Project Plan

Student Multi-Tool

02.27.2022

Team Marvel
Albert Toscano
Audrey Brio
Bradley Nickle
Devarsh Patel
Jacob Delgado
Joseph Cutri
Michael Kriesel (Team Leader)
SzeMan Tang

Introduction

The purpose of this document is to outline a plan for our project. It will include when we plan to have certain parts of our product to be completed as well as the expected cost and sprint plan.

Scope

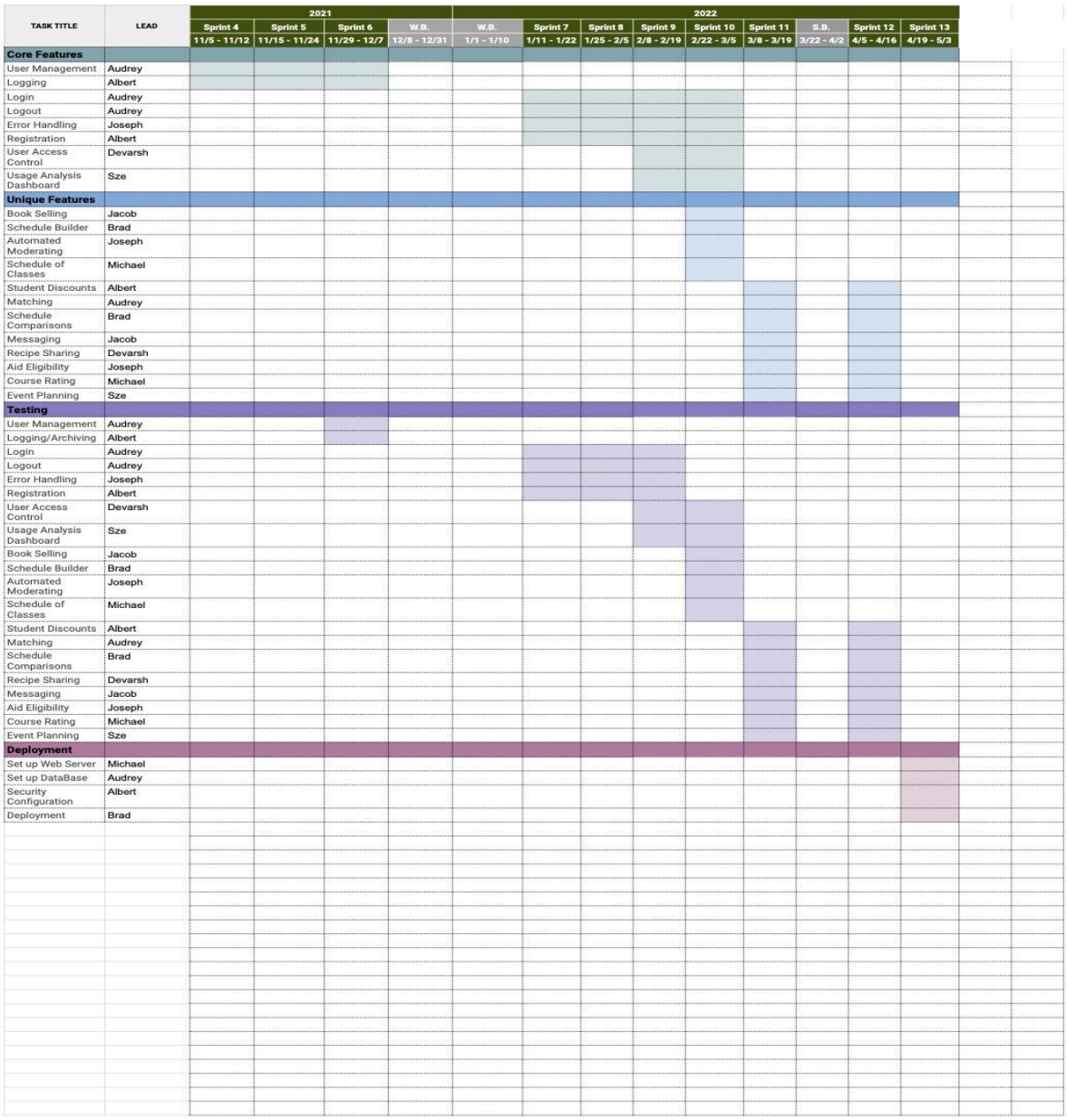
As our product is meant to be a web application, we plan to develop it for Chrome with the latest version available in January 2022. We plan on supporting each version until the latest version of Chrome available in April 2022. We will be creating this application using American English and American currency for any and all things dealing with money. Since our focus is on campuses in California, we will be using the Pacific Time Zone for times and date purposes. If there are any students using the web application that are not located in the Pacific Time Zone, dates and times will still show up in the Pacific Time Zone. We will be using a 24 hour clock system. An internet connection is required for all interactions with the system.

Deliverables

- Estimated Project Time - 1522 hours
    - Milestone 1 - Total Estimated Hours: 100
    - Milestone 2 - Total Estimated Hours: 349
    - Milestone 3 - Total Estimated Hours: 283 Hours
    - Milestone 4 - Total Estimated Hours: 490 Hours
    - Milestone 5 - Total Estimated Hours: 300 Hours
- Milestone One (completed by 12/15/2021)
    - Core Requirements
        - Logging/Arching - Total Estimated Hours: 50 Hours
        - User Management - Total Estimated Hours: 50 Hours
- Milestone Two (completed by 3/7/2022)
    - Core Requirements
        - Registration - Total Estimated Hours: 33 Hours
        - Login - Total Estimated Hours: 33 Hours
        - Logout - Total Estimated Hours: 33 Hours
        - Error Handling - Total Estimated Hours: 50 Hours
        - Network Communication -Total Estimated Hours: 50 Hours
        - Data Store Access - Total Estimated Hours: 50 Hours
        - User Access Control - Total Estimated Hours: 50 Hours
        - Usage Analysis Dashboard - Total Estimated Hours: 50 Hours
- Milestone Three (completed by 3/7/2022)
    - Application Features
        - Schedule of Classes - Total Estimated Hours: 70 Hours
        - Schedule Builder - Total Estimated Hours: 70 Hours
        - Book Selling - Total Estimated Hours: 70 Hours
        - Automated Moderating - Total Estimated Hours: 40 Hours
- Milestone Four (completed by 4/17/2022)
    - Application Features:
        - Student Discounts - Total Estimated Hours: 70 Hours
        - Messaging - Total Estimated Hours: 40 Hours
        - Schedule Comparison - Total Estimated Hours: 40 Hours
        - Course Difficulty - Total Estimated Hours: 70 Hours
        - Matching - Total Estimated Hours: 75 Hours
        - Aid Eligibility - Total Estimated Hours: 65 Hours
        - Recipe Sharing - Total Estimated Hours: 70 Hours
        - Event Planning - Total Estimated Hours: 60 Hours

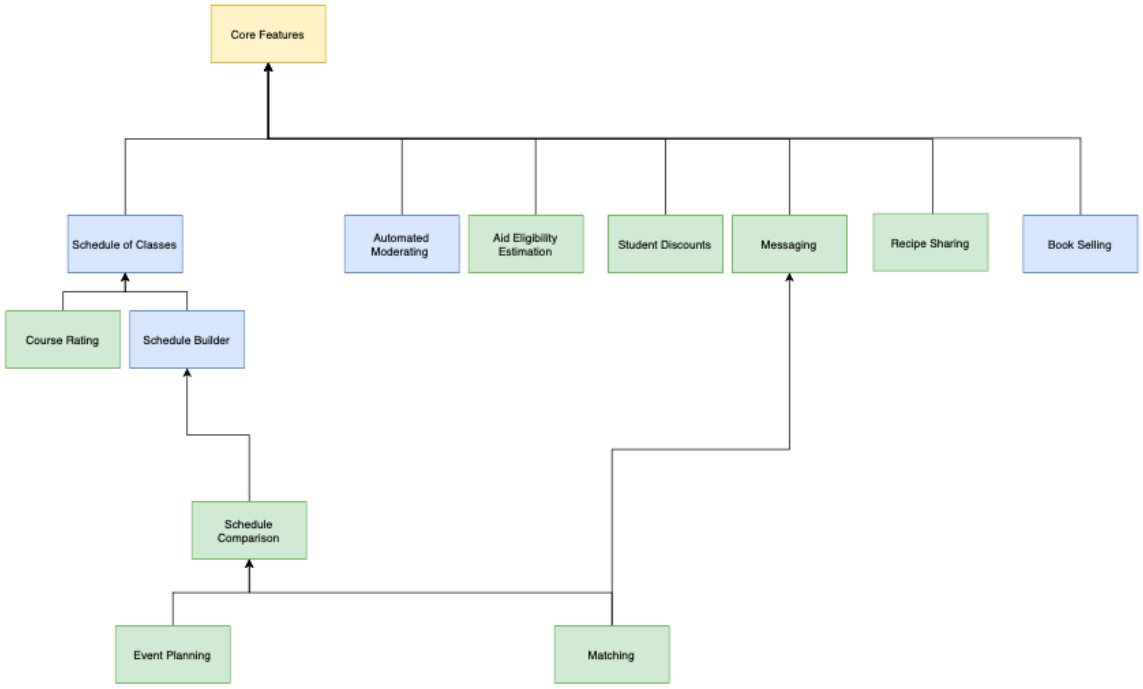
- Milestone Five (completed by 5/3/2022)
    - Testing
        - All system testing - Total Estimated Hours: 100 Hours
    - Deployment
        - Web server - Total Estimated Hours: 50 Hours
        - Database - Total Estimated Hours: 50 Hours
        - Security configuration - Total Estimated Hours: 50 Hours
        - Deployment - Total Estimated Hours: 50 Hours
- Deadline: Final Day possible to deploy 5/4/2022

Schedule



TASK TITLE	LEAD	2021				2022								
		Sprint 4	Sprint 5	Sprint 6	W.B.	W.B.	Sprint 7	Sprint 8	Sprint 9	Sprint 10	Sprint 11	S.B.	Sprint 12	Sprint 13
		11/5 - 11/12	11/15 - 11/24	11/29 - 12/7	12/8 - 12/31	1/1 - 1/10	1/11 - 1/22	1/25 - 2/5	2/8 - 2/19	2/22 - 3/5	3/8 - 3/19	3/22 - 4/2	4/5 - 4/16	4/19 - 5/3
Core Features														
User Management	Audrey													
Logging	Albert													
Login	Audrey													
Logout	Audrey													
Error Handling	Joseph													
Registration	Albert													
User Access Control	Devarsh													
Usage Analysis Dashboard	Sze													
Unique Features														
Book Selling	Jacob													
Schedule Builder	Brad													
Automated Moderating	Joseph													
Schedule of Classes	Michael													
Student Discounts	Albert													
Matching	Audrey													
Schedule Comparisons	Brad													
Messaging	Jacob													
Recipe Sharing	Devarsh													
Aid Eligibility	Joseph													
Course Rating	Michael													
Event Planning	Sze													
Testing														
User Management	Audrey													
Logging/Archiving	Albert													
Login	Audrey													
Logout	Audrey													
Error Handling	Joseph													
Registration	Albert													
User Access Control	Devarsh													
Usage Analysis Dashboard	Sze													
Book Selling	Jacob													
Schedule Builder	Brad													
Automated Moderating	Joseph													
Schedule of Classes	Michael													
Student Discounts	Albert													
Matching	Audrey													
Schedule Comparisons	Brad													
Recipe Sharing	Devarsh													
Messaging	Jacob													
Aid Eligibility	Joseph													
Course Rating	Michael													
Event Planning	Sze													
Deployment														
Set up Web Server	Michael													
Set up DataBase	Audrey													
Security Configuration	Albert													
Deployment	Brad													

Schedule Justification/Order of Feature Development



We have decided to develop our features in a particular order, using feature dependencies as our main justification for that order. To plan our development schedule, we have taken the following into consideration:

- If any of our core features are not completed on time, it will set us back on the application features, as the application features depend on the core features.
- In general, if any of the application features that have dependencies are not completed on time, it will set us back in developing those features that require it to be done. Specifically:
    - If we do not complete the schedule of classes on time, we will be behind schedule on completing the schedule builder, schedule comparison, and matching features.
    - If we do not complete messaging and schedule comparison on time, we will be behind schedule on completing the matching feature.
- The following features have no other features as dependencies, and are therefore less critical to our schedule:
    - Course Rating

- Matching
- Aid Eligibility Estimation
- Student Discounts
- Event Planning
- Recipe Sharing
- Book Selling
- Automated Moderating

While we would like to have dedicated sprints for testing and deployment, we expect that we will likely have features that are still in development and testing during sprints 12 and 13.

Features in blue correspond to features being done for milestone 3 and features in green correspond to those features being done for milestone 4.

Risk Identification & Mitigation

In this section we describe the initial risks of taking on this project. We have grouped our risks into categories: technical support, cost, and schedule. For each category, risks are ordered in terms of severity from highest to lowest. Severity uses a scale from 0 to 1, in which 1 is considered our most urgent risk(s), and 0 is not considered a risk at all.
We estimate the likelihood of each risk occurring on a scale from 0 to 1, in which 1 is a 100% chance of that risk actually occurring. A risk of 0 is a risk that has been resolved or fully mitigated.

Support Risks

1. RSU01: Access to scholarship and grant data
    a. Severity: 1
    b. Estimated Likelihood: 0.01
    c. Details: We need a way to access scholarship and grant data to fully implement the Aid Eligibility Estimation feature. Ideally, we would get this from a single source, but this may not be possible.
    d. Affected features
        i. Aid Eligibility Estimation
    e. Mitigation Strategy
        i. We have requested this data from CSULB enrollment services and they have approved the request.
2. RSU02: Access to schedule of classes data
    a. Severity: 1
    b. Estimated Likelihood: 0.8
    c. Details: We need a way to obtain the schedule of classes for a given school, which would make the Schedule Builder and Course Reviews simpler to use. Ideally, we would get this directly from the school, but this may not be possible.
    d. Affected features
        i. Schedule of Classes
        ii. Schedule Builder
        iii. Course Reviews
    e. Mitigation Strategy
        i. In case we do not get direct access to the data needed, we will develop a backup plan that uses web crawling
        ii. In case web crawling is not a viable option, we will develop a backup plan that uses crowdsourcing

3. RSU03: Lack of experience with web development
    a. Severity: 0.75
    b. Estimated Likelihood: 1
    c. Details: Most of our team is relatively unfamiliar with HTML, CSS, Javascript, apache, and other topics in web development.
    d. Affected features
        i. All
    e. Mitigation Strategy
        i. We will have time to learn the basic syntax and functionalities of these languages from November 2021 until late January 2022
        ii. We all have a fundamental understanding of object-oriented programming, and will be able to translate these skills to other languages as we go

Schedule Risks

1. RSS01: Documentation and implementation of backup plans for some features
    a. Severity: 0.5
    b. Estimated Likelihood: 0.75
    c. Details: Since some of our features require getting data from external sources, we may need to spend time creating a backup plan for one or more of those features.
    d. Affected features
        i. Aid Eligibility Estimation
        ii. Schedule of Classes
    e. Mitigation Strategy
        i. We are currently trying to get the data in a manageable way for each feature to ensure that we do not need to make backup plans
        ii. We have until late January to create and document our backup plans

Cost Risks

1. RSC01: AWS use fees
    a. Severity: 0.25
    b. Estimated Likelihood: 0.25
    c. Details: We consider the possibility of incurring AWS usage fees, whether expected or not, a risk. Since our estimated monthly cost is affordable for us for about 1 or 2 months, we do not consider this to be an urgent risk.
    d. Affected features
        i. All
    e. Mitigation Strategy

i. We have agreed to split costs at least 3 ways, reducing the cost to any one person.

Sprint Plan

Sprint	Start Date	End Date	Deliverables	Assigned
Sprint 5	11/15/2021	11/24/2021	Low Level Design Document	Audrey
			Logging/Archiving	Brad
			User Management	Audrey
Sprint 6	11/29/2021	12/07/2021	Low Level Design Document	Audrey
			Logging/Archiving	Brad
			User Management	Audrey
Sprint 7	1/11/2022	1/22/2022	Registration	Albert
			Login	Audrey
			Logout	Audrey
			Error Handling	Joseph
Sprint 8	1/25/2022	2/05/2022	Registration	Albert
			Login	Audrey
			Logout	Audrey
			Error Handling	Joseph
Sprint 9	2/08/2022	2/19/2022	Registration	Albert
			Login	Audrey
			Logout	Audrey
			Error Handling	Joseph
			User Access Control	Devarsh
			User Management	Jacob
			Usage Analysis Dashboard	SzeMan
Sprint 10	2/22/2022	3/05/2022	Schedule Builder	Brad

			Login/Logout	Audrey
			Schedule of Classes	Michael
			Registration	Albert
			Automated Moderating	Joseph
			Book Selling	Jacob
			Usage Analysis Dashboard	SzeMan
			User Access Control	Devarsh
Sprint 11	3/08/2022	3/19/2022	Matching	Audrey
			Messaging	Jacob
			Student Discounts	Albert
			Aid Eligibility	Joseph
			Course Review	Michael
			Schedule Comparison	Brad
			Event Planning	SzeMan
			Recipe Sharing	Devarsh
Spring Break	3/28/2022	4/01/2022		
Sprint 12	4/05/2022	4/16/2022	Matching	Audrey
			Student Discounts	Albert
			Aid Eligibility	Joseph
			Course Review	Michael
			Schedule Comparison	Brad
			Event Planning	SzeMan
			Recipe Sharing	Devarsh
			Messaging	Jacob

Sprint 13	4/19/2022	5/03/2022*	Deployment	Brad

*subject to change depending on the final deadline for completion of deployment.

Things we expect to be true of every sprint:

- Each Monday, before the start of the sprint on Tuesday, is the day of our sprint planning
- Retrospectives will be on the Sunday before the start of the next sprint on Tuesday
- We will continue to meet four times a week on Monday through Thursday
    - Days will be adjusted as needed
    - These meetings are where our daily standup meetings will occur
    - Meetings will occur on campus when possible and using Discord when necessary
- Sprint backlogs and a project backlog will be used to keep track of progress and work done
- Although we have a dedicated testing sprint, some testing will occur at the end of each sprint

Things we expect to accomplish during each sprint:

- Sprint 6
    - We plan on having user management and logging/archiving done by the end of sprint 6. We plan on doing user management and logging/archiving first since to meet the client's deadline for those features
- Sprint 7
    - We plan on working on registration, login, logout, and error handling during Sprint 7. Since Sprint 7 is planned for during break, we plan on beginning the features but are not expecting them to be finished by the end of the sprint.
- Sprint 8
    - We plan on continuing the work of Sprint 7 in Sprint 8.
    - While hit with something unexpected during this sprint, we had to readjust and move everything not finished this sprint to the next one
- Sprint 9
    - We plan on having most of the core features done with continuing working on usage analysis dashboard and user access control
- Sprint 10
    - We plan on having all the features done that we are going to present in the team code review done for this sprint
        - Those features are: Schedule of classes, schedule builder, automated

moderating, book selling, login/logout, registration, usage analysis dashboard and user access control
    - We plan on having done testing for all of these features as well
- Sprint 11
    - We plan on working on the rest of the application features during this sprint but do not expect to finish them by the end of this sprint
        - Those features are: Schedule comparison, messaging, matching, student discounts, aid eligibility, course review, recipe sharing, and event planning
    - We also plan on finishing any core requirements here if they are not finished already
- Spring Break
    - Our project plan includes not working during spring break. However, we can work during spring break to compensate for potential disruptions in our schedule.
- Sprint 12
    - We plan on finishing our application features during this sprint to be ready for individual code reviews
    - We plan on doing testing for these features here as well
- Sprint 13
    - At the end of milestone five, we plan on having deployed our project by the end of sprint 13. This will be the last sprint we have, and therefore our project should be done by the end of it.
    - At 14 days, sprint 13 is planned to be our longest sprint
    - We plan on doing all system testing here as well
    - The actual end date of this sprint is likely to change at the beginning of the spring 2022 semester, depending on when the client's deadline for deployment is.

Spring Break:

- Realistically, less work is likely to get done during a mid-semester break. Therefore, we are attempting to not over-rely on spring break by not planning specific work to be done at that time.
- However, we plan to use spring break as a buffer in case we fall behind on a feature or set of tasks. Ideally, we will not have to do this.

Deployment Plan

Item	Date of Completion
Set up Web Server	4/19/22
Set up DataBase	5/25/22
Security Configuration	4/30/22
Deployment	5/3/22

At this point in the process, everything will be ready to go and it will just be a matter of deploying a live version.

- Set up Web Server
    - Our web servers will hosted through Amazon Web Services (AWS)
- Set up DataBase
    - Our database will be hosted using Amazon RDS
    - Information for the database regarding some features must be present before deployment
- Security Configuration
    - This is a final step to ensure all security features are properly in place, and that the live product will be secure.
- Deployment
    - Our product will be officially deployed, and set live.
- After Deployment
    - After deployment we will continue to do regression testing
    - After deployment we will continue to monitor the status of the web app to verify that it is functioning properly

Resources

- Human
    - Team Marvel (5 people)
        - Albert
        - Audrey
        - Brad
        - Devarsh
        - Jacob
        - Joseph
        - Michael
        - SzeMan
- Time
    - Our time capacity is about 15 hours per person per week, or about 120 total hours per week. Over 23 weeks of development this is a total of 1725 hours (216 per person)
- Cost
    - Amazon Cloud - free tier
        - Overfee - see risk identification RSC01
    - Domain name - $12/year
    - Google Maps - Free tier
        - Up to 500 free requests per month
        - $0.01 for every request after
    - OS
    - IDE
    - Human Resources
        - 15 hours per week per person
        - About $5 per person
- Outside databases
    - Schedule of classes
    - Aid eligibility

References

- AWS General Reference
  https://docs.aws.amazon.com/general/latest/gr/Welcome.html
- Google Docs pricing
  https://rapidapi.com/googlecloud/api/google-maps-geocoding/pricing