On Call
Project Plan

A. Archer (aa), K. Hosaka (kah), A. Huque (ash), L. Jim (lmj), M. Terry (mht)

3-08-2020 - v3.10

Table of Contents

1. Project Plan Revision History

Date	Author	Description
2-12-2020	lmj	Created the initial document.
2-12-2020	kah	Wrote project plan.
2-12-2020	lmj	Created Initial Gantt chart.
2-13-2020	lmj	Updated task breakdown. Proofreading and minor edits.
2-13-2020	ash	Edited 6.3 and proofread section 3.
2-14-2020	aa	Created PERT Chart. Wrote section 5.
2-14-2020	lmj	Proofreading and minor edits.
2-14-2020	aa	Proofreading, small comments and edits.
2-14-2020	mht	Final edits and comments before submission.
2-15-2020	lmj	Updated task breakdown.
2-17-2020	ash	Added to section 7.3
2-18-2020	mht	Added meeting notes from the 16th of February.
2-18-2020	kah	Initial project plan completed.
2-19-2020	ash	Updated 5.2 with new Gantt chart, formatting, adding milestones
2-19-2020	lmj	Updated task breakdown.
2-19-2020	ash	Added section 6
2-19-2020	mht	Edited section 7
2-19-2020	ash	Adding 2-19-2020 meeting summary, updated Gantt chart
2-20-2020	ash	Fixed error with Gantt chart
2-20-2020	mht	Final edits before submission
2-26-2020	ash	Added updated Gantt Charts and 6.2., updated section 4 and 7.3.
2-27-2020	aa	Pert chart update 5.2.1., 5.2.2.
2-28-2020	ash	Updated task breakdown.
3-01-2020	kah	Minor edits.
3-03-2020	ash	Adding 3-01-2020 meeting notes
3-04-2020	ash	Updated testing, added meeting summaries, updated milestones
3-04-2020	kah	Updated section 6.2.5. Unit Testing - output.py
3-05-2020	ash	Wrote section 3, 6.2.2., 6.2.6., 6.2.7.
3-06-2020	aa	Updated section 4 pert charts for 3-04, 3-08
3-06-2020	ash	Edited section 2, reformatted project plan, added 3-06 meeting notes
3-07-2020	ash	Editing and proofreading
3-07-2020	lmj	Wrote section 6.2.1. Updated task breakdown. Proofreading.
3-08-2020	ash	Updated 5.2 and 7.4, proofreading
3-08-2020	mht	Updated section 7 and 6.2.3, proofreading

2. Project Plan

2.1. Management Plan

The *On Call* development team, Team LLC, has organized the project into a few main components. While all group members collaborated on all components, we assigned individuals as managers and main programmers for specific components to ensure project productivity.

Programming assignments:

- Alex programmed the weekend schedule generator.
- Kiana programmed the system output.
- Alyssa programmed the system input.
- Lily programmed the user interface which involved integration.
- Max programmed the weekday schedule generator and summary reporter.

Documentation assignments:

- Alex: SDS section 5, Project Plan PERT Charts
- Kiana: Project Plan, SDS section 4
- Alyssa: SRS section 2, Gantt chart and milestone spreadsheet for Project Plan
- Lily: SDS sections 2 - 4 including models
- Max: SRS section 3

Team LLC made decisions and communicated regularly throughout our development process. Our group met a minimum of twice a week, Sundays at 12:00 pm in Deschutes 100 and Wednesdays at 3:00 pm in PSC B004. During these meetings we made larger decisions, reviewed our progress from the past few days, and discussed the next steps to take. Outside of these meetings, we maintained communication through our mobile devices and Discord which provides cloud based text and voice chat. Our development team decided on programming in Python 3.7 and using a private Github repository for our development.

2.2. Breakdown Schedule

Our milestones and project schedule were as follows. See Section 5. “Project Timeline” for more details:

- **2-14:** Plan project (all).
- **2-14:** The SRS ConOps (Kiana, Alyssa), SDS System Overview and Architecture (Lily), Project Timeline (Alex, Lily) was completed.
- **2-14:** Initial Presentation (all) was completed.
- **2-19:** The initial SRS (Alyssa, Max) and SDS (Alex, Lily) was completed.
- **2-20:** Project Plan (Kiana, Alyssa) was completed.
- **2-25:** The RA Preferences Module (Alyssa), error checking (Alyssa) and Shift Assignments Module (Kiana) was completely coded.

- **2-26:** Weekend Scheduler Module (Alex) and basic On Call Viewer (Lily) was completely coded
- **3-1:** Weekday Scheduler Module (Max), On Call Viewer (Lily), and module integration (Lily, all) was completed
- **3-6:** Testing (all), finalized application (all), and Project Plan (Kiana, Alyssa), SRS (Alyssa, Max), SDS (Alex, Lily), and Technical Documentation (all) were completed.
- **3-8:** Documentation (all) was proofread and finalized.
- **3-11:** Project presentation (all) will be completed.

2.3. Monitoring and Reporting

Team LLC monitored individual and project progress throughout our development process. As mentioned, we used GitHub and biweekly meetings for monitoring and reporting. We created separate branches on GitHub for each group member. Above those branches, we created a "testing" branch that we merged to when we believe that our individual branches were working correctly. We then reviewed the testing branch at the beginning of our group meetings to get a verbal agreement from all group members to push the testing branch to the master branch. We also discussed our progress from the previous few days and what we were each planning on doing for the next few days. We updated the milestones Excel spreadsheet that contains our task schedules and tracked our progress throughout the project duration. See Section 5. "Project Timeline" for iterations of this spreadsheet.

2.4. Build Plan and Rationale

Team LLC decided that the sequence of our build plan was finalizing our initial documents, the RA Preferences Module, Shift Assignments Module, Weekend Scheduler Module, Weekday Scheduler Module, On Call Viewer Module, and then finalizing our final documents. Our rationale for this build plan is based on what we believed made sense in terms of temporal order. We broke them into these components because we believed that they were the larger milestones. By 2-25, we began the integration of our modules. The RA Preferences Module, Shift Assignments Module, Weekender Scheduler Module, and Weekday Scheduler Module were completed in its most basic form before the integration of the On Call Viewer module because the On Call Viewer module brought all of the modules together. Because we accounted for potential issues with integration, we allocated ourselves four days to complete the integration on 2-28. In reality, it took six days and was completed by 3-1. By then, we had over a week before the final submission date to make small changes to our application and conduct thorough testing. We integrated our modules in person to ensure that we have verbal agreements that we are prepared to do so and to ensure easier communication throughout the integration process. A risk that we faced was understanding the time it would take to complete our milestones. To reduce the chances of risk, we stated that the desired completion date was earlier than necessary for the final completion date. Additionally, we planned to finish *On Call* earlier than the intended due date, so any slippage that occured did not impact our ability to submit the project on time. We also ensured that we were updating the documentation throughout the development processes.

3. Project Plan Reflection

The *On Call* development team, Team LLC, found our project plan to be highly effective. Our team adhered to the majority of hard deadlines, deliverables, and milestones. It was anticipated there would be slippage so our initial project time planned on completing the project one week before the official deadline. This allowed slippage time to occur without inhibiting our team's ability to meet the class project deadline. Additionally, the modules in the project had loose coupling which allowed individuals to successfully complete modules in parallel and meet integration deadlines. Finally, our team was structured as an agile team which reduced errors, improved integration, and gave all members a better understanding of every mechanism in *On Call*. All these aspects provided our team with the proper tools and planning to follow our project timeline without missing any deadlines or encountering major delays.

Our team utilized soft and hard deadlines to ensure every member stayed on track and mitigate risk. Every Wednesday there was a soft deadline for work. This meant that, by Wednesday, it was expected that each member would make progress on their task and have work to show for it. However, it was acceptable if there were still bugs or errors by the soft deadline. The soft deadline ensured no individual procrastinated on their assignments and every member, at the very least, began thinking about how to best approach their task. Sundays were a hard deadline. A hard deadline meant that the assigned task must be fully completed. This allowed our group to evaluate all completed work during our Sunday meeting. If a hard deadline was not met, the entire project would be delayed. However, since there was a soft deadline on Wednesday we did not encounter this situation proving this method successfully worked as a risk mitigation technique.

The modularity of our project ensured individuals were allowed to successfully work on the project to meet deadlines. The project was divided into five main parts—RA Preferences, Weekday Scheduler, Weekend Scheduler, Shift Assignments, and On Call Viewer.

- Alyssa was in charge of RA Preferences. This consisted of parsing a given file and storing the information. Alyssa was responsible for writing the application's ability to import a file with the RA's preferences, permanently delete an RA from the system, reset the preferences stored in the system, update any adjustments made through the graphical user interface (GUI), and the undo function for RA preference changes. Additionally, Alyssa was in charge of error checking to make sure that the imported file is the correct format and that all the RAs in the system have made reasonable requests for preferences and weekends off.
- Max worked on the Weekday Scheduler module. This module had the algorithm for sorting RAs to a designated weekday based on their preferences, balancing the number of primary and secondary shifts each RA received, and implementing the method chosen by the user for breaking ties. Max also wrote the system's capabilities to generate a report of the schedule.
- Weekend Scheduler was assigned to Alex. Alex's responsibilities consisted of assigning RAs to weekend day and night shifts, balancing the number of primary and secondary

shifts each RA received, and ensuring no RA was assigned to a weekend they had requested off.

- Kiana coded the Shift Assignments module. Kiana was responsible for writing the application's ability to consolidate the weekend and weekday scheduler into one schedule, export a file with the schedule to be saved locally, reset the schedule, update any adjustments made through the GUI, and the undo function for schedule changes.
- The On Call Viewer module was the GUI. This module is everything the user interacts with. The viewer allows the user to interact with each module, provides the user with limited but necessary descriptions on how to use the system, and raises warnings when the user has made an invalid request.

Our team was structured like an agile team (van Vliet, 2008). This was successful since our team was small and all proficient in writing code as well documentation. As van Vliet describes, this method uses a pilot and a co-pilot and does not have a strict hierarchical structure.

- Alyssa was the pilot for documentation concerning project timelines and task schedules as well as interviews. This involved maintaining the task schedule, writing meeting summaries, interviewing people, and writing transcripts/notes of each interview.
- Kiana was the pilot for technical documentation making sure project technical documents were updated and accurate.
- Alex was the pilot for diagrams and proofreading.
- Max was the pilot for algorithm functionality and testing the application. This required testing system based testing to ensure the system did not break.
- Lily was the pilot for integration and creating the user interface.

All other members of the group assisted the pilots in their tasks, acting as co-pilots. This allowed everyone in our group to reach levels 2 and 3 of understanding, detaching and fluent respectively (van Vliet, 2008). Since all group members had a solid grasp of the various components of our project, we were able to write code that could be integrated faster. Additionally, since all members of the group reached these deeper levels of understanding, it allowed us to be more proactive to potential logic errors and have a faster response when unforeseen errors did occur, especially during integration.

4. Meeting Summaries

Date	Minute Taker	Description
2-12-2020	lmj	- All document templates created and content started - Private GitHub repository created - Alyssa explained how RA on call scheduling works - Assigned people to initial draft sections: Alyssa and Kiana to SRS ConOps, Lily to SDS System Overview and Architecture, Alex to Project Timeline, and Max to proofread everything. - Assigned people to initial documents: Alex to SDS section 5 and Project Plan charts (except Gantt), Lily to SDS section 2-4 and Gantt chart, Kiana to Project Plan and help as needed with SRS section 2, Alyssa to SRS section 2 and help as needed with Project Plan, and Max to SRS section 3. - System Architecture and code distribution decided upon: On Call Viewer (Lily), RA Preferences (Alyssa), Weekday Scheduler (Max), Weekend Scheduler (Alex), Shift Assignments (Kiana).
2-16-2020	mht	- Decided to include "snapshot" spreadsheets that show where the team was at any given week. - Decided we would be including three Gantt charts: the initial one, a Gantt chart halfway through the project, and a final Gantt chart at the end that reflects slippage time. - Came up with five non-mechanical human interactions (decisions that can only be made "in the head") that our program will reflect: updating preferences, modifying schedules (swapping shifts), flagging RAs from taking certain shifts, selecting one RA who gets their top preference, and a tie breaker prompt for when two people have identical preferences. - Went over how each module should work. The five modules in their most bare forms are: On Call Viewer (largely GUI interaction), RA preferences (file input), Weekday/Weekend Schedulers (the algorithm that assigns shifts), and Shift Assignments (file output). - We decided to use a dictionary for data storage. The key is the student's ID. - Made the standard that we will be using suggestion mode when making edits on other people's parts of the documentation. - The standard for code documentation will be a docstring at the top of each function which highlights the inputs/output types of the function, as well as a brief description.
2-19-2020	ash	- Reviewed diagrams and went over all three of the initial documents - All initial documents were finished, only needed to make grammar edits, agreed Max will submit on Thursday

		- Created a new discord channel to allow everyone to better communicate how long they spent on each task - Went over data structures and data types that will be utilized throughout the system - Established there will be a shift assignments dictionary and preference dictionary - Shift assignments dictionary = {week #: [[primary], [secondary]]}. primary/secondary = Sun_d, Sun_n, Mon, Tues, Wed, Thurs, Fri, Sat_d, Sat_n (list of ordered names for weeks 1–10) - Preference dictionary = {student ID: [FirstLastName, day1, day2, day3, end1, end2, end3,], the key 1 will contain the gold star RA student ID number, the key 2 will have the selected tiebreaker setting (0-2), the key 3 will have a list of lists of bad pairings (at most two)
2-23-2020	ash	- Alyssa discussed with the group the specific mechanisms of the settings with group - Settings need to accept None for no goldstar or bad pairing selected - decided "delete RA" should be a seperate button in the GUI - Lily walked everyone through what the GUI does so far - Discussed what error checking needs to be implemented - Discussed the output and necessary revisions: there should be 10 weeks instead of 11, there is excess whitespace in header, extra spaces after commas, unnecessary heading - Went over Alex's/Max's algorithm ideas - Reiterated next week's milestones (code done by 2/26 with the exception of GUI because of integration) - Integration/GUI will be done by 3/1
2-26-2020	ash	- discussed undo function and integration with it - Alyssa realize an update function was needed and implemented it at meeting - Algorithms are not finished, further discussion about how to sort people into weekdays - Finished undo functions - Addressed issues Alex was having with Git - Asked Alex how Alex was testing code, considering the issues he was having with Git
3-01-2020	ash	- Kiana added reset functionality to shift assignments module - Max verified the input to weekday scheduler and discussed when settings would be put into the dictionary and error checking - merged to master branch - Max walked everyone through the weekday algorithm and stated that only bag pairings and the two tiebreaker settings were left to implement - Discussed how to document testing, Alyssa will email Hornof for input - Went through GUI and gave Lily feedback

3-04-2020	ash	- Clarified vocabulary used throughout SDS (i.e. user vs. coordinator) - Decided against clearing out previously generated schedule when clearing out systems in case user forgot to export previous schedule - Discussed generating a report for user to see number of assigned shifts - Alyssa has implemented an error for user inputting more than 25 RAs - Went over what was needed from everyone to document testing - Will zip code file on 3-05-2020
3-06-2020	ash	- Discussed project plan formatting and best approach - Planned proofreading schedule - Need to add RA deletion to SDS - Add scheduler comparison to SRS - Brought up error with exporting summary, resolved it at meeting - Needed Lily and Max to add to their testing section - While code file is zipped, please continue to do system testing in order to be thorough.
3-08-2020	ash	- Ensured everyone proofread the documentation - Made sure the most recent version of each code file was included in the zip file - Discussed clarifying questions on SDS for accuracy - Alyssa double checked project plan formatting for ease of reading - Max brought up questions about SRS to make sure his understanding was accurate - submitting project 2! - Prepared for presentation

5. Project Timeline

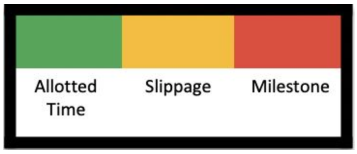
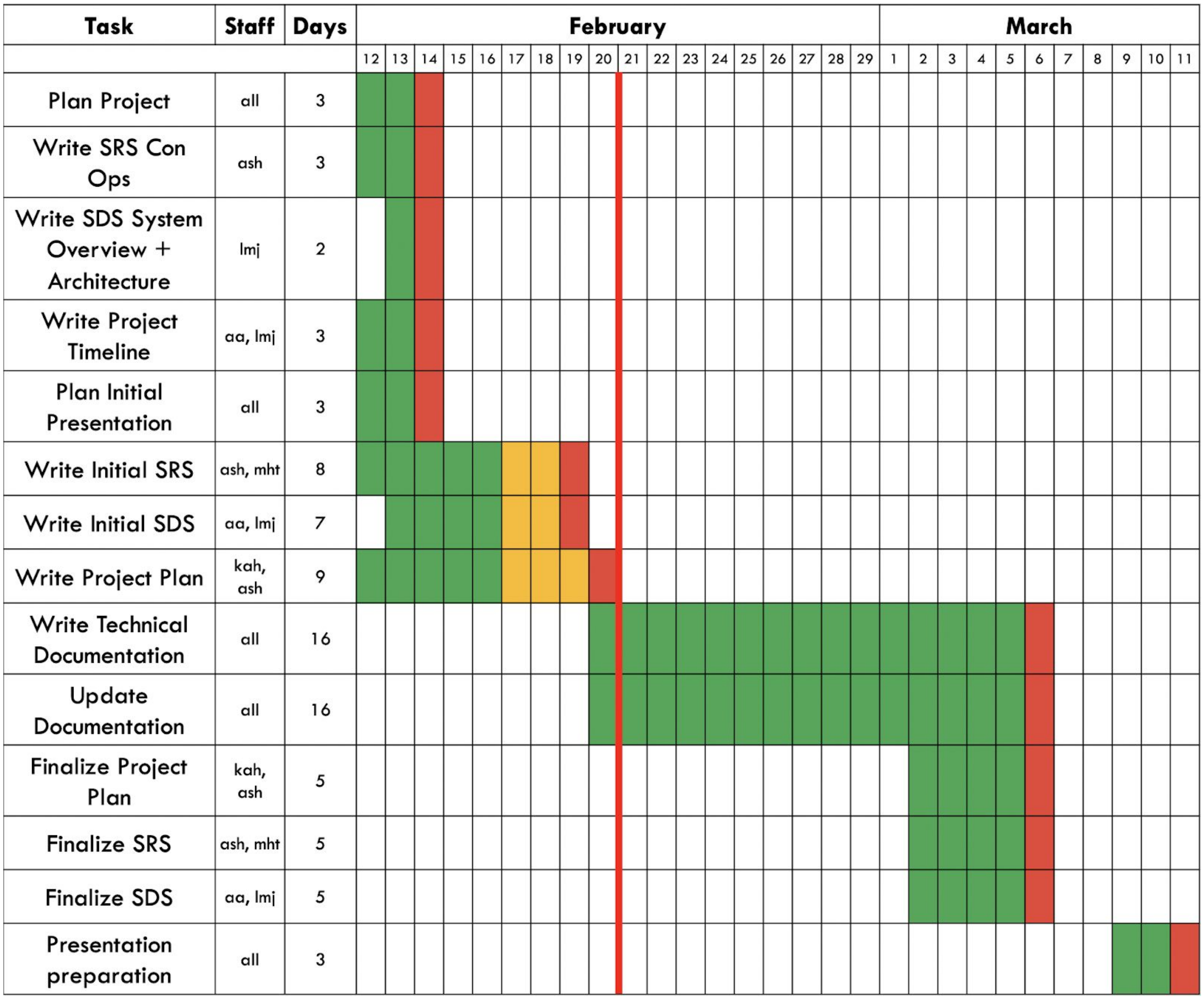


Figure 1. Color key. Displays the color coding used in the below Gantt charts.

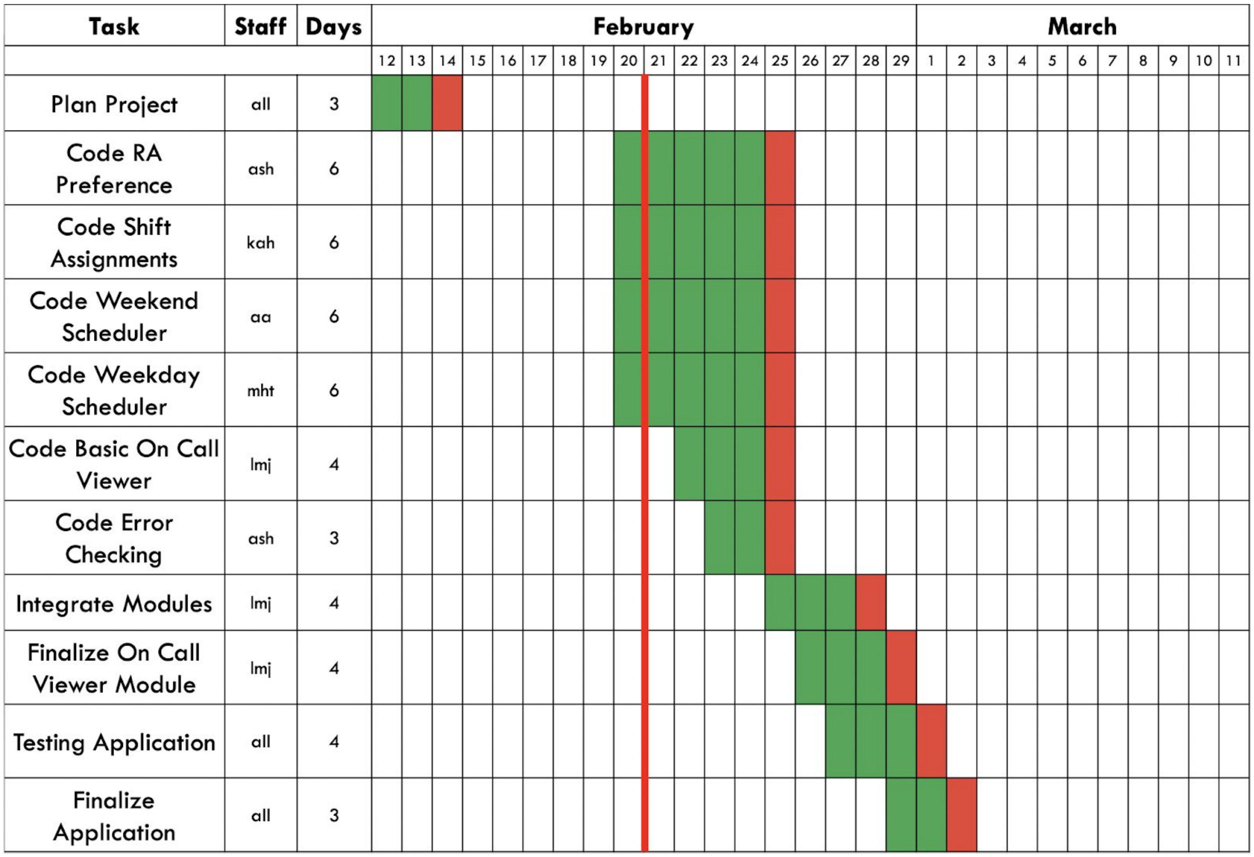
5.1. 2-20-2020

5.1.1. Gantt Chart



Task	Staff	Days	February																		March										
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11
Plan Project	all	3	🟩	🟩	🟥																										
Write SRS Con Ops	ash	3	🟩	🟩	🟥																										
Write SDS System Overview + Architecture	lmj	2		🟩	🟥																										
Write Project Timeline	aa, lmj	3	🟩	🟩	🟥																										
Plan Initial Presentation	all	3	🟩	🟩	🟥																										
Write Initial SRS	ash, mht	8	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟥																					
Write Initial SDS	aa, lmj	7		🟩	🟩	🟩	🟩	🟨	🟨	🟥																					
Write Project Plan	kah, ash	9	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟥																				
Write Technical Documentation	all	16									🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥					
Update Documentation	all	16									🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥					
Finalize Project Plan	kah, ash	5																				🟩	🟩	🟩	🟩	🟥					
Finalize SRS	ash, mht	5																				🟩	🟩	🟩	🟩	🟥					
Finalize SDS	aa, lmj	5																				🟩	🟩	🟩	🟩	🟥					
Presentation preparation	all	3																											🟩	🟩	🟥

Figure 2. Gantt chart for documentation. Shows the timeline during the development lifecycle for documenting our progress as of 2-20-2020.



Task	Staff	Days	February																		March										
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11
Plan Project	all	3	🟩	🟩	🟥																										
Code RA Preference	ash	6									🟩	🟩	🟩	🟩	🟩	🟥															
Code Shift Assignments	kah	6									🟩	🟩	🟩	🟩	🟩	🟥															
Code Weekend Scheduler	aa	6									🟩	🟩	🟩	🟩	🟩	🟥															
Code Weekday Scheduler	mht	6									🟩	🟩	🟩	🟩	🟩	🟥															
Code Basic On Call Viewer	lmj	4											🟩	🟩	🟩	🟥															
Code Error Checking	ash	3												🟩	🟩	🟥															
Integrate Modules	lmj	4														🟩	🟩	🟩	🟥												
Finalize On Call Viewer Module	lmj	4															🟩	🟩	🟩	🟥											
Testing Application	all	4																🟩	🟩	🟩	🟥										
Finalize Application	all	3																		🟩	🟩	🟥									

Figure 3. Gantt chart for coding. Shows the timeline during the development lifecycle for documenting our progress as of 2-20-2020.

5.1.2. PERT Chart

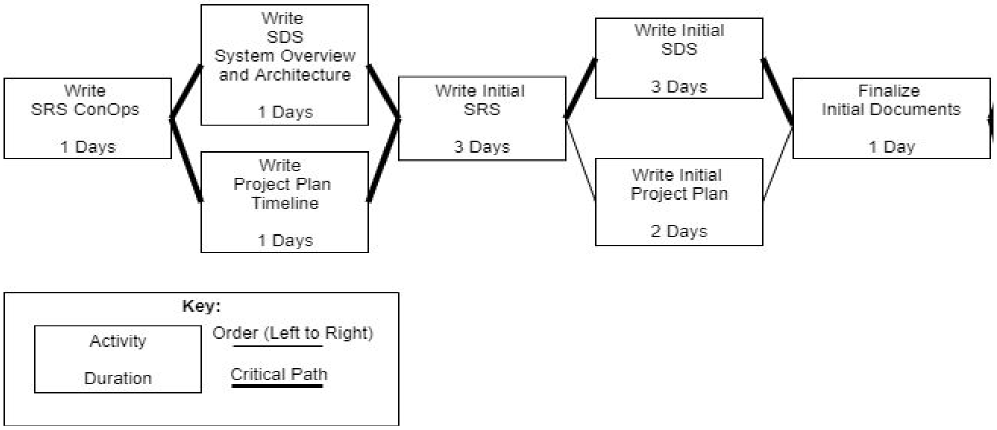


Figure 4. PERT diagram - part 1 of 2. The boxes represent activities.
The lines indicate the flow of task completion from left to right. The bolded lines show the critical path.

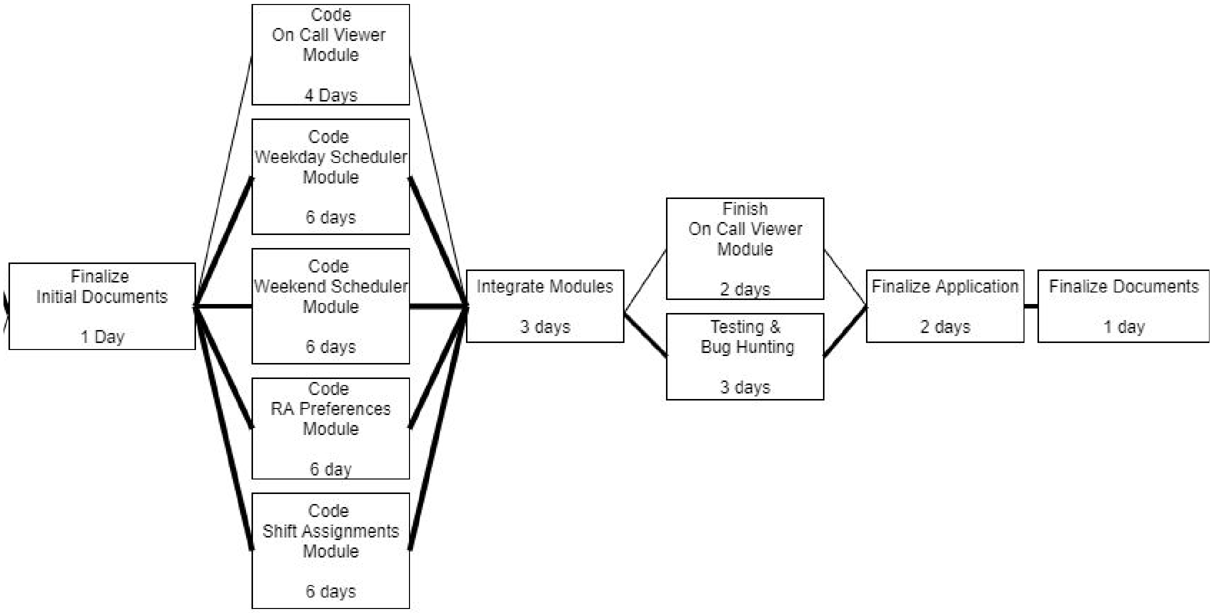


Figure 5. PERT diagram - part 2 of 2. The boxes represent activities.
The lines indicate the flow of task completion from left to right. The bolded lines show the critical path.

5.1.3. Task Schedule Spreadsheet

Assignment Chart				
Task/Milestone	**Assigned To**	**Date Assigned**	**Anticipated Time (Hours)**	**Date Due**
Week 6 (2/10-2/16)				
Form group	aa, kah, ash, lmj, mht	2/12/2020	0:20	2/12/2020
Discuss system requirements/design	aa, kah, ash, lmj, mht	2/12/2020	2:00	2/12/2020
Create GitHub	kah, lmj	2/12/2020	0:10	2/12/2020
Initial draft: SDS system overview and architecture	lmj	2/12/2020	0:30	2/14/2020
Initial draft: project timeline	aa	2/12/2020	0:45	2/14/2020
Initial draft: SRS ConOps	ash	2/12/2020	2:00	2/14/2020
Prepare for presentation	aa, kah, ash, lmj, mht	2/12/2020	0:30	2/14/2020
Brainstorm 5 human interactions with system	aa, ash, lmj, mht	2/14/2020	0:30	2/16/2020
Interview RA + transcript	ash	2/12/2020	1:00	2/19/2020
Assign tasks for initial submissions	aa, kah, ash, lmj, mht	2/16/2020	0:15	2/16/2020
Discuss interview	aa, ash, lmj, mht	2/16/2020	0:15	2/16/2020
Week 7 (2/17-2/23)				
Interview coordinator Erickson + transcript	ash	2/12/2020	2:00	2/19/2020
Interview coordinator Shafer + transcript	ash	2/12/2020	2:00	2/19/2020
Initial submission section 2 SRS	ash	2/16/2020	2:00	2/19/2020
Initial submission section 3 SRS	mht	2/16/2020	3:00	2/19/2020
Initial submission section 4 SRS	ash	2/15/2020	3:00	2/19/2020
Initial submission SDS	lmj	2/16/2020	5:00	2/19/2020
Initial submission project plan	kah	2/16/2020	2:00	2/19/2020
Project plan + SDS diagrams	aa	2/16/2020	3:00	2/19/2020
Start working on RA preference module	ash	2/20/2020	3:00	2/23/2020
Start working on shift assignments module	kah	2/20/2020	3:00	2/23/2020

Start working on weekend scheduler	aa	2/20/2020	2:00	2/23/2020
Start working on weekday scheduler	mht	2/20/2020	3:00	2/23/2020
Start working on on call viewer	lmj	2/20/2020	3:00	2/23/2020
Create week 8 milestones	ash	2/23/2020	1:00	2/23/2020

Table 1. Task Assignment Schedule for 2-19-2020.
Shows the dates and expectations of milestones.

Completion Chart

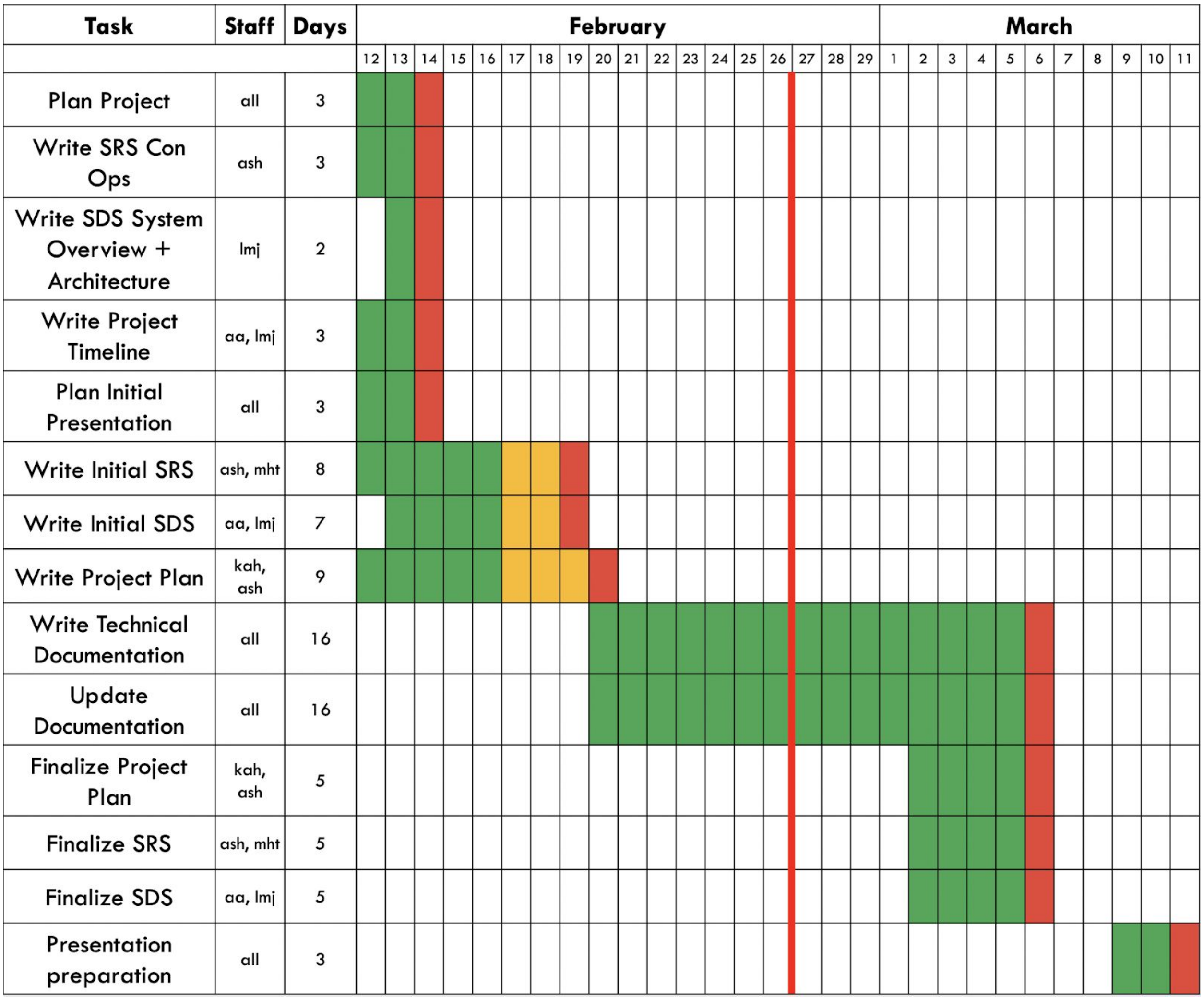
Task/ Milestone	Status	Date Completed	Completed by Whom	Time Spent (Hours)	Confirmed Completed	Date Confirmed Complete	Notes
Week 6 (2/10-2/16)							
Form group	completed	2/12/2020	aa, kah, ash, lmj, mht	0:15	ash	2/12/2020	
Discuss system requirements/design	completed	2/12/2020	aa, kah, ash, lmj, mht	1:30	ash	2/12/2020	
Create GitHub	completed	2/12/2020	kah	0:20	ash	2/12/2020	Had issues adding Alex to repo
Initial draft: SDS system overview and architecture	completed	2/14/2020	lmj	0:20	mht	2/14/2020	Max submitted on Canvas
Initial draft: project timeline	completed	2/14/2020	aa	1:00	mht	2/14/2020	Max submitted on Canvas
Initial draft: SRS ConOps	completed	2/14/2020	ash	8:00	mht	2/14/2020	Max submitted on Canvas
Prepare for presentation	completed	2/14/2020	aa, kah, ash, lmj, mht	1:00	ash	2/14/2020	Presentations were not given in class
Brainstorm 5 human interactions with system	completed	2/16/2020	aa, ash, lmj, mht	0:45	ash	2/16/2020	
Interview RA + transcript	completed	2/15/2020	ash	3:00	ash	2/15/2020	Interviewed two RAs
Assign tasks for initial submissions	completed	2/16/2020	aa, kah, ash, lmj, mht	0:45	ash	2/16/2020	
Discuss interview	completed	2/16/2020	aa, ash, lmj, mht	0:25	ash	2/16/2020	
Week 7 (2/17-2/23)							
Interview coordinator Erickson + transcript	completed	2/17/2020	ash	4:00	ash	2/17/2020	

Interview coordinator Shafer + transcript	completed	2/18/2020	ash	4:00	ash	2/18/2020	
Initial submission section 2 SRS	completed	2/18/2020	ash	4:00	ash	2/18/2020	
Initial submission section 3 SRS	completed	2/19/2020	mht	4:00	ash	2/19/2020	
Initial submission section 4 SRS	completed	2/18/2020	ash	2:00	ash	2/18/2020	
Initial submission SDS	completed	2/19/2020	lmj	6:00	lmj	2/19/2020	
Initial submission project plan	completed	2/18/2020	kah	3:00	ash	2/19/2020	Almost completed, only have to edit Gantt charts to be more legible
Project plan + SDS diagrams	completed	2/19/2020	aa	4:00	mht	2/19/2020	
Start working on RA preference module	not yet completed						
Start working on shift assignments module	not yet completed						
Start working on weekend scheduler	not yet completed						
Start working on weekday scheduler	not yet completed						
Start working on on call viewer	not yet completed						
Create week 8 milestones	not yet completed						

Table 2. Task Completion Schedule for 2-19-2020. Shows the status, Progress, and outcomes of milestones by 2-19-2020.

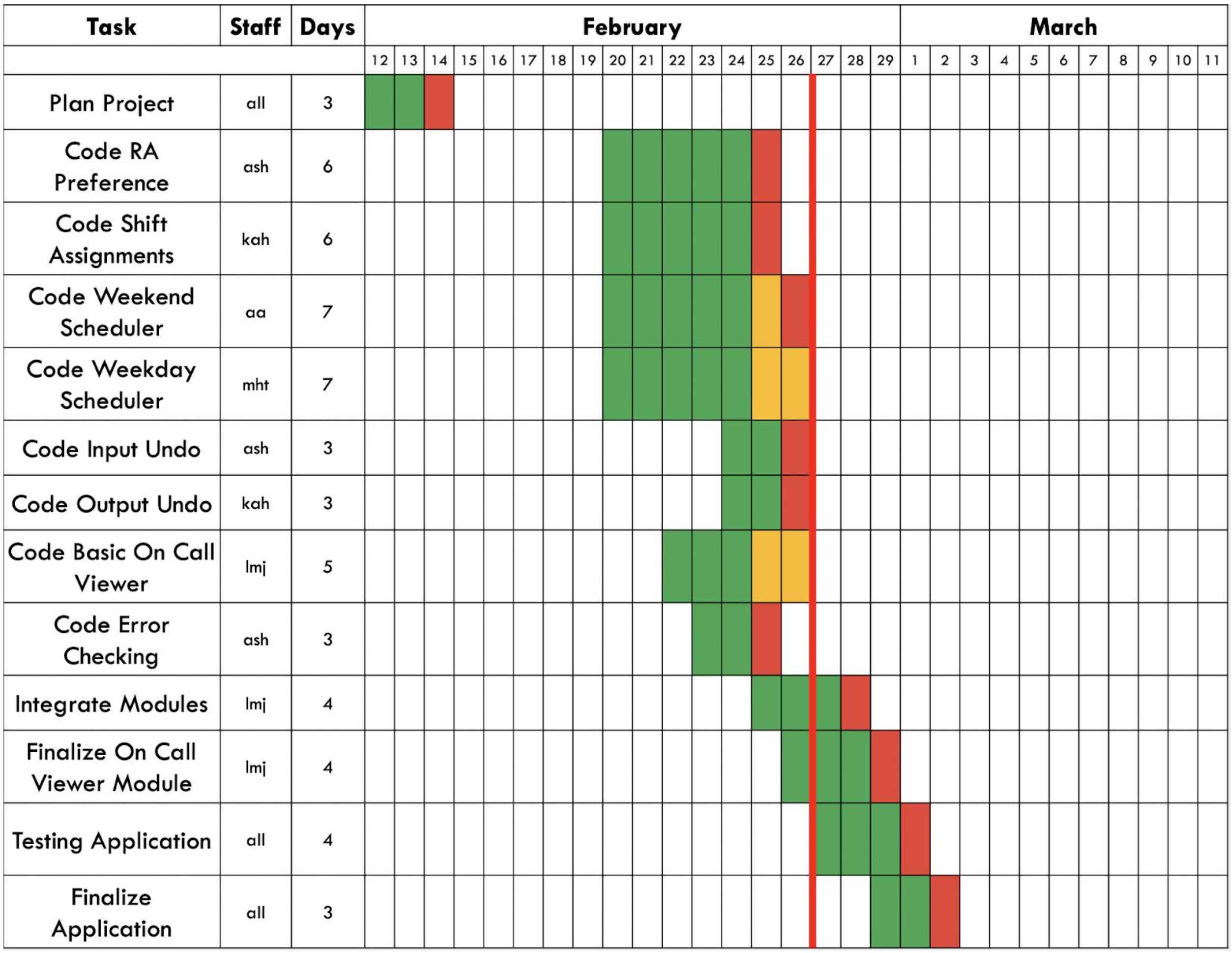
5.2. 2-26-2020

5.2.1. Gantt Chart



Task	Staff	Days	February																		March										
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11
Plan Project	all	3	🟩	🟩	🟥																										
Write SRS Con Ops	ash	3	🟩	🟩	🟥																										
Write SDS System Overview + Architecture	lmj	2		🟩	🟥																										
Write Project Timeline	aa, lmj	3	🟩	🟩	🟥																										
Plan Initial Presentation	all	3	🟩	🟩	🟥																										
Write Initial SRS	ash, mht	8	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟥																					
Write Initial SDS	aa, lmj	7		🟩	🟩	🟩	🟩	🟨	🟨	🟥																					
Write Project Plan	kah, ash	9	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟥																				
Write Technical Documentation	all	16									🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥					
Update Documentation	all	16									🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥					
Finalize Project Plan	kah, ash	5																				🟩	🟩	🟩	🟩	🟥					
Finalize SRS	ash, mht	5																				🟩	🟩	🟩	🟩	🟥					
Finalize SDS	aa, lmj	5																				🟩	🟩	🟩	🟩	🟥					
Presentation preparation	all	3																											🟩	🟩	🟥

Figure 6. Gantt chart for documentation. Shows the timeline during the development lifecycle for documenting our progress as of 2-26-2020.



Task	Staff	Days	February																		March										
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11
Plan Project	all	3	🟩	🟩	🟥																										
Code RA Preference	ash	6									🟩	🟩	🟩	🟩	🟩	🟥															
Code Shift Assignments	kah	6									🟩	🟩	🟩	🟩	🟩	🟥															
Code Weekend Scheduler	aa	7									🟩	🟩	🟩	🟩	🟩	🟨	🟥														
Code Weekday Scheduler	mht	7									🟩	🟩	🟩	🟩	🟩	🟨	🟨														
Code Input Undo	ash	3													🟩	🟩	🟥														
Code Output Undo	kah	3													🟩	🟩	🟥														
Code Basic On Call Viewer	lmj	5											🟩	🟩	🟩	🟨	🟨														
Code Error Checking	ash	3												🟩	🟩	🟥															
Integrate Modules	lmj	4														🟩	🟩	🟩	🟥												
Finalize On Call Viewer Module	lmj	4															🟩	🟩	🟩	🟥											
Testing Application	all	4																🟩	🟩	🟩	🟥										
Finalize Application	all	3																		🟩	🟩	🟥									

Figure 7. Gantt chart for coding. Shows the timeline during the development lifecycle for documenting our progress as of 2-26-2020.

5.2.2. PERT Chart

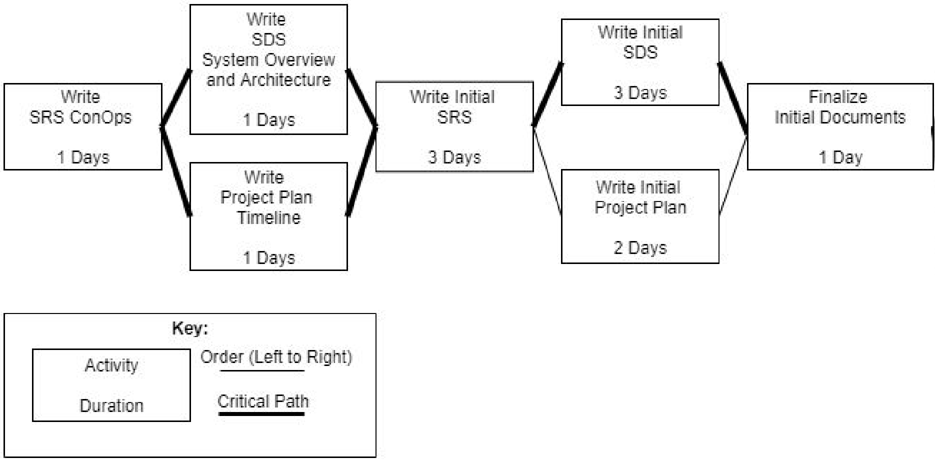


Figure 8. PERT diagram - part 1 of 2. The boxes represent activities.
The lines indicate the flow of task completion from left to right. The bolded lines show the critical path.
This section has no significant differences from the 2-20-20 chart.

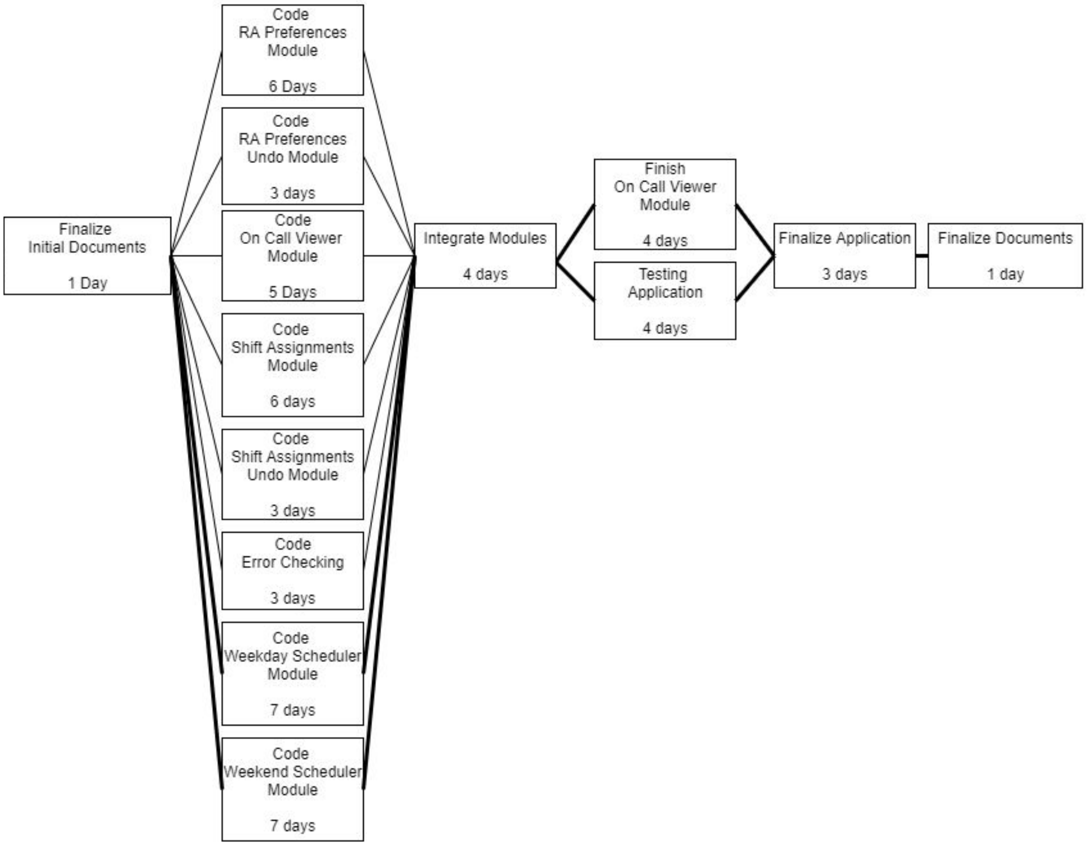


Figure 9. PERT diagram - part 2 of 2. The boxes represent activities.
The lines indicate the flow of task completion from left to right. The bolded lines show the critical path.
Key differences to the 2-20-20 chart include updates to day count, critical path, and the addition of the Undo modules.

5.2.3. Task Schedule Spreadsheet

Assignment Chart				
Task/Milestone	**Assigned To**	**Date Assigned**	**Anticipated Time (Hours)**	**Date Due**
Week 6 (2/10-2/16)				
Form group	aa, kah, ash, lmj, mht	2/12/2020	0:20	2/12/2020
Discuss system requirements/design	aa, kah, ash, lmj, mht	2/12/2020	2:00	2/12/2020
Create GitHub	kah, lmj	2/12/2020	0:10	2/12/2020
Initial draft: SDS system overview and architecture	lmj	2/12/2020	0:30	2/14/2020
Initial draft: project timeline	aa	2/12/2020	0:45	2/14/2020
Initial draft: SRS ConOps	ash	2/12/2020	2:00	2/14/2020
Prepare for presentation	aa, kah, ash, lmj, mht	2/12/2020	0:30	2/14/2020
Brainstorm 5 human interactions with system	aa, ash, lmj, mht	2/14/2020	0:30	2/16/2020
Interview RA + transcript	ash	2/12/2020	1:00	2/19/2020
Assign tasks for initial submissions	aa, kah, ash, lmj, mht	2/16/2020	0:15	2/16/2020
Discuss interview	aa, ash, lmj, mht	2/16/2020	0:15	2/16/2020
Week 7 (2/17-2/23)				
Interview coordinator Erickson + transcript	ash	2/12/2020	2:00	2/19/2020
Interview coordinator Shafer + transcript	ash	2/12/2020	2:00	2/19/2020
Initial submission section 2 SRS	ash	2/16/2020	2:00	2/19/2020
Initial submission section 3 SRS	mht	2/16/2020	3:00	2/19/2020
Initial submission section 4 SRS	ash	2/15/2020	3:00	2/19/2020
Initial submission SDS	lmj	2/16/2020	5:00	2/19/2020
Initial submission project plan	kah	2/16/2020	2:00	2/19/2020
Project plan + SDS diagrams	aa	2/16/2020	3:00	2/19/2020
Start working on RA preference module	ash	2/20/2020	3:00	2/23/2020
Start working on shift assignments module	kah	2/20/2020	3:00	2/23/2020

Start working on weekend scheduler	aa	2/20/2020	2:00	2/23/2020
Start working on weekday scheduler	mht	2/20/2020	3:00	2/23/2020
Start working on on call viewer	lmj	2/20/2020	3:00	2/23/2020
Create week 8 milestones	ash	2/23/2020	1:00	2/23/2020
Week 8 (2/24-3/01)				
Finish RA preference module	ash	2/23/2020	4:00	2/26/2020
Finish error checking	ash	2/23/2020	2:00	2/26/2020
Finish shift assignments module	kah	2/23/2020	4:00	2/26/2020
Finish weekend scheduler	aa	2/23/2020	5:00	2/26/2020
Finish weekday scheduler	mht	2/23/2020	5:00	2/26/2020
Finish GUI	lmj	2/26/2020	4:00	3/1/2020
Print out SRS, SDS, and project plan	aa	2/24/2020	0:10	2/24/2020
Meet with Professor Hornof	aa, kah, ash, lmj, mht	2/24/2020	1:00	2/24/2020
Add scheduler comparison to SRS	ash	2/24/2020	3:00	3/1/2020
Update Gantt chart	ash	2/24/2020	1:00	3/1/2020
Update PERT chart	ash	2/24/2020	0:30	3/1/2020
Write update/save/undo functions	ash	2/26/2020	4:00	3/1/2020
Update SRS meeting notes and chart	ash	2/26/2020	1:00	3/1/2020
Begin technical documentation	aa, kah, ash, mht	2/26/2020	10:00	3/1/2020
Integrate modules	lmj	2/26/2020	14:00	3/1/2020
Discuss how to implement undo function	aa, kah, ash, lmj, mht	2/26/2020	1:00	2/26/2020
Begin implementing undo function	kah, ash	2/26/2020	5:00	3/1/2020
Update SDS	kah, lmj	2/26/2020	3:00	3/1/2020
Update week 8 milestones	ash	2/26/2020	1:00	2/26/2020
Create week 9 milestones	ash	3/1/2020	1:00	3/1/2020

Table 3. Task Assignment Schedule for 2-26-2020.
Shows the dates and expectations of milestones.

Completion Chart							
Task/ Milestone	**Status**	**Date Completed**	**Completed by Whom**	**Time Spent (Hours)**	**Confirmed Completed**	**Date Confirmed Complete**	**Notes**
Week 6 (2/10-2/16)							
Form group	completed	2/12/2020	aa, kah, ash, lmj, mht	0:15	ash	2/12/2020	
Discuss system requirements/design	completed	2/12/2020	aa, kah, ash, lmj, mht	1:30	ash	2/12/2020	
Create GitHub	completed	2/12/2020	kah	0:20	ash	2/12/2020	Issues adding Alex to repo
Initial draft: SDS system overview and architecture	completed	2/14/2020	lmj	0:20	mht	2/14/2020	Max submitted on Canvas
Initial draft: project timeline	completed	2/14/2020	aa	1:00	mht	2/14/2020	Max submitted
Initial draft: SRS ConOps	completed	2/14/2020	ash	8:00	mht	2/14/2020	Max submitted
Prepare for presentation	completed	2/14/2020	aa, kah, ash, lmj, mht	1:00	ash	2/14/2020	Presentations not given
Brainstorm 5 human interactions with system	completed	2/16/2020	aa, ash, lmj, mht	0:45	ash	2/16/2020	
Interview RA + transcript	completed	2/15/2020	ash	3:00	ash	2/15/2020	Interviewed two RAs
Assign tasks for initial submissions	completed	2/16/2020	aa, kah, ash, lmj, mht	0:45	ash	2/16/2020	
Discuss interview	completed	2/16/2020	aa, ash, lmj, mht	0:25	ash	2/16/2020	
Week 7 (2/17-2/23)							
Interview coordinator Erickson + transcript	completed	2/17/2020	ash	4:00	ash	2/17/2020	
Interview coordinator Shafer + transcript	completed	2/18/2020	ash	4:00	ash	2/18/2020	

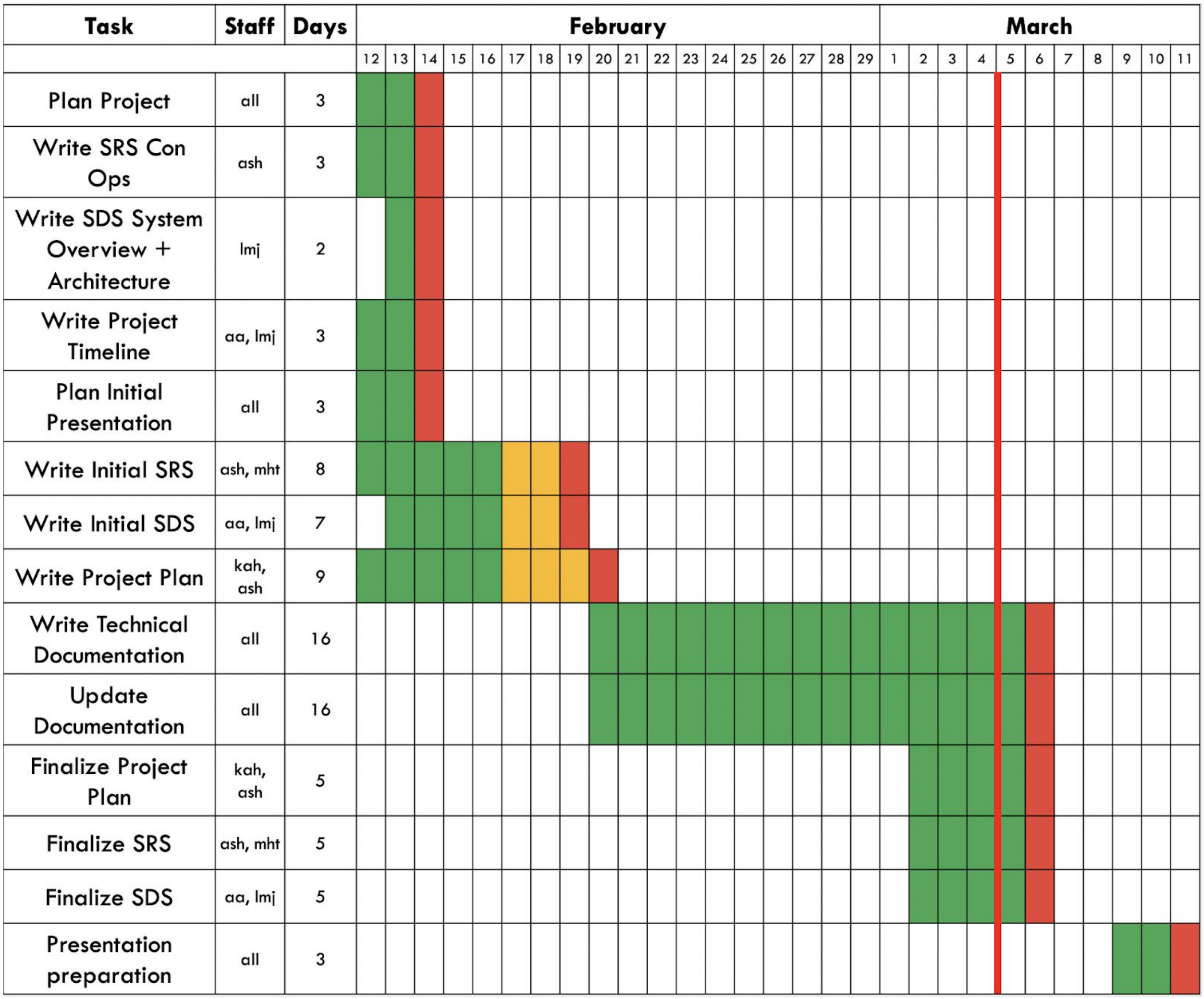
Initial submission section 2 SRS	completed	2/18/2020	ash	4:00	ash	2/18/2020	
Initial submission section 3 SRS	completed	2/19/2020	mht	4:00	ash	2/19/2020	
Initial submission section 4 SRS	completed	2/18/2020	ash	2:00	ash	2/18/2020	
Initial submission SDS	completed	2/19/2020	lmj	6:00	lmj	2/19/2020	
Initial submission project plan	completed	2/18/2020	kah	3:00	ash	2/19/2020	
Project plan + SDS diagrams	completed	2/19/2020	aa	4:00	mht	2/19/2020	
Start working on RA preference module	completed	2/22/2020	ash	3:00	ash	2/23/2020	
Start working on shift assignments module module	completed	2/22/2020	kah	4:00	ash	2/23/2020	
Start working on weekend scheduler	completed	2/24/2020	aa	5:00	lmj	2/24/2020	
Start working on weekday scheduler	completed	2/25/2020	mht	4:30	ash	2/26/2020	
Start working on on call viewer	completed	2/23/2020	lmj	3:30	ash	2/23/2020	
Create week 8 milestones	completed	2/23/2020	ash	1:00	ash	2/23/2020	
Week 8 (2/24-3/01)							
Finish RA preference module	completed	2/24/2020	ash	4:30	ash	2/26/2020	
Finish error checking	completed	2/25/2020	ash	4:00	ash	2/26/2020	
Finish shift assignments module	completed	2/25/2020	kah	5:30	ash	2/26/2020	
Finish weekend scheduler	completed	2/24/2020	aa	6:00	lmj	2/26/2020	
Finish weekday scheduler	not yet completed						

Finish GUI	not yet completed						
Print out SRS, SDS, and project plan	completed	2/24/2020	aa	0:20	aa, kah, ash, lmj, mht	2/24/2020	
Meet with Professor Hornof	completed	2/24/2020	aa, kah, ash, lmj, mht	0:30	aa, kah, ash, lmj, mht	2/24/2020	
Add scheduler comparison to SRS	not yet completed						
Update Gantt chart	not yet completed						
Update PERT chart	not yet completed						
Write update/save/undo functions	not yet completed						
Update SRS meeting notes and chart	not yet completed						
Begin technical documentation	not yet completed						
Integrate modules	not yet completed						
Discuss how to implement undo function	not yet completed						
Begin implementing undo function	not yet completed						
Update SDS	not yet completed						
Update week 8 milestones	completed	2/26/2020	ash	1:30	ash	2/26/2020	
Create week 9 milestones	not yet completed						

Table 4. Task Completion Schedule for 2-26-2020. Shows the status,
Progress, and outcomes of milestones by 2-26-2020.

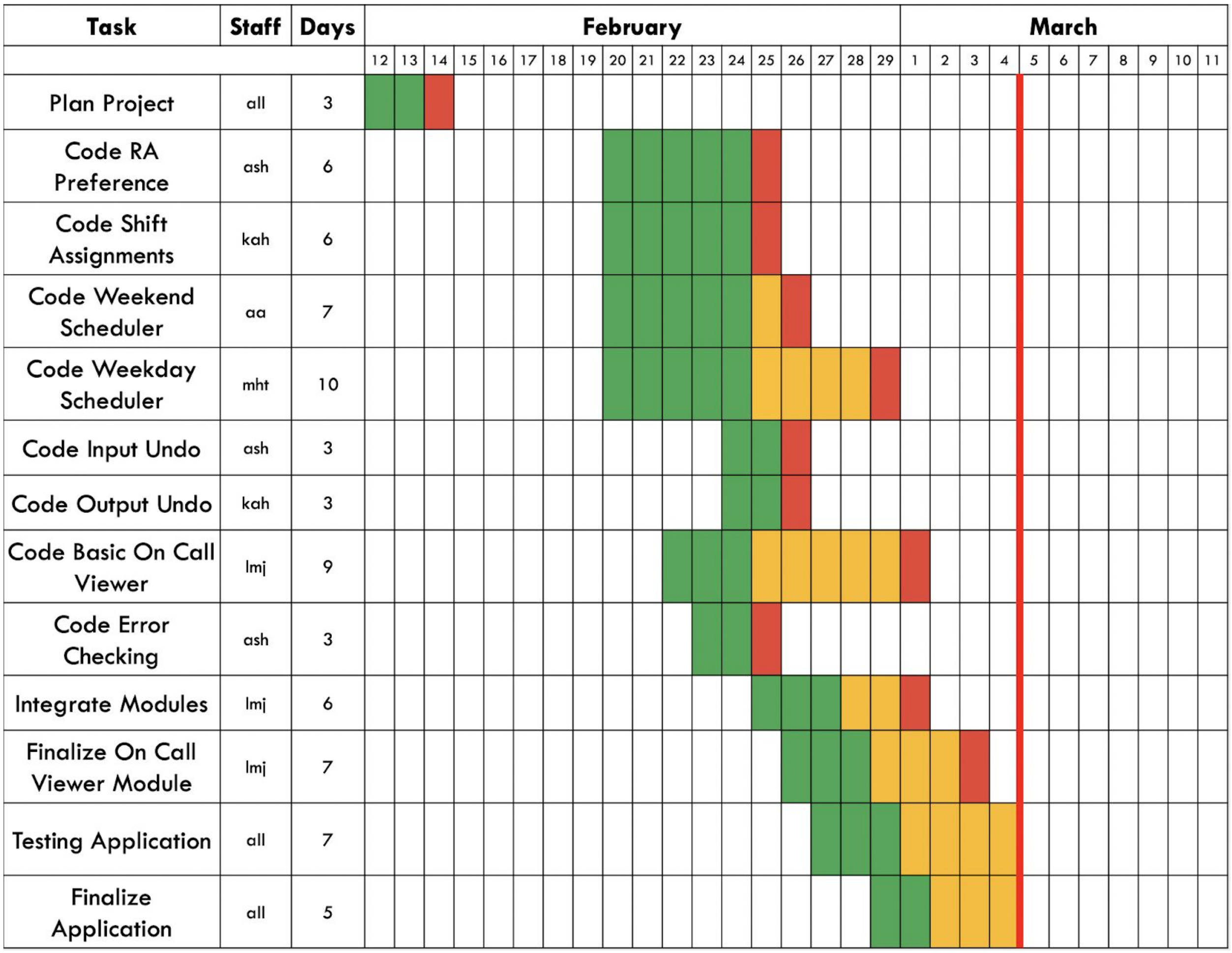
5.3. 3-04-2020

5.3.1. Gantt Chart



Task	Staff	Days	February																		March										
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11
Plan Project	all	3	🟩	🟩	🟥																										
Write SRS Con Ops	ash	3	🟩	🟩	🟥																										
Write SDS System Overview + Architecture	lmj	2		🟩	🟥																										
Write Project Timeline	aa, lmj	3	🟩	🟩	🟥																										
Plan Initial Presentation	all	3	🟩	🟩	🟥																										
Write Initial SRS	ash, mht	8	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟥																					
Write Initial SDS	aa, lmj	7		🟩	🟩	🟩	🟩	🟨	🟨	🟥																					
Write Project Plan	kah, ash	9	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟥																				
Write Technical Documentation	all	16									🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥					
Update Documentation	all	16									🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥					
Finalize Project Plan	kah, ash	5																				🟩	🟩	🟩	🟩	🟥					
Finalize SRS	ash, mht	5																				🟩	🟩	🟩	🟩	🟥					
Finalize SDS	aa, lmj	5																				🟩	🟩	🟩	🟩	🟥					
Presentation preparation	all	3																											🟩	🟩	🟥

Figure 10. Gantt chart for documentation. Shows the timeline during the development lifecycle for documenting our progress as of 3-04-2020.



Task	Staff	Days	February																		March										
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11
Plan Project	all	3	🟩	🟩	🟥																										
Code RA Preference	ash	6									🟩	🟩	🟩	🟩	🟩	🟥															
Code Shift Assignments	kah	6									🟩	🟩	🟩	🟩	🟩	🟥															
Code Weekend Scheduler	aa	7									🟩	🟩	🟩	🟩	🟩	🟨	🟥														
Code Weekday Scheduler	mht	10									🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟥											
Code Input Undo	ash	3													🟩	🟩	🟥														
Code Output Undo	kah	3													🟩	🟩	🟥														
Code Basic On Call Viewer	lmj	9											🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟥										
Code Error Checking	ash	3												🟩	🟩	🟥															
Integrate Modules	lmj	6														🟩	🟩	🟩	🟨	🟨	🟥										
Finalize On Call Viewer Module	lmj	7															🟩	🟩	🟩	🟨	🟨	🟨	🟥								
Testing Application	all	7																🟩	🟩	🟩	🟨	🟨	🟨	🟨							
Finalize Application	all	5																		🟩	🟩	🟨	🟨	🟨							

Figure 11. Gantt chart for coding. Shows the timeline during the development lifecycle for documenting our progress as of 3-04-2020.

5.3.2. PERT Chart

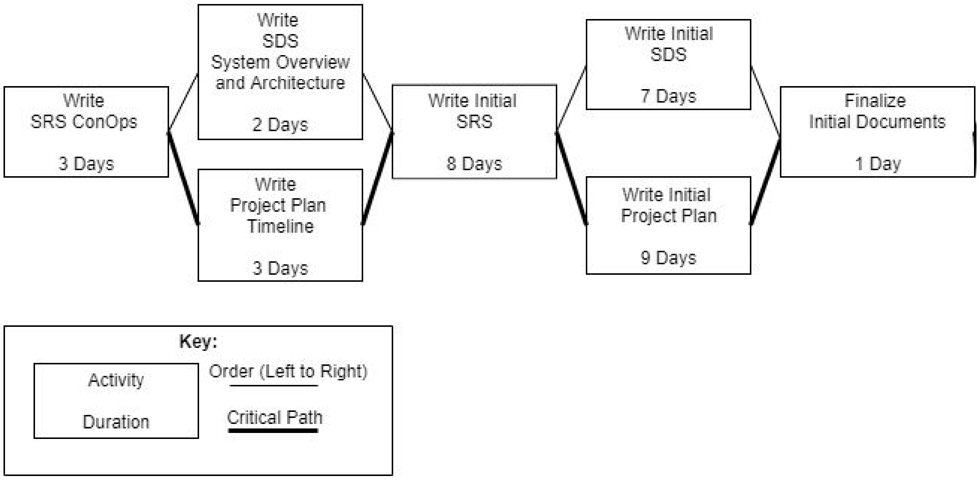


Figure 12. PERT diagram - part 1 of 2. The boxes represent activities.
The lines indicate the flow of task completion from left to right. The bolded lines show the critical path.
This section features updates to the time spent on planning and writing documents earlier in the project life compared to the 2-27-20 chart.

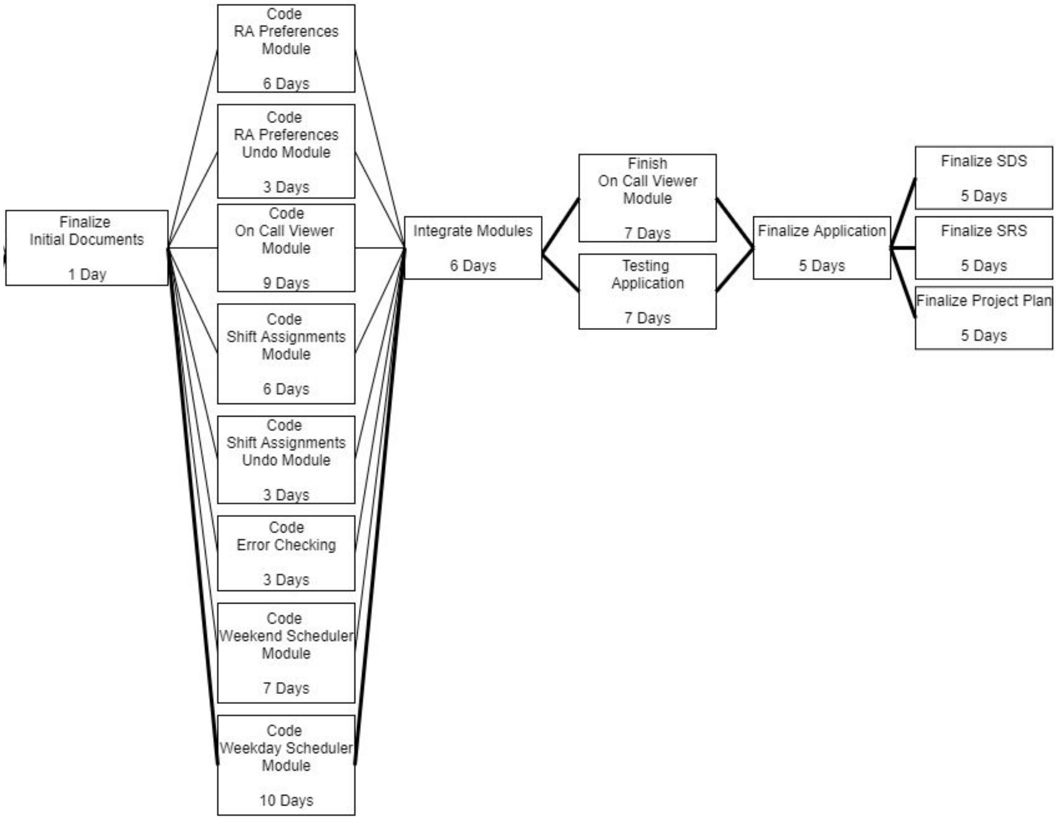


Figure 13. PERT diagram - part 2 of 2. The boxes represent activities.
The lines indicate the flow of task completion from left to right. The bolded lines show the critical path.
Key differences to the 2-27-20 chart include updates to day count and critical path is now through the weekday scheduler.

5.3.3. Task Schedule Spreadsheet

Assignment Chart

Task/Milestone	Assigned To	Date Assigned	Anticipated Time (Hours)	Date Due
Week 6 (2/10-2/16)				
Form group	aa, kah, ash, lmj, mht	2/12/2020	0:20	2/12/2020
Discuss system requirements/design	aa, kah, ash, lmj, mht	2/12/2020	2:00	2/12/2020
Create GitHub	kah, lmj	2/12/2020	0:10	2/12/2020
Initial draft: SDS system overview and architecture	lmj	2/12/2020	0:30	2/14/2020
Initial draft: project timeline	aa	2/12/2020	0:45	2/14/2020
Initial draft: SRS ConOps	ash	2/12/2020	2:00	2/14/2020
Prepare for presentation	aa, kah, ash, lmj, mht	2/12/2020	0:30	2/14/2020
Brainstorm 5 human interactions with system	aa, ash, lmj, mht	2/14/2020	0:30	2/16/2020
Interview RA + transcript	ash	2/12/2020	1:00	2/19/2020
Assign tasks for initial submissions	aa, kah, ash, lmj, mht	2/16/2020	0:15	2/16/2020
Discuss interview	aa, ash, lmj, mht	2/16/2020	0:15	2/16/2020
Week 7 (2/17-2/23)				
Interview coordinator Erickson + transcript	ash	2/12/2020	2:00	2/19/2020
Interview coordinator Shafer + transcript	ash	2/12/2020	2:00	2/19/2020
Initial submission section 2 SRS	ash	2/16/2020	2:00	2/19/2020
Initial submission section 3 SRS	mht	2/16/2020	3:00	2/19/2020
Initial submission section 4 SRS	ash	2/15/2020	3:00	2/19/2020
Initial submission SDS	lmj	2/16/2020	5:00	2/19/2020
Initial submission project plan	kah	2/16/2020	2:00	2/19/2020
Project plan + SDS diagrams	aa	2/16/2020	3:00	2/19/2020
Start working on RA preference module	ash	2/20/2020	3:00	2/23/2020
Start working on shift assignments module	kah	2/20/2020	3:00	2/23/2020

Start working on weekend scheduler	aa	2/20/2020	2:00	2/23/2020
Start working on weekday scheduler	mht	2/20/2020	3:00	2/23/2020
Start working on on call viewer	lmj	2/20/2020	3:00	2/23/2020
Create week 8 milestones	ash	2/23/2020	1:00	2/23/2020
Week 8 (2/24-3/01)				
Finish RA preference module	ash	2/23/2020	4:00	2/26/2020
Finish error checking	ash	2/23/2020	2:00	2/26/2020
Finish shift assignments module	kah	2/23/2020	4:00	2/26/2020
Finish weekend scheduler	aa	2/23/2020	5:00	2/26/2020
Finish weekday scheduler	mht	2/23/2020	5:00	2/26/2020
Finish GUI	lmj	2/26/2020	4:00	3/1/2020
Print out SRS, SDS, and project plan	aa	2/24/2020	0:10	2/24/2020
Meet with Professor Hornof	aa, kah, ash, lmj, mht	2/24/2020	1:00	2/24/2020
Add scheduler comparison to SRS	ash	2/24/2020	3:00	3/1/2020
Update Gantt chart	ash	2/24/2020	1:00	3/1/2020
Update PERT chart	ash	2/24/2020	0:30	3/1/2020
Write update/save/undo functions	ash	2/26/2020	4:00	3/1/2020
Update SRS meeting notes and chart	ash	2/26/2020	1:00	3/1/2020
Begin technical documentation	aa, kah, ash, mht	2/26/2020	10:00	3/1/2020
Integrate modules	lmj	2/26/2020	14:00	3/1/2020
Discuss how to implement undo function	aa, kah, ash, lmj, mht	2/26/2020	1:00	2/26/2020
Begin implementing undo function	kah, ash	2/26/2020	5:00	3/1/2020
Update SDS	kah, lmj	2/26/2020	3:00	3/1/2020
Update week 8 milestones	ash	2/26/2020	1:00	2/26/2020
Create week 9 milestones	ash	3/1/2020	1:00	3/1/2020
Week 9 (3/02-3/08)				
Customer testing	ash	3/3/2020	1:00	3/3/2020
Finish alphabetical and numerical by student ID tiebreakers	mht	3/2/2020	0:30	3/4/2020
Finish bad pairings	mht	3/2/2020	0:30	3/4/2020
Finish export report functionality	mht	3/2/2020	4:00	3/4/2020

Finish project plan	aa, ash	3/2/2020	2:00	3/8/2020
Finish SRS	ash, mht	3/2/2020	4:00	3/8/2020
Finish SDS	kah, lmj	3/2/2020	4:00	3/8/2020
Finish technical documentation	aa, kah, ash, lmj, mht	3/2/2020	5:00	3/8/2020
Test application	aa, kah, ash, lmj, mht	3/2/2020	5:00	3/8/2020

Table 5. Task Assignment Schedule for 3-04-2020.
Shows the dates and expectations of milestones.

Completion Chart

Task/ Milestone	Status	Date Completed	Completed by Whom	Time Spent (Hours)	Confirmed Completed	Date Confirmed Complete	Notes
Week 6 (2/10-2/16)							
Form group	completed	2/12/2020	aa, kah, ash, lmj, mht	0:15	ash	2/12/2020	
Discuss system requirements/design	completed	2/12/2020	aa, kah, ash, lmj, mht	1:30	ash	2/12/2020	
Create GitHub	completed	2/12/2020	kah	0:20	ash	2/12/2020	Had issues adding Alex to repo
Initial draft: SDS system overview and architecture	completed	2/14/2020	lmj	0:20	mht	2/14/2020	Max submitted on Canvas
Initial draft: project timeline	completed	2/14/2020	aa	1:00	mht	2/14/2020	Max submitted on Canvas
Initial draft: SRS ConOps	completed	2/14/2020	ash	8:00	mht	2/14/2020	Max submitted on Canvas
Prepare for presentation	completed	2/14/2020	aa, kah, ash, lmj, mht	1:00	ash	2/14/2020	Presentations were not given in class
Brainstorm 5 human interactions with system	completed	2/16/2020	aa, ash, lmj, mht	0:45	ash	2/16/2020	
Interview RA + transcript	completed	2/15/2020	ash	3:00	ash	2/15/2020	Interviewed two RAs
Assign tasks for initial submissions	completed	2/16/2020	aa, kah, ash, lmj, mht	0:45	ash	2/16/2020	
Discuss interview	completed	2/16/2020	aa, ash, lmj, mht	0:25	ash	2/16/2020	
Week 7 (2/17-2/23)							
Interview coordinator Erickson + transcript	completed	2/17/2020	ash	4:00	ash	2/17/2020	

Interview coordinator Shafer + transcript	completed	2/18/2020	ash	4:00	ash	2/18/2020	
Initial submission section 2 SRS	completed	2/18/2020	ash	4:00	ash	2/18/2020	
Initial submission section 3 SRS	completed	2/19/2020	mht	4:00	ash	2/19/2020	
Initial submission section 4 SRS	completed	2/18/2020	ash	2:00	ash	2/18/2020	
Initial submission SDS	completed	2/19/2020	lmj	6:00	lmj	2/19/2020	
Initial submission project plan	completed	2/18/2020	kah	3:00	ash	2/19/2020	Almost completed, edit Gantt chart
Project plan + SDS diagrams	completed	2/19/2020	aa	4:00	mht	2/19/2020	
Start working on RA preference module	completed	2/22/2020	ash	3:00	ash	2/23/2020	
Start working on shift assignments module	completed	2/22/2020	kah	4:00	ash	2/23/2020	
Start working on weekend scheduler	completed	2/24/2020	aa	5:00	lmj	2/24/2020	
Start working on weekday scheduler	completed	2/25/2020	mht	4:30	ash	2/26/2020	
Start working on on call viewer	completed	2/23/2020	lmj	3:30	ash	2/23/2020	
Create week 8 milestones	completed	2/23/2020	ash	1:00	ash	2/23/2020	
Week 8 (2/24-3/01)							
Finish RA preference module	completed	2/24/2020	ash	4:30	ash	2/26/2020	
Finish error checking	completed	2/25/2020	ash	4:00	ash	2/26/2020	
Finish shift assignment module	completed	2/25/2020	kah	5:30	ash	2/26/2020	

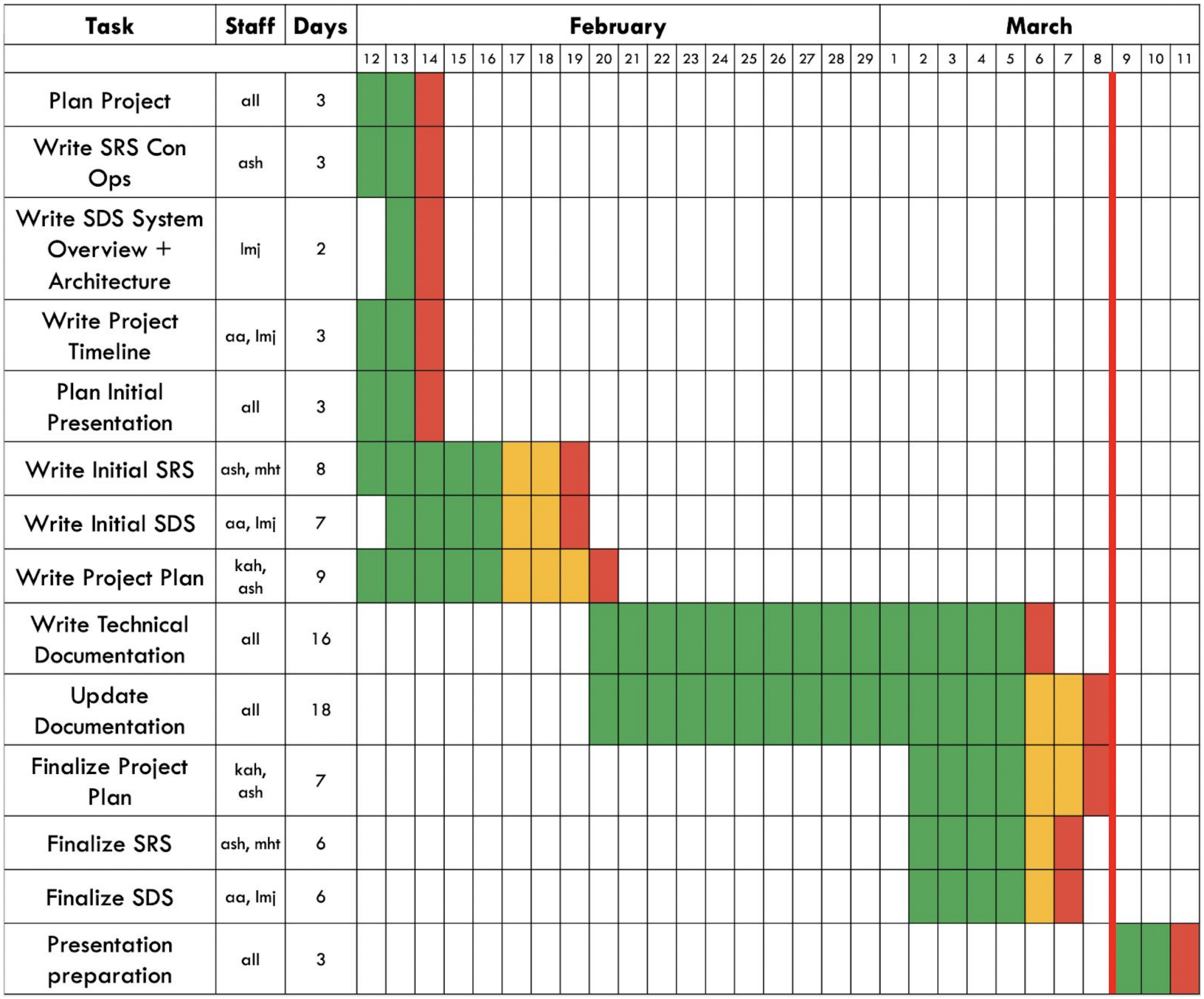
Finish weekend scheduler	completed	2/24/2020	aa	6:00	lmj	2/26/2020	
Finish weekday scheduler	completed	3/3/2020	mht	8:00	lmj	3/4/2020	
Finish GUI	completed	3/2/2020	lmj	6:00	ash	3/4/2020	
Print out SRS, SDS, and project plan	completed	2/24/2020	aa	0:20	aa, kah, ash, lmj, mht	2/24/2020	
Meet with Professor Hornof	completed	2/24/2020	aa, kah, ash, lmj, mht	0:30	aa, kah, ash, lmj, mht	2/24/2020	
Add scheduler comparison to SRS	not yet completed						
Update Gantt chart	completed	3/1/2020	ash	1:00	ash	3/1/2020	
Update PERT chart	completed	3/1/2020	aa	0:30	ash	3/1/2020	
Write update/save/undo functions	completed	2/26/2020	kah, ash	6:30	lmj	2/26/2020	
Update SRS meeting notes and chart	completed	3/1/2020	ash	1:30	ash	3/1/2020	
Begin technical documentation	completed	3/1/2020	aa, kah, ash, mht	14:00	ash	3/2/2020	
Integrate modules	completed	3/1/2020	lmj	7:00	aa, kah, ash, lmj, mht	3/1/2020	
Discuss how to implement undo function	completed	2/26/2020	aa, kah, ash, lmj, mht	1:00	aa, kah, ash, lmj, mht	2/26/2020	
Begin implementing undo function	completed	2/24/2020	kah, ash	6:30	lmj	2/26/2020	
Update SDS	not yet completed						
Update week 8 milestones	completed	2/26/2020	ash	1:30	ash	2/26/2020	
Create week 9 milestones	completed	3/1/2020	ash	1:00	ash	3/1/2020	

9 (3/02-3/08)							
Customer testing	completed	3/3/2020	ash	0:45	ash	3/3/2020	Revisions to system made based on customer input
Finish alphabetical and numerical by student ID tiebreakers	completed	3/3/2020	mht	1:30	lmj	3/4/2020	
Finish bad pairings	completed	3/3/2020	mht	2:00	lmj	3/4/2020	
Finish export report function	not yet completed						
Finish project plan	not yet completed						
Finish SRS	not yet completed						
Finish SDS	not yet completed						
Finish technical documentation	not yet completed						
Test application	not yet completed						

Table 6. Task Completion Schedule for 3-04-2020. Shows the status,
Progress, and outcomes of milestones by 3-04-2020.

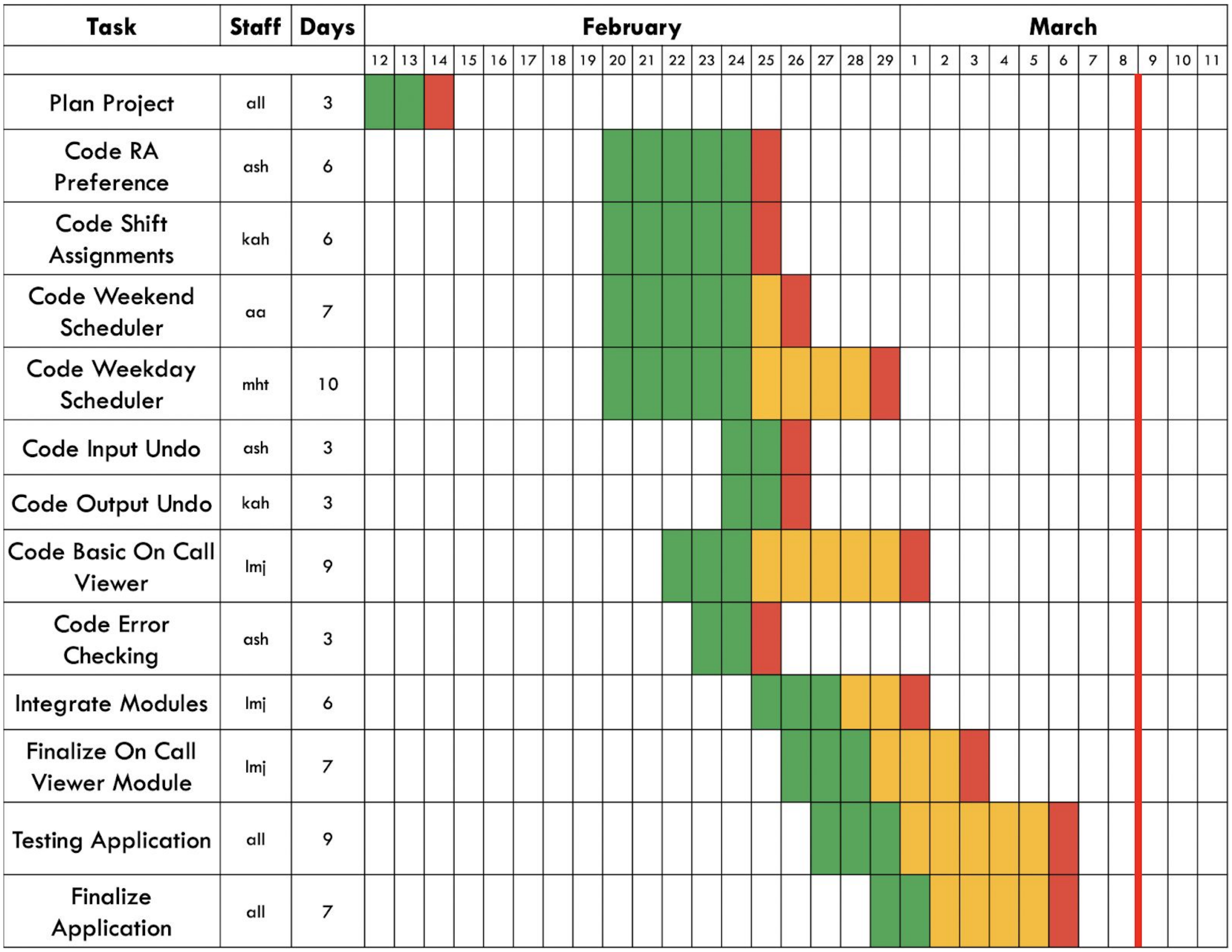
5.4. 3-08-2020

5.4.1. Gantt Charts



Task	Staff	Days	February																		March										
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11
Plan Project	all	3	🟩	🟩	🟥																										
Write SRS Con Ops	ash	3	🟩	🟩	🟥																										
Write SDS System Overview + Architecture	lmj	2		🟩	🟥																										
Write Project Timeline	aa, lmj	3	🟩	🟩	🟥																										
Plan Initial Presentation	all	3	🟩	🟩	🟥																										
Write Initial SRS	ash, mht	8	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟥																					
Write Initial SDS	aa, lmj	7		🟩	🟩	🟩	🟩	🟨	🟨	🟥																					
Write Project Plan	kah, ash	9	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟥																				
Write Technical Documentation	all	16									🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥					
Update Documentation	all	18									🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟥			
Finalize Project Plan	kah, ash	7																				🟩	🟩	🟩	🟩	🟨	🟨	🟥			
Finalize SRS	ash, mht	6																				🟩	🟩	🟩	🟩	🟨	🟥				
Finalize SDS	aa, lmj	6																				🟩	🟩	🟩	🟩	🟨	🟥				
Presentation preparation	all	3																											🟩	🟩	🟥

Figure 14. Gantt chart for documentation. Shows the timeline during the development lifecycle for documenting our progress as of 3-08-2020.



Task	Staff	Days	February																		March										
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11
Plan Project	all	3	🟩	🟩	🟥																										
Code RA Preference	ash	6									🟩	🟩	🟩	🟩	🟩	🟥															
Code Shift Assignments	kah	6									🟩	🟩	🟩	🟩	🟩	🟥															
Code Weekend Scheduler	aa	7									🟩	🟩	🟩	🟩	🟩	🟨	🟥														
Code Weekday Scheduler	mht	10									🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟥											
Code Input Undo	ash	3													🟩	🟩	🟥														
Code Output Undo	kah	3													🟩	🟩	🟥														
Code Basic On Call Viewer	lmj	9											🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟥										
Code Error Checking	ash	3												🟩	🟩	🟥															
Integrate Modules	lmj	6														🟩	🟩	🟩	🟨	🟨	🟥										
Finalize On Call Viewer Module	lmj	7															🟩	🟩	🟩	🟨	🟨	🟨	🟥								
Testing Application	all	9																🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟥					
Finalize Application	all	7																		🟩	🟩	🟨	🟨	🟨	🟨	🟥					

Figure 15. Gantt chart for coding. Shows the timeline during the development lifecycle for documenting our progress as of 3-08-2020.

5.4.2. PERT Charts

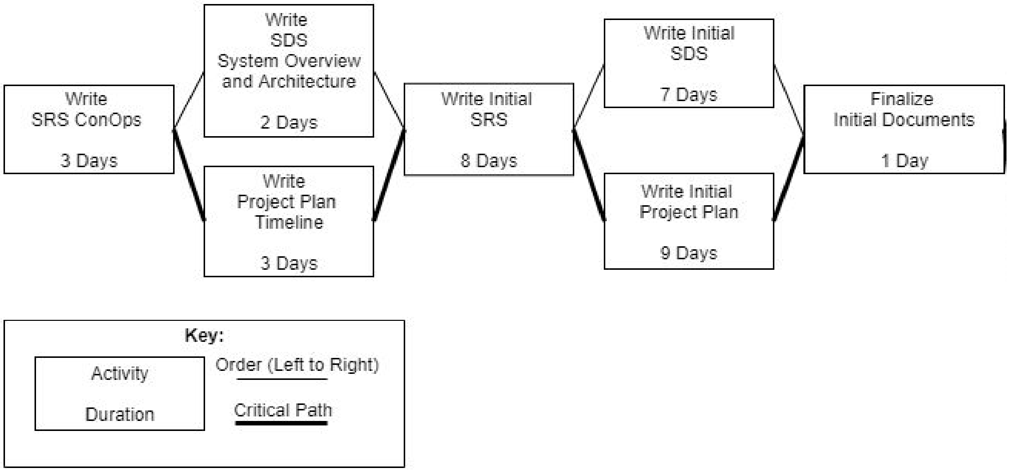

Write SRS ConOps 3 Days
Write SDS System Overview and Architecture 2 Days
Write Project Plan Timeline 3 Days
Write Initial SRS 8 Days
Write Initial SDS 7 Days
Write Initial Project Plan 9 Days
Finalize Initial Documents 1 Day
Key:
Activity Duration
Order (Left to Right)
Critical Path

Figure 16. PERT diagram - part 1 of 2. The boxes represent activities.
The lines indicate the flow of task completion from left to right. The bolded lines show the critical path.
This section features no differences to the 3-04-20 chart.

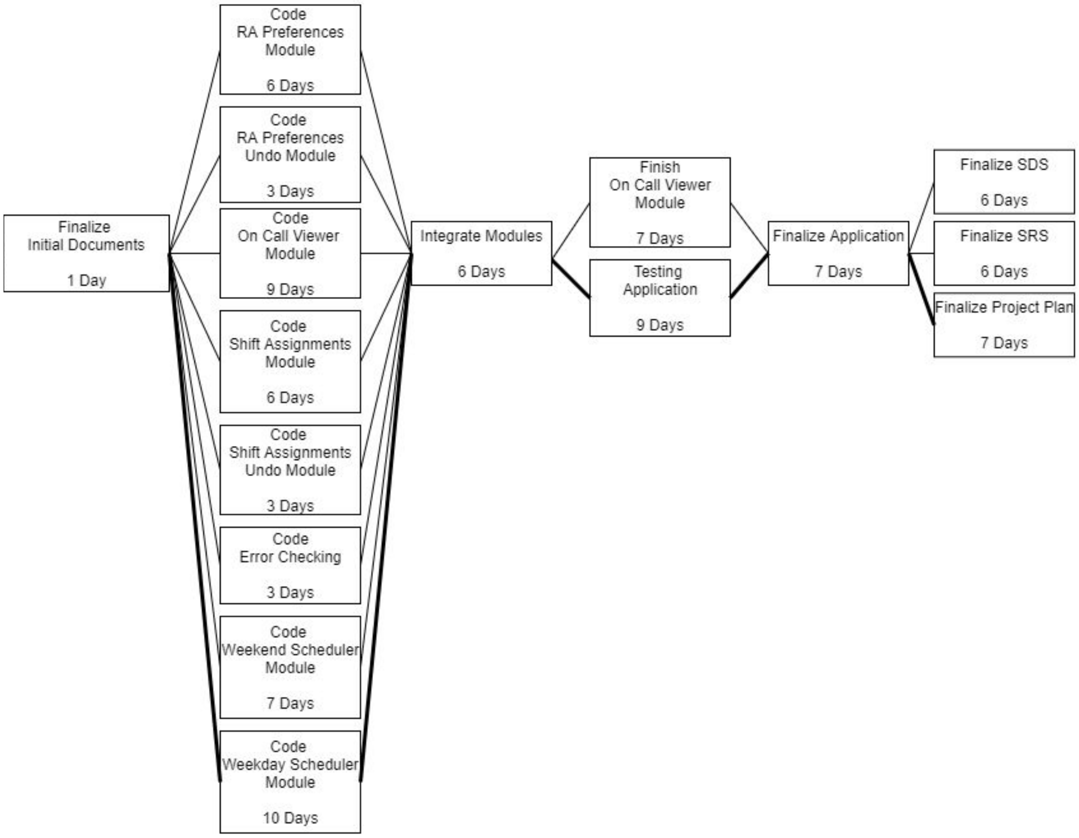

Finalize Initial Documents 1 Day
Code RA Preferences Module 6 Days
Code RA Preferences Undo Module 3 Days
Code On Call Viewer Module 9 Days
Code Shift Assignments Module 6 Days
Code Shift Assignments Undo Module 3 Days
Code Error Checking 3 Days
Code Weekend Scheduler Module 7 Days
Code Weekday Scheduler Module 10 Days
Integrate Modules 6 Days
Finish On Call Viewer Module 7 Days
Testing Application 9 Days
Finalize Application 7 Days
Finalize SDS 6 Days
Finalize SRS 6 Days
Finalize Project Plan 7 Days

Figure 17. PERT diagram - part 2 of 2. The boxes represent activities.
The lines indicate the flow of task completion from left to right. The bolded lines show the critical path.
Key differences to the 3-04-20 chart include updates to day counts and critical path through the end of the project.

5.4.3. Task Schedule Spreadsheet

Assignment Chart				
Task/Milestone	**Assigned To**	**Date Assigned**	**Anticipated Time (Hours)**	**Date Due**
Week 6 (2/10-2/16)				
Form group	aa, kah, ash, lmj, mht	2/12/2020	0:20	2/12/2020
Discuss system requirements/design	aa, kah, ash, lmj, mht	2/12/2020	2:00	2/12/2020
Create GitHub	kah, lmj	2/12/2020	0:10	2/12/2020
Initial draft: SDS system overview and architecture	lmj	2/12/2020	0:30	2/14/2020
Initial draft: project timeline	aa	2/12/2020	0:45	2/14/2020
Initial draft: SRS ConOps	ash	2/12/2020	2:00	2/14/2020
Prepare for presentation	aa, kah, ash, lmj, mht	2/12/2020	0:30	2/14/2020
Brainstorm 5 human interactions with system	aa, ash, lmj, mht	2/14/2020	0:30	2/16/2020
Interview RA + transcript	ash	2/12/2020	1:00	2/19/2020
Assign tasks for initial submissions	aa, kah, ash, lmj, mht	2/16/2020	0:15	2/16/2020
Discuss interview	aa, ash, lmj, mht	2/16/2020	0:15	2/16/2020
Week 7 (2/17-2/23)				
Interview coordinator Erickson + transcript	ash	2/12/2020	2:00	2/19/2020
Interview coordinator Shafer + transcript	ash	2/12/2020	2:00	2/19/2020
Initial submission section 2 SRS	ash	2/16/2020	2:00	2/19/2020
Initial submission section 3 SRS	mht	2/16/2020	3:00	2/19/2020
Initial submission section 4 SRS	ash	2/15/2020	3:00	2/19/2020
Initial submission SDS	lmj	2/16/2020	5:00	2/19/2020
Initial submission project plan	kah	2/16/2020	2:00	2/19/2020
Project plan + SDS diagrams	aa	2/16/2020	3:00	2/19/2020
Start working on RA preference module	ash	2/20/2020	3:00	2/23/2020
Start working on shift assignments module	kah	2/20/2020	3:00	2/23/2020

Start working on weekend scheduler	aa	2/20/2020	2:00	2/23/2020
Start working on weekday scheduler	mht	2/20/2020	3:00	2/23/2020
Start working on on call viewer	lmj	2/20/2020	3:00	2/23/2020
Create week 8 milestones	ash	2/23/2020	1:00	2/23/2020
Week 8 (2/24-3/01)				
Finish RA preference module	ash	2/23/2020	4:00	2/26/2020
Finish error checking	ash	2/23/2020	2:00	2/26/2020
Finish shift assignments module	kah	2/23/2020	4:00	2/26/2020
Finish weekend scheduler	aa	2/23/2020	5:00	2/26/2020
Finish weekday scheduler	mht	2/23/2020	5:00	2/26/2020
Finish GUI	lmj	2/26/2020	4:00	3/1/2020
Print out SRS, SDS, and project plan	aa	2/24/2020	0:10	2/24/2020
Meet with Professor Hornof	aa, kah, ash, lmj, mht	2/24/2020	1:00	2/24/2020
Add scheduler comparison to SRS	ash	2/24/2020	3:00	3/1/2020
Update Gantt chart	ash	2/24/2020	1:00	3/1/2020
Update PERT chart	ash	2/24/2020	0:30	3/1/2020
Write update/save/undo functions	ash	2/26/2020	4:00	3/1/2020
Update SRS meeting notes and chart	ash	2/26/2020	1:00	3/1/2020
Begin technical documentation	aa, kah, ash, mht	2/26/2020	10:00	3/1/2020
Integrate modules	lmj	2/26/2020	14:00	3/1/2020
Discuss how to implement undo function	aa, kah, ash, lmj, mht	2/26/2020	1:00	2/26/2020
Begin implementing undo function	kah, ash	2/26/2020	5:00	3/1/2020
Update SDS	kah, lmj	2/26/2020	3:00	3/1/2020
Update week 8 milestones	ash	2/26/2020	1:00	2/26/2020
Create week 9 milestones	ash	3/1/2020	1:00	3/1/2020
Week 9 (3/02-3/08)				
Customer testing	ash	3/3/2020	1:00	3/3/2020
Finish alphabetical and numerical by student ID tiebreakers	mht	3/2/2020	0:30	3/4/2020
Finish bad pairings	mht	3/2/2020	0:30	3/4/2020
Finish export report functionality	mht	3/2/2020	4:00	3/4/2020

Finish project plan	aa, ash	3/2/2020	2:00	3/8/2020
Finish SRS	ash, mht	3/2/2020	4:00	3/8/2020
Finish SDS	kah, lmj	3/2/2020	4:00	3/8/2020
Finish technical documentation	aa, kah, ash, lmj, mht	3/2/2020	5:00	3/8/2020
Test application	aa, kah, ash, lmj, mht	3/2/2020	5:00	3/8/2020

Table 7. Task Assignment Schedule for 3-08-2020.
Shows the dates and expectations of milestones.

Completion Chart

Task/ Milestone	Status	Date Completed	Completed by Whom	Time Spent (Hours)	Confirmed Completed	Date Confirmed Complete	Notes
Week 6 (2/10-2/16)							
Form group	completed	2/12/2020	aa, kah, ash, lmj, mht	0:15	ash	2/12/2020	
Discuss system requirements/design	completed	2/12/2020	aa, kah, ash, lmj, mht	1:30	ash	2/12/2020	
Create GitHub	completed	2/12/2020	kah	0:20	ash	2/12/2020	Had issues adding Alex to repo
Initial draft: SDS system overview and architecture	completed	2/14/2020	lmj	0:20	mht	2/14/2020	Max submitted on Canvas
Initial draft: project timeline	completed	2/14/2020	aa	1:00	mht	2/14/2020	Max submitted on Canvas
Initial draft: SRS ConOps	completed	2/14/2020	ash	8:00	mht	2/14/2020	Max submitted on Canvas
Prepare for presentation	completed	2/14/2020	aa, kah, ash, lmj, mht	1:00	ash	2/14/2020	Presentations were not given in class
Brainstorm 5 human interactions with system	completed	2/16/2020	aa, ash, lmj, mht	0:45	ash	2/16/2020	
Interview RA + transcript	completed	2/15/2020	ash	3:00	ash	2/15/2020	Interviewed two RAs
Assign tasks for initial submissions	completed	2/16/2020	aa, kah, ash, lmj, mht	0:45	ash	2/16/2020	
Discuss interview	completed	2/16/2020	aa, ash, lmj, mht	0:25	ash	2/16/2020	
Week 7 (2/17-2/23)							
Interview coordinator Erickson + transcript	completed	2/17/2020	ash	4:00	ash	2/17/2020	

Interview coordinator Shafer + transcript	completed	2/18/2020	ash	4:00	ash	2/18/2020	
Initial submission section 2 SRS	completed	2/18/2020	ash	4:00	ash	2/18/2020	
Initial submission section 3 SRS	completed	2/19/2020	mht	4:00	ash	2/19/2020	
Initial submission section 4 SRS	completed	2/18/2020	ash	2:00	ash	2/18/2020	
Initial submission SDS	completed	2/19/2020	lmj	6:00	lmj	2/19/2020	
Initial submission project plan	completed	2/18/2020	kah	3:00	ash	2/19/2020	Almost completed, edit Gantt chart
Project plan + SDS diagrams	completed	2/19/2020	aa	4:00	mht	2/19/2020	
Start working on RA preference module	completed	2/22/2020	ash	3:00	ash	2/23/2020	
Start working on shift assignments module	completed	2/22/2020	kah	4:00	ash	2/23/2020	
Start working on weekend scheduler	completed	2/24/2020	aa	5:00	lmj	2/24/2020	
Start working on weekday scheduler	completed	2/25/2020	mht	4:30	ash	2/26/2020	
Start working on on call viewer	completed	2/23/2020	lmj	3:30	ash	2/23/2020	
Create week 8 milestones	completed	2/23/2020	ash	1:00	ash	2/23/2020	
Week 8 (2/24-3/01)							
Finish RA preference module	completed	2/24/2020	ash	4:30	ash	2/26/2020	
Finish error checking	completed	2/25/2020	ash	4:00	ash	2/26/2020	
Finish shift assignment module	completed	2/25/2020	kah	5:30	ash	2/26/2020	

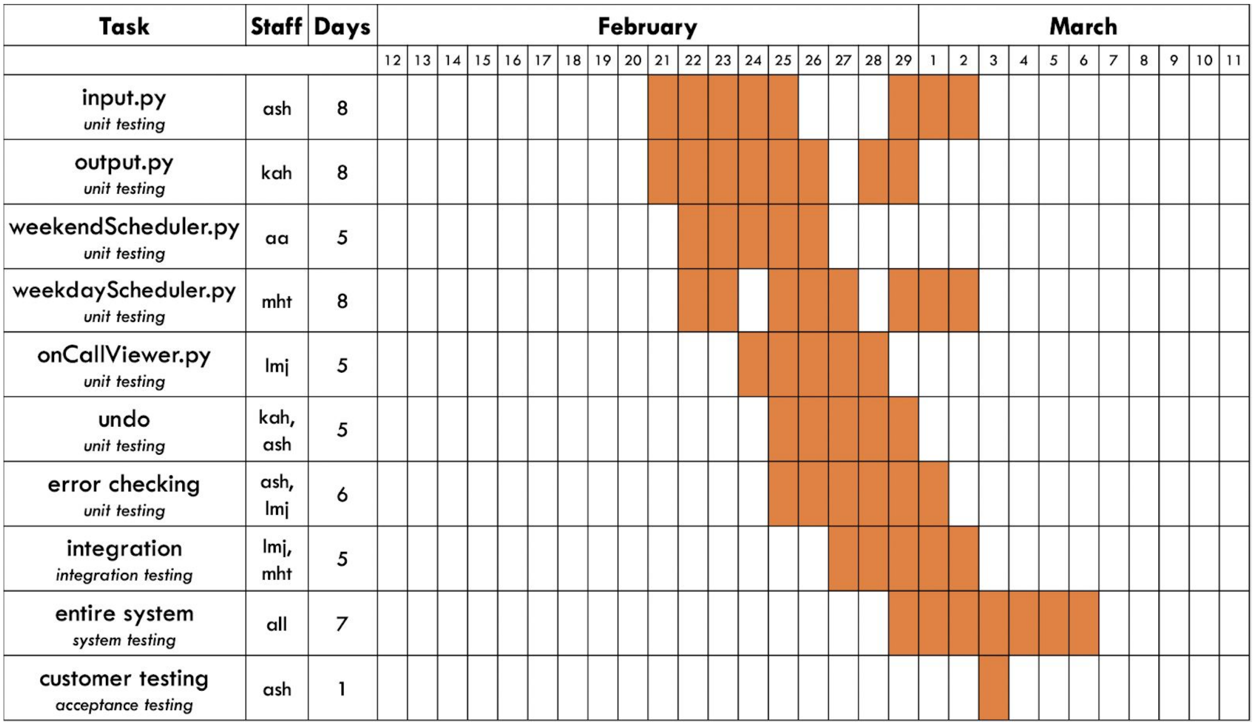
Finish weekend scheduler	completed	2/24/2020	aa	6:00	lmj	2/26/2020	
Finish weekday scheduler	completed	3/3/2020	mht	8:00	lmj	3/4/2020	
Finish GUI	completed	3/2/2020	lmj	6:00	ash	3/4/2020	
Print out SRS, SDS, and project plan	completed	2/24/2020	aa	0:20	aa, kah, ash, lmj, mht	2/24/2020	
Meet with Professor Hornof	completed	2/24/2020	aa, kah, ash, lmj, mht	0:30	aa, kah, ash, lmj, mht	2/24/2020	
Add scheduler comparison to SRS	completed	3/6/2020	ash	5:00	ash	3/6/2020	
Update Gantt chart	completed	3/1/2020	ash	1:00	ash	3/1/2020	
Update PERT chart	completed	3/1/2020	aa	0:30	ash	3/1/2020	
Write update/save/undo functions	completed	2/26/2020	kah, ash	6:30	lmj	2/26/2020	
Update SRS meeting notes and chart	completed	3/1/2020	ash	1:30	ash	3/1/2020	
Begin technical documentation	completed	3/1/2020	aa, kah, ash, mht	14:00	ash	3/2/2020	
Integrate modules	completed	3/1/2020	lmj	7:00	aa, kah, ash, lmj, mht	3/1/2020	
Discuss how to implement undo function	completed	2/26/2020	aa, kah, ash, lmj, mht	1:00	aa, kah, ash, lmj, mht	2/26/2020	
Begin implementing undo function	completed	2/24/2020	kah, ash	6:30	lmj	2/26/2020	
Update SDS	completed	3/5/2020	kah	4:00	kah	3/6/2020	
Update week 8 milestones	completed	2/26/2020	ash	1:30	ash	2/26/2020	
Create week 9 milestones	completed	3/1/2020	ash	1:00	ash	3/1/2020	

9 (3/02-3/08)							
Customer testing	completed	3/3/2020	ash	0:45	ash	3/3/2020	Revisions to system
Finish alphabetical and numerical by student ID tiebreakers	completed	3/3/2020	mht	1:30	lmj	3/4/2020	
Finish bad pairings	completed	3/3/2020	mht	2:00	lmj	3/4/2020	
Finish export report functionality	completed	3/5/2020	mht	9:00	lmj	3/6/2020	
Finish project plan	completed	3/8/2020	aa, kah, ash, lmj, mht	4:00	aa, kah, ash, lmj, mht	3/8/2020	
Finish SRS	completed	3/8/2020	aa, kah, ash, lmj, mht	6:00	aa, kah, ash, lmj, mht	3/8/2020	
Finish SDS	completed	3/8/2020	aa, kah, ash, lmj, mht	5:00	aa, kah, ash, lmj, mht	3/8/2020	
Finish technical documentation	completed	3/8/2020	aa, kah, ash, lmj, mht	4:00	aa, kah, ash, lmj, mht	3/8/2020	
Test application	completed	3/6/2020	aa, kah, ash, lmj, mht	5:00	aa, kah, ash, lmj, mht	3/6/2020	

Table 8. Task Completion Schedule for 3-08-2020. Shows the status,
Progress, and outcomes of milestones by 3-08-2020.

6. Testing

6.1. Testing Schedule



Task	Staff	Days	February																		March										
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11
input.py *unit testing*	ash	8										■	■	■	■	■				■	■	■									
output.py *unit testing*	kah	8										■	■	■	■	■	■		■	■											
weekendScheduler.py *unit testing*	aa	5											■	■	■	■	■														
weekdayScheduler.py *unit testing*	mht	8											■	■		■	■	■		■	■	■									
onCallViewer.py *unit testing*	lmj	5													■	■	■	■	■												
undo *unit testing*	kah, ash	5														■	■	■	■	■											
error checking *unit testing*	ash, lmj	6														■	■	■	■	■	■										
integration *integration testing*	lmj, mht	5																■	■	■	■	■									
entire system *system testing*	all	7																		■	■	■	■	■	■	■					
customer testing *acceptance testing*	ash	1																					■								

Figure 18. Timetable for testing code. Shows the timeline during the development lifecycle for documenting our progress as of 3-08-2020.

6.2. Testing Summaries

6.2.1. Unit Testing - onCallViewer.py

onCallViewer.py was tested by repeatedly using the application. Since onCallViewer.py is what creates the user interface, actually running the application was the best way to test it. When writing the code, everytime a new widget was added to the application, the addition was tested. If the layout changed, the application was tested for usability. Occasionally, an existing widget would be altered—such as remapping a button or changing the text of a label—so the application would be run to test this change. This testing mainly focused on usability and appearance. It checked to make sure all of the functionality was present and the layout made sense. Part of testing the appearance of the application involved having others use it and getting feedback from them.

onCallViewer.py is what effectively integrates the rest of the code. This meant the testing for onCallViewer.py also included testing integration and the application as a whole. After connecting widgets, typically buttons, to functions in other modules, the application would be run to test if the expected behavior occurred. For activities such as importing RAs, the application would be run several times using different inputs to test how the system reacted to different information. Another example is how different settings would be chosen to make sure various combinations all worked and behaved as expected. Testing widgets, such as the undo buttons, required stepping through other parts of the system before being able to test the intended widget. This meant portions of the application were used several times in different patterns allowing the chance for more bugs to be discovered.

Section 6.2.7. details the system testing. Most of onCallViewer.py's testing happened as a system test at different stages of development.

6.2.2. Unit Testing - input.py

input.py was tested with example input files that can be found in testInput. This file parses through a given file and, if it is formatted correctly, stores the information in the raPreferences dictionary. This parsing functionality was testing with example files that fit the formatting: <student ID>,<first and last name>,<weekday preference 1>,<weekday preference 2>,<weekday preference 3>,<weekend requested off 1>,<weekend requested off 2>,<weekend requested off 3>. Additionally, whenever a file is imported it checks if the given file has any shared keys in the current raPreferences dictionary. If it is, meaning one RA is having their information twice imported, it will update the relevant information in raPreferences to the most recent import. This update functionality was tested with updatedexample.csv in testInput.

input.py also raises errors if the input is incorrectly formatted to be parsed. Error checking was created and checked for by using faulty inputs that would create an error if parsed. Some examples of this can be found in the testInput file (Same weekday chosen.csv, Same Weekend Off.csv, or Max RAs.csv).

6.2.3. Unit Testing - weekdayScheduler.py

weekdayScheduler.py was initially tested using example raPreferences dictionaries constructed by the tester. These tests ensured that the algorithm was working as expected and that a weekday schedule could be produced. Prior to integration with the user interface, weekdayScheduler.py utilized print statements in the terminal to show the produced schedule. A simple examination was done on the file's output to make sure it appeared reasonable.

Once weekdayScheduler.py was able to produce a schedule, a summary report feature was created to test the application more extensively for accuracy. This summary report initially only counted the number of shifts assigned to each person. After the acceptance test, this summary functionality was integrated with the overall system and produced a more comprehensive summary.

After ensuring the algorithm was doing its basic functionality, edge cases were created to find situations in which the algorithm would crash. Upon finding these errors, the algorithm was adjusted to accommodate for this or it was communicated to other modules to generate an error should these cases occur. This was done to account for cases in which the algorithm could potentially crash.

6.2.4. Unit Testing - weekendScheduler.py

weekendScheduler.py was initially tested with a file containing an example dictionary of RA Preferences. This example dictionary was of the same format that would be generated by input.py. At group meetings we discussed the exact format for the RA Preferences dictionary so both weekendScheduler.py could be coded and tested without needing input.py to be completely finished. Print statements were used throughout the code to ensure a schedule was being outputted and the algorithm was working as expected. Since RAs were chosen randomly in a while loop, the program had to be run dozens of times to identify any errors that resulted in a timeout. The logic to enforce the number of shifts and the balance between primary and secondary shifts was tested by creating a function to print each RA's distribution. By running the program with a dozen different raPreferences the conditions for the while loops tightened. Further the variance between each RAs average number of shifts, and between the number of primary to secondary shifts dropped. The program was also tested to run on the same raPreferences dictionary multiple times to produce new results each time if possible. This resulted in catching several errors related to minimum sized raPreferences that were ultimately accounted for by the raPreferences input acceptance. Keyword arguments were also experimented with to try and account for additional exporting functionality that was later adopted elsewhere and dropped. weekendScheduler.py was tested by creating edge cases and ensuring that potential errors were accounted for prior to generating the schedule to prevent algorithm crashes.

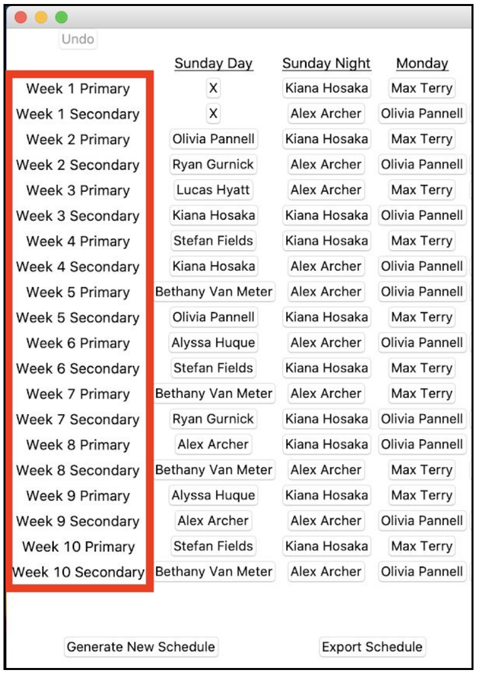
6.2.5. Unit Testing - output.py

output.py was tested with test files containing example shift assignments in the correct format that would be returned by weekendScheduler.py and weekdayScheduler.py. These test files were used for testing rather than the output of weekendScheduler.py and weekdayScheduler.py to achieve loose coupling. By doing this, any errors found in output.py would be sure to have occured in output.py rather than in weekendScheduler.py or weekdayScheduler.py. Coverage testing was conducted to ensure that every function was called at least once. Error based testing was conducted to ensure thorough testing was done in error prone areas. Places that were considered to be error prone were places where function from other files were called, such as weekendShifts and weekdayShifts from weekendScheduler.py and weekdayScheduler.py. Because these functions returned the true shift assignments, it was thoroughly tested to ensure that the format was compatible with what was written in output.py. The undo function in output.py was also tested thoroughly because this function was susceptible to various errors that aren't entirely visible to the programmer.

6.2.6. Acceptance Testing - Ethan Shafer

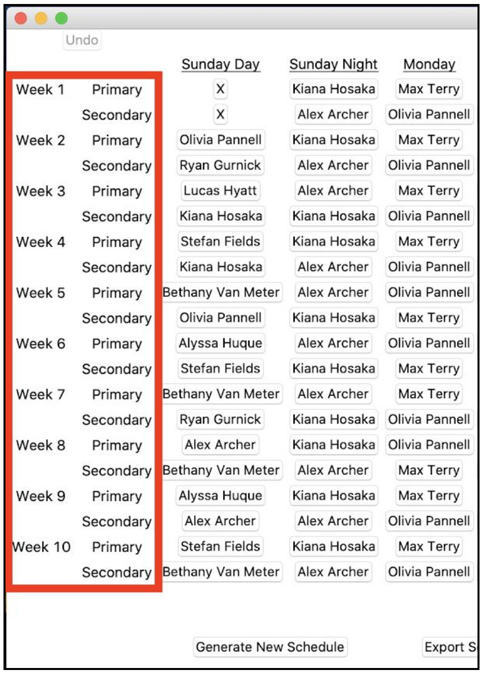
Alyssa Huque conducted an acceptance test with the targeted consumer audience of *On Call*, an RA coordinator. Alyssa met with coordinator Ethan Shafer on March 3, 2020 to try out the system. Alyssa provided no guidance to Shafer on how to use the system at first. Shafer was able to successfully import a file, update preferences in the GUI, undo preference changes, generate a schedule, and export the schedule without any input from Alyssa. While interacting with *On Call*, Shafer brought up some areas for system improvement that was incorporated into our final product:

- RA Preferences: Shafer asked for the warning when deleting an RA to state that deletion was permanent. This would prevent coordinators from attempting to undo a deletion.
- Shift Assignments: stated the method of displaying the assignments “Week 1 Primary,” “Week 1 Secondary,” “Week 2 Primary, “Week 2 Secondary,” and so on was difficult to read. Asked that the week number and primary/secondary label be in two different columns.



	Sunday Day	Sunday Night	Monday
Week 1 Primary	X	Kiana Hosaka	Max Terry
Week 1 Secondary	X	Alex Archer	Olivia Pannell
Week 2 Primary	Olivia Pannell	Kiana Hosaka	Max Terry
Week 2 Secondary	Ryan Gurnick	Alex Archer	Olivia Pannell
Week 3 Primary	Lucas Hyatt	Alex Archer	Max Terry
Week 3 Secondary	Kiana Hosaka	Kiana Hosaka	Olivia Pannell
Week 4 Primary	Stefan Fields	Kiana Hosaka	Max Terry
Week 4 Secondary	Kiana Hosaka	Alex Archer	Olivia Pannell
Week 5 Primary	Bethany Van Meter	Alex Archer	Olivia Pannell
Week 5 Secondary	Olivia Pannell	Kiana Hosaka	Max Terry
Week 6 Primary	Alyssa Huque	Alex Archer	Olivia Pannell
Week 6 Secondary	Stefan Fields	Kiana Hosaka	Max Terry
Week 7 Primary	Bethany Van Meter	Alex Archer	Max Terry
Week 7 Secondary	Ryan Gurnick	Kiana Hosaka	Olivia Pannell
Week 8 Primary	Alex Archer	Kiana Hosaka	Olivia Pannell
Week 8 Secondary	Bethany Van Meter	Alex Archer	Max Terry
Week 9 Primary	Alyssa Huque	Kiana Hosaka	Max Terry
Week 9 Secondary	Alex Archer	Alex Archer	Olivia Pannell
Week 10 Primary	Stefan Fields	Kiana Hosaka	Max Terry
Week 10 Secondary	Bethany Van Meter	Alex Archer	Olivia Pannell

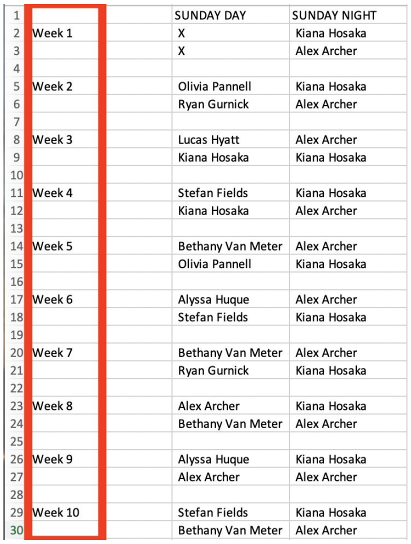
Figure 19. The previous formatting of the y-axis. This formatting allowed a user to look at the row and see which week and shift the RAs in that row were assigned to.



		Sunday Day	Sunday Night	Monday
Week 1	Primary	X	Kiana Hosaka	Max Terry
	Secondary	X	Alex Archer	Olivia Pannell
Week 2	Primary	Olivia Pannell	Kiana Hosaka	Max Terry
	Secondary	Ryan Gurnick	Alex Archer	Olivia Pannell
Week 3	Primary	Lucas Hyatt	Alex Archer	Max Terry
	Secondary	Kiana Hosaka	Kiana Hosaka	Olivia Pannell
Week 4	Primary	Stefan Fields	Kiana Hosaka	Max Terry
	Secondary	Kiana Hosaka	Alex Archer	Olivia Pannell
Week 5	Primary	Bethany Van Meter	Alex Archer	Olivia Pannell
	Secondary	Olivia Pannell	Kiana Hosaka	Max Terry
Week 6	Primary	Alyssa Huque	Alex Archer	Olivia Pannell
	Secondary	Stefan Fields	Kiana Hosaka	Max Terry
Week 7	Primary	Bethany Van Meter	Alex Archer	Max Terry
	Secondary	Ryan Gurnick	Kiana Hosaka	Olivia Pannell
Week 8	Primary	Alex Archer	Kiana Hosaka	Olivia Pannell
	Secondary	Bethany Van Meter	Alex Archer	Max Terry
Week 9	Primary	Alyssa Huque	Kiana Hosaka	Max Terry
	Secondary	Alex Archer	Alex Archer	Olivia Pannell
Week 10	Primary	Stefan Fields	Kiana Hosaka	Max Terry
	Secondary	Bethany Van Meter	Alex Archer	Olivia Pannell

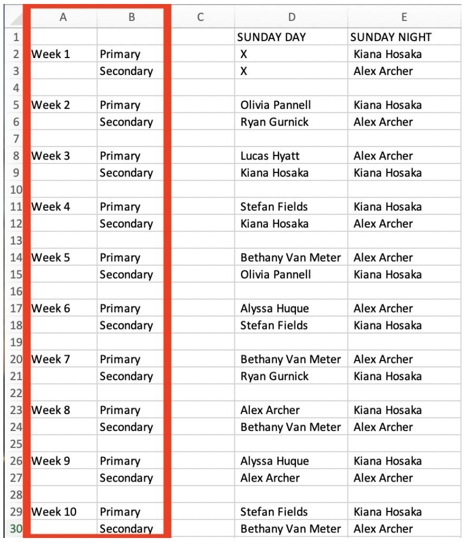
Figure 20. The updated formatting of y-axis based on Shafer's feedback. This formatting is neater with less information in each column.

- Shift Assignments: found an error with the day labeling. Instead of one column labeled as "Saturday Day" and the next column labeled as "Saturday Night," there were two columns that were both labelled "Saturday Day" by mistake.
- Shift Assignments: Asked that the exported schedule CSV explicitly state which row was a primary and secondary shift.



1			SUNDAY DAY	SUNDAY NIGHT
2	Week 1		X	Kiana Hosaka
3			X	Alex Archer
4				
5	Week 2		Olivia Pannell	Kiana Hosaka
6			Ryan Gurnick	Alex Archer
7				
8	Week 3		Lucas Hyatt	Alex Archer
9			Kiana Hosaka	Kiana Hosaka
10				
11	Week 4		Stefan Fields	Kiana Hosaka
12			Kiana Hosaka	Alex Archer
13				
14	Week 5		Bethany Van Meter	Alex Archer
15			Olivia Pannell	Kiana Hosaka
16				
17	Week 6		Alyssa Huque	Alex Archer
18			Stefan Fields	Kiana Hosaka
19				
20	Week 7		Bethany Van Meter	Alex Archer
21			Ryan Gurnick	Kiana Hosaka
22				
23	Week 8		Alex Archer	Kiana Hosaka
24			Bethany Van Meter	Alex Archer
25				
26	Week 9		Alyssa Huque	Kiana Hosaka
27			Alex Archer	Alex Archer
28				
29	Week 10		Stefan Fields	Kiana Hosaka
30			Bethany Van Meter	Alex Archer

Figure 21. The previous formatting of the y-axis. This formatting only allowed a user to see what week an RA was assigned to and required a coordinator to infer that the first row was a primary shift and the second row was a secondary shift.



	A	B	C	D	E
1				SUNDAY DAY	SUNDAY NIGHT
2	Week 1	Primary		X	Kiana Hosaka
3		Secondary		X	Alex Archer
4					
5	Week 2	Primary		Olivia Pannell	Kiana Hosaka
6		Secondary		Ryan Gurnick	Alex Archer
7					
8	Week 3	Primary		Lucas Hyatt	Alex Archer
9		Secondary		Kiana Hosaka	Kiana Hosaka
10					
11	Week 4	Primary		Stefan Fields	Kiana Hosaka
12		Secondary		Kiana Hosaka	Alex Archer
13					
14	Week 5	Primary		Bethany Van Meter	Alex Archer
15		Secondary		Olivia Pannell	Kiana Hosaka
16					
17	Week 6	Primary		Alyssa Huque	Alex Archer
18		Secondary		Stefan Fields	Kiana Hosaka
19					
20	Week 7	Primary		Bethany Van Meter	Alex Archer
21		Secondary		Ryan Gurnick	Kiana Hosaka
22					
23	Week 8	Primary		Alex Archer	Kiana Hosaka
24		Secondary		Bethany Van Meter	Alex Archer
25					
26	Week 9	Primary		Alyssa Huque	Kiana Hosaka
27		Secondary		Alex Archer	Alex Archer
28					
29	Week 10	Primary		Stefan Fields	Kiana Hosaka
30		Secondary		Bethany Van Meter	Alex Archer

Figure 22. The updated formatting of y-axis based on Shafer's feedback.
This formatting makes the primary and secondary shift assignment explicit.

- Shift Assignments: requested that there was an ability to generate a report so the coordinator could see which weekday an individual was assigned to and how many shifts each RA had.

Shafer also made some suggestions that were not incorporated into *On Call* but could be during the maintenance stage of this product.

- Produce a calendar, rather than a schedule, that was specific to the dates in the term.
- Allow for 'special shifts,' such as a holiday, that might require an extra RA to be on call or day shift on a weekday (i.e. MLK Day requires a Monday day shift)
- Allow RAs to specify if they prefer a 12 or 24 hour shift on a weekend and incorporate their preference when scheduling the weekend shifts.
- Allows RAs to give a date range to request off rather than a week number.

Shafer also clarified that when updating preferences in the user interface it does not edit the original file. Alyssa explained it would not, edits made in the system would stay in the system and Shafer was happy with this. Additionally, Shafer was excited by the gold star RA and bad pairings feature as he did not anticipate our system having this requirement and was glad the scheduling process was not entirely automated. Shafer seemed to be pleased as it allowed his understanding of the team dynamic to impact the schedule output.

Shafer was able to use *On Call* with ease and little guidance and was highly satisfied with the system.

6.2.7. System Testing - *On Call*

Alyssa, Kiana, and Alex did coverage-based testing of *On Call*. This required Alyssa, Kiana, and Alex to independently walk through the system as a coordinator would to see if any errors arise or unexpected cases are encountered. Through this method many errors were found. Some examples of them are:

- When exporting the summary report, an index error was occurring the first time someone exported a summary report due to a logic error.
- The screen needs to be resized when there are over 18 RAs in order for the delete button to be seen.
- An infinite loop was generated when the gold star RA was also a bad pairing RA. This was due to an edge case that was not accounted for in the weekend scheduler algorithm.
- An error was not generated when the total RAs in the system was greater than 25, an unreasonably large size for an RA team that would only occur if the coordinator made a mistake when importing RAs to the system.

Lily and Max did fault-based testing of *On Call*. This involved both Lily and Max using the system with the intent to generate an error. Some of the errors found were:

- Alyssa forgot to account for the case when a user edits an RA, deletes the RA, and then presses undo. This would cause an error because the undo function would try to change an RA that does not exist anymore.
- Kiana forgot to account for the case when a user edits the schedule, deletes the schedule, and creates a new schedule. The undo button could still be clicked, causing the new schedule to change a shift to an RA who was there in a previous schedule.
- No error is generated when there are too many fields given in the input.
- Error catching was needed when a user puts the same RA as a pair for the bad pairings setting.
- *On Call* was unable to handle when None was given for the gold star RA.

While doing system testing, the summary report feature, which calls functions from exportSummary.py, helped to see the validity of our system. This improved testing as it informed the tester how well the algorithm was functioning for both the system and edge cases. All the issues that were discovered during testing, both the coverage-based and fault-based testing, were resolved before the final submission of *On Call*.

7. Individual Task Breakdown Assignments

7.1. Alex Archer

Task	Assigned Date	Completed Date	Time Spent (Hours)
Write Timeline Description	2-12-2020	2-13-2020	1
Create PERT Chart	2-12-2020	2-13-2020	3
Proofreading Initial Documents	2-12-2020	2-13-2020	1
Create Use Case Diagrams	2-14-2020	2-18-2020	6
Code Weekend Scheduler	2-24-2020	2-26-2020	7
Proofreading Final Documents	2-24-2020	3-07-2020	8
Update PERT Charts	2-26-2020	3-07-2020	2
Write Technical Documentation Section 5.5	3-01-2020	3-5-2020	1.5
Update Use Case Diagrams	3-01-2020	3-06-2020	3
Debug Weekend Scheduler	3-02-2020	3-03-2020	2
Update SDS 4.4.1	3-05-2020	3-06-2020	2

Table 9. Assignment Breakdown for Alex Archer. Shows the tasks, date assigned, date completed, and time spent on those tasks.

7.2. Kiana Hosaka

Task	Assigned Date	Completed Date	Time Spent (Hours)
Start SDS	2-12-2020	2-12-2020	1
Start SRS Section 2	2-12-2020	2-12-2020	2.5
Write Project Plan Section 2	2-12-2020	2-18-2020	3
Code Shift Assignment Module	2-12-2020	2-20-2020	5
Code Error Checking	2-12-2020	2-26-2020	0.75
Code Undo function for Shift Assignments	2-24-2020	2-26-2020	3
Comment Code	2-24-2020	2-26-2020	1
Write Technical Documentation	2-24-2020	3-1-2020	4
Editing SDS	3-1-2020	3-6-2020	9
Fixing Integration Bug	3-1-2020	3-1-2020	3
Testing Documentation	3-4-2020	3-4-2020	0.5
System Testing	3-4-2020	3-6-2020	4
Proofread Section 4 of Technical Documentation	3-4-2020	3-6-2020	2
Review Suggestions in Section 5.6 of Technical Documentation	3-7-2020	3-7-2020	0.5
Edited Section 2 of SRS	3-7-2020	3-7-2020	2
Edited Section 4 of the SRS	3-7-2020	3-7-2020	1

Edited Section 5 of the SRS	3-7-2020	3-8-2020	1
Proofread Section 5.6 and 5.7 of Technical Documentation	3-8-2020	3-8-2020	.5
Proofread and Edited Section 6 of Project Plan	3-8-2020	3-8-2020	1

Table 10. Assignment Breakdown for Kiana Hosaka. Shows the tasks,
date assigned, date completed, and time spent on those tasks.

7.3. Alyssa Huque

Task	Assigned Date	Completed Date	Time Spent (Hours)
Write SRS section 2	2-12-2020	2-13-2020	12
Interview RA Werts	2-12-2020	2-15-2020	1
Type RA Werts transcript	2-12-2020	2-15-2020	3
Interview coordinator Joseph Erickson	2-12-2020	2-17-2020	0.75
Type Erickson transcript	2-12-2020	2-17-2020	3.5
Interview coordinator Ethan Shafer	2-12-2020	2-18-2020	0.5
Type Shafer transcript	2-12-2020	2-18-2020	3
Update milestones spreadsheet	2-12-2020	3-8-2020	2
Proofread initial documentation	2-14-2020	2-20-2020	3
Write SRS section 4	2-15-2020	2-18-2020	2
Maintain project plan section 7	2-16-2020	3-8-2020	2
Update Gantt chart	2-18-2020	2-19-2020	3.5
Code RA preference module	2-20-2020	2-25-2020	6
Comment code	2-20-2020	2-26-2020	1
RA preferences unit testing	2-20-2020	3-2-2020	4
Code error checking	2-23-2020	2-25-2020	4
Code undo function for RA preferences	2-24-2020	2-26-2020	3.5

Utilizing proprietary scheduling software	2-24-2020	3-1-2020	2
Add scheduler comparison to SRS	2-24-2020	3-4-2020	8
Update SRS	2-24-2020	3-8-2020	4
Update Project Plan	2-24-2020	3-8-2020	10
Testing documentation	2-24-2020	3-4-2020	4
Write technical documentation	2-26-2020	3-8-2020	10
Acceptance testing with coordinator	3-3-2020	3-3-2020	0.5
System testing	3-3-2020	3-6-2020	8

Table 11. Assignment Breakdown for Alyssa Huque. Shows the tasks, date assigned, date completed, and time spent on those tasks.

7.4. Lily Jim

Task	Assigned Date	Completed Date	Time Spent (Hours)
Create All Document Templates	2-12-2020	2-12-2020	2
Create Gantt Chart	2-12-2020	2-12-2020	1
Create Sample Input and Output Templates	2-12-2020	2-12-2020	1
Write SDS sections 2-3	2-12-2020	2-13-2020	5
Catch Alex up on Missed Meeting	2-12-2020	2-12-2020	6
Assist Alex with PERT Chart and Timeline Description	2-12-2020	2-13-2020	4
Create Initial Technical Documentation (File and Function Names)	2-13-2020	2-13-2020	1
Proofreading Initial Draft Documents	2-13-2020	2-14-2020	4
Write SDS sections 4, excluding models	2-12-2020	2-16-2020	6
Update SDS with additional functionality	2-16-2020	2-18-2020	4
Create SDS section 4 models	2-12-2020	2-19-2020	4
Proofreading Initial Documents	2-19-2020	2-19-2020	4
Create basic outline of Python files	2-19-2020	2-19-2020	1

Write initial onCallViewer.py	2-22-2020	2-23-2020	4
Work on integration	2-22-2020	2-26-2020	3
Initial Code Review	2-22-2020	2-28-2020	4
Continue GUI	2-26-2020	3-1-2020	6
Continue Integration	2-26-2020	3-1-2020	1
Finalize Integration	3-1-2020	3-7-2020	6
Finalize GUI	3-1-2020	3-7-2020	10
Write Technical Documentation for onCallViewer.py	3-1-2020	3-6-2020	3
Write Technical Documentation sections 2, 3, and 4	3-4-2020	3-6-2020	2
Proofread and Fix Technical Documentation for Consistency and Accuracy	3-4-2020	3-7-2020	3
Update SDS Diagrams	3-4-2020	3-7-2020	1
Assist Alex with Section 5 of the SDS	3-4-2020	3-7-2020	1.5
Review and Update SDS	3-4-2020	3-7-2020	6
Assist Alex to Meet Good Writing Standards	3-4-2020	3-7-2020	3
Proofread All Documents	3-7-2020	3-7-2020	4

Table 12. Assignment Breakdown for Lily Jim. Shows the tasks,
date assigned, date completed, and time spent on those tasks.

7.5. Max Terry

Task	Assigned Date	Completed Date	Time Spent (Hours)
Wrote SRS Section 3	2-12-2020	2-18-2020	5
Proofreading initial documentation	2-12-2020	2-19-2020	4
Initial weekday scheduler	2-20-2020	2-29-2020	10
Weekday scheduler day assignment algorithm	2-20-2020	2-29-2020	11
Weekday scheduler primary/secondary balance	2-20-2020	2-29-2020	1
Write technical documentation	2-20-2020	3-1-2020	3
Error testing weekday scheduler	2-22-2020	3-2-2020	10
Code basic algorithm summary report	2-25-2020	3-5-2020	14
Error based system testing	2-27-2020	3-6-2020	4
Project Plan 6.2.3. and 7.5	3-1-2020	3-6-2020	2
Code summary report for entire system	3-2-2020	3-5-2020	2
Proofread final documentation	3-6-2020	3-8-2020	3

Table 13. Assignment Breakdown for Max Terry. Shows the tasks, date assigned, date completed, and time spent on those tasks.

8. References

van Vliet, Hans. (2008). Software Engineering: Principles and Practice, 3nd edition, John Wiley & Sons.

9. Acknowledgments

This document layout is based on the On Deck Development Team's Project Plan document from project one in Anthony Hornof's CIS 422 class at the University of Oregon during winter term 2020.

Information from van Vliet's textbook is used for section 3.