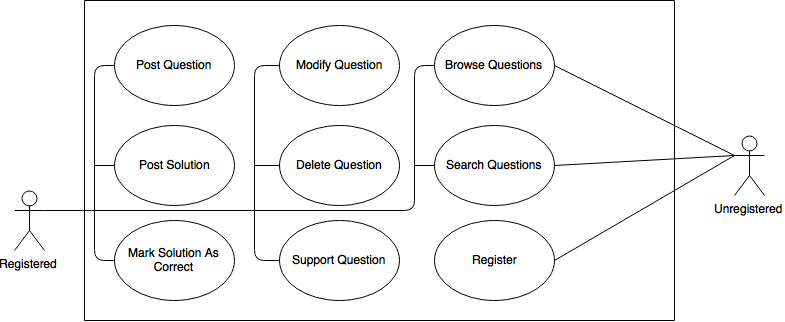
**Assignment 1 Alex Bisbach**

Use Case Diagram:



Use case specification cards:

|  |
| --- |
| Name: Register |
| Actors: Unregistered |
| Trigger: Unregistered user decides to register |
| Preconditions: The user is on the website |
| Post conditions: The user is either registered, or unregistered |
| Success Scenario:   1. The user selects to register 2. The system asks for necessary information 3. The user provides information 4. The system checks for valid information 5. The system creates an account for the user |
| Alternative Flows:   * 1. The information is not valid, in which case the user is sent back to step 2   2. The user already has an account, in which case it logs the user in if password was correct, or takes them to the log in page otherwise |

|  |
| --- |
| Name: Search Questions |
| Actors: Registered, Unregistered |
| Trigger: User decides to search for questions |
| Preconditions: The user is on the website |
| Post conditions: The user is either no longer on the website, or is using another use case |
| Success Scenario:   1. The user selects to search for questions 2. The system asks for the search term(s) 3. The user provides the search term(s) 4. The system searches for and displays questions relevant to the term(s) 5. The user then starts browsing the questions |
| Alternative Flows: None |

|  |
| --- |
| Name: Browse Questions |
| Actors: Registered, Unregistered |
| Trigger: User decides to browse through questions |
| Preconditions: The user is on the website |
| Post conditions: The user is either no longer on the website, or is using another use case |
| Success Scenario:   1. The user selects a criterion for the sorting (unanswered first, answered first, oldest, newest, most supporters, etc) 2. The system sorts the questions by that criteria, and displays the first page of them to the user 3. The user then browses until they decide to enter another use case, or close the page |
| Alternative Flows: None |

|  |
| --- |
| Name: Support Question |
| Actors: Registered |
| Trigger: Registered User decides to support someone else’s question, telling others that they are having the same problem |
| Preconditions: The user is registered |
| Post conditions: The user has a list of questions supported, possibly empty, or the same as before this use case |
| Success Scenario:   1. The user selects that they would like to support the question they are looking at, because they have the same problem 2. The system increases the number of people supporting the question 3. The system adds the question to the users supported questions list |
| Alternative Flows: None |

|  |
| --- |
| Name: Delete Question |
| Actors: Registered |
| Trigger: Registered User decides to delete a question they have previously posted |
| Preconditions: The user is registered and has a posted question |
| Post conditions: The user has either just as many posted questions as before, or one less |
| Success Scenario:   1. The user selects to delete one of their questions 2. The system asks for confirmation 3. The user confirms the action 4. The system deletes the questions and all solutions from its list |
| Alternative Flows:   * 1. The user does not confirm, in which case the use case ends |

|  |
| --- |
| Name: Modify Question |
| Actors: Registered |
| Trigger: Registered User decides to modify one of their questions |
| Preconditions: The user is registered and has a question posted |
| Post conditions: The user has the same number of questions as before the use case |
| Success Scenario:   1. The user selects to modify their question 2. The system displays the question with the previously entered info already entered 3. The user modifies any of the info they would like, or can add more info 4. The user selects to save the question 5. The system updates the question in its list |
| Alternative Flows:   * 1. The user does not provide the required components, such as a title, in which case go back to use case step 2. |

|  |
| --- |
| Name: Mark Solution as Correct |
| Actors: Registered |
| Trigger: Registered User decides to mark a solution as the correct solution |
| Preconditions: The user is registered and the question they are looking at has at least one posted solution |
| Post conditions: The question still exists, either unanswered, or answered |
| Success Scenario:   1. The user selects an answer to mark as correct to their question 2. The system marks the solution as correct 3. The system closes the question to more solutions |
| Alternative Flows: None |

|  |
| --- |
| Name: Solution Question |
| Actors: Registered |
| Trigger: Registered User decides to post a solution |
| Preconditions: The user is registered and is looking at an unanswered question |
| Post conditions: The question has either just as many posted solutions as before, or has one more |
| Success Scenario:   1. The user selects to post a solution 2. The system asks for the details, and any files the user would like to include 3. The user provides the necessary information 4. The user selects to post the solution 5. The system adds the solution to that questions list of solutions |
| Alternative Flows:   * 1. The user does not provide the required components, such as any details, in which case go back to use case step 2. |

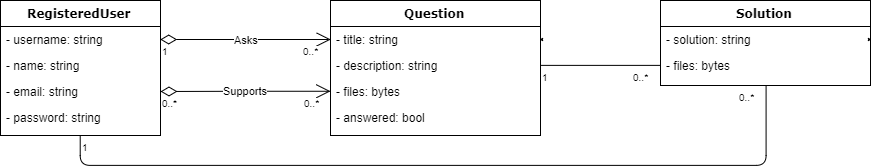
|  |
| --- |
| Name: Post Question |
| Actors: Registered |
| Trigger: Registered User decides to post a question |
| Preconditions: The user is registered |
| Post conditions: The registered user has a possibly empty list of questions with at least as many questions as before the use case |
| Success Scenario:   1. The user selects to post a question 2. The system asks for the title, question, details, and any files the user wants to include 3. The user provides the necessary information 4. The user selects to post the question 5. The system adds the question to its list of questions |
| Alternative Flows:   * 1. The user does not provide the required components, such as a title, in which case go back to use case step 2. |

Noun Extraction:

Suppose you are required to implement a Programming Question-Solution System. This system needs to provide a way for registered users to post programming questions and to provide solutions to questions asked by other users. A user can manage the questions posted by him or herself (modify or delete). Both registered and unregistered users can browse and search any questions and solutions. Registered users should also be able to mark a solution to their question as correct, and support someone else’s question, rather than asking it again.

1. Registered user
2. Question
3. Solution

Initial Class Diagram:



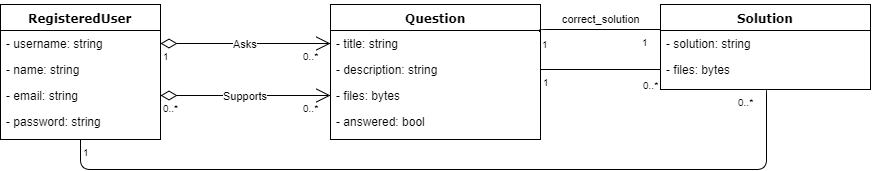
CRC cards:

|  |  |
| --- | --- |
| RegisteredUser | |
| Maintains data about him/herself  Manages a list of questions asked  Manages a list of questions supported  Posts solutions to questions | Question  Solution |

|  |  |
| --- | --- |
| Question | |
| Maintains data about the question  Maintains a list of Solutions  Maintains a correct solution | Solution |

|  |  |
| --- | --- |
| Solution | |
| Maintains information about the solution  Maintains the poster of the solution | RegisteredUser |

Refined Class Diagram:



State Chart:

