InfiniMuse Project Plan

Project Plan

Hello World

Updated - April 17, 2021

Team Leader: Matt Caponi 011161928

Ian Macabulos 016401240

Jakub Koziol 016523856

Shaan Shariff 015741750

Wonsuk Seo 018660874

Table Of Contents

Project Plan 1

Table Of Contents 2

Revision History 2

Introduction 4

Statement of Intent 4

Goals 4

Scope 4

Budget 6

Risk Mitigation 6

Quality Control 12

Deliverables 13

Milestone 1 13

Milestone 2 13

Milestone 3 13

Code Review 1 14

Code Review 2 14

Code Review 3 15

Final Presentation 15

Sprint Analysis 15

Average Sprint 15

Task-Time Estimate Breakdown 15

Average Team Capacity 18

Average Velocity 19

Project Roadmap 27

Gantt Chart 27

Semester 2(Sprint 13 - Sprint 24) 27

Revision History

Date	Version	Description	Author
10/20/20	<0.1>	Started Outline	Matthew Caponi

1/18/21	<.2>	Started New Project Plan	Matthew Caponi, Wonsuk Seo, Ian Macabulos
1/20	<.3>	Tentative Schedule	Matthew Caponi, Wonsuk Seo, Ian Macabulos
1/21	<.4>	Finalized Risks	Ian Macabulos
1/23	<.5>	Added Gantt Chart and New Analysis	Matthew Caponi, Wonsuk Seo
1/27	<1.0>	Redid Time Estimation	Matthew Caponi
1/29	<1.1>	Reworked Component Chart, Reworked Time Estimation, Redesigned Gantt Chart	Wonsuk Seo, Matthew Caponi
1/30	<1.2>	Finalize	Matthew Caponi
2/05	<2.0>	Updated Gantt Chart	Wonsuk Seo
2/05	<2.1>	Updated Component List	Matthew Caponi
3/03	<3.0>	Redistributed components based on team member withdrawal.	Matthew Caponi
3/26	<3.1>	Updated component list	Matthew Caponi
3/26	<3.2>	Updated component list	Matthew Caponi
4/27/21	<3.3>	Updated gantt chart	Matthew Caponi

1. Introduction

1.1. Statement of Intent

- This project plan document was made to explain the overall, high-level process of building our system from the ground up. Our intent for making this document is to provide the client and other team members an overall view of the process that we will take in order to deliver a fully working system to the client within the given time period. This document will be used as guidance to ensure that our timeline is realistic and to hold us accountable to our deadline. This document also lays out our goals, budget, schedule, deliverables, and the potential risks we might face and the mitigation steps that our team has planned to address potential risks.

1.2. Goals

1.3. Project Goal

- Our project's goal is to build a web application that makes it easy for a user to match and collaborate with other users who are also looking for matches in similar areas. The core functionality of this application is an intelligent search engine that gives the user the ability to search for other users as if they were searching Google. This centralized, intelligent application eliminates the need to use a variety of different websites to find the matches that are needed, and makes it easy to find what one is looking for using natural language. It can be thought of as a central hub where users can connect and meet new friends, teammates, contributors, project collaborators, coaches, mentors, therapists, dates, and anyone else who shares an aligned purpose.

1.4. Scope

1.4.1. Project Documents

- Business Requirement Document
- High Level Design
- Tech Specs
- Site Map
- Test Plan Document
- Project Plan Document
- Network Diagram

- Low Level Design
- Logging
- User Management

1.4.2. Project Tasks

- Prepare planning documents
- Researching Technologies
- Learning programming languages, software frameworks, software architecture, and database design.
- Designing the application architecture and overall structure of the system.
- Develop each component.
- Test through an iterative process.
- Deploy

1.4.3. Confirmed Deadlines

- Milestone 1
    - October 10, 2020
- Milestone 2
    - October 31, 2020
- Milestone 3
    - December 12, 2020
- Retrospective 1
    - February 20, 2021
- Code Review 1
    - March 6, 2021
- Retrospective 2
    - March 20, 2021
- Code Review 2
    - April 3, 2021
- Retrospective 3
    - April 17, 2021
- Code Review 3
    - May 1, 2021
- Final Presentations
    - May 15, 2021

1.4.4. Final Goal of the Project

- The final goal of this project is to have a fully functional and secure web application that satisfies all business requirements (outlined in the Business Requirements Document) that is built to commercial software application standards. The application should be ready to deploy and use by the final project deadline.

2. Budget

Name	Rate	Quantity	Total Cost
Software			
Azure App Service	$0.10	500	$50.00
Azure Student Account	-$100	1	-$100
Subtotal			$0
Total Amount Left			$0

3. Risk Mitigation

3.1. Schedule Risks

Risk Criteria

- **Potential Risk** - Risks that might happen in the future. This can include expected risks based on decisions that have been made.
- **Resolved Risk** - These are risks that have already been dealt with. Risks can only be considered as resolved if the problem has been completely taken care of
- **Occurring Risk** - These are risks that are currently being dealt with by the team members. These risks can either be from the potential risks section or are completely unexpected and have to be addressed on the spot.

Risk Levels

- **Severe-** These risks are the high impact problems that can push the team back and have to be addressed first.
- **High-** These risks are still impactful problems that need to be addressed immediately but not before risks marked as highly severe.
- **Moderate -** These risks ,while impactful, can be dealt with rather easily and can usually be resolved quickly.
- **Low -** These are the risks that can be easily resolved within the group. These risks are usually fixed by proper communication and do not result in hours loss towards other tasks.

3.2. Missed Deadlines

- **Risk Criteria:** Potential risk.
- **Severity**
    - **High** - Missing deadlines will lead to point deduction and risk failing the class entirely. This will also affect upcoming sprints, as incomplete tasks will have to be moved to the next sprint, causing the team to fall further behind. This may also delay the project's timeline, which would delay the client's expected date of completion.
    - **Irreparability**
        - **Sprint' point of irreparability**
            - The sprint becomes unrecoverable if 3 out of 5 members of the team fails to complete their assigned tasks by the end of the sprint. This is because their tasks would then have to roll over to the next sprint causing a backup in tasks and ultimately causing the entire team to be behind schedule.
        - **Component' point of irreparability**
            - A component becomes unrecoverable if the person misses the code review deadline for that component due to lack of knowledge and research needed to complete that specific component.
        - **Project' point of irreparability**
            - The project becomes unrecoverable if 3 major components(intelligent search engine,Traditional Listing ,Traditional Listing Search) were not done by the deadline. The project also becomes unrecoverable if any 5 or more smaller components are not done by the deadlines even if the 3 major components were completed.

            - **Risk Mitigation**
                - Our team's plan to mitigate this risk is as follows:
                    - Ensure that every team member is working on work items as consistently as possible.
                    - Ensure that team members are seeking iterative feedback.
                    - Ensure that the team capacity and individual capacities are taken into consideration while sprint planning.
                    - Making sure that the work-load is proportionally distributed amongst the team members, based upon their individual capacities.
                    - Encouraging team members to communicate openly and often, seeking feedback when they need it, and being fully

transparent about any roadblocks that may impede their progress.

3.3. Operational Risks

3.3.1. Lack of Team Cohesiveness

- **Risk Criteria:** Potential risk

3.3.2. Severity

- **Severe**- This risk can cause tension, confusion, stress among team members, which will lead to lower quality work.
- **Sprint' point of irreparability**
  - The sprint becomes unrecoverable if 4 of the members of the team faced roadblocks that affected their ability to work with the team. Sprint becomes unsuccessful and past the point of recovery if work has not been done due to the factors listed above
- **Component's point of irreparability**
  - A component becomes unrecoverable if the reasons listed above affected the team members heavily to the point where it is unrealistic for the team members to work together cohesively to meet the complete the component
- **Project's point of irreparability**
  - The entire project becomes unrecoverable if the inner conflicts within the team go unresolved for multiple sprints. This is because with the scope and complexity of the project, any standstill caused by confusion and burnout will make the finishing of the project unrealistic.

- **Risk Mitigation**
  - Our team's plan to mitigate this risk, broken down by the roles responsibilities, are as follows:

    Team Lead
    - Clear expectations presented by the team lead.
    - Explicit delegation of tasks.
    - Small and tangible deadlines.
    - Frequent meetings to help assess the team status and clear roadblocks.
    - Iterative feedback to team members on tasks worked on and completed.
    - Individual attention paid to team members who may be struggling in certain areas.
    - Properly balanced constructive criticism that both aims to guide toward improvement while also building up team member confidence when work is done well.
    - Willingness to be patient with, work with, and accommodate team members who may not be fully meeting expectations.

- Resolve to take deceive and strong action when a team member is consistently refusing to meet expectations, despite repetitive attempts to help them.
- Willingness to listen to feedback the team may have about themselves, to better improve their leadership abilities and better help the team members succeed.
- Willingness to accept responsibility and admit fault if a team leader makes a major mistake.
- Enthusiasm for the project.

Team Members

- ○ Consistent, clear, and open communication among team members.
- ○ Honest and open communication about roadblocks with team members and the team lead.
- ○ Honest and open communication about any major personal struggles that are preventing the team member from being able to meet expectations.
- ○ Consistent effort put toward tasks and toward meeting expectations.
- ○ Consistent effort put toward constant improvement in both learning and team organizational skills.
- ○ Honest assessment of individual capacities, expertise, and personal capacities so that the team member does not overpromise and undeliver.
- ○ Willingness to help out other team members that may be struggling.
- ○ Willingness to take on tasks that may not be assigned but may need to be completed.
- ○ Willingness to accept responsibility and admit fault for under-delivering on expectations or on making serious mistakes.
- ○ Willingness to listen to feedback, including constructive criticism.
- ○ Commitment to improving upon feedback.
- ○ Enthusiasm for the project.

3.4. Technical Risks

3.4.1. Changes in software to be used for the project

- **Risk Criteria:** Occurring and Potential risk
- **Severity**
  - **Moderate**- This risk can impact some team members but not all, depending on everyone’s knowledge and skills. This risk can push back some team members as they will have to prioritize learning any new software that will be needed now or in the future. If appropriate attention is not paid to learning the technologies in advance, it can significantly impede progress on tasks that involve

the use of those technologies, as well as the quality of work for those tasks.

    - **Sprint' point of irreparability**
        - The sprint is deemed unrecoverable or irreparable if 3 out of the 5 members are behind due to this risk. This can cause the focus of the sprint itself from getting the work done to learning how to get the work done thus changing the original sprint plan.
    - **Component' point of irreparability**
        - The component itself is deemed unrecoverable if the person in charge of the component has determined that it is unrealistic to finish the component due to lack of both time and knowledge. The team will decide that this is unrecoverable if 2 other members are unable to help out with completing the component.
    - **Project' point of irreparability**
        - The entire project is deemed unrecoverable or irreparable if any 2 or more major components and/or any 5 or more minor components are affected by this risk and left undone to the point that it is unrealistic to finish the project by its last deadline.
- **Risk Mitigation**
    - Our team's plan to mitigate this risk is as follows:
      Team Lead
        - Scheduling ample time in some sprints for all team members to learn the technologies that they do not have expertise in.
        - Scheduling these "learning tasks" well in advance of when the relevant technological skills will be needed.
        - Requiring team members who are tasked with learning these tasks to practice and demonstrate their newly acquired skills through short projects.
        - Clearly defining the scope of the learning tasks, what needs to be learned and what is not relevant to implementation of the associated component/feature, within an individual technology.

      Team Members
        - Consistently working toward learning the technologies in accordance with the learning task deadlines, guidelines, and requirements.
        - Applying a curious and enthusiastic approach when it comes to learning the required tasks.

- Ensuring that they are leaning with the goal of learning in mind, not just with the goal of marking the task as done.
- Being open and honest about any roadblocks that may prevent them from learning the task on time or properly.

3.5. External Risks

3.5.1. Someone Leaving the team

- **Risk Criteria:** Potential risk
- **Severity**
  - **Severe -** Someone leaving the team due to any reason will result in more workload being put onto the other group members, as no new team member will be expected to join to alleviate the work of the person who leaves the team. Team capacity and velocity will also be lowered.
  - **Sprint' point of irreparability**
    - The sprint becomes unrecoverable if 2 out of the remaining 5 members has to leave the team for any reason. This will cause the complexity and scope of the project to be reduced however, having 3 people work on a complex application is not ideal.
  - **Component' point of irreparability**
    - A component becomes unrecoverable if the person working on that specific component leaves the team. This is because the other members will not be able to cover the amount that the person left and the component will have to be completely removed from the project.
  - **Project's point of irreparability**
    - The project becomes unrecoverable if 2 out of the last 5 members of the team has to leave the team for any reason. Though the scope and complexity of the project will have to be reduced and matched according to the number of members left, the workload might simply be too much for the remaining members.
- **Risk Mitigation**
  - Our team's plan to mitigate this risk is as follows:
    - **Team Lead**
      - Communicate clear and as often as possible to reduce the risk of tension as a result of miscommunication.
      - Give clear instructions so as to avoid miscommunication that leads to failed deadlines and team members dropping out of frustration.

        - Take into consideration the team members have other responsibilities and obligations outside of the project.
        - Treat team members with respect and don’t push them to do things the team lead themselves would not want to do.
        - Provide encouragement, use positive reinforcement, and work on building team members’ confidence.
        - Be attentive to team member needs and reach out if a team member is believed to be struggling.
        - Consistently and properly organize the project so that team members have confidence that the team lead has things under control, and thus does not leave for fear of team failure.
    - **Team Members**
        - Treat other team members with respect and dignity.
        - Communicate efficiently with other team members.
        - Avoid falling behind on work, so as to avoid having to offload onto other team members.
        - Be reliable, so that other team members do not have to stress about their own work as well as the work of others.

4. Quality Control

4.1. Project Documents Quality Management

- Our project goes or will go through a number of quality assurance checks before their deadlines to ensure that everyone on the team is happy with the current state of any deliverable before it has to be deployed. Each document made goes through constant feedback and revisions in order to ensure that every part of it is up to the expected standard of quality. These feedbacks can range from overall outline of the documents to checking and catching any errors made with the information given. This is done so that the important documents that explain our intent, goals,structure,and design are as clear as we want it to be.

4.2. Project Implementation Quality Management

- Software Quality Management
    - Software implementation for this project must be up to the standard of consumer grade systems that are being built with the purpose of selling the product. Our code must be checked to make

sure that the design of our solutions are efficient and not forced or hard coded. This quality assurance is made to ensure that our system passes all of the tests that are described in our test plan document.

5. Deliverables

Semester One

5.1.Milestone 1

5.1.1.Business Requirement Document

- Deadline
  - October 10, 2020 at 8:00 am

5.1.2.High Level Design

- Deadline
  - October 10, 2020 at 8:00 am

5.1.3.Tech Specs

- Deadline
  - October 10, 2020 at 8:00 am

5.1.4.Site-map

- Deadline
  - October 10, 2020 at 8:00 am

5.2.Milestone 2

5.2.1.Project Plan

- Deadline
  - October 31, 2020 at 8:00 am

5.2.2.Test Plan

- Deadline
  - October 31, 2020 at 8:00 am

5.2.3.Network Diagram

- Deadline
  - October 31, 2020 at 8:00 am

5.3.Milestone 3

5.3.1.Low-Level Design

- Deadline
  - December 12, 2020 at 8:00 am

5.3.2.User Management

- Deadline
  - December 12, 2020 at 8:00 am

5.3.3.Logging

- Deadline
    - December 12, 2020 at 8:00 am

Semester Two

5.4. Code Review 1

5.4.1. Registration
- Deadline
    - March 06, 2021 at 10:00 am

5.4.2. Traditional Listing
- Deadline
    - March 06, 2021 at 10:00 am

5.4.3. Notifications
- Deadline
    - March 06, 2021 at 10:00 am

5.4.4. Private Messaging
- Deadline
    - March 06, 2021 at 10:00 am

5.4.5. Research Intelligent Search Engine Algorithms
- Deadline
    - March 06, 2021 at 10:00 am

5.4.6. Refactor UM/Logging
- Deadline
    - March 06, 2021 at 10:00 am

5.4.7. User-Interactions
- Deadline
    - March 06, 2021 at 10:00 am

5.4.8. Security
- Deadline
    - March 06, 2021 at 10:00 am

5.5. Code Review 2

5.5.1. Login/Logout
- Deadline
    - April 3, 2021 at 10:00 am

5.5.2. Traditional Listing Search
- Deadline
    - April 3, 2021 at 10:00 am

5.5.3. Friends List
- Deadline
    - April 3, 2021 at 10:00 am

5.5.4. User Account Settings
- Deadline
    - April 3, 2021 at 10:00 am

5.5.5. Intelligent Search Engine

- Deadline
  - April 3, 2021 at 10:00 am

5.5.6. Security

- Deadline
  - April 3, 2021 at 10:00 am

5.6. Code Review 3

5.6.1. Archiving

- Deadline
  - May 8, 2021 at 10:00 am

5.6.2. User Analysis Dashboard

- Deadline
  - May 8, 2021 at 10:00 am

5.6.3. Group Pages

- Deadline
  - May 8, 2021 at 10:00 am

5.6.4. User Profile

- Deadline
  - May 8, 2021 at 10:00 am

5.6.5. Content Moderation

- Deadline
  - May 8, 2021 at 10:00 am

5.6.6. Security

- Deadline
  - May 8, 2021 at 10:00 am

5.7. Final Presentation

5.7.1. Deployment

- Deadline
  - May 15, 2021 at 10:00 am

5.7.2. Final Testing

- Deadline
  - May 15, 2021 at 10:00 am

5.7.3. Presentation Preparation

- Deadline
  - May 15, 2021 at 10:00 am

6. Sprint Analysis

6.1. Average Sprint

6.1.1. Task-Time Estimate Breakdown

- Time Estimation
  - Sprint 0

Total Estimated: 109 hours
Total Actual: 83 hours
Percent Error: 23.85%

  - Sprint 1

Total Estimated: 318.5 hours
Total Actual: 179.5 hours
Percent Error: 43.64%

  - Sprint 2

Total Estimated: 85 hours
Total Actual: 49.5 hours
Percent Error: 41.76%

  - Sprint 3

Total Estimated: 97.5 hours
Total Actual: 56 hours
Percent Error: 42.56%

  - Sprint 4

Total Estimated: 103.5
Total Actual: 105.6
Percent Error: -2.05%

  - Sprint 5

Total Estimated: 69
Total Actual: 48.2
Percent Error: 30.14%

  - Sprint 6

Total Estimated: 52
Total Actual: 57.5
Percent Error: -10.58%

  - Sprint 7

Total Estimated: 92
Total Actual: 101.8
Percent Error: -10.65%

  - Sprint 8

Total Estimated: 45
Total Actual: 55.25
Percent Error: -22.78%

- Completed Tasks
  - Sprint 0

Total Tasks Assigned: 58
Total Tasks Completed: 29
Percent Error: 50%

  - Sprint 1

Total Tasks Assigned: 72

Total Tasks Completed: 34
Percent Error: 52.78%

- Sprint 2

  Total Tasks Assigned: 18
  Total Tasks Completed: 13
  Percent Error: 27.78%

- Sprint 3

  Total Tasks Assigned: 25
  Total Tasks Completed: 33
  Percent Error: 24.4%

- Sprint 4

  Total Tasks Assigned: 24
  Total Tasks Completed: 15
  Percent Error: 37.5%

- Sprint 5

  Total Tasks Assigned: 19
  Total Tasks Completed: 12
  Percent Error: 36.84%

- Sprint 6

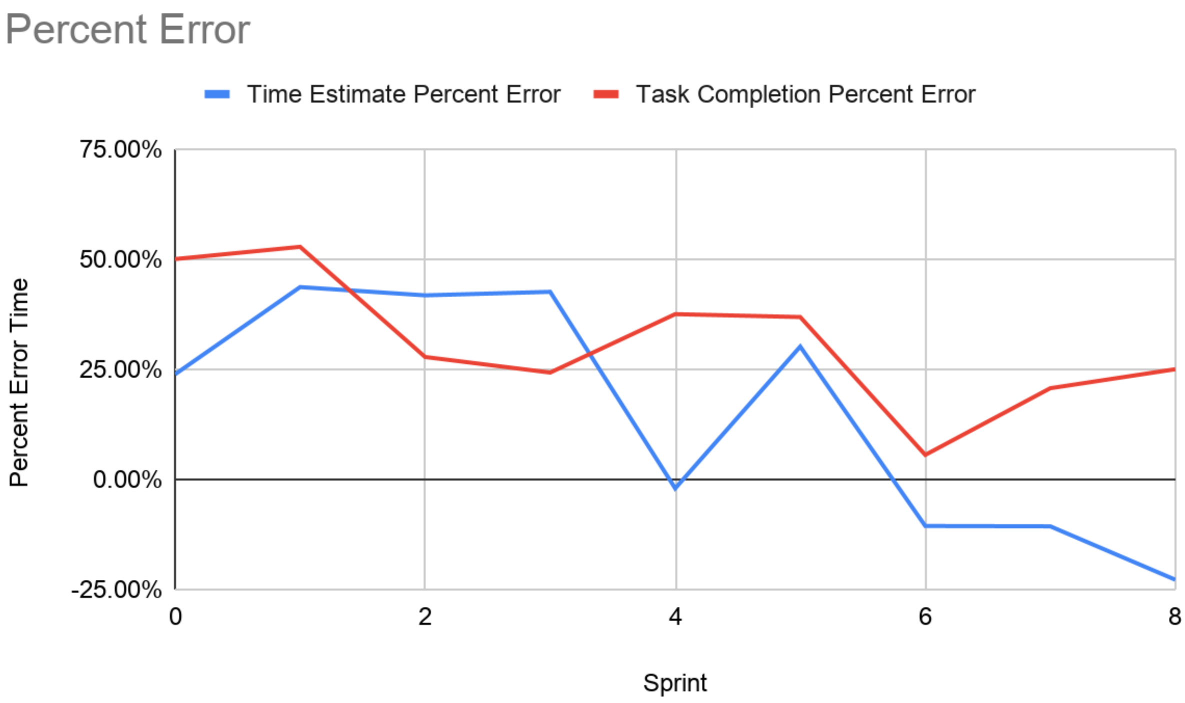
  Total Tasks Assigned: 18
  Total Tasks Completed: 17
  Percent Error: 5.56%

- Sprint 7

  Total Tasks Assigned: 20
  Total Tasks Completed: 23
  Percent Error: 20.69%

- Sprint 8

  Total Tasks Assigned: 12
  Total Tasks Completed: 9
  Percent Error: 25.00%

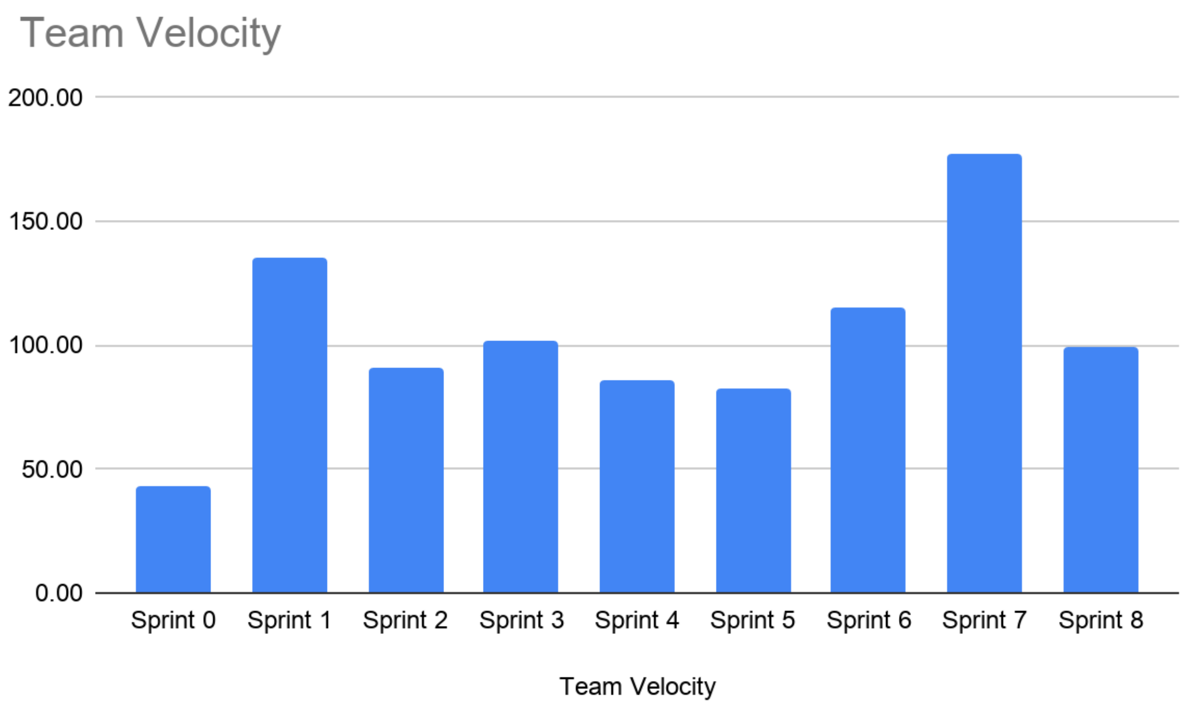


7. Average Team Capacity

Team Capacity (Normalized for 2 Week Sprints	
Sprint 0	120
Sprint 1	154
Sprint 2	154
Sprint 3	154
Sprint 4	154
Sprint 5	100
Sprint 6	120
Sprint 7	200
Sprint 8	100
Average Team capacity	139.56

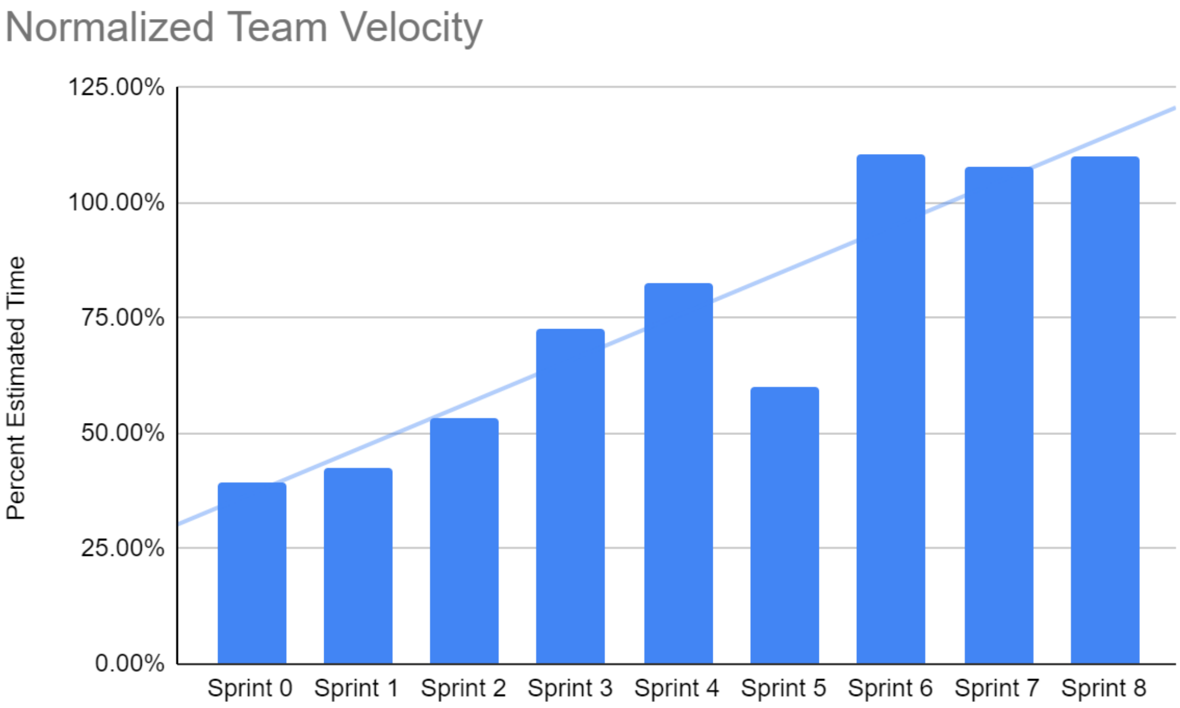
8. Average Velocity

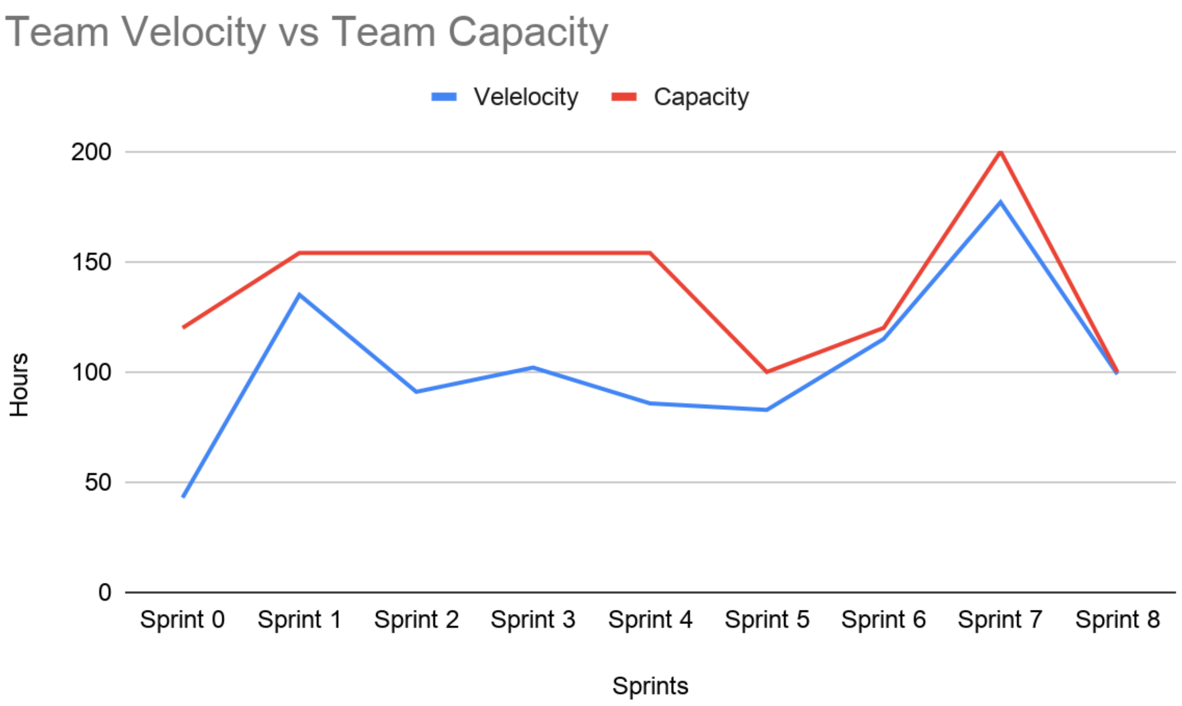
Team Velocity (Normalized for 2 Week Sprints)	**# Completed Hours**	**Total Hours**
Sprint 0	43.00	83.00
Sprint 1	135.00	180.00
Sprint 2	91.00	100.00
Sprint 3	102.00	110.00
Sprint 4	85.75	105.00
Sprint 5	82.80	96.40
Sprint 6	115.00	115.00
Sprint 7	177.00	203.60
Sprint 8	99.00	110.50
Average Sprint Velocity	103.39	122.61



Estimated Time	Normalized Team Velocity (Percentage of Estimated Time)
109	39.45%
318	42.45%
170	53.53%
140	72.86%
103.5	82.85%

138	60.00%
104	110.58%
164	107.93%
90	110.00%





Sprint #	Proportion of Velocity to Capacity
Sprint 0	35.83%
Sprint 1	87.66%
Sprint 2	59.09%
Sprint 3	66.23%
Sprint 4	55.68%
Sprint 5	82.80%
Sprint 6	95.83%
Sprint 7	88.50%
Sprint 8	99.00%
Sprint 9	NA
Sprint 10	NA
Sprint 11	NA
Sprint 12	NA

9. Project Estimations

9.1. Sprints

9.1.1. Number of Sprints Remaining: 9

9.1.2. Mode Sprint Time: 2 Weeks

9.2. Sprint Timeline

Sprint #	**Sprint Start**	**Sprint End**
Sprint 0	September 7	September 20
Sprint 1	September 21	October 4
Sprint 2	October 5	October 18
Sprint 3	October 19	October 25
Sprint 4	October 26	October 31
Sprint 5	November 17	November 23
Sprint 6	November 23	November 30
Sprint 7	December 1	December 9
Sprint 8	December 10	December 14
Sprint 9	January 5	January 12
Sprint 10	January 12	January 20
Sprint 11	January 20	January 22
Sprint 12	January 22	January 23
Sprint 13	January 24	January 30
Sprint 14	January 31	February 6
Sprint 15	February 7	February 20
Sprint 16	February 21	March 6
Sprint 17	March 7	March 20
Sprint 18	March 21	April 3
Sprint 19	April 4	April 17
Sprint 20	April 18	May 1
Sprint 21	May 2	May 15

10. Estimated Hours and Velocity

Estimated Project Hours Remaining	1640
Total Project Velocity Remaining	827.12
Estimated Hours Needed Per Sprint	205
Average Velocity	103.39
Estimated Sprint Velocity Needed Per Person	41.00
Current Trajectory Sprint Velocity Per Person	20.68

11. Component Point Estimations

11.1. We made an estimate of the points that each component will be worth based on our interpretation of their individual complexity.

11.2. These estimations were based upon the syllabus and code review sheet that Professor Vong had posted to his Github.

		Code Review 1	**Not Graded**	**Code Review 2**	**Not Graded**	**Code Review 3**	**Not Graded**	**Total Points**
Name	*Shaan*							
Component		***Registration***		***Login/Logout***		***Archiving***		
Estimated Points		100		100		50		250
Name	*Ian*							
Component		***Traditional Listing***		***Traditional Listing Search***		***User Analysis Dashboard***	***Admin Dashboard***	
Estimated Points		75		100		100		275

Name	*Jake*							
Component		***User Account Settings***	**Security**	***Messaging***		***User Profile***	***Friends List***	
Estimated Points		100		125		75		300
Name	*Matt*							
Component		***User Access Control/ User Access Control Dashboard***	***UM/Logging***	***Content Moderation***	***Notifications***	***Intelligent Search Engine***		
Estimated Points		50		125	**User Interactions**	125		275

12. Analysis

12.1. Current Outlook

12.1.1. The current trajectory of our sprint velocity indicates that we won't be able to meet the necessary hours to complete all components, however, we believe we can far exceed our current velocity.

12.2. Extenuating Circumstances

12.2.1. Group conflict and the need to restart the project caused significant delays that resulted in less improvement than we would have otherwise had during sprints 5-8.

12.3. Positive Trends

12.3.1. When looking at the change in rate of the normalized velocity, there is a trend of consistent growth. While it stalled a little bit during the last 3 sprints, that was due to the extenuating circumstances.

12.3.2. When looking at the change of the proportion of the sprint velocity to the sprint capacity, the difference between the two has a trend of narrowing, with the difference being only 1% on the last sprint, even during the stalled time.

12.4. Possible Ways to Reduce Estimated Hours Needed

12.4.1. Analysis

- Cut out tasks

    - While cutting out tasks would help, we could only at most cut one moderate task, as cutting out anymore would jeopardize the ability for all of our team members to reach 250 points.
- Reduce complexity of tasks
    - Same as cutting out tasks, we could only reduce the complexity by a moderate amount of one task to avoid jeopardizing the ability for all team members to reach 250 points.

12.4.2. Conclusion

- Given the time it would take to rework our BRD and project plan to factor in the changes and the time it would take to get approval for these changes, and also given the little benefit to doing so, it would not be worth it to cut out tasks or reduce complexity. Instead, we need to assess whether our rate of sprint metrics indicate the ability to rise to meet the challenge.

12.5. Changes to Facilitate Necessary Velocity

12.5.1. While the trends are positive, it takes more than just trends to ensure that we meet our needed velocity. We plan on taking the following actions to ensure we can complete the project on time:

- More frequent scrums
    - More frequent scrums ensure that we keep each other on task and can’t procrastinate.
- Team Code Reviews
    - Assigning tasks whereby group members review the code of other group members to assist with any bugs and to ensure quality.
- Plan Ahead
    - In addition to the plans made in this project plan, the team understands that it will take on average, 2.93 hours per day of work on this project in order to meet our time estimate. By understanding exactly what it takes, team members can plan ahead and make sure they work consistently at the right amount to meet our goals and finish on time.
- Shift in Mindset
    - After an entire semester worth of experience, the team has already had a shift in mindset in the work required to succeed on this project. After working on a new BRD and Project Plan together, the team now understands the vast amount of work that lays ahead and is already in the process of shifting their mindset to be able to handle this work, and thus increase their estimated capacities.
- Learn from Experience

    - Our team has also had an entire semester worth of experience in making mistakes. By learning from these mistakes, we can learn from to improve this next semester. This should continue to contribute to our growth in team velocity to match our growth in estimated capacity.

12.5.2. Conclusion

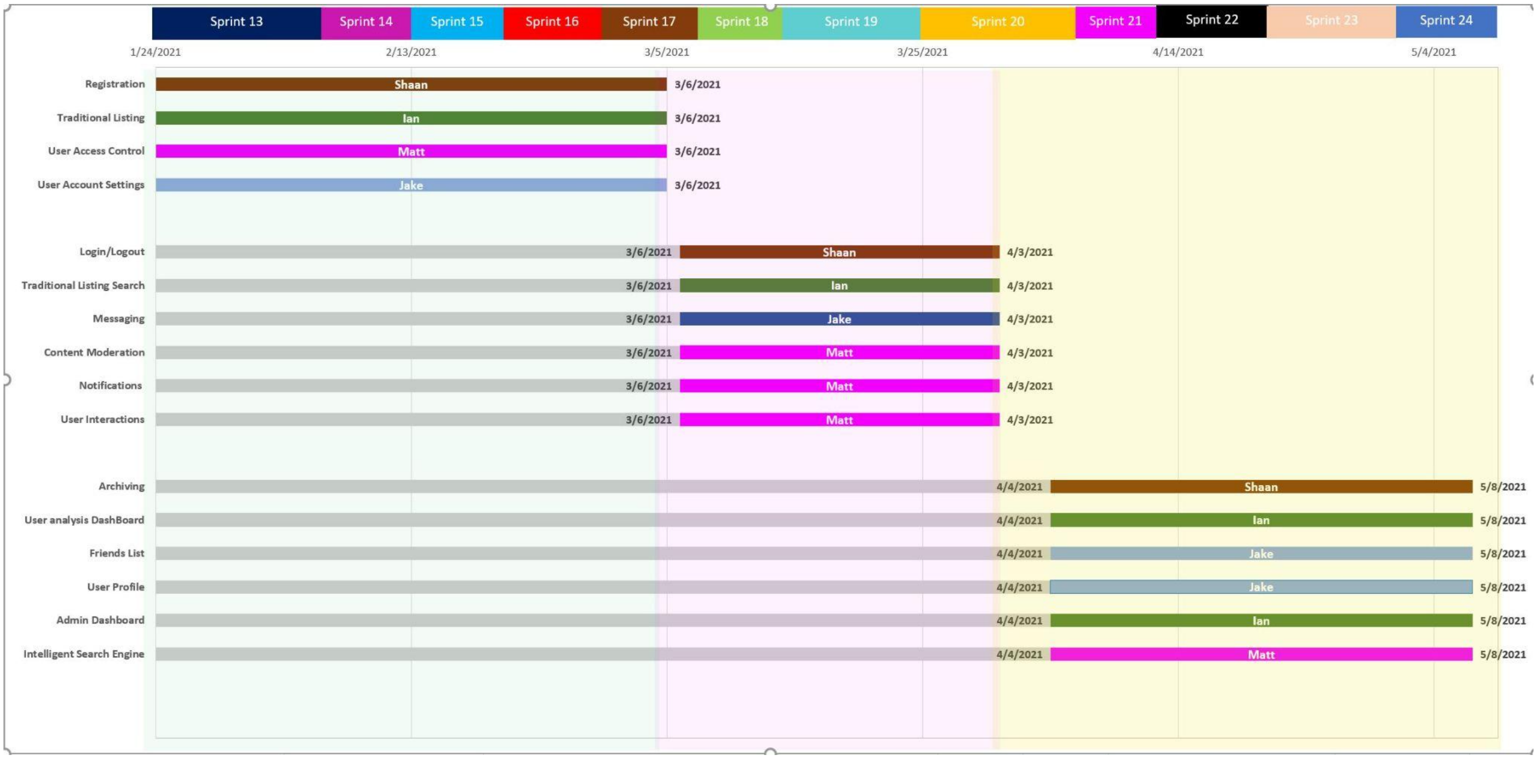
- Based on our analysis, we should be able to meet the challenge of completing all components we have assigned in order to complete this project by the deadline.

13. Project Roadmap

13.1. **Gantt Chart**

This schedule was developed by determining the hours we estimate each task to complete over the course of our project, and then determining how many tasks we could fit in per sprint, based upon our estimated average project velocity.

13.1.1. Semester 2(Sprint 13 - Sprint 24)


Sprint 13
Sprint 14
Sprint 15
Sprint 16
Sprint 17
Sprint 18
Sprint 19
Sprint 20
Sprint 21
Sprint 22
Sprint 23
Sprint 24
1/24/2021
2/13/2021
3/5/2021
3/25/2021
4/14/2021
5/4/2021
Registration
Shaan
3/6/2021
Traditional Listing
Ian
3/6/2021
User Access Control
Matt
3/6/2021
User Account Settings
Jake
3/6/2021
Login/Logout
3/6/2021
Shaan
4/3/2021
Traditional Listing Search
3/6/2021
Ian
4/3/2021
Messaging
3/6/2021
Jake
4/3/2021
Content Moderation
3/6/2021
Matt
4/3/2021
Notifications
3/6/2021
Matt
4/3/2021
User Interactions
3/6/2021
Matt
4/3/2021
Archiving
4/4/2021
Shaan
5/8/2021
User analysis DashBoard
4/4/2021
Ian
5/8/2021
Friends List
4/4/2021
Jake
5/8/2021
User Profile
4/4/2021
Jake
5/8/2021
Admin Dashboard
4/4/2021
Ian
5/8/2021
Intelligent Search Engine
4/4/2021
Matt
5/8/2021