San Francisco State University
SW Engineering CSC 648/848 Fall 2022
Req Check
TEAM #6

Names	**Role**	**Emails**
Alex Sanchez	Team Lead	asanchez26@mail.sfsu.edu
Victoria Wilson-Anumudu	GitHub Master	vwilsonanumudu@mail.sfsu.edu
Syed Faiz	Back End Lead	sfaiz@mail.sfsu.edu
Vivek Santoki	Front End Lead	vsantoki@sfsu.edu
Eric Falk	Front End	efalk1@mail.sfsu.edu
Erik Rodriguez	Front End	erodriguez7@mail.sfsu.edu

Milestone 2

Milestone Version	Date
M2V1	October 20, 2022

TABLE OF CONTENTS

Contents

Data Definitions

Student:
First Name
Last Name
Student ID number
Student email
Student cell

Professor:
First Name
Last Name
Employee number
Professor email

Course:
Course Number
Course Title
Units/Credit
Semester
Location

Records:
Transcripts
Student ID

Road Map:
Course Number
Course Title
Semester

Enrollment:
Credits
Course
Student ID
Space

Grade:
Letter
Student ID
Course Number
Course Title
Semester

Transfer Credits:
Institution Name
Course Number
Course Title
Units/Credits
Student ID

Equivalency Appeal:
Appeal ID
Outcome

Notification:
Email
Course
Section

Attendance
Present
Absent
Total Count

Waitlisted Student
Student first name
Student last name
Student ID #
Email

Dropped Students
Student first name
Student Last name
Student ID
Email

Deadlines
Last day to add
Last day to drop

Course Hierarchy:
Course Number
Course Title
Semester

Request
Request ID
Request type
Request description
Request timestamp
Outcome

Student Courses
Course Number
Course Title
Course Description
Course Credit

Syllabus
Course
Description

Prerequisites List
List of Courses

Prioritized Functional Requirement

Over here expand on the functionalist requirements. In addition, these requirements will be implemented on priority 1, 2 and 3.

Function	Priority	Actors	Function
17	1	Admin	Admin shall add the prerequisite on a defined hierarchy of courses.
20	1	Admin	Admin shall have a list of all the courses.
21	1	Admin	Admin shall have a list of courses taken by a student.
4	1	Application	The application should protect the encapsulated user data and disclose information based on roles.
6	1	Application	The application should automatically verify a student's academic records via the administrator.
7	1	Application	The application should be able to authenticate user identity by the university.
43	1	Course	Each course shall be given a category of major, minor, GE, or elective.
44	1	Course	Each course shall be appointed based on a student’s grade level (current, transfer, new).
47	1	Course	Each course shall have numerical options of perquisites.
48	1	Course	Each course shall have an enrollment status based whether a grade has been given or not.
51	1	Course	Each course shall display the semester’s it is being offered for.
1	1	Degree Planner	To use the degree planner feature, students must fill the appropriate subjects that they want to take for their degree.
59	1	Person	An authenticated person shall use the software.
37	1	Rec Checker	The Req checker shall be available on the WWW platform.
42	1	Requirement	Each requirement change shall update the overall course completion right away.
52	1	Semester	Each semester shall display the courses that can be taken prior to enrollment.
58	1	Student	Students shall be authenticated by the university to use the system.
57	1	System	System will be available during all hours of the day including weekends and holidays.
8	1	System	System shall provide all information on passed courses, professors, grades, and sessions.
25	1	System	The system shall display the course requirements for graduation.
26	1	System	The system shall differentiate courses by requirements and electives.
28	1	System	The system shall update all courses based on students input data.
29	1	System	The system shall inform students of course exemptions.
31	1	System	The system shall inform the student any prerequisites required prior to enrolling in a course.
32	1	System	The system shall update any prerequisite requirements.

34	1	System	The system shall enforce any prerequisites prior to enrolling in courses.
35	1	System	The system shall require satisfaction of prerequisites to enroll in courses.
36	1	System	The system shall give alternatives to satisfying prerequisites if applicable.
49	1	System	The system shall not allow students to enroll in courses they do not qualify for.
50	1	System	The system shall require students to complete prerequisites to enroll.
30	1	System	The system shall inform the student the number of semesters left to graduate.
16	2	Admin	Admin shall change any courses/prerequisites as per the demand of the admin panel.
14	2	Admin	Admin shall set new records and shall make a hierarchy of prerequisite courses.
15	2	Admin	Admin shall arrange all the courses department wise.
19	2	Course	Course shall have less dependency to any other entity.
2	2	Degree Planner	The degree planner shall give a timeline on how much time students need to complete all the courses and prerequisites for a degree.
45	2	Exemption	Each exemption shall be based on course requirements and student grade level.
53	2	Main Page	The main page shall display the user's schedule and classes with professor’s name, date, and class number.
61	2	People	The amount of people shall be defined by the college administration.
12	2	Professor	A professor shall fill up a seat once a student had dropped the course.
10	2	Professor	Professors shall only approve a defined number of students to their courses.
11	2	Professor	A professor shall fill in dropped students’ seats using a waitlist of students.
22	2	Student	Students shall see the list of professors for a course.
13	2	Student	Each student shall have a chance to request enrollment and each enrollment seat shall be filled up by a first come first serve basis.
33	2	System	The system shall be notified if a course has not been passed.
23	2	System	The system shall display the student’s grade level (current, transfer, new).
24	2	System	The system shall inform students of any changes in department requirements.
41	2	System	The system shall give a grade level status based on overall course completion.
27	2	System	The system shall notify students of all newly required courses.
46	2	System	The system shall give numerical status on the number of semesters left before graduating.
3	2	Timeline	The timeline shall depend on how many units a particular student must complete to obtain their degree.
60	2	Website	The website shall serve a limited number of people at a time.
5	3	User	The user (student) must be given information on whether a specific course is transferable from a certain school.
18	3	Student	Students shall have a selection of departments they must choose from.
9	3	System	System shall send notifications of recommended courses and waitlisted courses for users to add.

38	0	N/A	
39	0	N/A	
40	0	N/A	
54	0	N/A	
55	0	N/A	
56	0	N/A	

Initial list of functional requirements

1. To use the degree planner feature, students must fill the appropriate subjects that they want to take for their degree.
    1.1. There will be a table where students can select each course based on semester, division, area, department, etc.
2. The degree planner shall give a timeline on how much time students need to complete all the courses and prerequisites for a degree.
    2.1. There will be a roadmap table for each semester (Fall, Winter, Spring, Summer) and each academic year; this table will show the courses needed to graduate.
3. The timeline shall depend on how many units a particular student must complete to obtain their degree.
    3.1. Based on grade level or courses inputted, the roadmap table will update accordingly.
4. The application should protect the encapsulated user data and disclose information based on roles.
    4.1. This is more technical and will be included.
5. The user (student) must be given information on whether a specific course is transferable from a certain school.
    5.1. When submitting an equivalency appeal, the student will be notified of the appeal decision.
6. The application should automatically verify a student's academic records via the administrator.
    6.1. This should happen, but since we don’t have access to the SFSU system, it’ll be by default.
7. The application should be able to authenticate user identity by the university.
    7.1. This should happen, but since we don’t have access to the SFSU system, it’ll be by default.
8. System shall provide all information on passed courses, professors, grades, and sessions.
    8.1. Should be displayed in the user’s profile.
9. System shall send notifications of recommended courses and waitlisted courses for users to add.
    9.1. This is 2 parts; first, recommended courses are by default. And waitlisted courses notifications are optional when adding courses.
10. Professors shall only approve a defined number of students to their courses.
    10.1. Amount should be created by the professor in their dashboard.
11. A professor shall fill in dropped students’ seats using a waitlist of students.
    11.1. Should take place in the Professor dashboard for each unique course.
12. A professor shall fill up a seat once a student has dropped the course.
    12.1. Should take place in the Professor dashboard for each unique course.
13. Each student shall have a chance to request enrollment and each enrollment seat shall be filled up by a first come first serve basis.

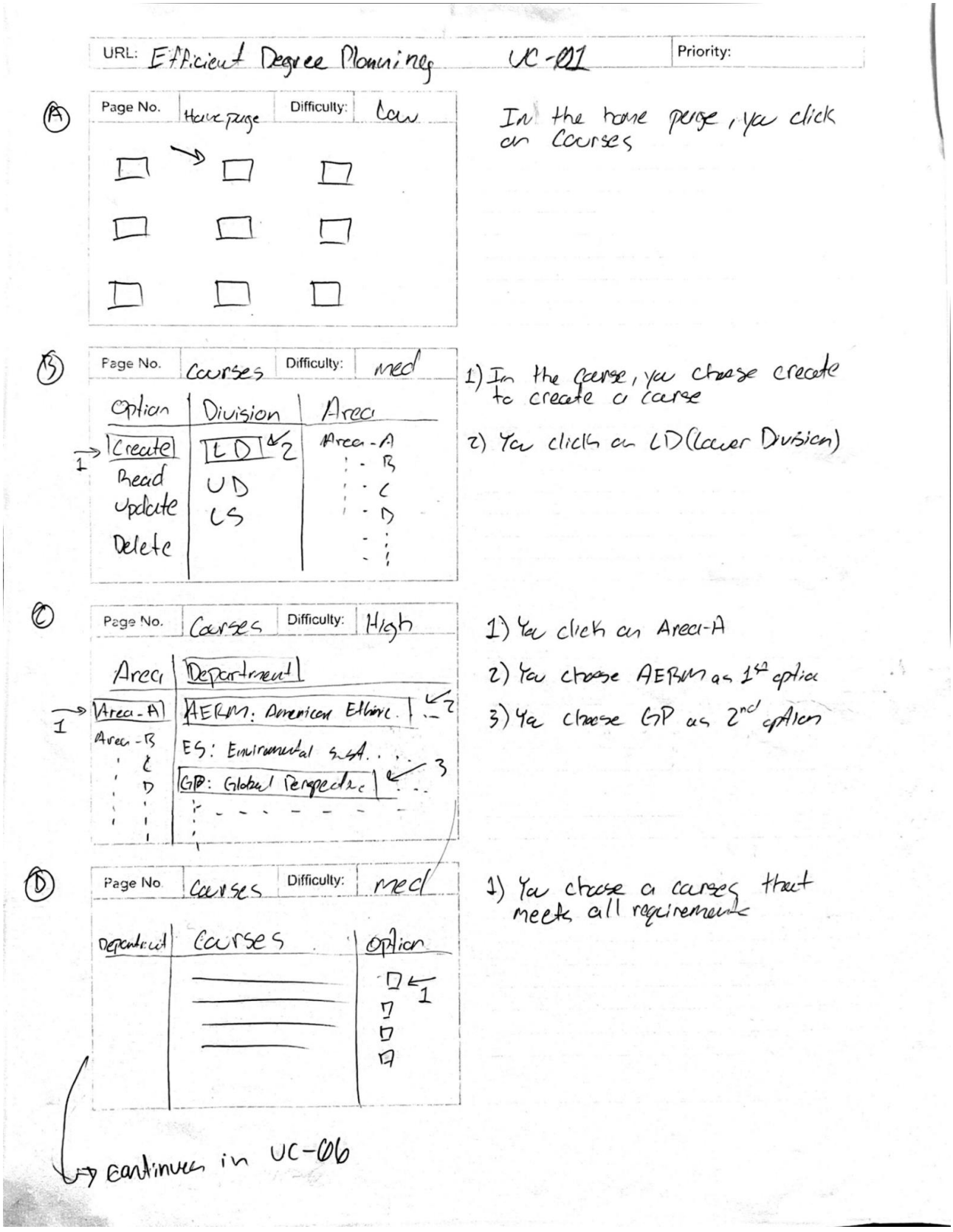
13.1. This is 2 parts; first, it shouldn’t request enrollment if there’s no waitlist. Second, filling a seat is by default a first come first serve. Otherwise, use a waitlist at professors' discretion.

14. Admin shall set new records and shall make a hierarchy of prerequisite courses.
    14.1. Admin will only receive prerequisites approved by all professors teaching the course.
15. Admin shall arrange all the courses department wise.
    15.1. Arrange as in arranging the tables?
16. Admin shall change any courses/prerequisites as per the demand of the admin panel.
    16.1. Only professors could do this, not the admin.
17. Admin shall add the prerequisite on a defined hierarchy of courses.
    17.1. Only professors could do this, not the admin.
18. Students shall have a selection of departments they must choose from.
    18.1. Should be shown in the course page.
19. Course shall have less dependency to any other entity.
    19.1. Not clear.
20. Admin shall have a list of all the courses.
    20.1. By default, yes.
21. Admin shall have a list of courses taken by a student.
    21.1. By default, yes.
22. Students shall see the list of professors for a course.
    22.1. Shown in the course page for students.
23. The system shall display the student’s grade level (current, transfer, new).
    23.1. Shown in the student's profile.
24. The system shall inform students of any changes in department requirements.
    24.1. Notification should be shown in the student's profile.
25. The system shall display the course requirements for graduation.
    25.1. Should be shown in both the Roadmap page and course page.
26. The system shall differentiate courses by requirements and electives.
    26.1. This should be shown in the course page.
27. The system shall notify students of all newly required courses.
    27.1. Should be shown in the student's profile as well as in the course page.
28. The system shall update all courses based on students input data.
    28.1. Inputs should update each student’s overall course table for graduation.
29. The system shall inform students of course exemptions.
    29.1. Based on equivalency courses approved, this should update the overall course table.
30. The system shall inform the student the number of semesters left to graduate.
    30.1. This should be default in the student’s profile and roadmap.
31. The system shall inform the student any prerequisites required prior to enrolling in a course.
    31.1. This should be default when searching for courses.
32. The system shall update any prerequisite requirements.
    32.1. This should be default

33. The system shall be notified if a course has not been passed.
    33.1. This should be shown in the student's profile
34. The system shall enforce any prerequisites prior to enrolling in courses.
    34.1. This should be the default when searching for courses to enroll in.
35. The system shall require satisfaction of prerequisites to enroll in courses.
    35.1. This should be the default when searching for courses to enroll in.
36. The system shall give alternatives to satisfying prerequisites if applicable.
    36.1. This should be the default when searching for courses to enroll in.
37. The Req checker shall be available on the WWW platform.
    37.1. This should be the default.
38. ~~The Website can serve the number of people at a time and the number will be defined by the college administration~~. Moved to non-functional
39. ~~As we used an optimized library to build the website so it will give the performance even if the user has low internet.~~ Moved to non-functional
40. ~~Web sites can have the changes in terms of functionality and design wise but will make sure that all the changes will be found in our user manual if there are any updates in the website~~. Moved to non-functional.
41. The system shall give a grade level status based on overall course completion.
    41.1. Should be default and shown in the student profile and roadmap.
42. Each requirement change shall update the overall course completion right away.
    42.1. Should be the default.
43. Each course shall be given a category of major, minor, GE, or elective.
    43.1. Should be the default and be in the course page.
44. Each course shall be appointed based on a student's grade level (current, transfer, new).
    44.1. Should take in consideration all course inputs, equivalency appeals, etc.
45. Each exemption shall be based on course requirements and student grade level.
    45.1. Should happen in the course page when searching and requesting a course.
46. The system shall give numerical status on the number of semesters left before graduating.
    46.1. Should be shown in both student profile and roadmap.
47. Each course shall have numerical options of perquisites.
    47.1. Should be the default and shown in the course page.
48. Each course shall have an enrollment status based whether a grade has been given or not.
    48.1. Each student should wait until grades are posted to see enrollment status.
49. The system shall not allow students to enroll in courses they do not qualify for.
    49.1. Should be the default in the course page.
50. The system shall require students to complete prerequisites to enroll.
    50.1. Should be the default in the course page.
51. Each course shall display the semester's it is being offered for.
    51.1. Should be the default in the course and roadmap page.
52. Each semester shall display the courses that can be taken prior to enrollment.
    52.1. Should be the default in the roadmap page.

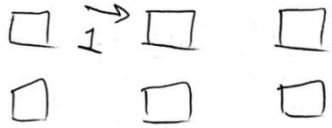
53. The main page shall display the user's schedule and classes with professor’s name, date, and class number.
    53.1. I want to say this should be in the roadmap page.
54. ~~All classes that can be selected will be green while classes that can not be taken will be red. If the course being taken is a prereq the following course will show yellow.~~ Moved to non-functional
55. ~~System will be able to support a large number of users at one time.~~ Moved to non-functional
56. ~~System will run on Windows, Mac and Lynx.~~ Moved to non-functional
57. System will be available during all hours of the day including weekends and holidays.
    57.1. This should be the default.
58. Students shall be authenticated by the university to use the system.
    58.1. This should happen, but since we don’t have access to the SFSU system, it’ll be default.
59. An authenticated person shall use the software.
    59.1. For the register accounts only, and to keep track of unique students.
60. The website shall serve a limited number of people at a time.
    60.1. For enrolled students only, otherwise it’ll cause storage problems.
61. The amount of people shall be defined by the college administration.
    61.1. For only the CS Department for now since we're dealing with a specific department only.

UI Mockups and Storyboards



URL: students satisfying prerequisites for electives UC-04 | Priority:

Page No. | Homepage | Difficulty:



1) You click on courses

Page No. | Courses | Difficulty: med

option | course | Prequisite | Status | Grade | Enrollment

Create
Read
Update
Delete

Yes ☑ No ☐

Planned
In-Progress
Completed

Pass
No-Pass
Credit
No Credit
Withdraw

Summer
Fall
Spring

1 2 3

1) choose update
2) Get your course list
3) Is it a prequisite,

Page No. | | Difficulty: high

status | Grade | Alternate Enrollment | update/create | clear

Planned
In-Progress
Completed

Pass
No-Pass
Credit
No Credit
Withdraw

Summer - 20XX
Fall - 20XX
Spring - 20XX

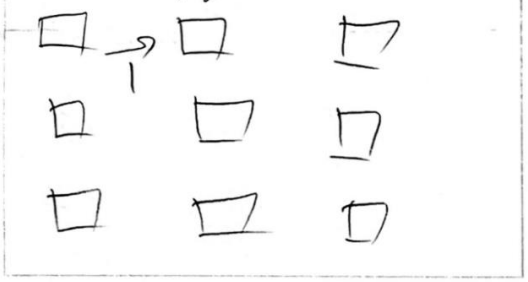
1 2 3 4

1) choose completed and get steam
2) Grade, choose "No-Pass" and get steam
3) Alternate enrollments, choose an option
4) Either update/create your course or clear and redo.

Page No. | | Difficulty:

URL: Grade Check UC-05 | Priority:

Page No. | Homepage | Difficulty:



1) In the home page, you choose courses

Page No. | courses | Difficulty:

option	Course	Prequisit	Grade	Future Course
Create		Yes ☑	A B C- D F W C	
Read		No ☐		
update				
Delete				

↑1 ↑2 ↑3 ↑4 ↑5

1) Choose Read

2) And get shown all completed courses?

3) is it a prequisite

4) your grade received

5) Future courses that needs the courses

Page No. | Course | Difficulty:

Future Course	Enrollment	Petition
	Granted ☐	Yes ☑ ← 4
	Denied ☑ ↑2	No ☐

→ 1

3

2) Select the future course you want

2) and be notified of possible enrollment

3) You can now petition to retake future course

4) and click yes

Page No. | Course | Difficulty:

Petition	Notify	Upload	Permission
	Faculty ☐	Input	D-G
	Dean ☐		D-G
	Department ☐		D-G

↑1 ↑2 ↑3

1) Allow you to notify everyone to allow you to retake the course

2) Upload any files if needed

3) Get notified whether you have receive permission or not from all three levels.

URL: Fastest Route to Graduation UC-08	Priority:

A)

Page No.	Courses	Difficulty:	

option | status | Type | View by
Create | Full Time | Required | Division
Semester

1

This is based on usecase #7, part B.

1) You choose by semester

Page No.	Courses	Difficulty:	

View by | Semester | Division
Division | Winter
Semester | Spring
Summer
Fall

1 2

1) You choose winter

2) You go to usecase #1, part B

REDUNDANT! Redo usecase #1 -add semester b4 Division

Page No.		Difficulty:	

Page No.		Difficulty:	

URL: Semester Courses are Offered UC-09 | Priority:

Page No. Homepage | Difficulty: low

1) Click on course

Page No. Courses | Difficulty: med

Edit	Course	Prequisite	Status	Passed	Proof	Share
Create		Yes ☑	Planned	Yes ☑	Grade	Yes ☐
Read			In Progress			
Update		No ☐	Completed	No ☐	Input	No ☐
Delete						
Verify						
↑1		↑2	↑3	↑4	↑5	↑6 ↑7

1) Click on verify
2) Click on a taken course
3) Is it a prequisite
4) Give status
5) See current grade
6) See current grade or input proof
7) Share your grade

Page No. Dashboard/prof. | Difficulty: med

Enrolled	List	Prequisite met	Information	Permission
Course A		Y - N	Request/See	Granted / Denied
Course B		Y - N	See	
Course C				
↑1	↑2	↑3	↑4	↑5

1) Professor selects a course
2) Select a student
3) Was prequisite met
4) request proof or see it
5) Give permission to enroll in class

Page No. | Difficulty:

URL: Prequisite Road Map Chart	UC-10	Priority:

Page No.	Homepage	Difficulty:

1) Clicks on Roadmap

Page No.	Roadmap	Difficulty:

with?
1 → Choose length 3,4,5,6 years
2 → Choose status Full-Time or Half-Time
3 → Courses From profile, Randomize
4 → Generate Roadmap

1) Choose length of actual attendance
2) Choose attendance status
3) Import courses from profile or randomize
4) Generate Roadmap.

Page No.	Generate Rm	Difficulty:

	20XX	20XX	20XX	20XX	
Fall	=	=	=		
Winter					
Spring	=	=	=		
Summer		=			

1) A road map of courses from profile

Page No.		Difficulty:

URL: To set the Prequisite courses	UC-12	Priority:

Admin view

Page No.	Homepage	Difficulty:

1) On home page, view courses

Page No.	Courses	Difficulty:

Courses	Division
Create	LD
Read	UD
update	CS
Delete	

1) similar to student, the admin will be able to edit all courses, prequisite & non-prequisite

Page No.		Difficulty:	

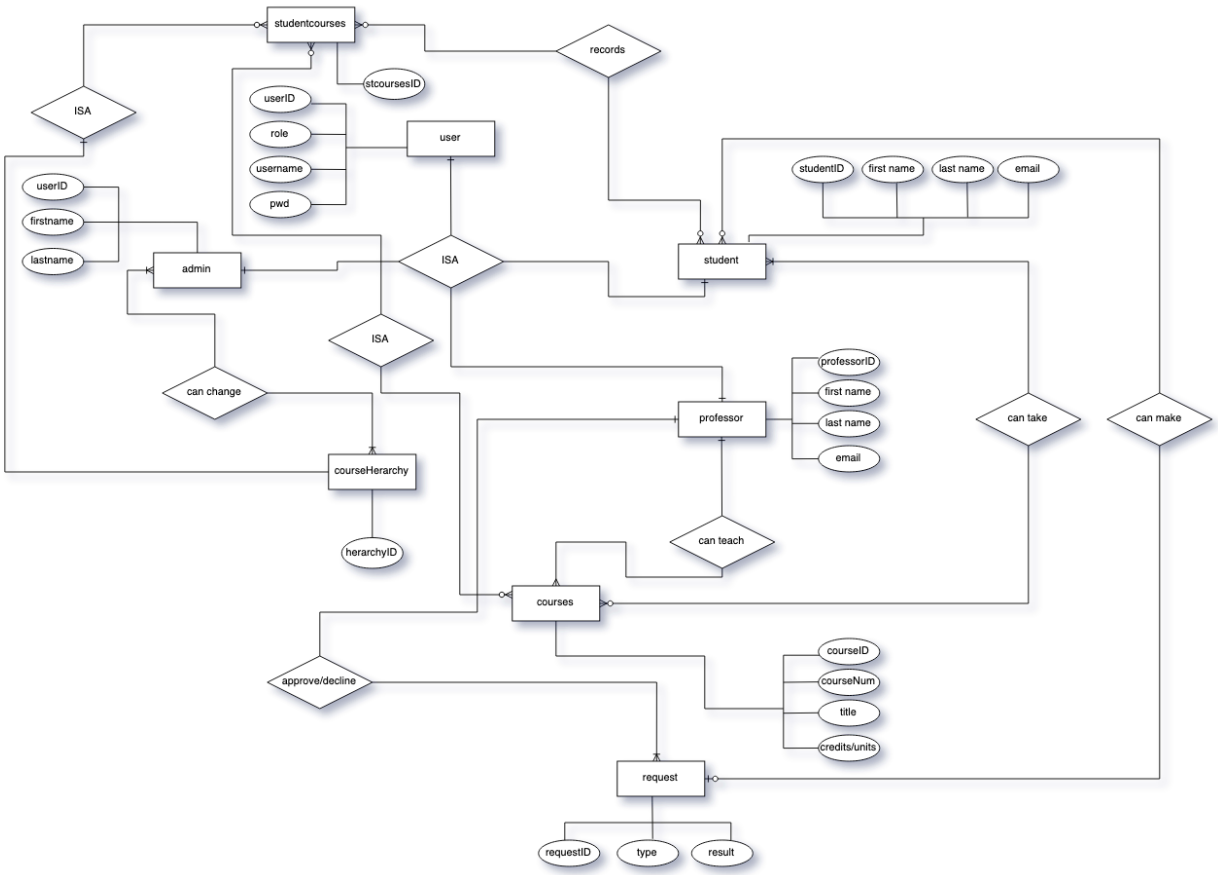
Page No.		Difficulty:	

High-level DB Architecture and Organization

Database Requirements

1. The admin shall be the user super.
2. The user should have one of these roles: admin, professor or student.
3. All users must have login credentials.
4. Students shall request to get many prerequisites waived.
5. Professor must approve or decline many prerequisite waiver requests.
6. The admin shall update the many course hierarchies for the students.
7. Student courses shall have information about many courses.
8. Students courses shall have be linked to many courses to show one or many prerequisites
9. A professor can teach one or many courses.
10. A student prerequisite request shall be terminated after a defined time.

Entity Relationship Diagram



Entities and Attributes

Entity 1: courses
courseID, primary key and not null.
courseNumber, numeric.
course title, numeric.
credit /units numeric.

Entity 2: admin
empID, primary key and not null.
First name, varchar alphanumeric.
Last name, varchar alphanumeric.

Entity 3: user
userID, primary key and not null.
Role, varchar
Username, alphanumeric and not null
Password, alphanumeric.

Entity 4: student
studentID, primary key, unique, and not null.
First name, alphanumeric.
Last name, alphanumeric.
Email, alphanumeric.

Entity 5: professor
professorID, primary key, unique, and not null.
First name, varchar alphanumeric.
Last name, varchar alphanumeric.
Email, varchar alphanumeric.

Entity 6: student-courses
studentID, fk key
courseID fk key

Entity 7: request
requestID, primary key
Request type, alphanumeric

Result, bool and not null

Entity 8: courseHierarchy
hierarchyID, primary key
courseID, fk key
courseIDs, array

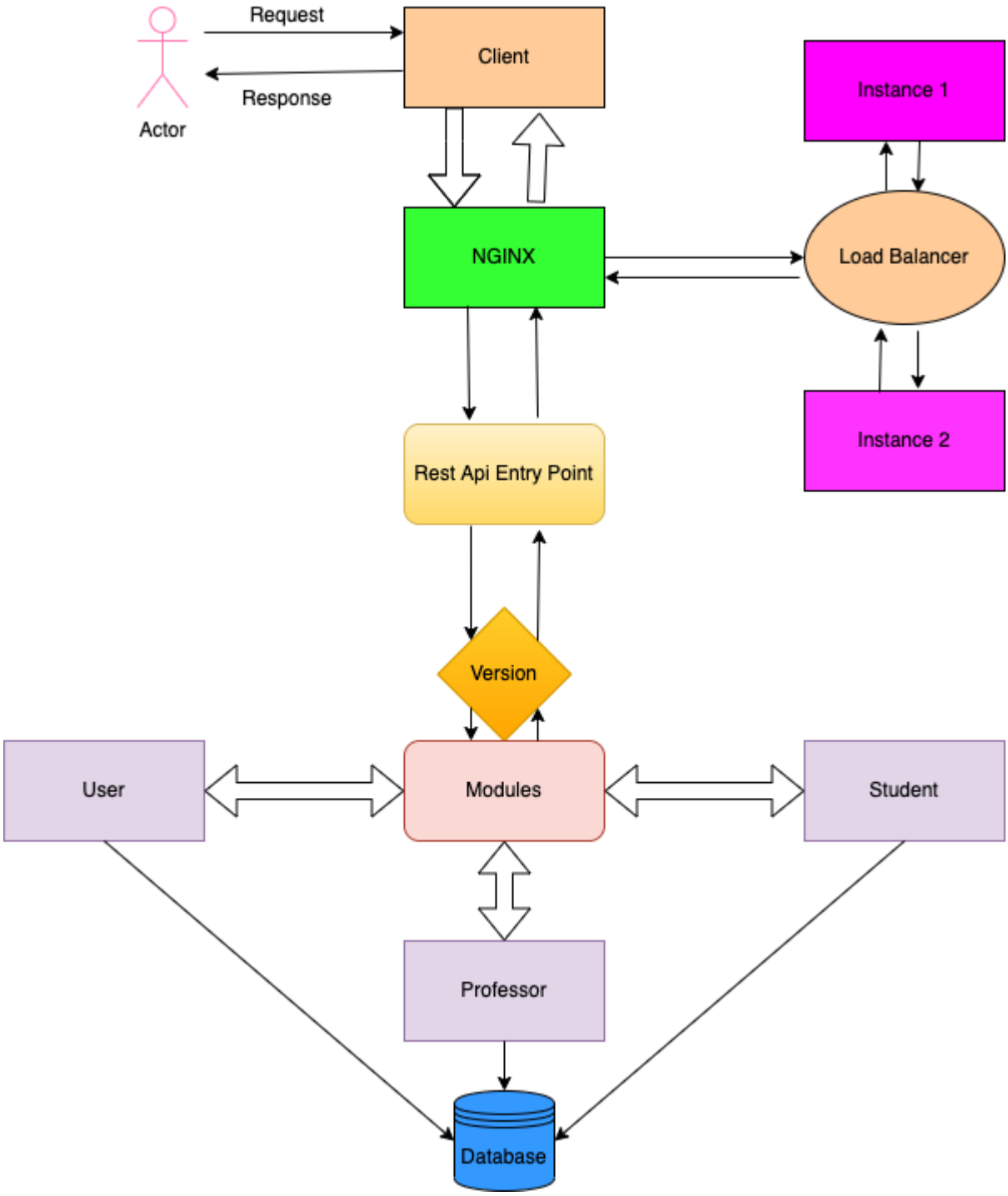
For database requirements and architecture, our team will use MySQL workbench to create tables, add data to the tables, create triggers, procedures, and queries for data management. For the backend/database search algorithm, we'll use mysql workbench and create a table courses using SQL statements manually since we do not have EER. We used insert statement to add data into the table and used queries such as %s and %i to retrieve data to verify if the exists and database is functional.

High level API and Main Algorithms

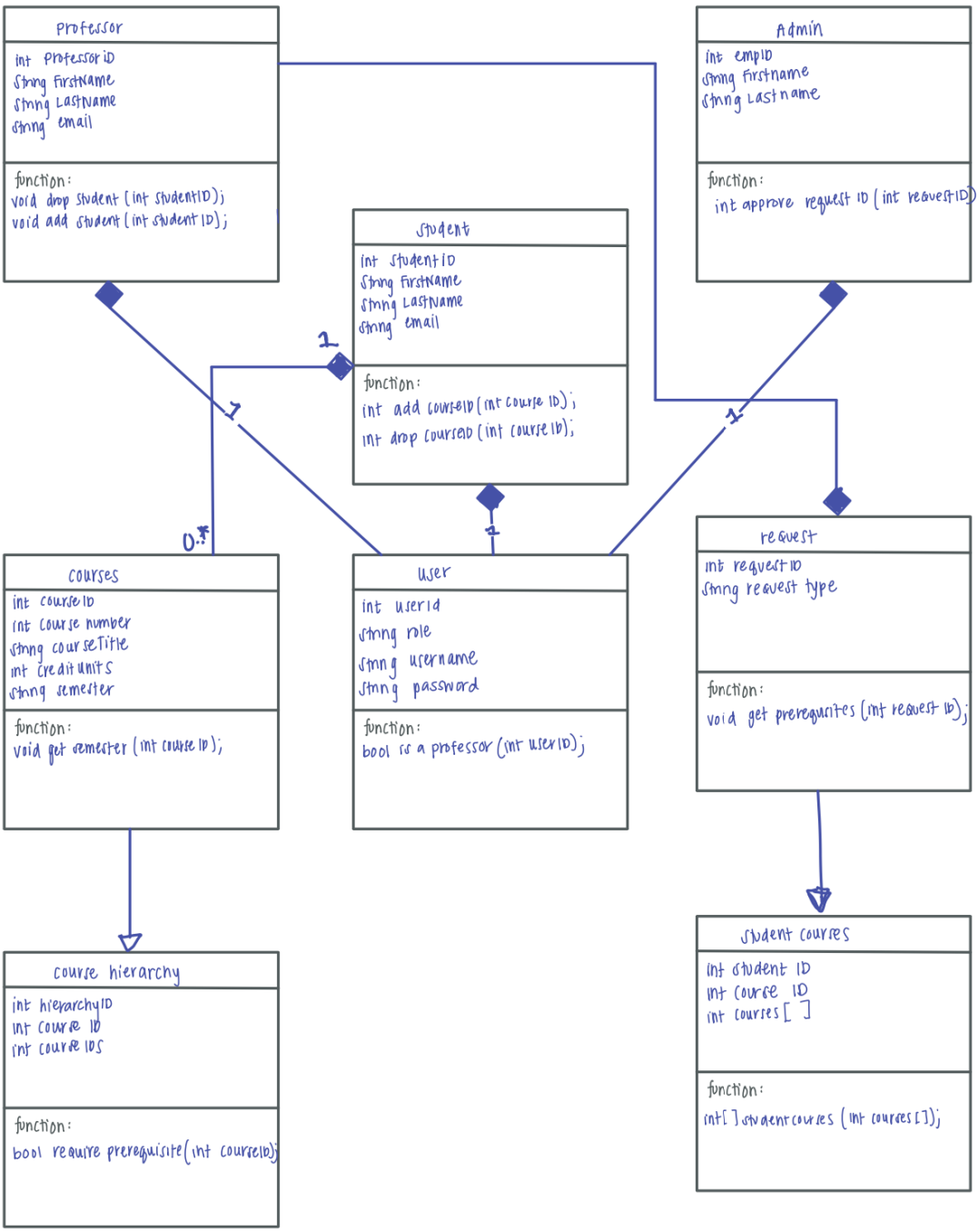
Backend Architecture :- Monolithic architecture

Frontend Architecture:- Atom architecture

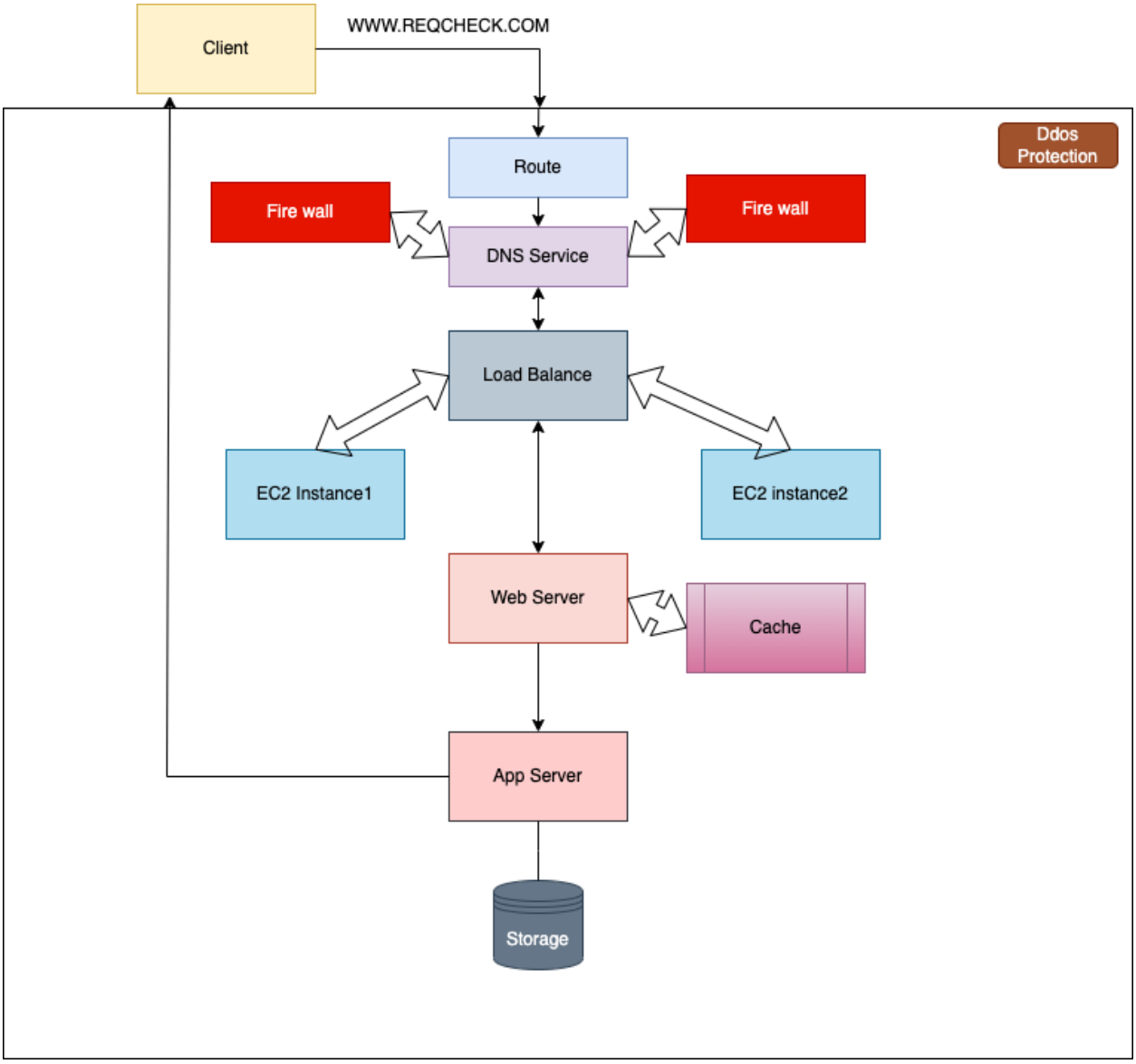
Algorithms:- We will mainly use recursive binary search tree to extract the data which are stored as hierarchy.



High level UML Diagram



High-level Application Network & Deployment Diagram



Key Risk

- Skills Risk
    - Identify Risk & Explain
        - Most of our team members are exposed to frontend development, but very few in backend development. Not many are fully familiar with React, as well as deployment and cloud platforms. The more technical aspects might need some quick rundown for everyone due to how difficult it is.
    - Solution to issue
        - We'll need to at least get everyone exposed to the full app development; go over the frontend, backend and deployment of an app. We'll probably need to individually go over YouTube links for React and reinforce the knowledge by going over previous milestone prototype code.

- Schedule Risk
    - Identify Risk & Explain
        - Some of the team members will have to make time and allocate the resource due to additional academic courses and non-academic commitments and priorities.
    - Solution to issue
        - We primarily use discord to communicate, and so far, it's working. We just need to communicate more, define our roles/responsibilities, and clarify what needs to be done. Trello would eventually be used as well, and we have a schedule to input our availability.
            - https://www.when2meet.com/?16641975-7JBDt

- Technical Risk
    - Identify Risk & Explain
        - There is not much technical risk to be identified.
    - Solution to issue
        - Any technical risk will be dealt with as soon as possible.

- Teamwork Risk
    - Identify Risk & Explain
        - The primary risk is communication; we have a severe communication issue that is still present and will need to be addressed. We primarily use discord, but I believe we will need to include more face-to-face meetings as well as Zoom meetings. Secondly, our individual roles and responsibilities are not as clear as they should be as well as creating a set timeline to meet deadlines and planning.
    - Solution to issue
        - We'll be implementing changes to how we communicate; meeting over zoom, whether in groups or individuals should be carried out. At least 3x

a week if possible and taking notes on said meetings in order to share with others. We will also need to clarify what our roles and responsibilities will be when we take on a milestone. We'll also need to make aware of what our planning and progress will be like to better understand how the team is doing overall. Also, everyone should support one another by bringing more participation and energy to meetings.

- Legal/Content Risk
    - Identify Risk & Explain
        - There is not much legal risk to be identified since our work is derived from open source.
    - Solution to issue
        - Any legal risk will be dealt with as soon as possible.

Project Management

This will be in two parts, before and after the changes in team lead. I would like to acknowledge Vivek's team leadership, so I'll leave his part here.

In order to tackle the work we will divide each module across all the team members. Each team member will have their tasks which are divided into modules. To reduce the dependency, we break the module in such a way that other team member will be not blocked or dependent.To reduce the dependency, we have to give priority to each task.To maintain the task we will use the Trello where all the people will have their ticket which we will includes all details related to the task including attachment.We will sue the development cycle. After completion of each task we will deploy them into the development server.Respective team members will do the unit testing after that task will be reviewed by the team lead and after doing successful testing we will deploy that feature in production. Every team member will finish the high priority task first and so on.We will use Discord platform to communicate with other team member.Every production bug will have the priority to solve that bug.We will have weekly virtual meetings in order to make sure every team member has the idea of overall development.

For project management, we should have used Trello more actively; the usage of DIscord will and did help, but given the time constraints due to leadership, Discord became the primary point of communication. For future tasks, Trello will be used for visual confirmation of individual accomplishments; Discord will be used for group communications in order to verify tasks and ask any questions. Finally, we will use Zoom to meetup and refine and agree to our roles and responsibilities. With Trello, Discord, and Zoom, I hope we can better collaborate and improve our communications.

List of Contributions

Part A: Team Lead's evaluation of individual team member's contributions.
Team Lead: Alex Sanchez

Team Member Name	Contributions	Rating
Syed Faiz	Backend Lead	9 / 10
Victoria Wilson-Anumudu	GitHub Master	6 / 10
Vivek Santoki	Front End Lead	7 / 10
Eric Falk	Front end	5 / 10
Erik Rodriguez	Front end	2 / 10

Rating Scale: 1 (No work at all) – 10 (Contributed to the maximum capacity)

- Syed Faiz
    - Contributions
        - Documentation
            - Task Assigned: 4
        - Prototype
            - Setup the AWS database, setup table to connect to frontend when possible.
            - Added an individual "about me" section.
    - Rating: 9 / 10
        - Documentation: 5 / 5
        - Prototype: 4 / 5
            - Implemented backend part for prototype
            - Needs to implement search feature for backend
            - Added an "about me" code to web app

- Victoria Wilson-Anumudu
    - Contributions
        - Documentation
            - Task Assigned: 1, 6
        - Prototype
            - Added an individual "about me" section.
    - Rating: 6 / 10
        - Documentation: 5 / 5
        - Prototype: 1 / 5
            - Added "about me" code to web app

- Vivek Santoki
    - Contributions
        - Documentation
            - Task Assigned: 5, 7
        - Prototype
            - Created the first prototype; full working app, deployed to AWS.
            - Will add an individual “about me” section.
    - Rating: 7 / 10
        - Documentation: 5 / 5
        - Prototype: 2 / 5
            - Implement first prototype; full working app, deployed to AWS, but will not be used.
            - Added an “about me” code to web app

- Eric Falk
    - Contributions
        - Documentation
            - Task Assigned: None
            - Originally all tasks were assigned at the beginning of the milestone, and none were given to Eric.
                - I did not change this due to how close the due dates were when I took over as team lead.
        - Prototype
            - Created the backup prototype; full working app, deployed to both Google Cloud and AWS.
            - Will add an individual “about me” section.
    - Rating: 5 / 10
        - Documentation: 0 / 5
            - No documentation tasks assigned.
        - Prototype: 5 / 5
            - Implement backend prototype; full working app, deployed to AWS & Google Cloud. This prototype will be chosen for submission.

- Erik Rodriguez
    - Contributions
        - Documentation
            - Task Assigned: None
            - Originally all tasks were assigned at the beginning of the milestone, and none were given to Erik.
                - I did not change this due to how close the due dates were when I took over as team lead.
            - I want to note that Erik contributed to task 03, UI Mockups through verbal feedback on UX/UI, but that is about it.
        - Prototype
            - Will add an individual “about me” section.
    - Rating: 2 / 10
        - Documentation: 1 / 5
            - No documentation tasks assigned.
            - Contributed to task 03 verbally.
        - Prototype: 1 / 5
            - Added an individual “about me” section.
- Note
    - I would like to rectify how each team member contributed to both documentation and prototype in the next milestone. At the point I took over as team leader, I erred on not reassigning both documentation and prototype tasks since I prioritize finishing the documentation first before developing the prototype. With time constraints, I prioritize the overall milestone objective over individual contributions. However, this will change and I will make sure every team member will participate, contribute, and understand both documentation and prototype in the next milestone.
    - Given the ratings and notes, I hope my team members will understand my position.