Agile Action

LOGO

**Introduction**

This document outlines collaborative work completed by our team as part of the project planning exercise. We followed a structured approach to scheduling and resource management, incorporating techniques covered in this module. Each section below captures a key deliverables we worked on as a group.

**Summary of team discussion**

Our team agreed final solutions for Question 2, after collaborative input on OpenStudio and the small team forum.

Firstly we validated the *Activity-on-Node diagram*. Multiple team members independently arrived at a duration of 22 weeks, no discrepancies there. The chosen version stands out due to its clean design and intuitive colour scheme, which makes task relationships easier to interpret.

Next, we evaluated the *list of activities*. The team opted for a version enhanced with notes and assigned staff fields. These provided better task, clearly displaying number of each developer type.

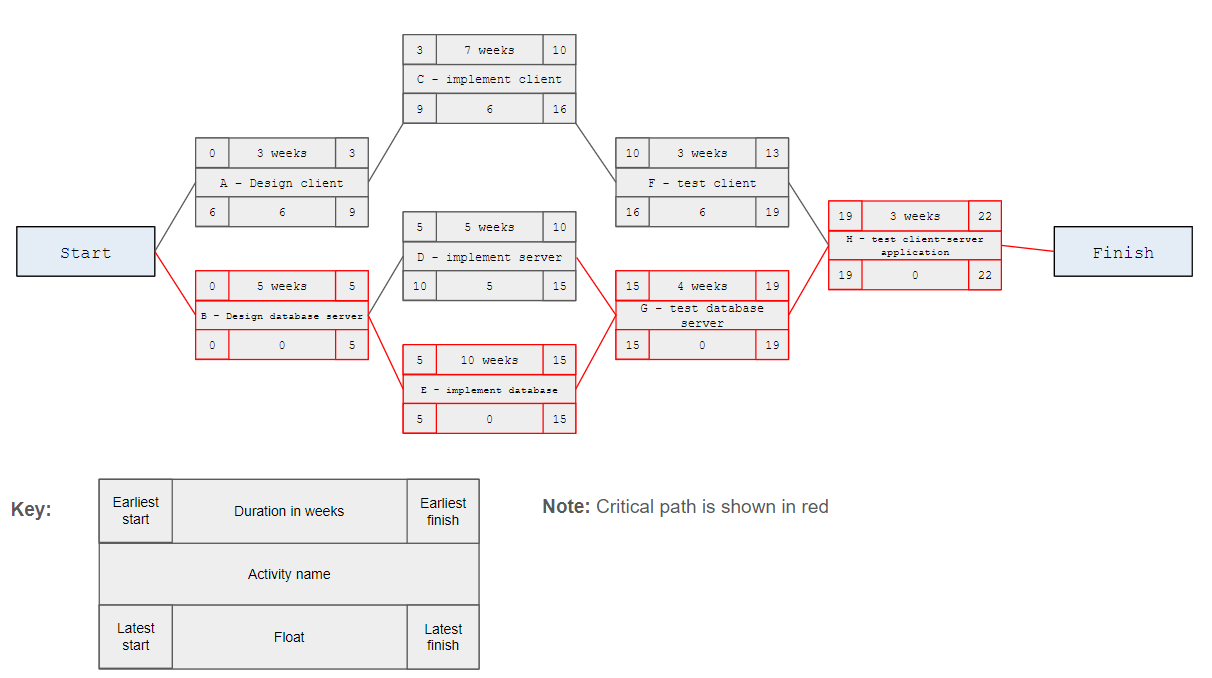
We also selected a specific *activity bar chart* that highlights float values clearly. That will help with further activities like smoothing.

Finally, we chose a *resource table* that simplifies interpretation by using zeroes to indicate unassigned weeks. This reduces visual noise and makes staff allocation clear. It also outlines periods when developers are scheduled to work at the same time. (150 words ) Having difficult time trying to shorten it!

**Activity-on-node network**

Activity On Node

The activity-on-node (AoN) network diagram shows all eight tasks as nodes which are connected based on their relationship and order. Each node displays its earliest start (ES), duration, earliest finish (EF), latest start (LS), float and latest finish (LF) values. There are two paths (A→C→F) and (B→E→G) that both merge at the testing node (H). The critical path, highlighted in red, follows B → E → G → H with zero float, defining the minimum project duration of 22 weeks. The floats on non-critical activities allowed us to find areas for scheduling flexibility and resource smoothing. This AoN network is based on our bar chart and the resource table.



PicX. The diagram shows AoN with critical path in red colour, duration of 22 weeks.

**List of activities and roles**

-table of activities

-assigned roles

-estimated duration of each

Peter’s work

**Initial activity bar chart and resource table**

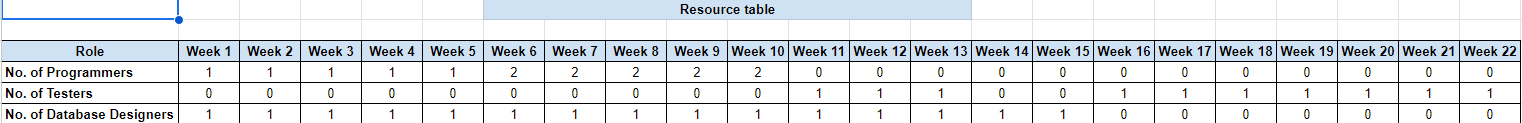
-chart and resource table

-identifying bottlenecks

-weekly number of developers needed, which weeks are busiest

A diagram of a project

AI-generated content may be incorrect.

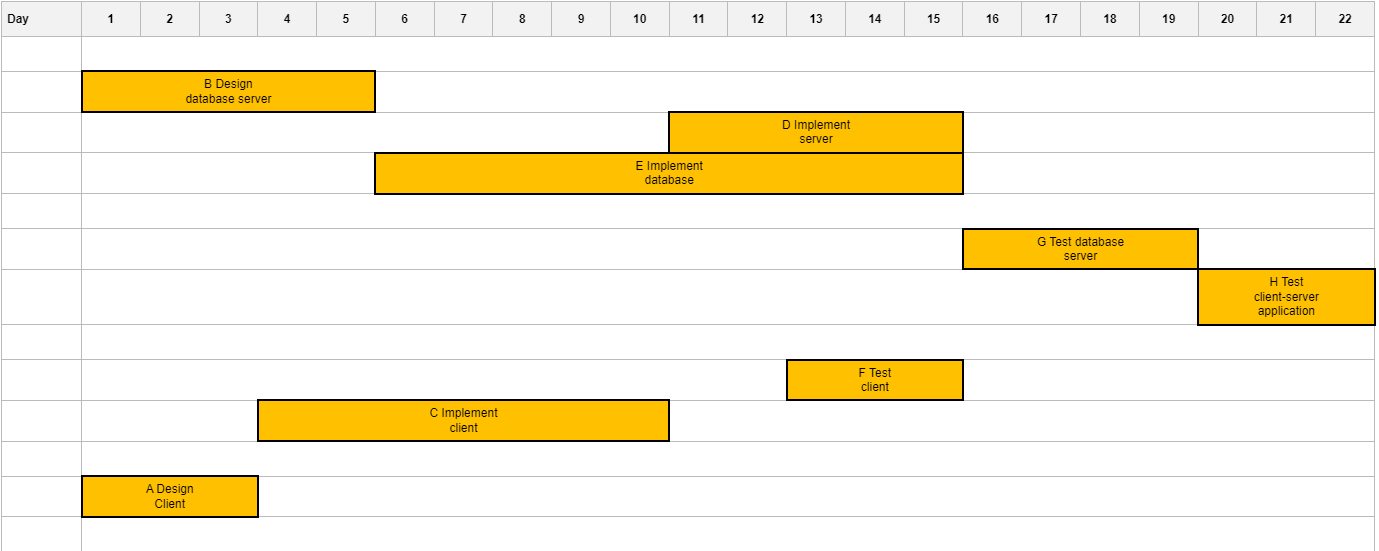


**Smoothing**

As a team, we decided to go with smoothing for this project mainly to reduce costs. Without smoothing, we’d have needed both programmers, and that just wouldn’t be cost-effective. *Darnley* costs twice as much, and since there’s no detailed info about their skills, like whether they’re full-stack, frontend or backend, we assumed *Samira* would be able to handle the work jus as well. From experience, I’ve seen that on smaller projects, having one person often leads to better integration of the different parts, like connecting client-side and server-side code, especially on small projects, and this project I assume is not big.

We also rearranged the tester’s tasks to be closer together in time. This way, we can keep just one tester working for 10 days straight, which can be frustrating and demotivating for them.

The best part is, none of these changes affect the critical path tasks, so the project will still be finished on the same day. We managed this by making use of the floats in the non-critical paths (basically shifting things around where we has some wiggle room).



Pic?. Adjusted activity bar, without the floats visible.



Pic?. Adjusted resource table.

**Final work schedule and cost analysis**

-final schedule reflecting agreed changes and final activity timing

-weekly and cumulative costs

-two line charts

**Conclusion**