Group 30

FDM Mentor Matching App

ECS506U Software Engineering Group Project

Requirements Elicitation Report

1. List of Requirements

Accounts

ID	Requirement	Type	Priority	Use Case
1	Each customer must be identified with a unique username.	Functional (D)	Core	Register
2	Customers must be able to login with a username and password.	Functional (D)	Core	Login
3	The system must have two types of users: customers and mentors.	Functional	Core	Register
4	The app could enable mentees and mentors to customise their profiles	Functional	Core	Edit Profile
5	FDM Employees, consultants and alumni can become mentors	Functional	Core	Register
6	Guest users must be able to register an account	Functional (D)	Core	Register
7	Profiles include users' education	Functional(D)	Core	Register
8	Profiles include users' work experience	Functional(D)	Core	Register
9	Profiles include users' field	Functional(D)	Core	Register
10	Users should have the ability to choose the visibility of their profile and ensure privacy	Functional	Optional	-
11	User fill in their requirements for mentor/mentee	Functional (D)	Core	Register
12	Mentees able to see mentor credentials and experience	Functional	Core	View Profiles
13	Guest users could be able to browse limited pages of the app	Functional	Optional	Browse limited pages
14	Profiles could store the number of reports/complaints received	Functional	Optional	View profiles

Matching

ID	Requirement	Type	Priority	Use Case
15	The app should have a matching algorithm that matches mentees to mentors and shows users the matches in a list.	Functional	Core	-
16	Alumni, consultants and employees should be matched up with ex-military, graduates and returners to work for reverse mentoring	Functional	Core	-
17	Administrators could be able to create mentoring opportunities	Functional(D)	Optional	Create Mentorship Opportunities
18	Administrators shall be able to edit mentorship opportunities	Functional (D)	Core	Edit mentorship opportunities
19	Administrators shall be able to delete mentorship opportunities	Functional (D)	Core	Delete mentorship opportunities
20	Users are able to filter search based on various criteria such as location.	Functional	Optional	-
21	The app could display how compatible a mentor is to a user in percentages	Functional	Optional	View profiles/find match

Mentor/Mentee Relationship

ID	Requirement	Type	Priority	Use Case
22	The app should provide a platform for mentees and mentors to communicate with each other via messaging.	Functional	Core	Use chat room
23	The app could allow mentees and mentors to set goals for their mentoring relationship and track their progress towards those goals.	Functional	Optional	Set goals
24	The app could allow mentees to track their progress towards goals set by mentors	Functional	Optional	Track goals
25	The app could provide a platform for mentees to provide feedback and evaluate their mentor, and vice versa.	Functional	Optional	Provide Feedback on Mentor/Mentee
26	The app could have a resource library of documents where mentors can share articles, tools, and other resources with their mentees.	Functional	Optional	Use chat room
27	Administrators could be able to track the progress of mentoring relationships.	Functional (D)	Optional	Track goals
28	Mentors and Mentees are not able to message or view other user profiles while they are matched up	Functional	Core	-
29	Mentor and Mentee can end mentorship relationship	Functional	Core	End mentor/mentee Relationship
30	Mentors able to create programs with specific curriculum	Functional	Optional	Create Curriculum
31	The Learning and development team can create forums for more specific and wider groups	Functional	Optional	-
32	Users should be able to block or report other users	Functional	Optional	-

Non-Functional Requirements

ID	Requirement	Type	Priority	Use Case
33	The FDM Mentor Matching App could be accessible 99% of the time in a 24 hour period	Non-Functional	Optional	-
34	The FDM Mentor Matching App could be accessible 99.5% of the time in a 24 hour period	Non-Functional	Optional	-
35	The FDM Mentor Matching App could be accessible 99.99% of the time in a 24 hour period	Non-Functional	Optional	-
36	The FDM Mentor Matching App should display correctly on the latest version of Google Chrome, Safari and Firefox.	Non-Functional	Core	-
37	The app should be secure and protect user data from unauthorised access and ensure confidentiality.	Non-Functional	Optional	-
38	The app should be able to handle a large number of users and match them effectively across a global scale.	Non-Functional	Optional	-
39	The app should be accessible to users with disabilities and comply with accessibility guidelines.	Non-Functional	Optional	-
40	The app should be user-friendly and easy to navigate for both mentees and mentors.	Non-Functional	Core	-
41	The app should have a vigorous backup of data and recovery plan to ensure data is not lost	Non-Functional	Optional	-
42	The app should be able to adapt to different screen resolutions	Non-Functional	Core	-

43	The app shall be accessible via browser, regardless of what operating system is used	Non-Functional	Core	-
44	The app should be easy to maintain through regular updates, backup and testing	Non-Functional	Core	-

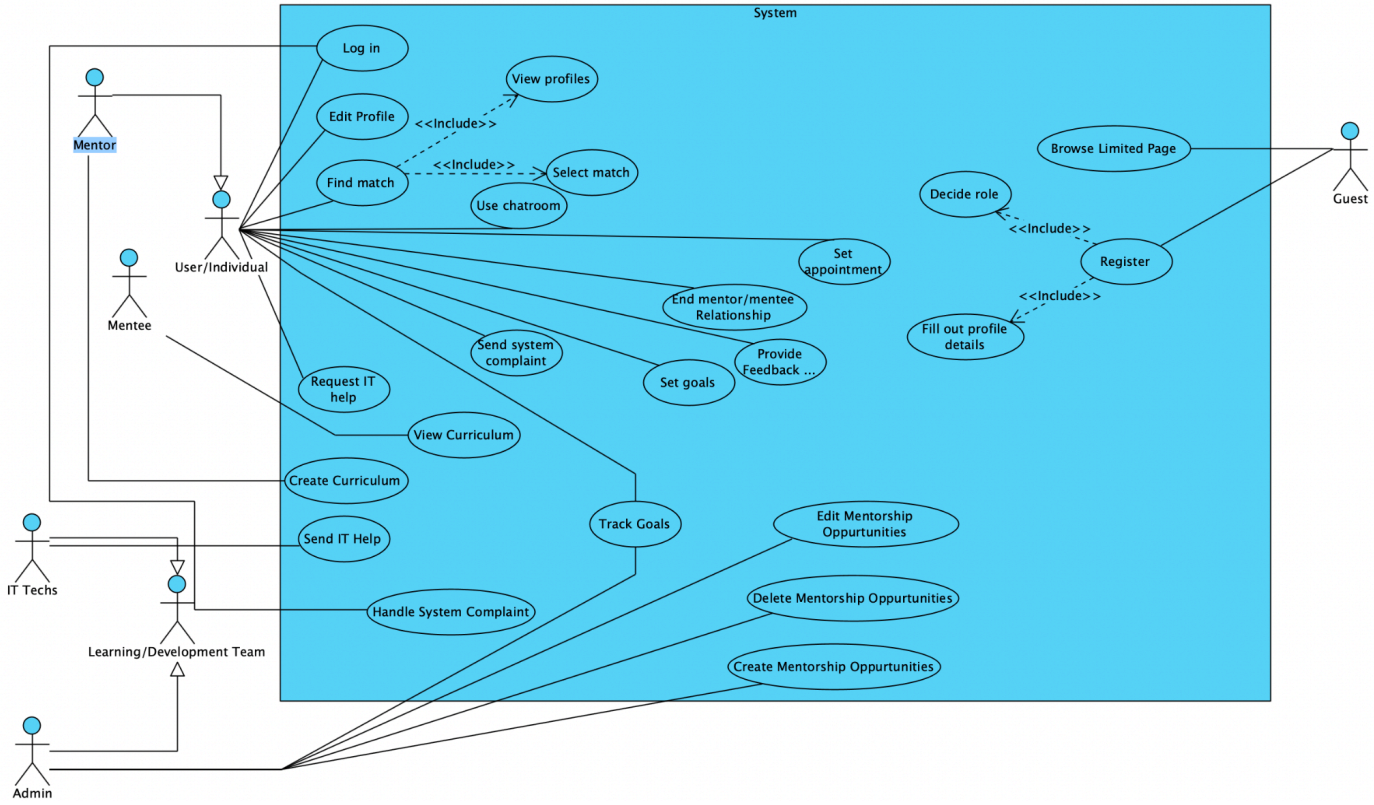
Help and FAQ

ID	Requirement	Type	Priority	Use Case
45	Mentors and Mentees could be able to send complaints	Functional (D)	Optional	Send system complaint
46	Administrators and IT Technicians could be able to handle complaints.	Functional (D)	Optional	Handle system complaint
47	Users should be able to provide feedback through a survey function to help improve the app	Functional	Optional	-
48	Users are able to access help and FAQ section for additional support	Functional	Optional	-

User Experience

ID	Requirement	Type	Priority	Use Case
49	The app could notify mentees and mentors of upcoming events.	Functional	Optional	-
50	The app could support multiple languages to accommodate FDM employees from different parts of the world.	Functional	Optional	-
51	Users are able to sync appointments to personal calendars	Functional	Optional	Set appointment
53	The app could notify mentees and mentors of messages received	Functional	Optional	-
54	The app could notify mentees and mentors of program updates	Functional	Optional	-
55	The app could notify mentees and mentors of relevant news.	Functional	Optional	-

2. Use-Case Diagram



3. Use Case Description

Register

1. **Brief Description:**
    a. This use case describes how a user registers for the application
2. **Actors:**
    a. Guest
3. **Preconditions:**
    a. There is an active internet connection
4. **Basic Flow:**
    a. The use case begins when the user chooses to register for the application
    b. The user is asked to create a username and password
    c. The user decides to be a mentor or mentee
    d. The user fills in their field
    e. The user fills in their education
    f. The user fills in their work experience
    g. The user fills in their requirements for the mentor/mentee
    h. The profile is filled in
    i. The account is created
    j. Use case ends successfully
5. **Alternate Flows:**
    a. Username/password taken
        i. In step 4.b, if the username is taken, then the user will be asked to use a different username.

b. Password isn't strong enough
    i. In step 4.b, if the password isn't strong enough, then the user will be asked to create a stronger password that fills the requirements.
c. One of the fields is left blank
    i. In step 4.d to 4.g, if the user fails to fill in the given fields, the use case will end in a failure condition.

6. **Key Scenario:**
    a. N/A
7. **Post Conditions:**
    a. Successful Completion:
        i. The user has created an account and can proceed with the application
    b. Failure Conditions:
        i. The user has failed to create an account

Find Match

1. **Brief Description:**
    a. The use case describes how a mentor/mentee find a match
2. **Actors:**
    a. Users/Individuals
3. **Preconditions:**
    a. The user has an account
4. **Basic Flow:**
    a. The use case begins when the user clicks "find match"
    b. The user will then be shown a list of mentors/mentees that match their requirements
    c. The user can view their profiles in detail by clicking on them
    d. They can choose to match with the mentor/mentee or go back to viewing profiles
    e. If the user wants to match with a mentor/mentee, they can send a match request to the mentor/mentee.
    f. The mentor/mentee that receives the request gets to view the senders profile.
    g. The mentor/mentee can then accept the request.
    h. The use case ends successfully.
5. **Alternate Flows:**
    a. Mentor/mentee doesn't like the matches they have
        i. In the case that the mentor/mentee doesn't like any of the given mentor/mentee profiles, they can choose to either edit their details in more detail, and proceed from step 4.a
    b. Mentor/mentee ignores request
        i. In the case that the mentor/mentee ignores the request, the use case ends in a failure condition
6. **Key Scenario:**
    a. Mentor/Mentee don't find match that they like
7. **Post Conditions:**
    a. Successful Completion:
        i. Mentor/mentee finds match
    b. Failure Conditions:
        i. Mentor/mentee can't find match

Set appointment

1. **Brief Description:**
    a. This use case describes how the mentor/mentee can set appointments with each other for a learning session.
2. **Actors:**
    a. Users/Individuals
3. **Preconditions:**
    a. Both mentor and mentee are matched
4. **Basic Flow:**
    a. The use case begins when either mentor or mentee begin using the chat room.
    b. The user receiving the message will receive a notification that they've been communicated.
    c. The user will then respond and discuss details with the mentor/mentee
    d. The user can then set an appointment for a session between mentor/mentee
    e. The system will then save the date and time for the appointment.
    f. The system will then send a notification on the time of the appointment
    g. The use case ends successfully
5. **Alternate Flows:**
    a. The user isn't online
    b. If the user isn't online then the system will send a notification to them and continue from step 4.c
6. **Key Scenario:**
    a. Either Mentor or Mentee aren't online at the moment
7. **Post Conditions:**
    a. Successful Completion:
        i. Mentor and mentee have successfully made an appointment for their session
    b. Failure Conditions:
        i. Mentor and mentee fail to create an appointment for their session

Curriculums

1. **Brief Description:**
    a. This use case describes how the mentor/mentee create and view curriculums
2. **Actors:**
    a. Users/Individuals
3. **Preconditions:**
    a. Both mentor and mentee are matched
4. **Basic Flow:**
    a. The use case begins when either mentor or mentee begin using the chat room.
    b. The user receiving the message will receive a notification that they've been communicated.
    c. The user will then respond and discuss details with the mentor/mentee
    d. The mentor can then describe to the mentee the curriculum.
    e. The mentee can then accept the curriculum
    f. Both mentor and mentee put the goals of the curriculum into "set goals", so that it can be tracked later on.
    g. The use case ends successfully

5. **Alternate Flows:**
    a. The mentee doesn't like the curriculum:
        i. In the case that the mentee doesn't like the given curriculum, they can discuss it with the mentor, until an agreement is reached, and then proceed from step 4.e.
        ii. If the mentee doesn't reach an agreement with the mentor, then the use case will end in a failure condition
6. **Key Scenario:**
    a. Mentee doesn't like curriculum
7. **Post Conditions:**
    a. **Successful Completion:**
        i. Mentor and mentee have successfully agreed on a curriculum.
    b. **Failure Conditions:**
        i. Mentor and mentee fail to agree on curriculum, and most likely end up "ending relationship"

4. Risk Assessment

We have identified and tried to assess the threat of risks to our project at different stages of its completion. All of these risks consider risks caused by algorithmic decisions, management decisions or decisions on how the application will operate. These risks include events where the application could potentially fail to operate as intended in the future and are considered to be poor design decisions which result in a low quality product and a failed project. Whilst some risks are unlikely, they have been considered so that our group can work to include the preventative designs where possible.

4.1 Risks when developing the application

Risk	Likelihood	Severity	Impact	Preventative / Mitigating Actions
Poor Time Management	High	High	Unfinished product	Evaluate progress frequently and assign deadlines for task completion
Missing Member(s) due to Health Issues	High	Medium	Unfinished / low quality product	Evaluate progress frequently and adjust task distribution where possible
Underestimating Steps or Layers to a Process	Medium	High	Low Quality Product: Algorithm lacks detail	Create a list that details the subtasks involved in making the procedure work
Overestimating Steps or Layers to a Process	Medium	High	Low Quality Product: Algorithm becomes too complex	Create a list that details the subtasks involved in making the procedure work
Poor Communication	Medium	High	Unfinished / low quality product with clashing or non-complementary ideas or procedures	Schedule regular progress checks and provide detailed updates of tasks completed or changes made

4.2 Risks when managing the application

Risk	Likelihood	Severity	Impact	Preventative / Mitigating Actions
Stakeholder Conflict over Proposed Changes	High	High	Low Quality Product: Slow or no progress made when changes are needed	Come to an agreement by considering benefits of each idea and voting for the optimal choice
Ambiguity of Change Requests between Administrators and IT Technicians	Medium	High	Low Quality Product: Miscommunication causing wrong updates to be made that clash with requirements or design	Make note of change requests with any necessary given details and have them verified before execution
Change Requests Conflict with Requirements - Project scope is limited	Low	Medium	Low Quality Product: Difficulty in embedding change into design	Consider more suitable changes before deciding to diverge from requirements

Source Code Difficult to Understand and Adapt - Project did not account for flexibility	Medium	Medium	Low Quality Product: Difficulty in changing code for updates perhaps due to poor design or due to changes in future (e.g. new dominating operating systems or languages)	Flexible and clear code, commented and sectioned per part, with an easy to navigate design - larger future changes cannot be taken into consideration as of now
High Frequency of Intervention due to Failed Algorithm	Medium	High	Low Quality Product: Need for manual matching by Learning and Development Team despite application being created	Ensure algorithm is objective and creates optimal matches and work to avoid risks mentioned in 4.3.2

4.3 Risks when using the application

4.3.1 User-side Risks

Risk	Likelihood	Severity	Impact	Preventative / Mitigating Actions
Falsified Data - Project did not account for poor data input	Low	Medium	Low Quality Product: Algorithm would not be affected and would still create optimal matches but matches are less likely to be accurate	Minimise scope of error by limiting data input (drop down information where possible or create multiple choice quiz style form for matching)
Falsified Verification - Project did not account for fake identification	Low	High	Low Quality Product: Information Breaches or if done in an attempt to access application then other security breaches or risks are possible	Verifying mentors (consultants, alumni or employees) can be checked against the database and given details checked for safety. Verifying mentees (ex-forces, returners to work and graduates) show id and proof of each situation and provide a reference.
Human Error - Project did not account for ways to minimise human error	High	High	Low Quality Product: Accidentally inputs revealing information, contacts someone, makes match by miss-clicking or objective algorithm is disrupted and fails	Allow users to change profile, always provide a confirmation page where they check the input details are accurate and allow matches to be withdrawn and re-selected
Difficulty to Navigate	Medium	Medium	Low Quality Product: Users would not wish to use the application and request matches to be made manually thus the designed application fails to meet its purpose	Format application clearly and avoid adding too much complexity to design
Information Security	Medium	High	Low Quality Product: User	Increase levels of

Breach			data is leaked or has been compromised	authentication (password verification, email and phone verification - 2fa) required for login and registration

4.3.2 Algorithm-side Risks

Risk	Likelihood	Severity	Impact	Preventative / Mitigating Actions
Too Few Users - Project did not set a limit on how many mentors or mentees needed for matches to be optimal	Low	Medium	Low Quality Product: Although algorithm makes optimal matches, if there are only 1 or 2 mentors or mentees then it would match them even if their profiles are not similar	Create a minimum limit for matching algorithm that ensures matching begins after a certain number of users are involved - for small numbers, manual matching is better and less time consuming
Not Enough Information Provided - Project did not set limit on how much information is needed for matches to be optimal	Medium	Medium	Low Quality Product: Profile data provided would have optional input boxes so whilst some users may provide a lot of detail, others may not - the algorithm would either match users because the limited information overlaps but depending on what information this is, matches made will not be optimal	Understand and split matching profile over multiple categories that are considered - include compulsory input boxes (preferably with drop down option) so the algorithm can make matches even if users choose an option “n/a” for some data
No Overlap of given Data - Project did not account for lack of information overlap that could make the matches less credible	Medium	Medium	Low Quality Product: The algorithm attempts to make optimal matches but if there is no overlap in data then matches are quite random and the algorithm would fail to provide an optimal selection designed for each user	Algorithm still makes matches despite no overlap but should create a notice when listing matches that there was no or very little overlap
Objective Algorithm Out-dated	Medium	Medium	Low Quality Product: Algorithm would not consider new important criteria considered in future (e.g. new hobbies need to be considered or new entry level experience requirements etc)	Perhaps create a database of options and add to it every time a new detail needs to be considered. This would be slightly time consuming to create at first but adding details would be quicker after it is complete
User Inactivity - Project did not consider what would occur if accounts are inactive	Low	Medium	Low Quality Product: Algorithm would keep suggesting matches with inactive accounts if we have not specified a rule to ignore matching with accounts that	Ensure users deactivate account after programme is complete and create a time limit for inactivity (after a number of days, the algorithm does not consider

			have been inactive for a certain number of days	the user until there is some activity detected again) - also provide an option for users to say they will be inactive
Information Security Breach	Medium	High	Low Quality Product: User database is leaked or compromised due to poor design security	Increase levels of authentication (password, email verification or phone verification - 2fa) for administrators and IT technicians