Section 1 Schedule So-far:

We have tightly followed the schedule made in Final Project Proposal and have completed the sequential version of our code. Based on the course schedule, we decide to work on our final project following such timeline before the poster session:

	Meeting 1 (Mon.)	Meeting 2 (Thurs.)	Instructor Communication (Wed.)
W1(Mar 31st to Apr. 7th)	Finish the final project proposal, decide main focus of the project, and assign work to each week	Search for resources (papers/ relevant open-source project information) and perform different ideations and decide which one to work on	Discuss various ideations with instructors during OH and ask for their advice (which model could be better to implement and if any other resources/APIs could be considered when implementing the model
W2(Apr.10th to Apr. 14th)	Implement basic structure of the model with available resources and discuss obstacles encountered, and prepare to implement the whole model and **start Milestone Report**	Done with the general model implementation as first prototype and recheck the defects of the model, think about possible ways of optimization while starting performance testing	Show the first prototype to instructors and ask for their advice and possible ways to optimize. Again, think about any more resources that can be added to the prototype during future optimization periods. Moreover, ask about any other aspects of program could be tested but not considered yet
W3(Apr. 17th to Apr. 21th)	Firstly finish implementing parallel version of the algorithm and then time the performance and optimizing the first prototype and complete second version of model with performance test and brainstorm other possible ways of optimization and **submit Milestone Report**	Finish second prototype and refine the model to final version with advice from instructors and **start writing poster and final report**	Ask instructors for help on optimizing the second prototype and maybe more ways of testing and optimization
W4(Apr. 24th to Apr. 28th)	Test final performance/ improvements brought by our model and write the report and design poster, finishing it before meeting instructors	Refine final report based on advice given by instructors and redesign the poster according to final report of new version	Check details of report and poster with instructors and ask for advice on refining
W5(May 1st to May 5th)	Ready to **submit the**	**Finish the poster**	Final advice on poster and check

	final report		if anything needs to be changed

Section 2 Work Summary:

As a project that we start from scratch, an important part of our work is to build up the network which includes a random data generator, all the structures needed for our algorithm, input/output file managing codes and testing codes. Thus, for the first three weeks, we completed these parts as mentioned in our schedule. More specifically they are:

- Basic structures: adjacency matrix input/output files, structures for people, connection between people, read/write files functions, makefile
- Codes: random generator for adjacency matrix input data, network randomization (changes in each iteration of the loop of our simulation steps)

Besides, we have finished implementing all the sequential versions of our algorithm:

1. Network_simulator_seq.cpp is a script where one degree of friendship is considered (people only gather information from their direct friend and make evaluation based on these data) and no randomization of friends took place in the iterations.
2. Network_simulator_seq2.cpp is a script where n>1 degrees of friendship is considered (people not only gather information from their direct friend, but also gather information from the friends of friends multiplying a discounting factor) and uses randomization of connections to simulate a more realistic human network connection.

Section 3 Goals and Deliverables:
In our Final project proposal, we state that the goal of the system is to parallelize the computation of social networks with multiple degrees and compare the speedup with the corresponding sequential version of the code. At the same time, since the like-value of each person changes completely randomly (implemented with a random number generated algorithm) and then we will always test how different scheduling approaches will affect the performance of the system. For now, we have successfully implemented the sequential version of the algorithm that runs with different degrees and are working on implementing MPI version of code and trying to achieve better speedup with parallelism. Our current is coherent to the purpose stated before and will incorporate performances of different scheduling approaches. Overall, the new list of goals will be similar to what we have before but with slightly different modifications:

- Speedup comparison between sequential and parallel version (implemented by MPI)
- Incorporate with different scheduling approaches
- Analyze the reason behind the speedup brought by parallelism under different scenarios (dense/sparse graph, intense/mild randomness, etc)

Section 4: Demo presentations:

In our final demo session, we plan to show a speedup chart which describes the speedup we achieve with parallelism using MPI in comparison with the sequential version we have. We will demonstrate how and why these speedups are achieved and how these results are meaningful to our understanding of human networking and the usage of parallelism in network computations.

Section 5 Preliminary Results:
As we have only implemented sequential versions of our code and are still working on parallelizing the algorithm using MPI, the performance analysis can merely describe how runtime changes when the degree increases and will not provide too much useful information toward our final result or speedup. Therefore, we leave our current preliminary result as blank and will fill this out immediately after we have done the whole project.

Section 6 Concerns:

As we have not yet completed the parallel algorithm, here are some of the concerns that we have:

1. Figuring out commands that would allow us to run our scripts in multi-core: makeFile
2. what and how the system overall is able to benefit from the parallelism: we may encounter large overhead because the messaging is huge in this project
3. Exact ways to parallelize the computation and how we can send messages to each processor (detailed implementation): each person may expect a different number of connections/messages.
4. In what ways can we best distribute work evenly to the workers: we have ideas about predicting the individual workload but how that will benefit the time cost is still unknown.

Section 7 Instructor Meetup:
We have signed up for a meeting with the instructor on Wednesday Morning to discuss our current progress and any modifications needed to better complete the project.