Albert Lee

MSDS 460

Assignment 2 – Network Model – Project Management

**Project Introduction**

The project of creating a consumer-faced restaurant recommendation system would be a custom application (apps) since it's something that is currently not available in the market. The custom application would require a variety of skills that would not be traditionally available to a Data Scientist. These types of custom apps typically require a variety of languages needed for development outside of Python or R which include C#, Java, JavaScript, CSS, etc. Currently the dataset "restaurants-v001.json" is static and the system requires monthly feeds which would require a data storage. For data store since I'm first starting out and am trying to limit my infrastructure costs, it would make sense for the system to be hosted on the cloud vs. on-prem. For a cloud provider, I'm recommending AWS because of it's industry leading and has the largest market share. This would also allow me to find data engineers easier at average market rates vs the other cloud platforms where they may be more in demand. The data raw, core, and semantic layer should then all be accessible on a PostgreSQL database. Again, similar to AWS, PostgreSQL is an industry leader and should allow me to find folks to help augment in data engineering easier. The back-end will be primarily developed C# with Python as the analytics engine. C# is versatile and can be used various types of computers. The front-end development will be primarily developed in JavaScript since it can allow for more interactive and dynamic features. To keep costs low, I will wear multiple hats and serve as both the data scientist and project manager coordinating development across the various developers. The other roles (front-end developer, back-end developer, and data engineer) will be hired on as subcontractors for the project.

**Project Plan**

According to a Forbes article titled *"Software Development Estimation: How Long Should It Take to Develop a Product?"* by Vlad Kytainyk estimated that typical software development takes ~4 months to ~9 months. For this project since there's also work outside of just development, I'm factoring in ~5 months (best case) to ~11 months (worst case) within the project plan. I wanted to ensure that the project was completed within a year of bidding on the proposal so that the client could have a product within a year. The expected completion on the project is ~8 months. For simplicity on the project, I assumed the average hourly rate for all positions is $62 per hour. For this type of project, I would also propose a fixed-price type contract and the hourly rates for the positions would not be shown anyways. At $62 per hour average rates, I would also include ~1.75X burdens to account for my fringe, OVH, and G&A costs bringing the fully burdened hourly rate to $108.5. When bidding, I would also bid at the average of "expectedHours" and "worstCaseHours" which comes out to 1,650.5 hours. I would then apply a 20% fee and every hour beyond the 1,650.5 hours, my fee % decreases putting my profit at risk. This is why I need to ensure the hours and deliverables are being met for the client. One thing that I would alter on the assignment and project plan would be to add a new task which is Hypercare. These types of services help provide support as the client is learning how to use the new tool and helps address any break/fix issues. I was able to successfully draw the network diagram of the project plan using Python.

**Linear Programming Model / Python Code**

Within the project plan dataset, the decision variables was the start times for each task or the “TaskID”. The objective function was to minimize the total project completion. The constraints were ensuring that no task can start before it’s predecessor finishes and no negative start times. Before I could create a Linear Programming Model, I first needed to clean the dataframe. TaskID “D” is a roll-up or a subset of multiple subtasks (D1-D8) under Develop product prototype and did not include any predecessors. I had to drop any missing or any 0’s to ensure the model would calculate correctly. Then using Python’s PuLP package and some assistance from ChatGPT, the programming model of “LpProblem('Project_Scheduling', LpMinimize)” will be used once the decision variables, objective function, and constraints are added. The decision variables of start time for each task was added next. I also added a completion time variable to add completed tasks. The objective function of minimizing total project duration was added using “model += completion_time, 'Minimize_Project_Completion_Time'”. Lastly, the constraints of task predecessors was added using a loop to first check if there were any taskIDs that had multiple predecessors and if not, take the predecessor while simultaneously ensuring that the predecessors are valid taskID’s. The final constraint was to ensure that the completion time must be at least as large as the latest finish time.

**Solution**

This resulted in optimal hours of 1,378. Similar code was then used to determine the optimal bestCaseHours, expectedHours (which should be the same as the 1,378 shown above), and worstCaseHours. The decision variables, objective function, and constraints were all the same. The Best Case optimal hours was 913 and the Worst Case Hours was 1,923 hours. The critical path for all 3 scenarios is A > D1 > D2 > D4 > D6 > D7 > D8 > F > H. The critical path helped determine which tasks were the most important and cannot be delayed. If those tasks were delayed, the overall project timeline would be extended. Again, using ChatGPT's assistance, a Gantt Chart was created in Python instead of Excel for all 3 scenarios.

**Project Overview**

For the delivery of a custom software/application that creates a consumer-focused recommendation system for more than 100 hundred restaurants in the Marlborough, MA area, I would charge a Fixed Price contract of $860,000 with a delivered prototype in 42 weeks completion (or 1,680 hours). Realistically, a prototype could be done in short as 35 weeks BUT the 7 extra weeks allows provides slack within the project timeline. If additional subcontractors were considered, then the project could be done sooner but at a higher cost. Because it's a Fixed Price contract, any delays to the project beyond the 42 weeks, the company will absorb the additional costs. Typical projects like this can last between ~5 to ~11 months. I am confident that the team can complete this within 42 weeks. If the client requires a Hypercare support period seen in many product releases like this, that would come at additional costs and a new SoW (Statement of Work) would need to be developed.

**Alternatives**

Since I'm not as familiar with creating a monte-carlo simulation in Python, I leveraged ChatGPT to see if it would be possible to do so. It appears so and what it does is it defines task durations as probability distribution. It uses a triangular distribution to model variability and samples random duration between bestCaseHours, expectedHours, and worstCaseHours. The simulation is then run 10,000 times and it solves each simulation using Linear Programming.

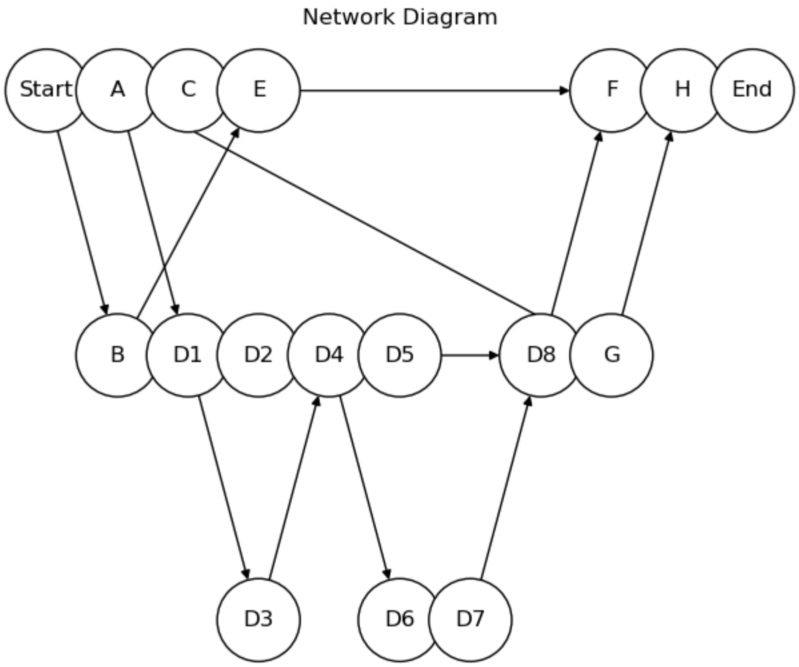
## References

- Kytainyk, V. (2024, August 13). Software development time estimation: How long should it take to develop a product? Forbes. https://www.forbes.com/councils/forbesbusinesscouncil/2022/12/02/software-development-time-estimation-how-long-should-it-take-to-develop-a-product/
- Armas, L. F. P., PhD. (2024, June 21). Efficient Project Scheduling with Python. The Critical Path

  Method. *Medium*. https://medium.com/@luisfernandopa1212/efficient-project-scheduling-with-python-the-critical-path-method-19a3f8235f91
- Armas, L. F. P., PhD. (2024a, May 17). Optimizing Project Schedules: An Expert Approach to the Resource Constrained Project Scheduling Problem (RCPSP) with Python and Pyomo. *Medium*. https://python.plainenglish.io/solving-the-resource-constrained-project-scheduling-problem-rcpsp-with-python-and-pyomo-001cffd5344a

# Appendix

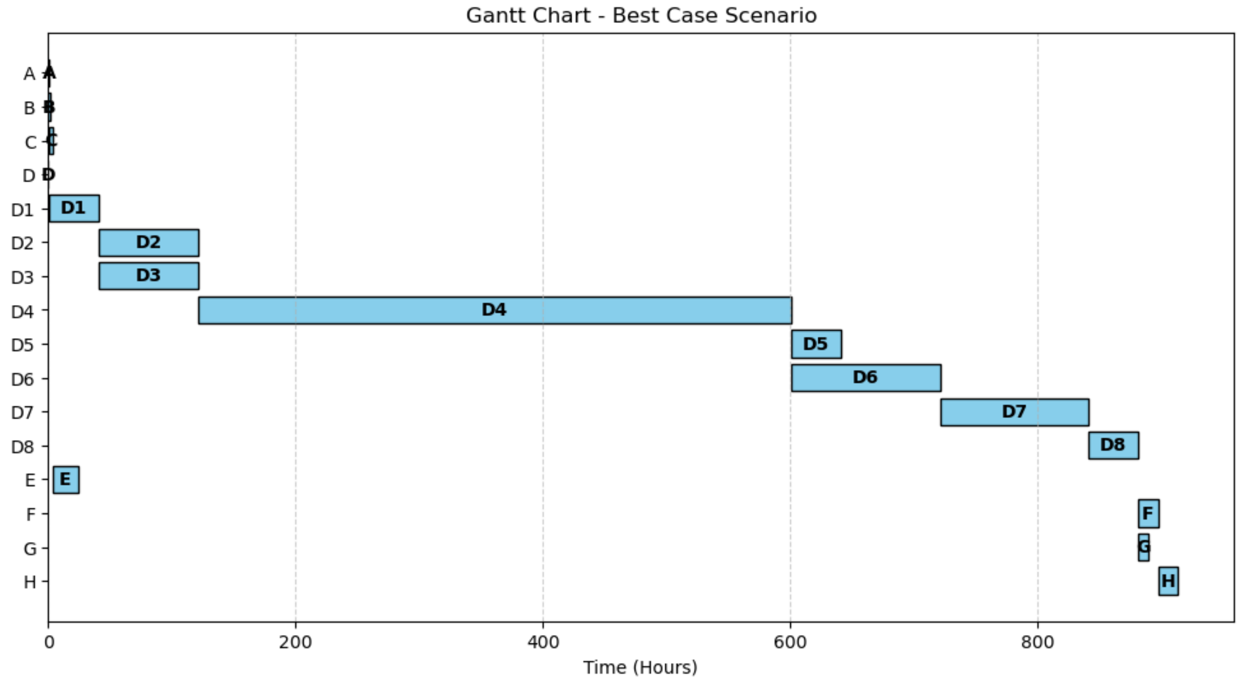
## Network Diagram



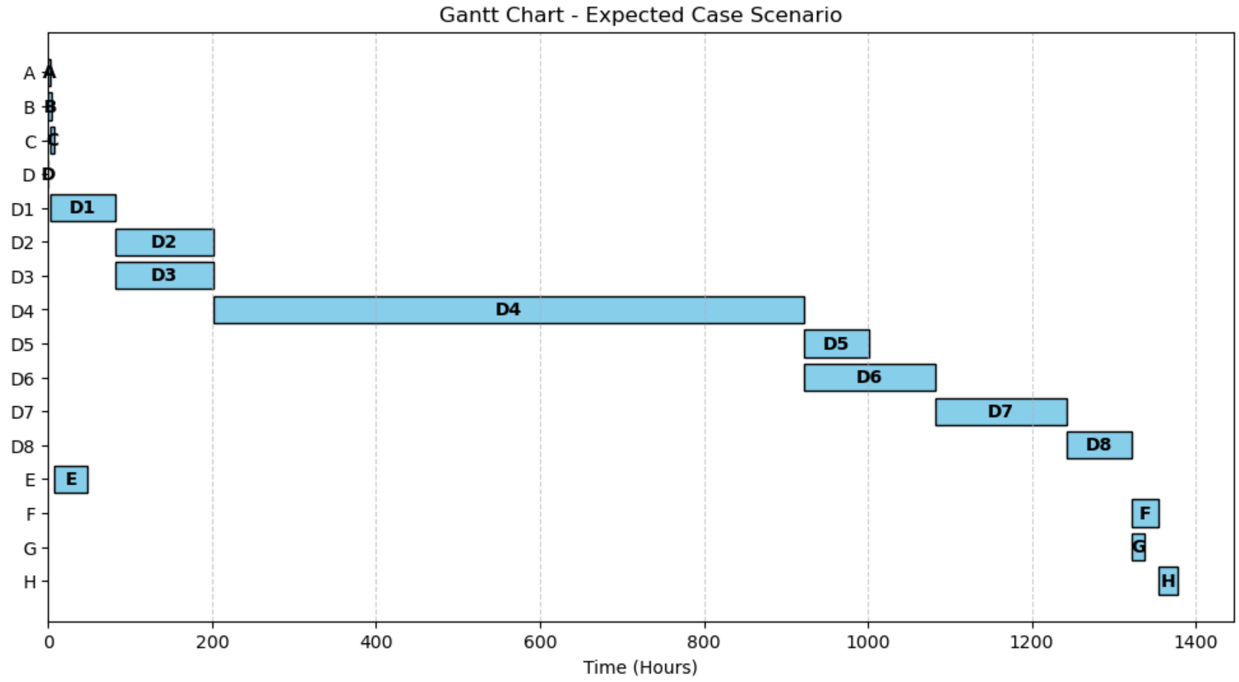
## Project Plan

| taskID | task | predecessorTaskIDs | bestCaseHours | expectedHours | worstCaseHours | projectManager | frontendDeveloper | backendDeveloper | dataScientist | dataEngineer |
|---|---|---|---|---|---|---|---|---|---|---|
| A | Describe product | | 1 | 2 | 3 | $62 | $62 | $62 | $62 | $62 |
| B | Develop marketing strategy | | 2 | 4 | 6 | $62 | $62 | $62 | $62 | $62 |
| C | Design brochure | A | 3 | 6 | 9 | $62 | $62 | $62 | $62 | $62 |
| D | Develop product prototype | | | | | | | | | |
| D1 | Requirements analysis | A | 40 | 80 | 120 | $62 | $62 | $62 | $62 | $62 |
| D2 | Software design | D1 | 80 | 120 | 160 | $62 | $62 | $62 | $62 | $62 |
| D3 | System design | D1 | 80 | 120 | 160 | $62 | $62 | $62 | $62 | $62 |
| D4 | Coding | D2, D3 | 480 | 720 | 960 | $62 | $62 | $62 | $62 | $62 |
| D5 | Write documentation | D4 | 40 | 80 | 120 | $62 | $62 | $62 | $62 | $62 |
| D6 | Unit testing | D4 | 120 | 160 | 240 | $62 | $62 | $62 | $62 | $62 |
| D7 | System testing | D6 | 120 | 160 | 240 | $62 | $62 | $62 | $62 | $62 |
| D8 | Package deliverables | D5, D7 | 40 | 80 | 120 | $62 | $62 | $62 | $62 | $62 |
| E | Survey potential market | B, C | 20 | 40 | 60 | $62 | $62 | $62 | $62 | $62 |
| F | Develop pricing plan | D8, E | 16 | 32 | 48 | $62 | $62 | $62 | $62 | $62 |
| G | Develop implementation plan | A, D8 | 8 | 16 | 24 | $62 | $62 | $62 | $62 | $62 |
| H | Write client proposal | F, G | 16 | 24 | 32 | $62 | $62 | $62 | $62 | $62 |

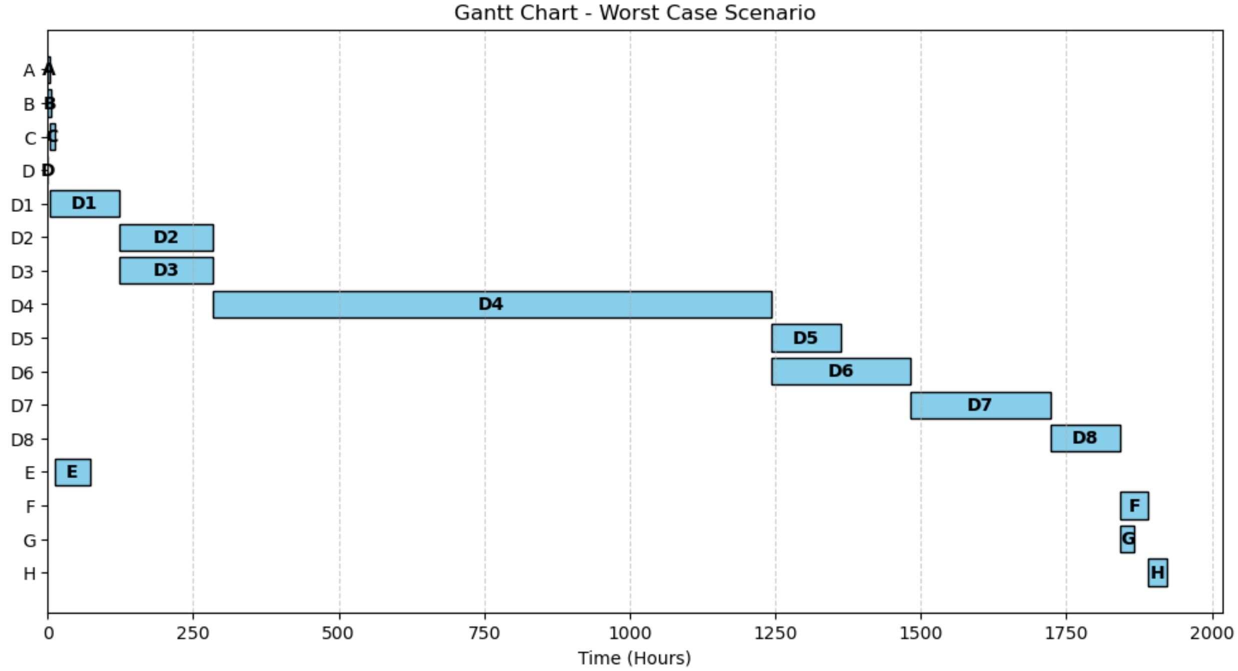
Gantt Chart – Best Case Hours



Gantt Chart – Expected Hours



Gantt Chart – Worst Case Hours


Gantt Chart - Worst Case Scenario
A
B
C
D
D1
D2
D3
D4
D5
D6
D7
D8
E
F
G
H
Time (Hours)