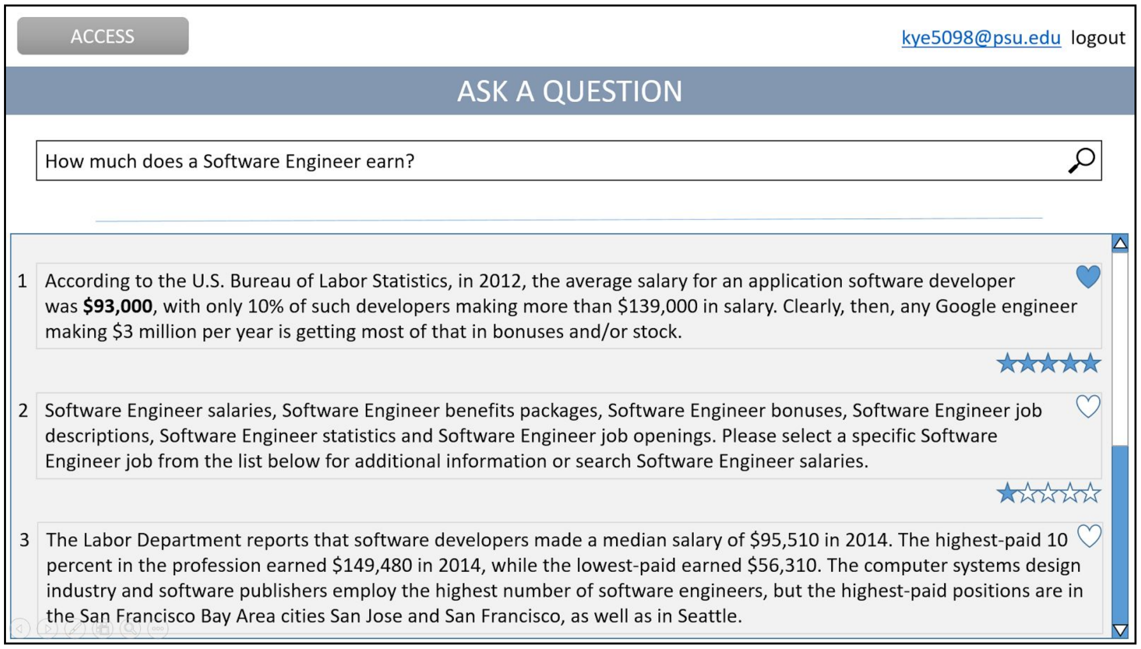
*Figure 34: **Assessment:** allows user to view their personalized assessment and to send it, partially or in full, to an email address.*



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

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

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



How much does a Software Engineer earn?

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.

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

Structural Alternative Designs

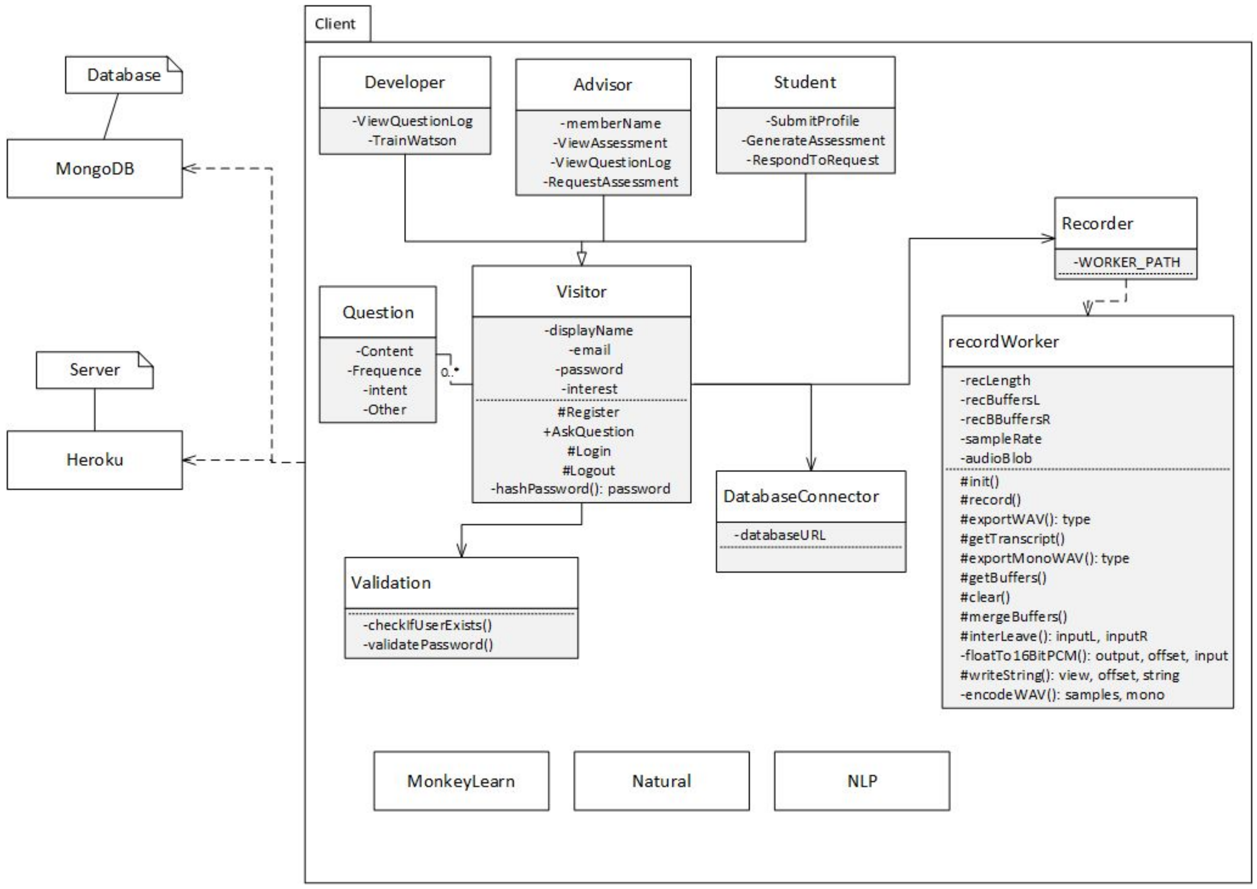


Figure 37: alternative structural design

- ➔ 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, programing 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

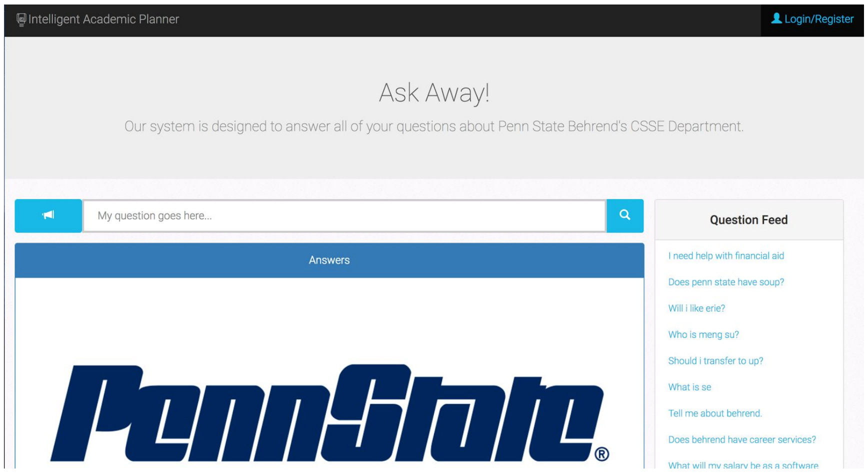
NodeJS: main server platform, version require v6.0 and up
MongoDB: version require 3.0.x and up
Modern Web Browser: Firefox, Chrome, Edge, Safari, Opera, Iceweasel
Bluemix
Heroku-Github

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.



Ask Away!

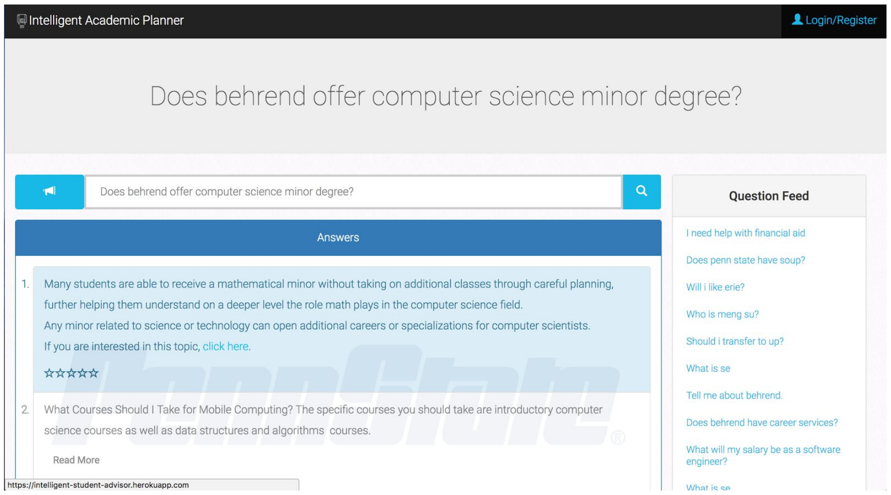
Our system is designed to answer all of your questions about Penn State Behrend's CSSE Department.

My question goes here...

I need help with financial aid
Does penn state have soup?
Will i like erie?
Who is meng su?
Should i transfer to up?
What is se
Tell me about behrend.
Does behrend have career services?
What will my salary be as a software

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.



Does behrend offer computer science minor degree?

Does behrend offer computer science minor degree?

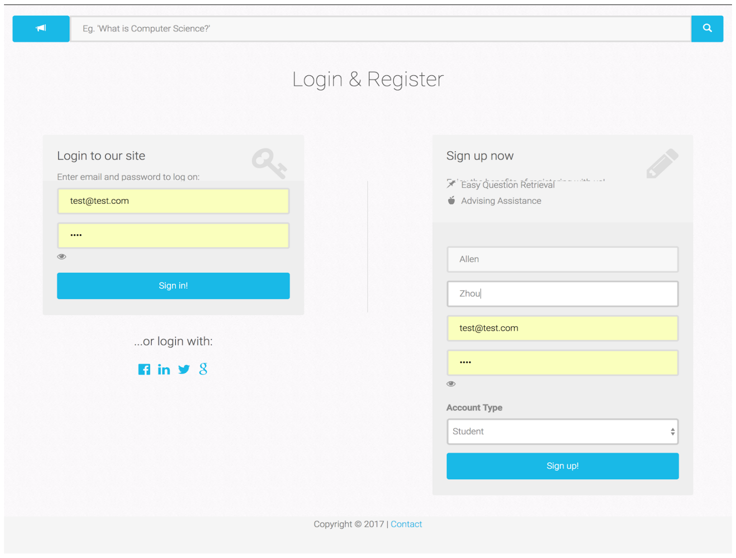
1. Many students are able to receive a mathematical minor without taking on additional classes through careful planning, further helping them understand on a deeper level the role math plays in the computer science field.
Any minor related to science or technology can open additional careers or specializations for computer scientists.
If you are interested in this topic, click here.
☆☆☆☆☆
2. What Courses Should I Take for Mobile Computing? The specific courses you should take are introductory computer science courses as well as data structures and algorithms courses.

https://intelligent-student-advisor.herokuapp.com

I need help with financial aid
Does penn state have soup?
Will i like erie?
Who is meng su?
Should i transfer to up?
What is se
Tell me about behrend.
Does behrend have career services?
What will my salary be as a software engineer?
What is se

Figure 39: Index page with question asked

4. 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.



Eg. 'What is Computer Science?'

Login & Register

Login to our site

Enter email and password to log on:

test@test.com

····

Sign in!

...or login with:

Sign up now

Easy Question Retrieval

Advising Assistance

Allen

Zhou

test@test.com

····

Account Type

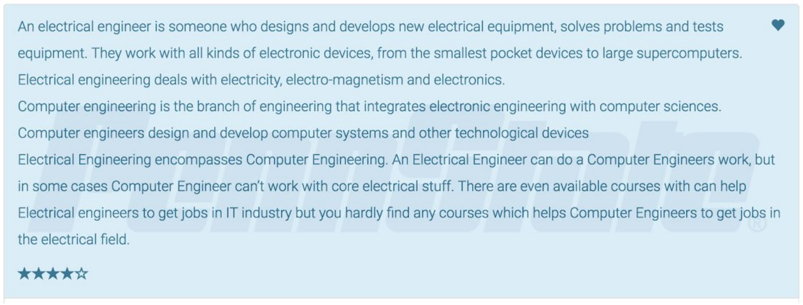
Student

Sign up!

Copyright © 2017 | Contact

Figure 40: Login / Register Page

5. 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

6. 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.

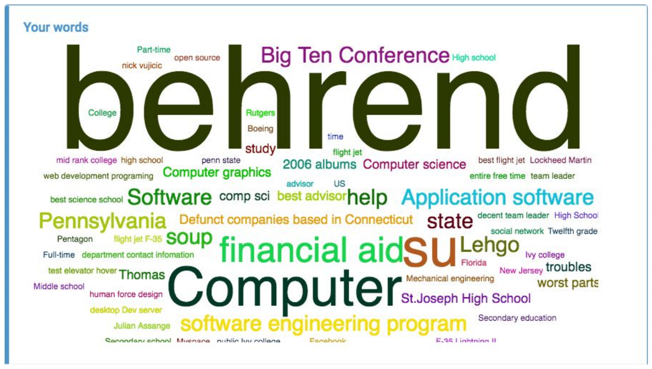
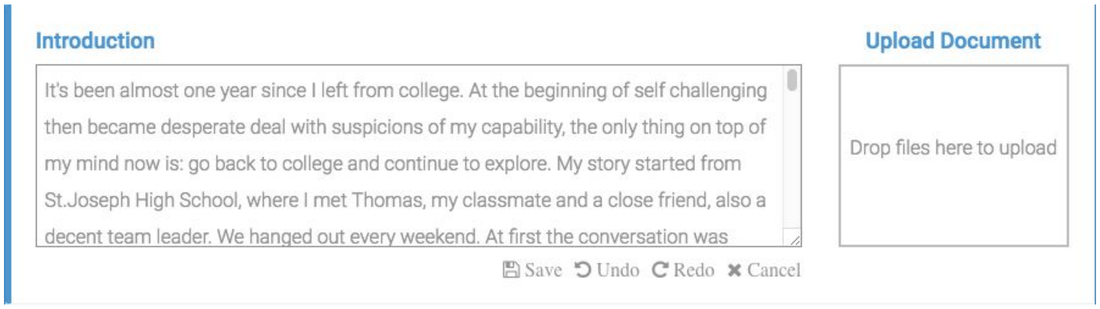


Figure 42: InterestCloud

   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

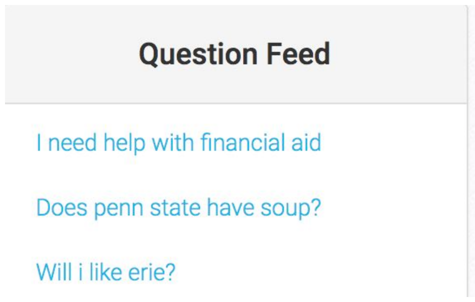
Drop files here to upload

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.



Question Feed

I need help with financial aid

Does penn state have soup?

Will i like erie?

Figure 45: Question Feed

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

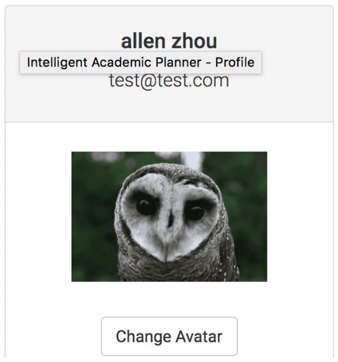


Figure 46: Change avatar

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.

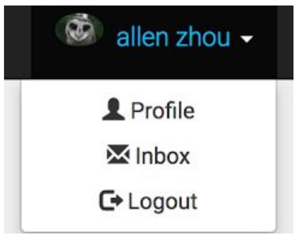


Figure 47: Drop-down menu

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 http://psbehrend.psu.edu/school-of-engineering/academic-programs/computer-engineering (accessed November 11, 2016).
27) Penn State Erie, The Behrend College https://en.wikipedia.org/wiki/Penn_State_Erie,_The_Behrend_College (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/secsdiff.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).