Brainstorming and Organizing

Use Cases – everyone picks 3, yellow = complete

1. Create account
2. Log in
3. Log out
4. Reset Password
5. Find recommended courses based off classification year /course
6. Filter search results by reviews/ratings/workloads
7. Look up a specific professor/course (Josh)
   1. Post review of a professor/course
8. Rate a professor (1-5 stars?) (Combined with reviewing professor?)
9. Rate a professor by workload (scale of 1-10)
10. Add professor to “Saved Professors” list
11. Remove professor from “Saved Professors” list
12. Add course to “Saved Courses” list
13. Remove course from “Saved Courses” list
14. “Save Changes?”exit message
15. Post to a discussion board
16. Reply to a discussion
17. Upvote a discussion on the discussion board
18. Flag review (was “request to remove review”)

All Requirements (for Traceability Matrix)

1. The user can create a new account
2. The user can log in and out of their account
3. The user can find recommended courses based off their classification year
4. The user can add a review of a professor
5. The user can search reviews of professors/courses
6. The user can filter search results by workload, rating, etc.
7. The user can request a review to be removed
8. The user can have a list of saved professors
9. The user can have a list of saved courses
10. The system saves or deletes changes upon logout
11. The user can post questions and answers in the discussion board
12. The user can reply to discussions in the discussion board
13. The user can upvote questions in the discussion board

Concepts for domain model

1. Course – attributes could be name, course code, workload
2. Professor – attributes could be name, rating, workload
3. List of Professors
4. List of Courses
5. User – attributes username, email, gender, name, dob, password
6. Review with attributes such as helpfulness, workload

|  |
| --- |
| **ID: UC Create Account** |
| **Scope:** Course/Professor Planning |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system  Database Service - Site database with degree plan info and professor reviews info |
| **Preconditions:**  The user is not logged in |
| **Flow of events:**  1. A new user wants to create an account  2. User clicks on Create Account  3. The system asks for Baylor email  4. The user enters their Baylor email  5. The system sends a verification email  6. The user clicks the link and is brought back to the screen  7. The system asks to enter the password and other information  8. The user enters the information and click submit |
| **Extensions:**  4a. The user does not enter Baylor Email  1. System will display "Please enter Baylor Email"  2. System will return to the start of step three  7a. The entered password does not meet the requirements  1. System will display "Please enter a new password” |
| **Postconditions/Success:**  Account is created successfully |

|  |
| --- |
| **ID: UC Login** |
| **Scope:** Course/Professor Planning |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system  Database Service - Site database with correct login information |
| **Preconditions:**  The user has an account created |
| **Flow of events:**  1. The user clicks on login  2. The system asks for Baylor email and password  3. If the login credentials exist, the user is directed to the home page |
| **Extensions:**  2a. The user enters an email which does not exist  1. System will display "Please enter Baylor Email"  2. System will return to the start of step two  2b. The entered password does not match  1. System will display "Please re-enter”  2. If the user has had 3 tries, the system redirects to reset password |
| **Postconditions/Success:**  User is logged in |

|  |
| --- |
| **ID: UC Logout** |
| **Scope:** Course/Professor Planning |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system  Database Service - Site database with correct login information |
| **Preconditions:**  The user is logged in |
| **Flow of events:**  1. The user clicks on logout  2. The system confirms from the user about logging out and tells the user to save any changes  3. The user confirms  4. The system directs to the home page |
| **Extensions:**  a\* Anytime the system does not respond  1. User will restart the application |
| **Postconditions/Success:**  User is logged out |

|  |
| --- |
| **ID: UC Reset Password** |
| **Scope:** Course/Professor Planning |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system  Database Service - Site database with correct login information |
| **Preconditions:**  The user has an account created and is not logged in |
| **Flow of events:**  1. The user clicks on “Reset Password”  2. The system asks for Baylor email  3. The system link to their email  4. The user clicks the link and is brought back to reset password page  5. The system asks the user to enter the email and new password  6. The user enters and clicks on submit  7. The system updates the information in the database |
| **Extensions:**  2a. The user enters an email which does not exist  1. System will display "Please enter Baylor Email"  2. System will return to the start of step two  4a. The user does not respond to the link in 24 hours  1. The link expires.  5a. The password does not meet the requirements  1. The system asks the user to enter a new password |
| **Postconditions/Success:**  The password is reset. |

|  |
| --- |
|  |

|  |
| --- |
| **ID: UC Professor Lookup** |
| **Scope:** Professor Lookup & Rating System |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system  Database Service - Site database with professor info |
| **Preconditions:**  User must be logged into the website |
| **Flow of events:**   1. User opens the site main page 2. User enters search terms into the search box, and chooses a search mode (by professor name or by class) 3. Site redirects user to a list of all professors matching search terms 4. User selects the desired professor 5. Site redirects user to selected professor’s page |
| **Extensions:**  3a. If no professors match search terms, site offers to search by other search mode |
| **Postconditions/Success:**  User is directed to professor’s page by lookup system  Site displays professor’s reviews and ratings to user |

|  |
| --- |
| **ID: UC Review & Rate Professor** |
| **Scope:** Professor Lookup & Rating System |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system  Database Service - Site database with professor info |
| **Preconditions:**  User must be logged into the website |
| **Flow of events:**   1. User uses professor lookup system to navigate to desired professor’s page 2. Site displays professor’s page 3. User selects “Rate and Review Professor” 4. Site opens popup with text box and rating selection buttons 5. User types review into text box 6. User selects review rating with rating selection buttons 7. Site saves review and returns user to the professor’s page |
| **Postconditions/Success:**  User’s review and rating are stored in the site’s database  User’s review and rating are now visible on the professor’s page on the site |

|  |
| --- |
| **ID: UC Rate Professor Workload** |
| **Scope:** Professor Lookup & Rating System |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system  Database Service - Site database with professor info |
| **Preconditions:**  User must be logged into the website |
| **Flow of events:**   1. User uses professor lookup system to navigate to desired professor’s page 2. Site displays professor’s page 3. User selects “Rate Professor Workload” 4. Site opens popup with rating selection buttons 5. User selects rating with rating selection buttons 6. Site saves rating and returns user to the professor’s page |
| **Postconditions/Success:**  User’s workload rating is stored in the site’s database  User’s workload rating is now visible on the professor’s page on the site |

|  |
| --- |
| **ID: UC Save Professor** |
| **Scope:** Professor Lookup and List of saved professors |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system |
| **Preconditions:**  User is logged into the website, and has successfully looked up a list of professors |
| **Flow of events:**   1. The user selects a professor from the search results 2. The system displays a page with that professor’s info 3. The user selects the “Save Professor” icon on the professor’s window 4. If the professor is not currently stored in the user’s “Saved Professors” list, then    1. The system displays a notification that the professor has been added to the end of the user’s list 5. Else,    1. The system displays a notification that the professor has already been saved to the list |
| **Postconditions/Success:**  The user’s “Saved Professors” list has one more professor at the end of the list, if it is not already in the list |

|  |
| --- |
| **ID: UC Remove Professor** |
| **Scope:** List of saved professors and Professor Lookup |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system |
| **Preconditions:**  User must be logged into the website |
| **Flow of events:**   1. The user opens up their “Saved Professors” list 2. The system displays the user’s list 3. If their list is not empty, then    1. The user selects the “Remove Professor” icon next to a professor in their list    2. The system removes that professor from the user’s list and displays the updated list to the user |
| **Postconditions/Success:**  The user’s “Saved Professors” list has one less professor, or remains empty if originally empty |

|  |
| --- |
| **ID: UC Save Course** |
| **Scope:** Course Lookup and List of saved courses |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system |
| **Preconditions:**  User is logged into the website, and has successfully looked up a list of courses |
| **Flow of events:**   1. The user selects a course from the search results 2. The system displays a page with that course’s info 3. The user selects the “Save Course” icon on the course's window 4. If the course is not currently in the user’s “Saved Courses” list, then    1. The system displays a notification that the professor has been added to the end of the user’s list 5. Else,    1. The system displays a notification that the course has already been saved to the list |
| **Postconditions/Success:**  The user’s “Saved Courses” list has one more course at the end of the list, if it is not already in the list |

|  |
| --- |
| **ID: UC Remove Course** |
| **Scope:** List of saved courses and Course Lookup |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system |
| **Preconditions:**  User must be logged into the website |
| **Flow of events:**   1. The user opens up their “Saved Courses” list 2. The system displays the user’s list 3. If their list is not empty, then    1. The user selects the “Remove Course” icon next to a course in their list    2. The system removes that course from the user’s list and displays the updated list to the user |
| **Postconditions/Success:**  The user’s “Saved Courses” list has one less course, or remains empty if originally empty |

|  |
| --- |
| **ID: UC Filter Results** |
| **Scope:** Professor Reviews |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system  Database Service - Site database with reviews |
| **Preconditions:**  The user is on the search professor page of the site |
| **Flow of events:**   1. User selects a professor’s name from dropdown menu to search for reviews 2. Site displays the reviews of the given professor 3. If user wants to filter results the user can select a filter from the dropdown menu 4. The system filters the results, and the site displays filtered info |
| **Postconditions/Success:**  Site displays the reviews with the given input from the user |

|  |
| --- |
| **ID: UC Recommendations** |
| **Scope:** Course/Professor Planning |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system  Database Service - Site database with degree plan info and professor reviews info |
| **Preconditions:**  The user is on the planning page of the site |
| **Flow of events:**   1. User selects type of recommendation 2. System records the type of rec selected 3. If user choses course rec    1. User enters the semester they are in    2. Displays list of courses with professors 4. If user chooses prof rec    1. User enters the course they are in    2. Displays list of top rated profs for the selected course |
| **Postconditions/Success:**  Site displays recommended courses along with the associated professors |

|  |
| --- |
| **ID: UC Flag Review** |
| **Scope:** Professor reviews |
| **Level:** User Goal |
| **Actors:**  User - Student interacting with system  Database Service - Site database with reviews |
| **Preconditions:**  The user has searched for professor reviews from the search professor page of the site |
| **Flow of events:**   1. The user selects a review from the professor review search results 2. The system displays a page with the selected review 3. The user selects the “Flag Review” icon on the review page 4. The system updates the flag icon to reflect that the review has been reported to admins |
| **Postconditions/Success:**  The review is marked as flagged and is reported to the admins |