MentorWeb Request Analysis Document (RAD)

Joseph Richardson, Daniel Trebe,
Silas McCroskey, Halin Gordon, Clifford Chan

March 5, 2014

Prepared for SE/CMPE 133: Software Engineering II, San Jose State
Spring 2014

Contents

0 Revision History

Revision	Originator	Date	Comments
1	HammerSmash	2014-02-17	Initial Revision
2	HammerSmash	2014-02-20	Adding in requirements
3	HammerSmash	2014-02-24	Adding Use Cases
4	HammerSmash	2014-03-02	Adding Sequence Diagrams, use case models, and non-fucntional requirements
5	HammerSmash	2014-03-04	Moved into LaTeX file, added Definitions, Acronyms, and Abbreviations section

1 Introduction

1.1 Purpose

MentorWeb is a social media web-based application for working professionals. MentorWeb provides a way for working professionals to develop a Mentorand Mentee relationship. Any user can be a mentorand a mentee at anytime. Users can follow mentors based on what the user's goals are matched to the mentors background.

1.2 Scope

MentorWeb attempts to establish a stronger network for working professionals.

MentorWeb will create a mentorand mentee relationship by allowing users who choose to sign up as a mentee to follow mentors.

MentorWeb will suggest certain mentors for each mentee by matching the mentee's aspirations to the mentor's background.

Each mentee will be able to communicate to their mentors in a variety of ways without requiring additional tools.

1.3 Objectives and Success Criteria

MentorWeb users will have a stronger professional network and professional foundation.

1.4 Definitions, Acronyms, and Abbreviations

Term	Definition
Mentor	A professional with industry experience who volunteers to advise a mentee.
Mentor	A student or worker new to the industry who seeks advising from a mentor.
Sponsor	A professional with industry experience who not only advises a Mentee, but also is involved in their career through sponsorship means, giving opportunities more directly
MentorWeb	A web-based system which aims to facilitate the mentor-mentee relationships described above.
The Program	Used in this document where not otherwise obvious: the program MentorWeb.
Shall	When used in a requirement text, signifies that the program in question is incomplete or incorrect when not in compliance with this requirement.
Should	When used in a requirement text, signifies a desirable property of the target system, but one without which the system can still be considered complete and correct.
API	Application Programming Interface

1.5 References

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications

1.6 Overview

MentorWeb attempts to allow the user to establish a stronger professional network through facilitating mentor-mentee relationships.

2 Current System

MentorWeb as a system has yet to be implemented; this document covers a new system which does not replace an existing one.

3 Proposed System

This section describes the requirements and specifications of MentorWeb.

3.1 Overview

MentorWeb attempts to allow the user to establish a stronger professional network through a mentor-mentee relationship.

3.2 Functional Requirements

ID	Description	Priority
3.2.0.1	Users shall be able to decide to keep their conversations off of server storage	EF 1
3.2.0.2	MentorWeb shall allow sponsors with other companies	EF 2
3.2.0.3	Administrative users shall be able to suspend users for abusing MentorWeb	EF 3
3.2.0.4	In the event the user opts to deactivate his or her account, MentorWeb shall support this, but shall not delete user data to provide for the case that the user opts to re-register	EF 4
3.2.0.5	MentorWeb shall be a web-based project, and keep an open API for expandability	EF 5
3.2.0.6	MentorWeb should be designed to easily migrate to a mobile application.	DF 1
3.2.1.1	MentorWeb shall have a login process and a registration process for users	EF 6
3.2.1.2	MentorWeb shall allow login with the social media outlets Facebook and LinkedIn	EF 7
3.2.2.1	MentorWeb users shall be able to registor as just a mentor, just a mentee, or both	EF 8
3.2.3.1	MentorWeb users shall be able to input and display their professional information (background, aspirations, skills), and/or pull such information from an existing LinkedIn profile	EF 9
3.2.3.2	MentorWeb shall store converations not taken offline for one year	EF 10

3.2.3.3	Users shall have membership periods for mentor-mentee relationships that can be extended	EF 11
3.2.3.4	MentorWeb users should be able to block communication from other registered users	DF 2
3.2.3.5	MentorWeb users shall be able to make some of their information private	EF 12
3.2.4.1	MentorWeb shall display information about number of mentors currently online, total number of registered mentors, how many pairs of mentor-mentee exist, how often conversations occur, and duration of membership	DF 3
3.2.4.2	MentorWeb should have a news channel on users' start page displaying current articles related to the user's interests	DF 4
3.2.4.3	MentorWeb shall have a mentor rating system	EF 13
3.2.4.4	MentorWeb shall include a comments section for other mentors or mentees to be able to leave comments on their effectiveness, etc.	EF 14
3.2.4.5	MentorWeb shall show a history of past relationships of users and their mentors	EF 15
3.2.4.6	MentorWeb mentors can have several mentees and vice-versa	EF 16
3.2.5.1	MentorWeb shall allow searching for mentors or mentees from user home page	EF 17
3.2.5.2	MentorWeb shall send users email notifications when making requests to be a mentee for any mentor	EF 18
3.2.5.3	MentorWeb shall not force members to be in a mentor-mentee relationship; it is up to the mentor and mentee involved to both agree on the relationship	EF 19
3.2.6.1	MentorWeb shall run on target browsers Firefox and Internet Explorer	EF 20
3.2.7.1	MentorWeb shall allow for various online communication methods, such as appointment systems (calendars), chat, messaging, and/or audio/video calls	EF 21

3.2.7.2	MentorWeb shall have communication methods embedded into the website	EF 22
3.2.8.1	MentorWeb shall match mentees with mentors based on server-side cloud code matching mentee aspirations with mentor background and/or match mentee background with mentor background	EF 23

3.3 Nonfunctional Requirements

3.3.1 Usability

There should be a help feature and options to guide new users through usage of the system

3.3.2 Reliability

Website should be reliable.

Website must display correctly in the target web browsers. (see 3.2.6.1)

3.3.3 Performance

Must provide enough internet bandwidth. There should be a balance of speed and cost.

The functions of the website should run quickly.

Any displayed information should be updated within a reasonable amount of time (no more than 1 minute lag)

Website should be able to scale to accomodate a large user base

3.3.4 Supportability

Guided troubleshooting of common errors should be included in the software system so users do not have to complain to get a problem fixed.

Event logging

Complete documentation of how to use the software

System monitoring (network, memory usage)

3.3.5 Implementation

Internet access is required

Software must be secure (safe from being broken by hackers or incorrect inputs)

User input into the system should be validated and sanitized to ensure security and avoid injections

3.3.6 Interface

Website should be aesthetically pleasing and look professional

Users should not have to search for a button or be confused as to how to use a function

Placement of buttons should be obvious for users

3.3.7 Packaging

Include an easy one-click interface to install software

Include a cloud option ot access the software but make sure user has necessary drivers and software to run our services (automatic check)

3.3.8 Legal

Must avoid copyright infringement of any kind

Avoid offending users, all content should be checked and approved

Software should not break any law or enable users to break any laws

Privacy of user information marked private should be maintained

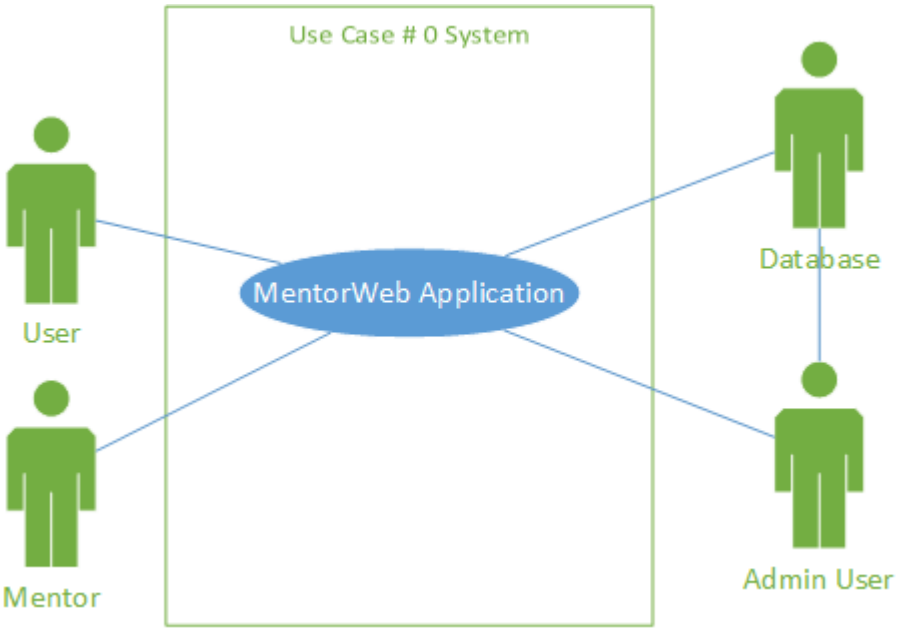
User must accept a legal agreement before using the software

3.4 System Models

This section displays several models that describe various aspects of the system to be implemented.

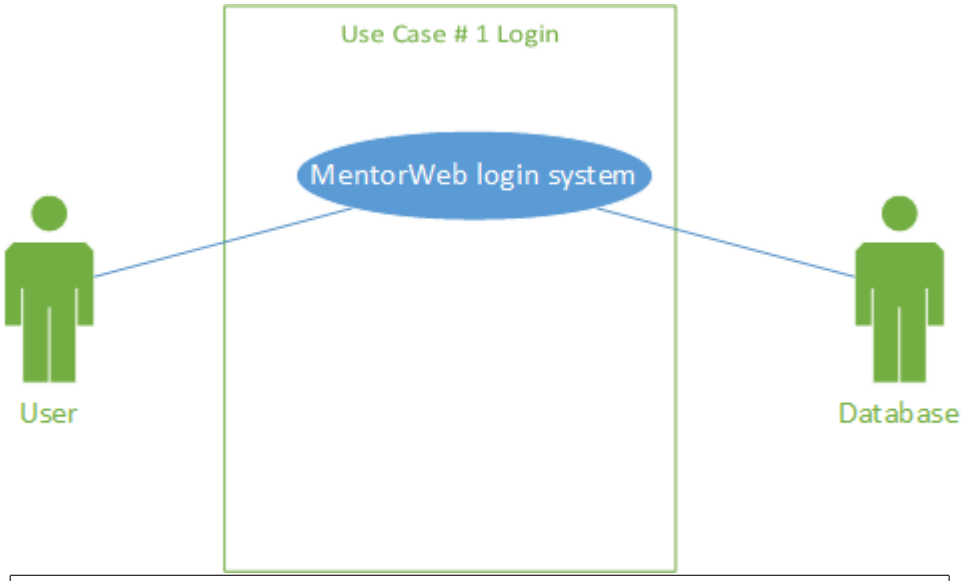
3.4.1 Scenarios

3.4.2 Use Case Model



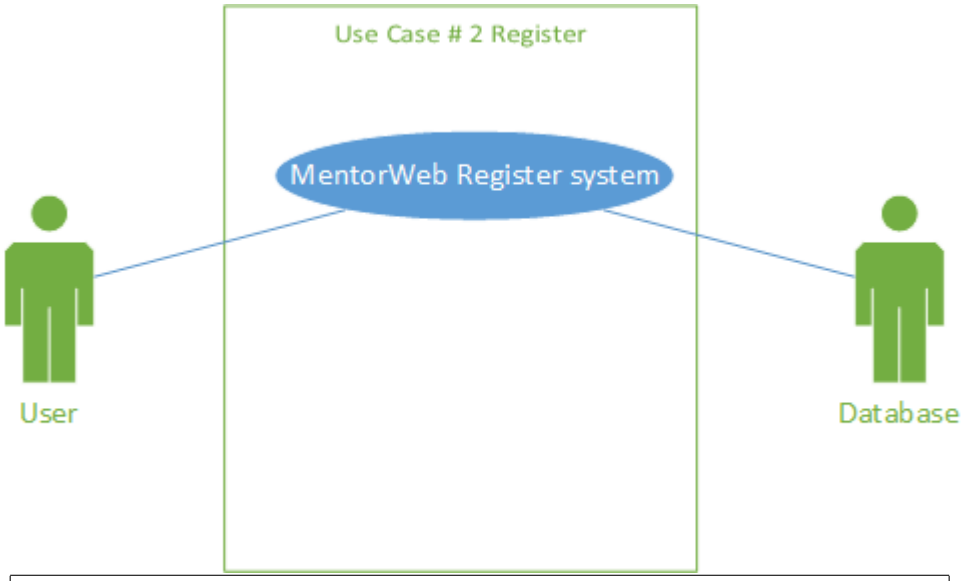
Name: System
Actors: User, Mentor, Database, Admin User
Entry Condition: User turns on MentorWeb System
Flow of Events: 1. User brings up MentorWeb in their browser 2. User logs in (see Use Case 1)
Exit condition: User leaves MentorWeb site
Exceptions: Lost internet connection

Figure 1: Use Case Diagram for Use Case 0: System



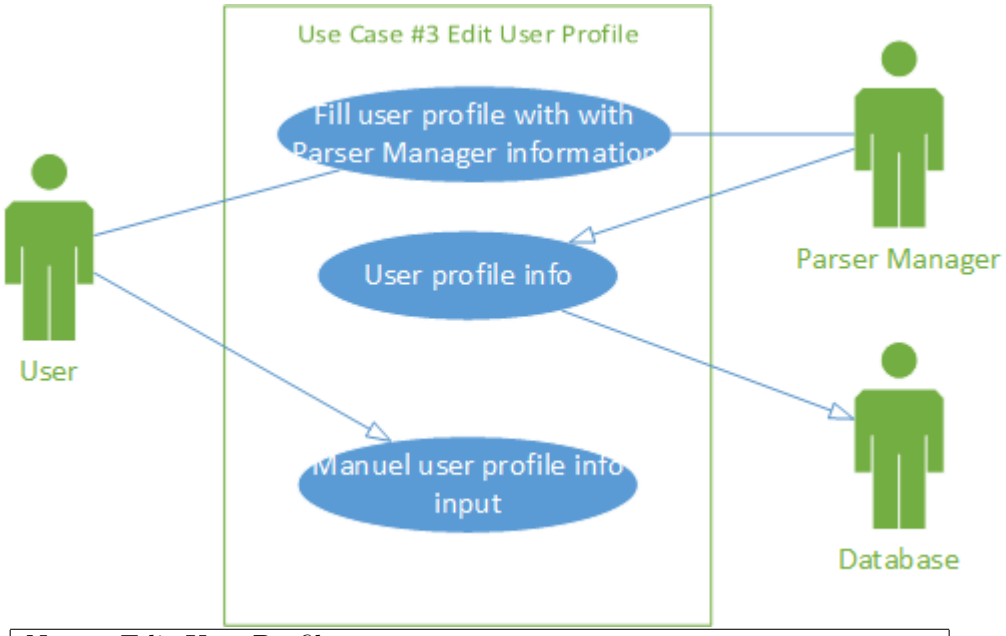
Name: Login
Actors: User, Database
Entry Condition: User has established connection with MentorWeb system
Flow of Events: 1. User attempts to log into MentorWeb 2. User either successfully logs in or user credentials do not work
Exit condition: User successfully logs in
Exceptions: User has no account and must register (see Use Case 2)

Figure 2: Use Case Diagram for Use Case 1: Login



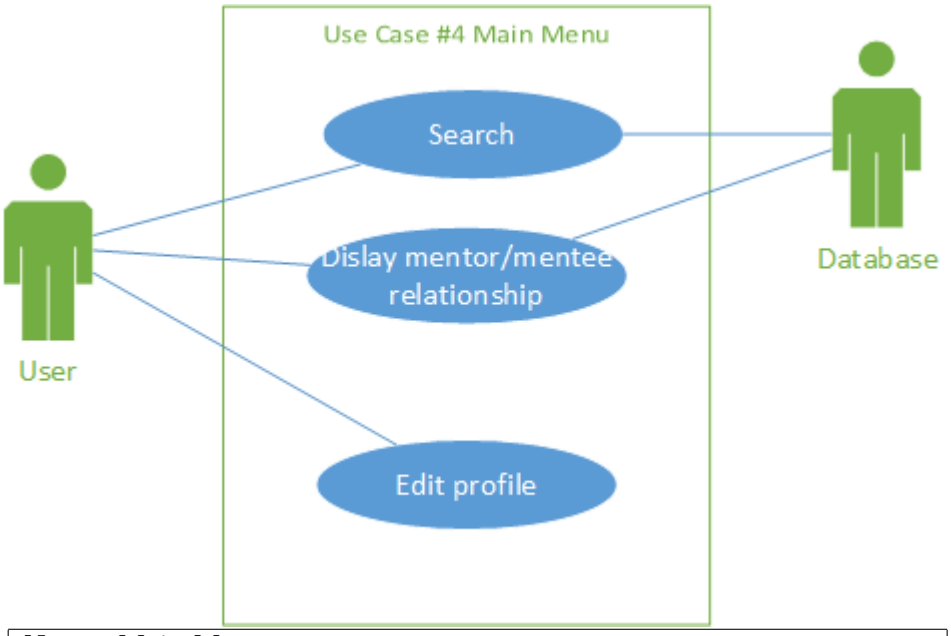
Name: Register
Actors: User, Database
Entry Condition: User is connected to, but does not have an account with, the MentorWeb system
Flow of Events: 1. User enters in a valid email address to which a validation email is sent 2. User creates a secure password 3. User enters validation code from validation email
Exit condition: User has successfully created a new account
Exceptions: Account already exists

Figure 3: Use Case Diagram for Use Case 2: Register



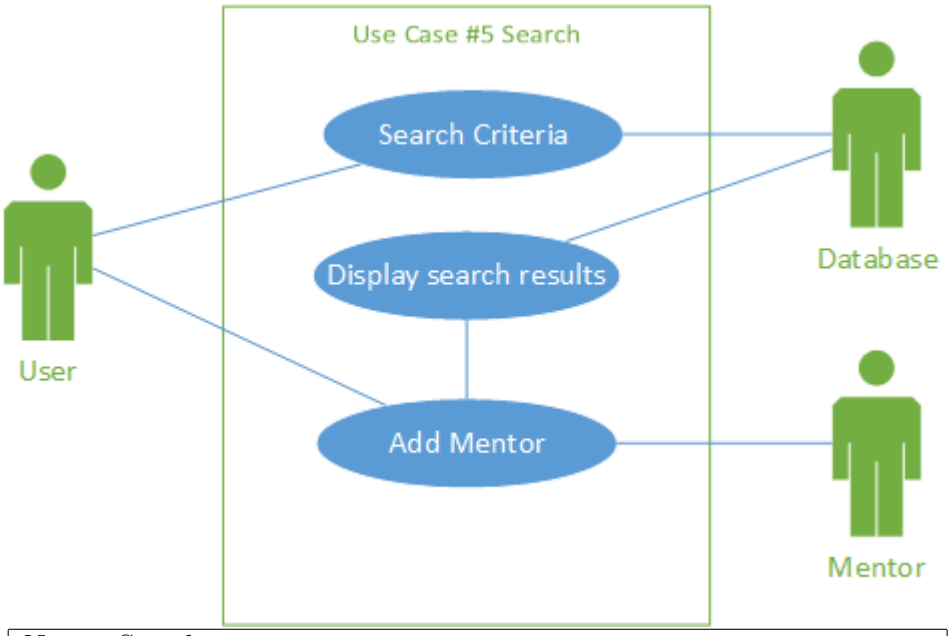
Name: Edit User Profile
Actors: User, Parse Manager, Database
Entry Condition: User has logged into MentorWeb system
Flow of Events: 1. User optionally chooses to import information from LinkedIn account to fill user information fields 2. User edits information fields manually to insert or correct information 3. User submits changes to information 4. Information is uploaded to the database
Exit condition: Database is updated
Exceptions: Edit is cancelled

Figure 4: Use Case Diagram for Use Case 3: Edit User Profile



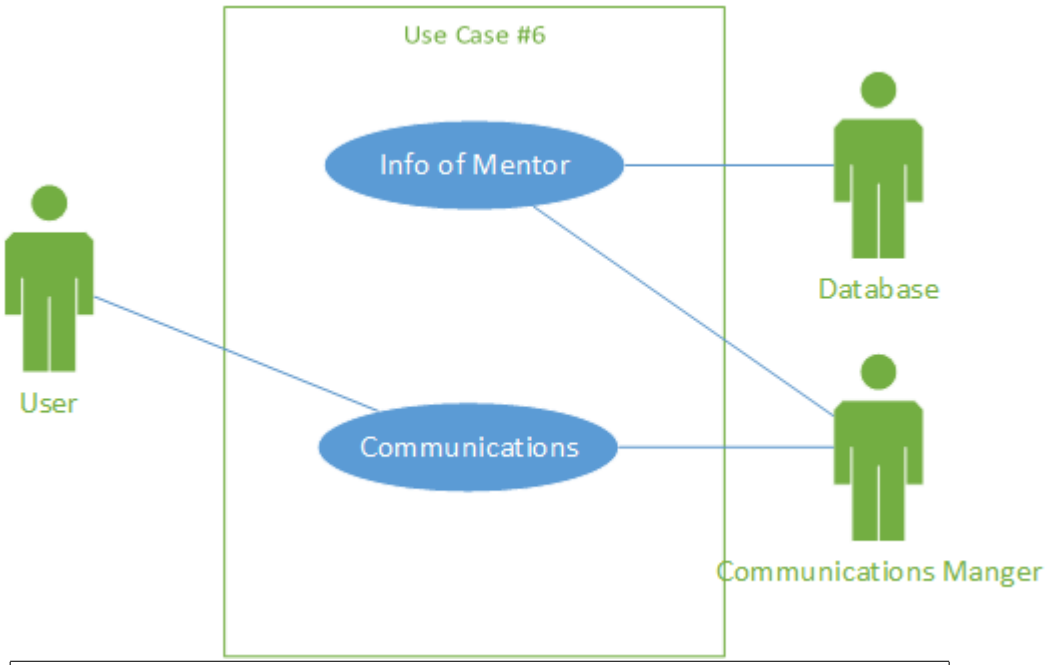
Name: Main Menu
Actors: User, Database
Entry Condition: User has logged into MentorWeb system
Flow of Events: 1. User MentorWeb network displayed 2. User has options to search for mentors, view their mentor's information, or edit their user profile
Exit condition: User navigates to other MentorWeb components
Exceptions: none

Figure 5: Use Case Diagram for Use Case 4: Main Menu



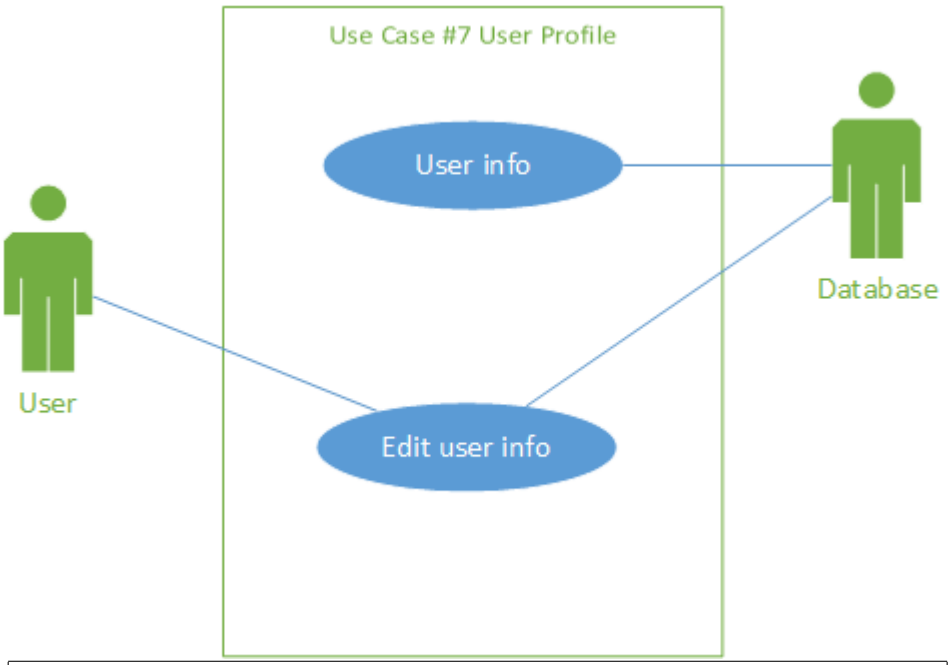
Name: Search
Actors: User, Database, Mentor
Entry Condition: User selects to search for mentors
Flow of Events: 1. User enters search criteria for mentors they want to connect with 2. System searches database and displays search results 3. User selects which mentors they want to connect with 4. Mentor consents to connection with mentee
Exit condition: Mentor-mentee connection is established
Exceptions: No mentors interesting to the user are found, or connection is denied by mentor

Figure 6: Use Case Diagram for Use Case 5: Search



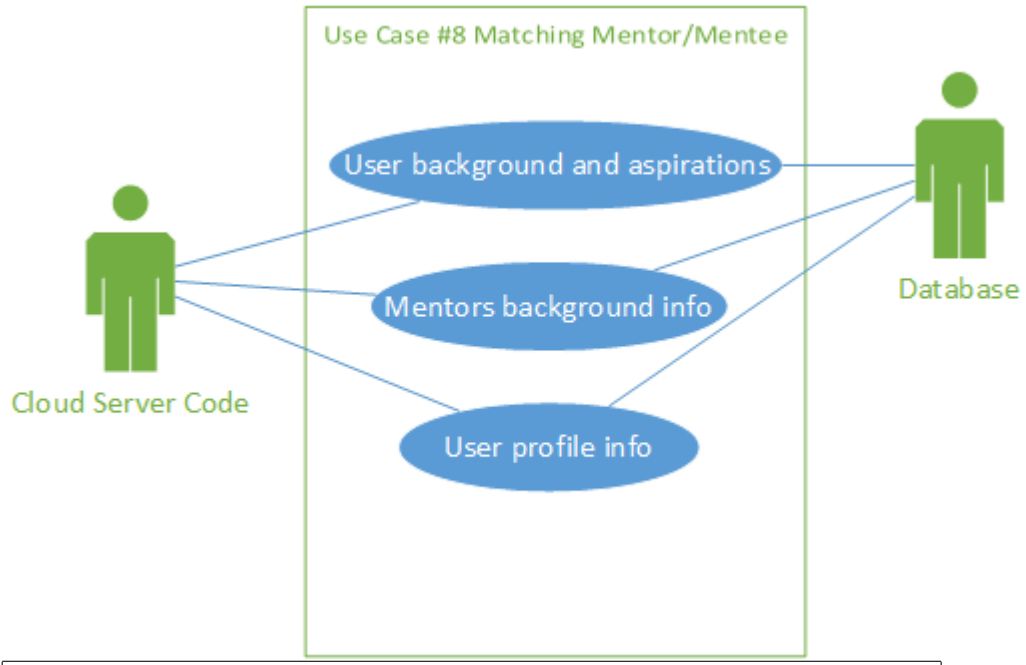
Name: Mentor and Mentee Comunication
Actors: User, Database, Communications Manager
Entry Condition: User selects to search for mentors
Flow of Events: 1. User chooses Mentor to view their information 2. User selects to contact selected mentor by utilizing Communications Manager
Exit condition: Communication occurs
Exceptions: Mentor/Mentee unavailable for communication

Figure 7: Use Case Diagram for Use Case 6: Mentor-Mentee Communication



Name: User Profile
Actors: User, Database
Entry Condition: User logged in to MentorWeb
Flow of Events: 1. Database displays information about user 2. User optionally selects to edit their profile
Exit condition: User leaves or begins editing profile
Exceptions: none

Figure 8: Use Case Diagram for Use Case 7: User Profile



Name: Matching Mentor with Mentee
Actors: Cloud Server Code, Database
Entry Condition: User has entered their background information and aspirations
Flow of Events: 1. Cloud Server Code parses through user's background and aspiration information for keywords 2. Cloud Server Code parses through background information from list of Mentors based on parsed user information 3. Cloud Server Code matches user background and aspiration information with Mentor's background information 4. Cloud Server Code sends luser list of recommended Mentors
Exit condition: Cloud Server Code makes recommendations to user
Exceptions: No applicable matches

Figure 9: Use Case Diagram for Use Case 8: Cloud Matching

3.4.3 Object Model

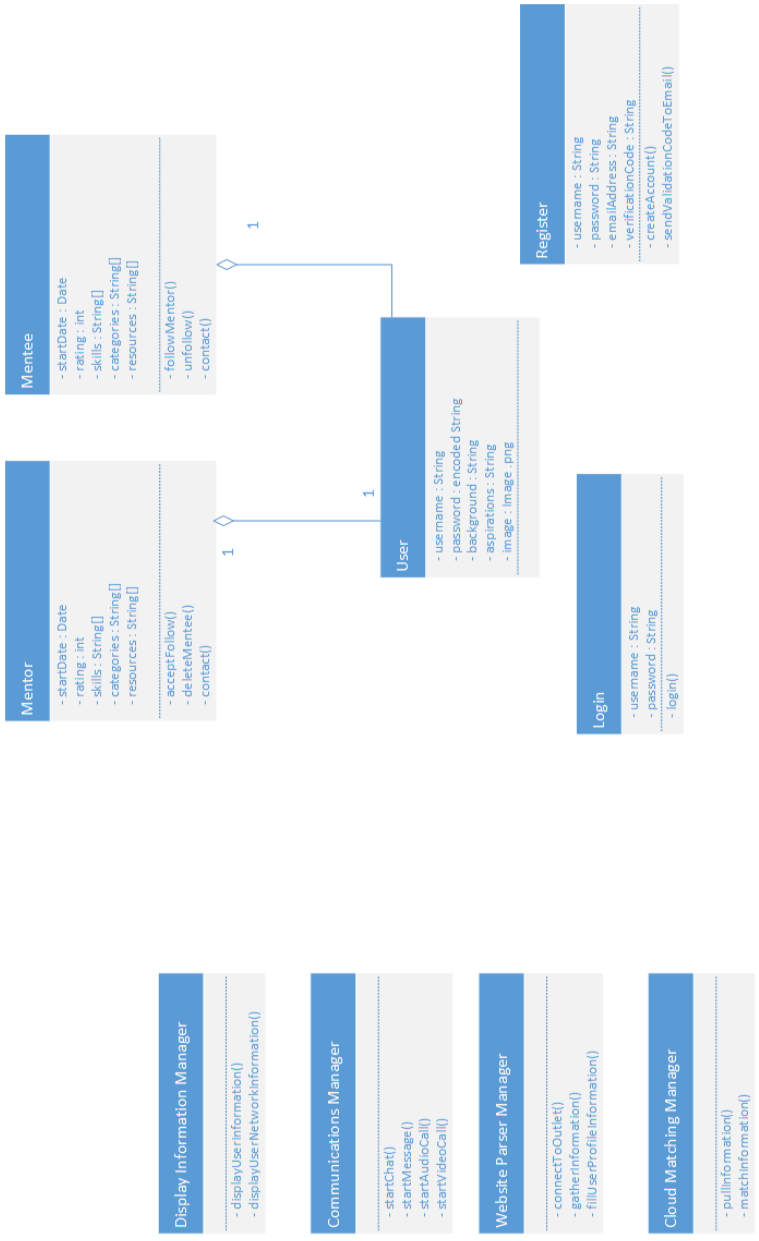


Figure 10: Class Diagram for MentorWeb

3.4.4 Dynamic Model

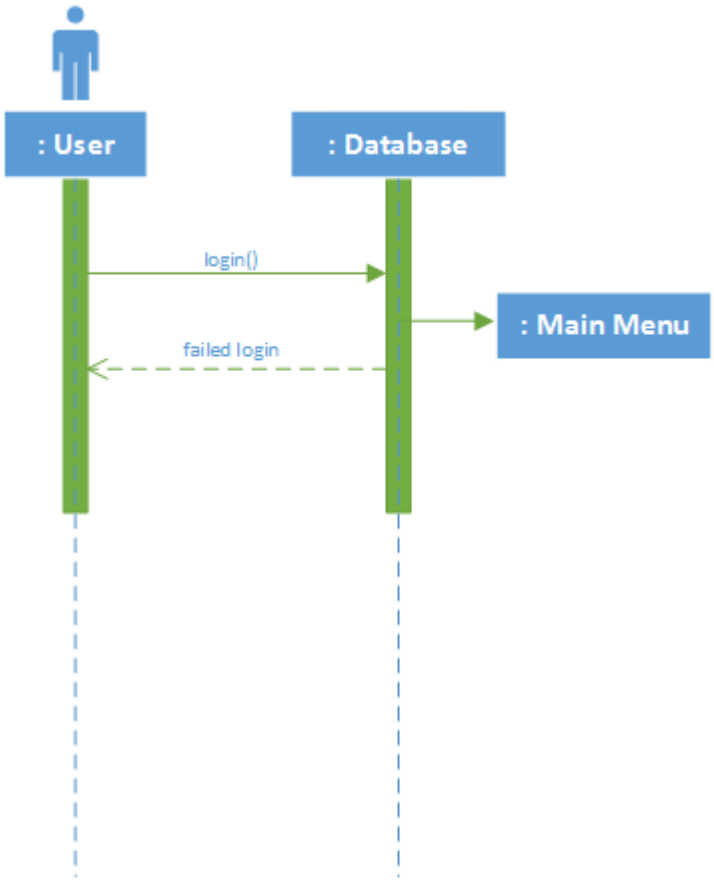


Figure 11: Sequence Diagram for Use Case 1: Login

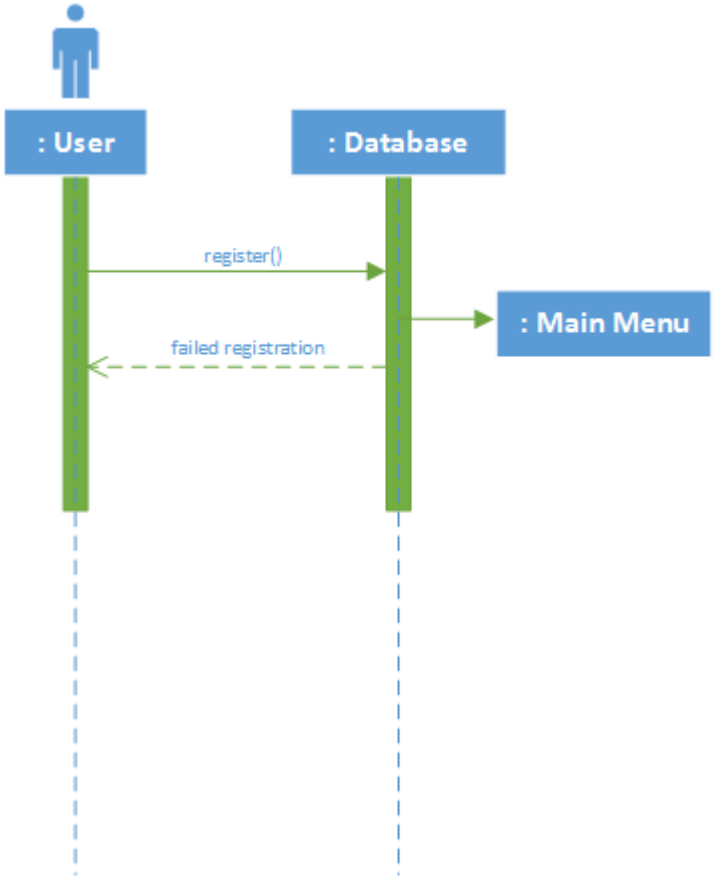


Figure 12: Sequence Diagram for Use Case 2: Register

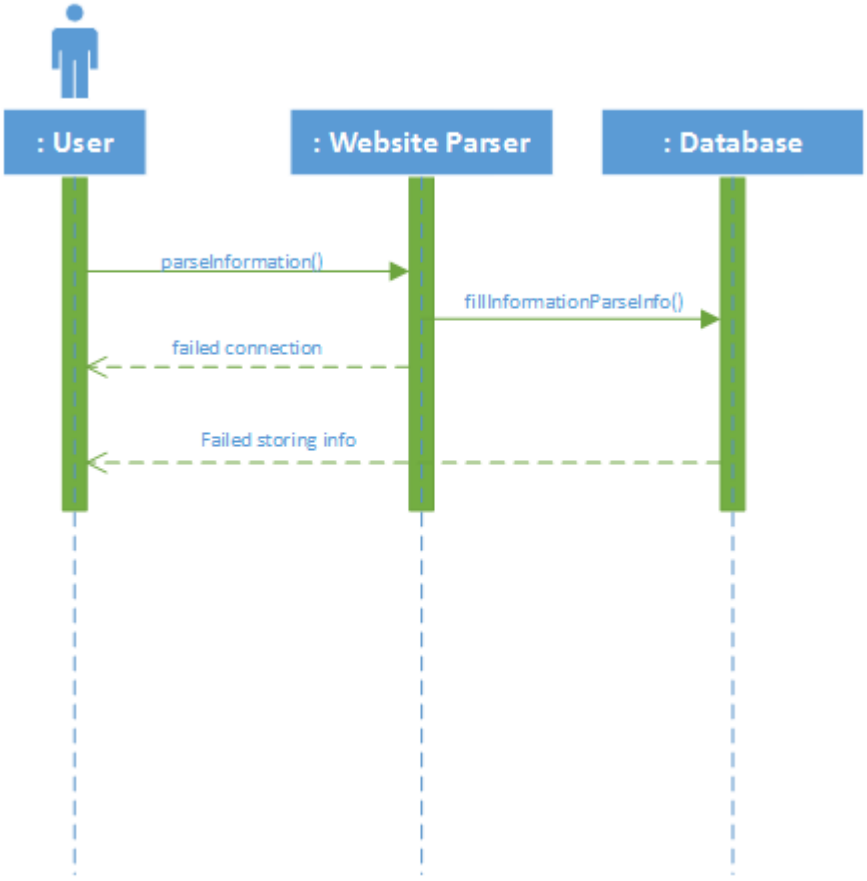


Figure 13: Sequence Diagram for Use Case 3: Edit User Profile

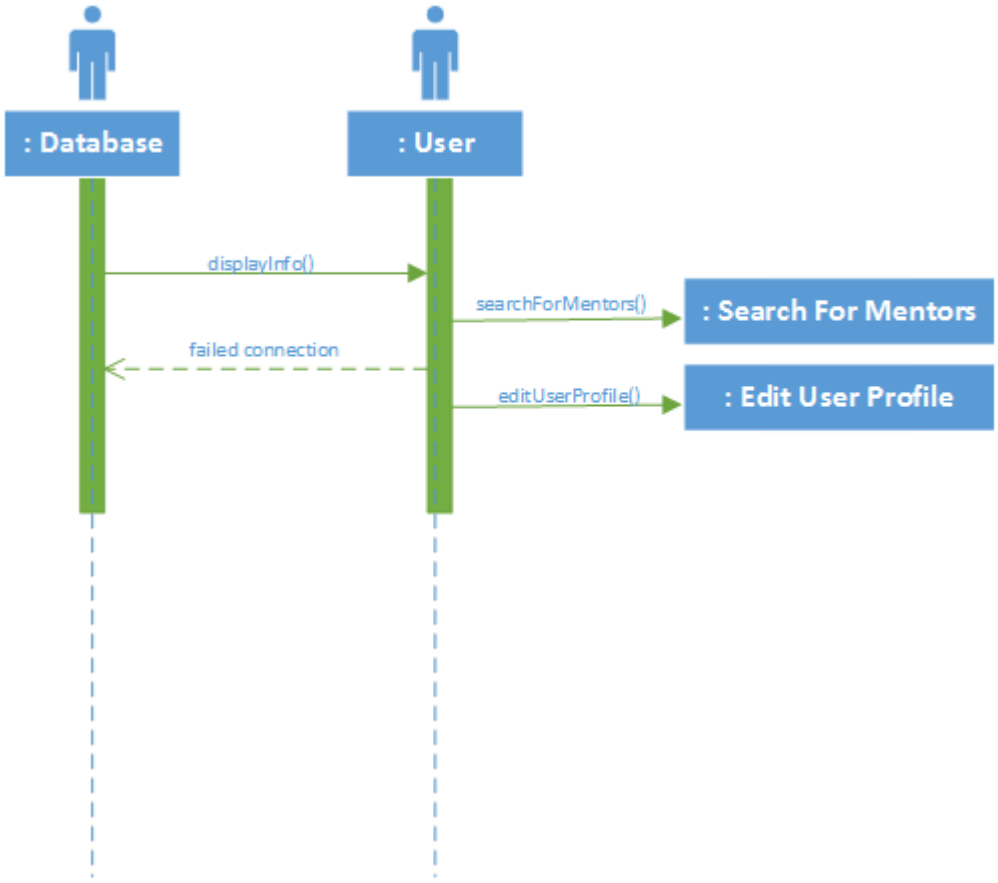


Figure 14: Sequence Diagram for Use Case 4: Main Menu

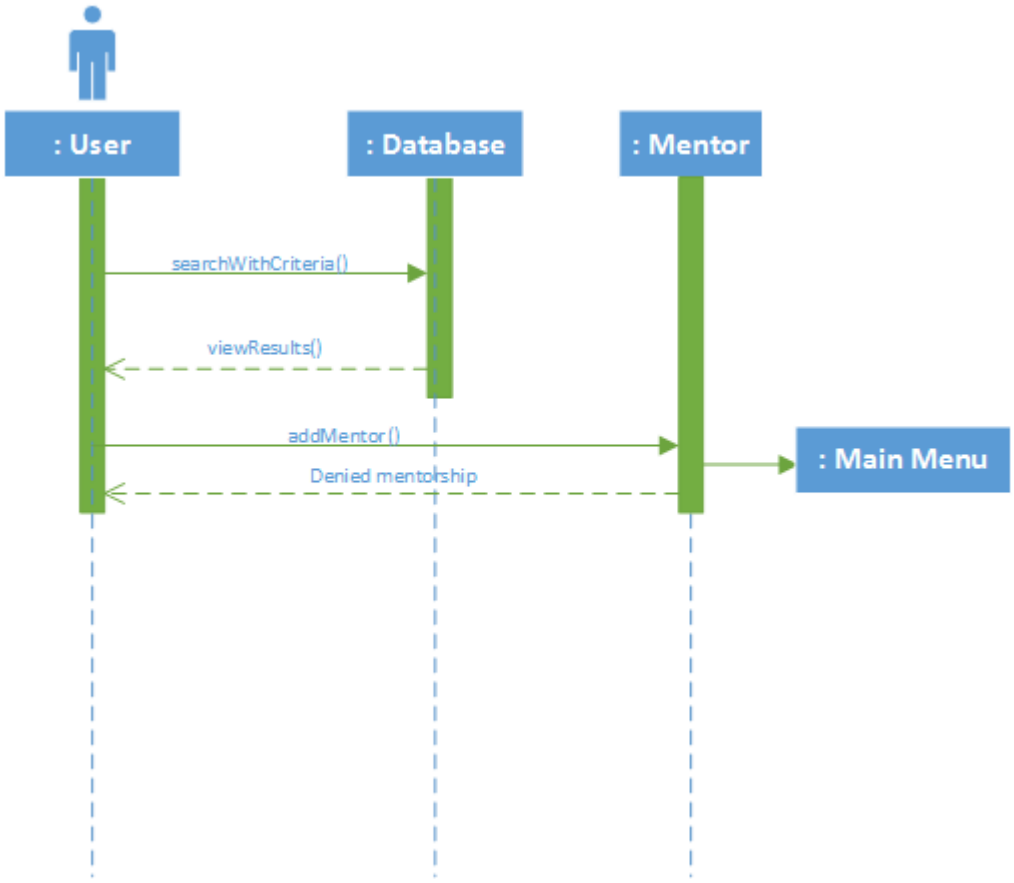


Figure 15: Sequence Diagram for Use Case 5: Search

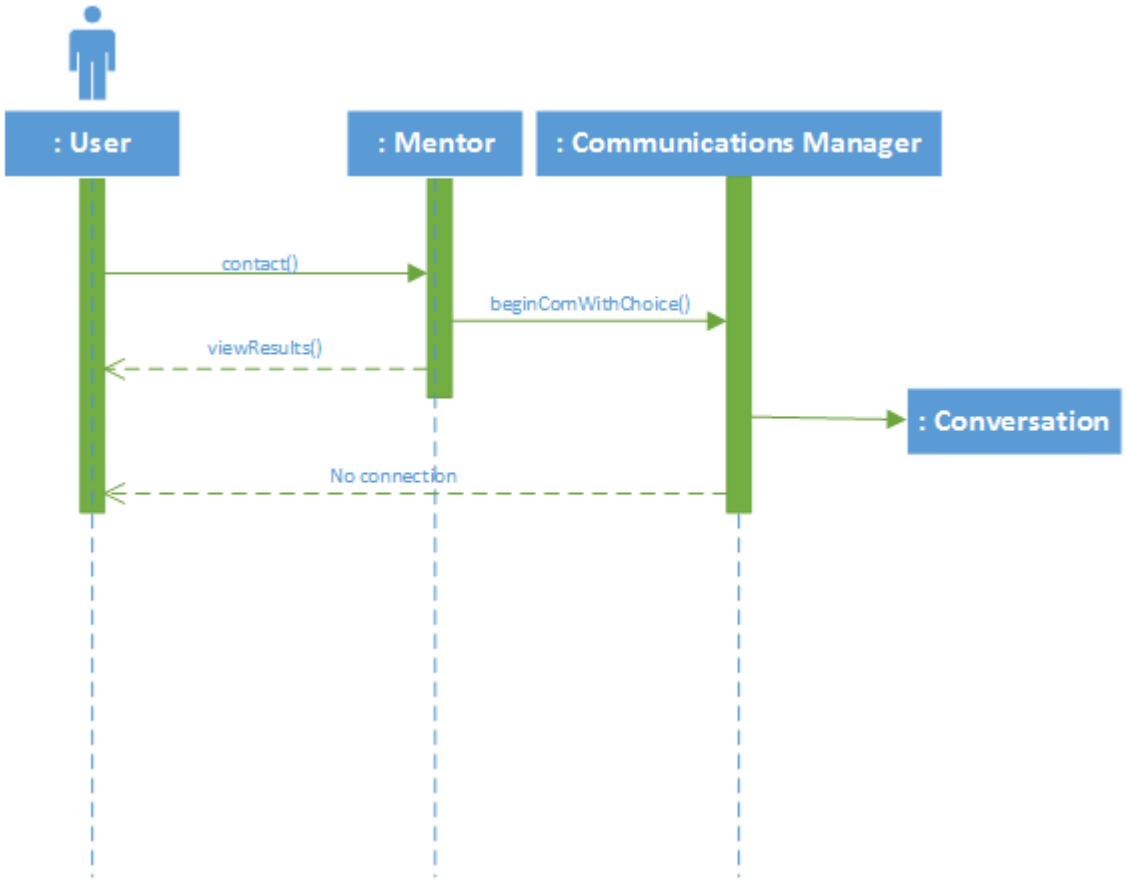


Figure 16: Sequence Diagram for Use Case 6: Communication

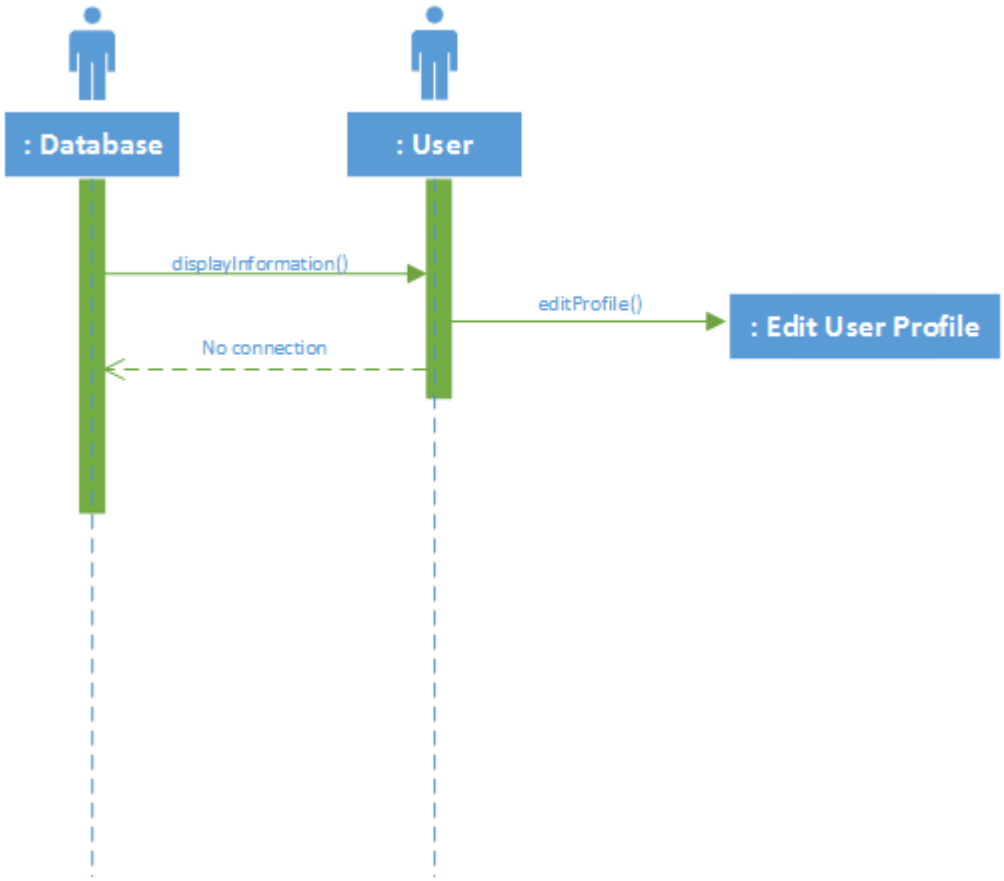


Figure 17: Sequence Diagram for Use Case 7: User Profile

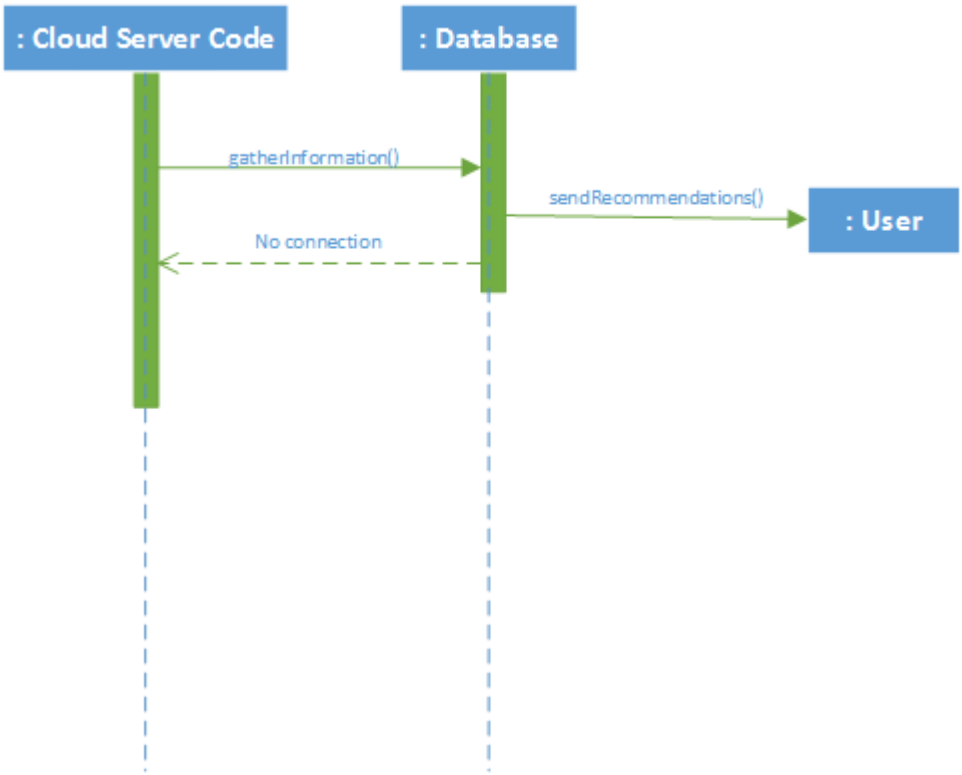


Figure 18: Sequence Diagram for Use Case 8: Cloud Matching

3.4.5 User Interface

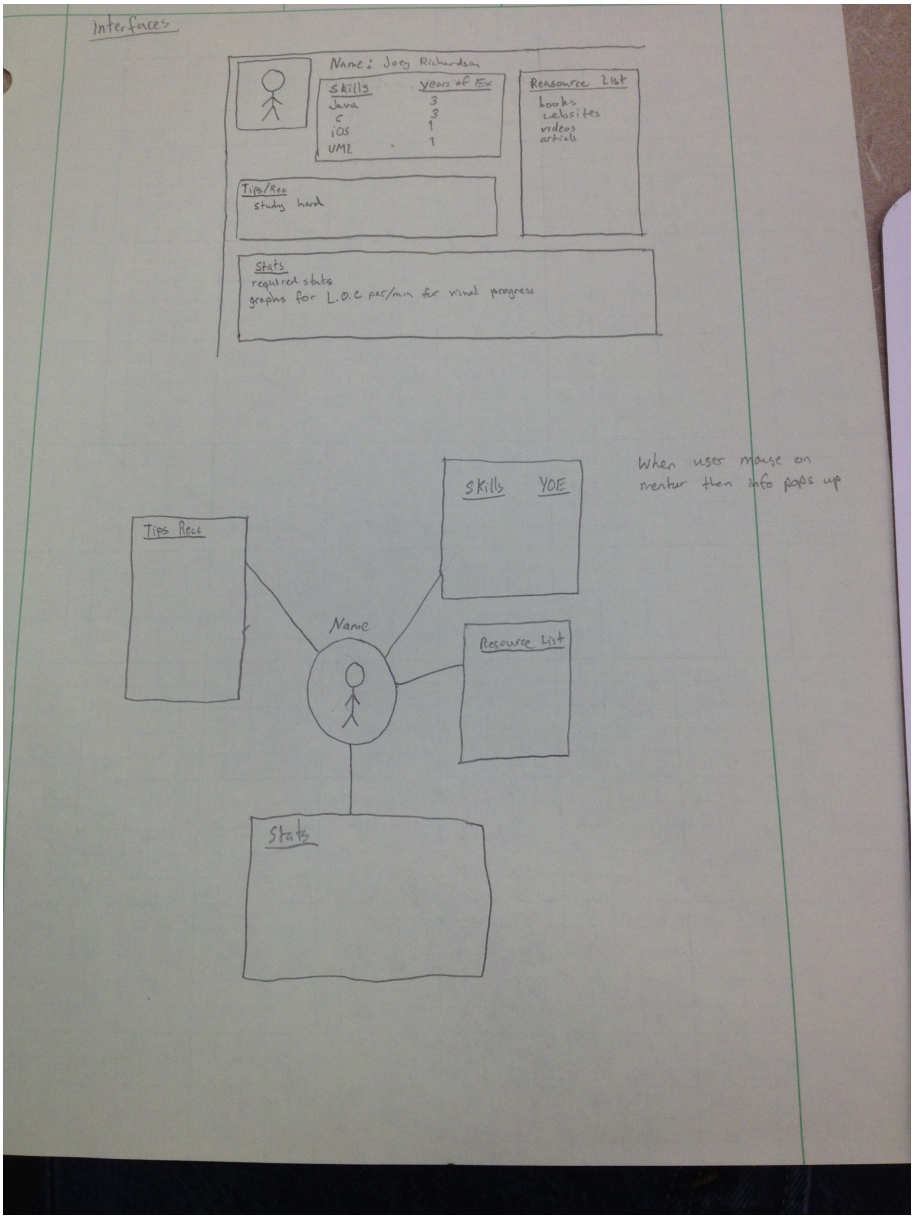


Figure 19: Mockup of a possible user interface

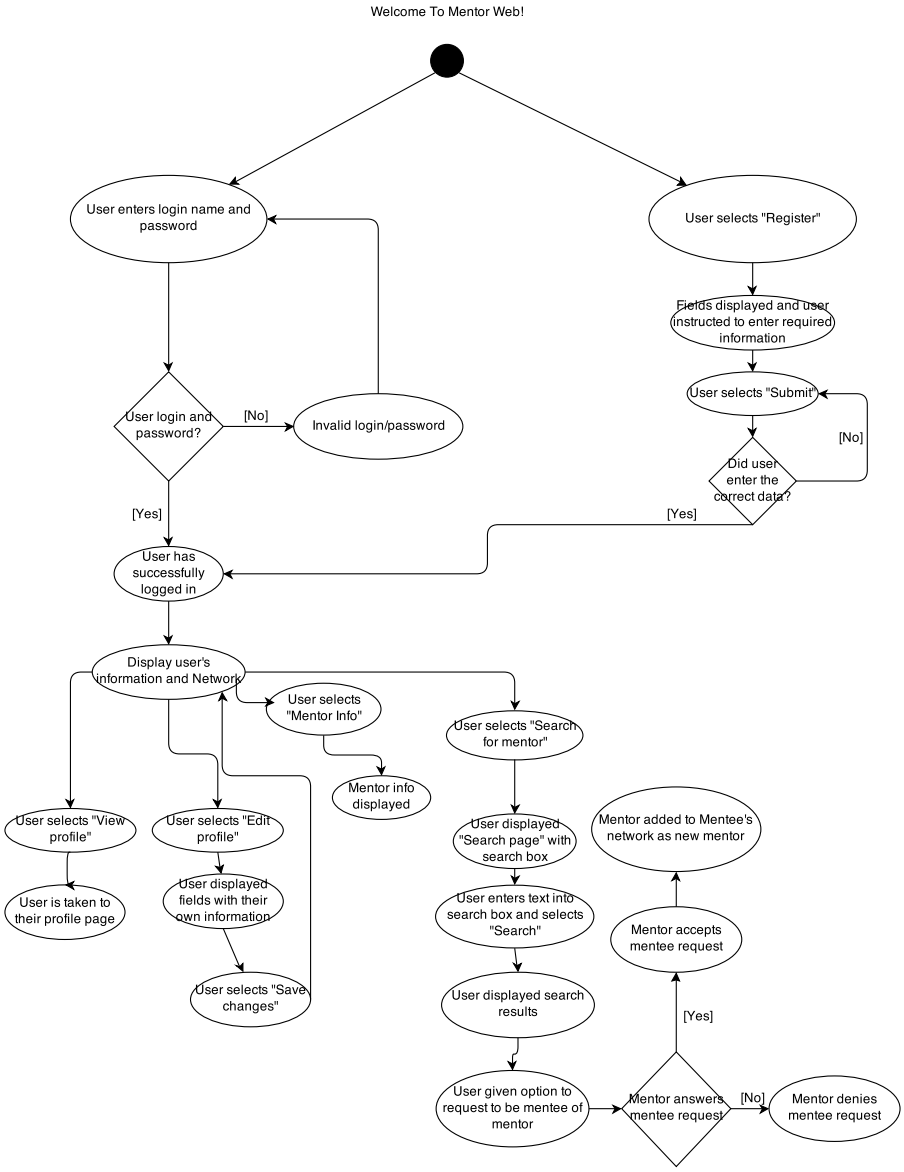


Figure 20: Activity Diagram

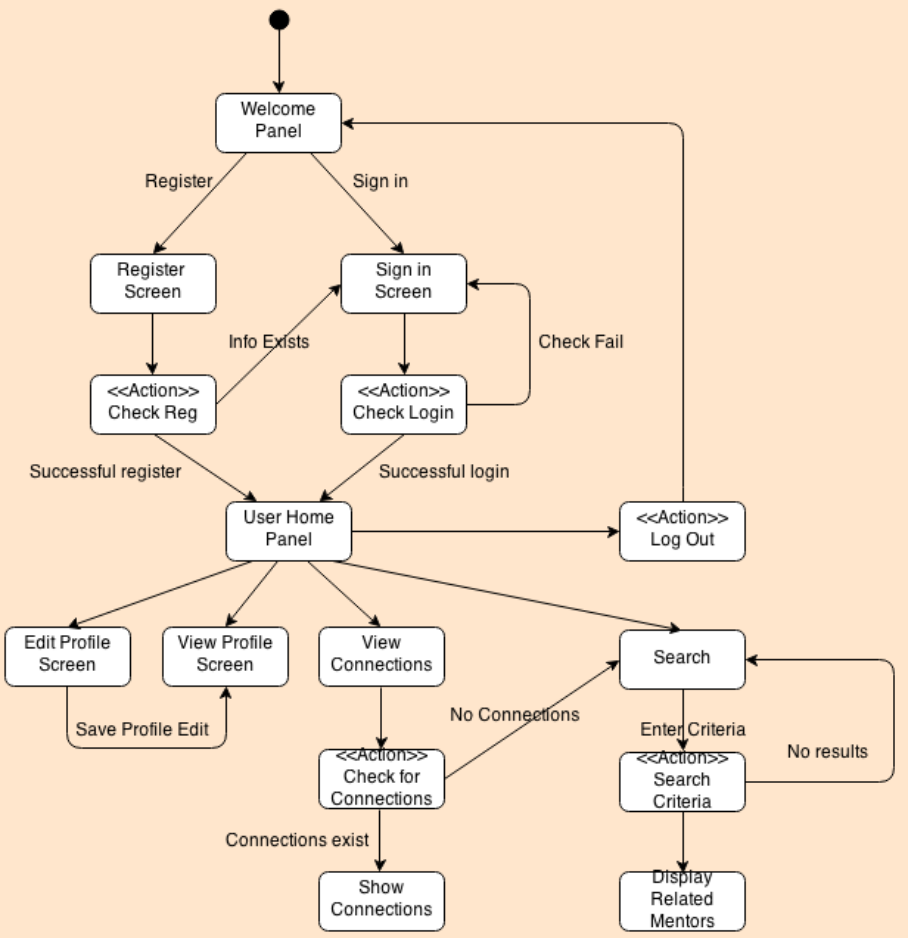


Figure 21: State Diagram

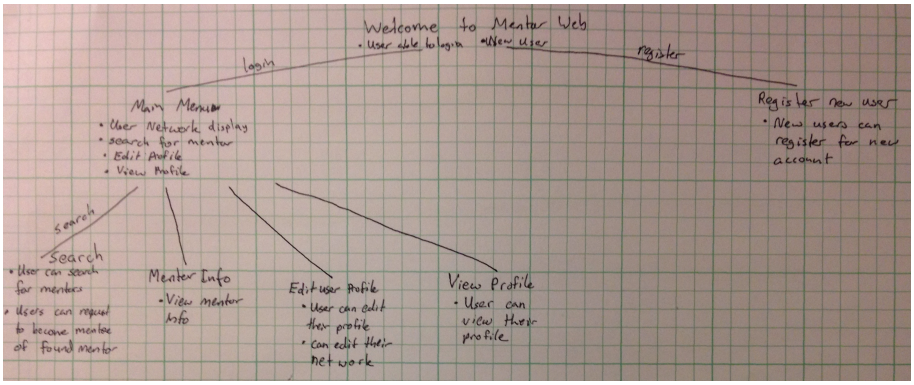


Figure 22: Mentor Web Navigation

4 Glossary

5 Appendices

5.1 Hardware Requirements

5.2 Project Plan

5.3 Team Staffing and Responsibilities

5.4 Notebook Log

5.5 Index

Index