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 question 2 solutions have been agreed based on the forum discussions and comments in the OpenStudio.

**Activity-on-node diagram** can be verified for correctness as the critical path has been calculated by multiple team members to be 22 weeks. Further to that the chosen diagram has an easy to read colour palette to quickly digest the information.

The Chosen **List of activities** has been selected by the team for added fields Notes and Assigned staff. This improved clarity and will be useful in future planning as it shows concurrent tasks.

We decided on the best **Activity bar chart** as it clearly shows float values and using visually varied colour patterns it is easy to see tasks based on the groups of staff.

**The Resource table** we have chosen has 0s as values where no staff is utilized, to make reading the chart easier and less ambiguous. This chart also shows where developers work concurrently in specific weeks.

**Activity-on-node network**

-diagram showing all activities

-dependencies between activities

-critical path highlighted and explained

-project duration

KAMIL’s work

**List of activities and roles**

-table of activities

-assigned roles

-estimated duration of each

Peter’s work

**Initial activity bar chart and resource table**

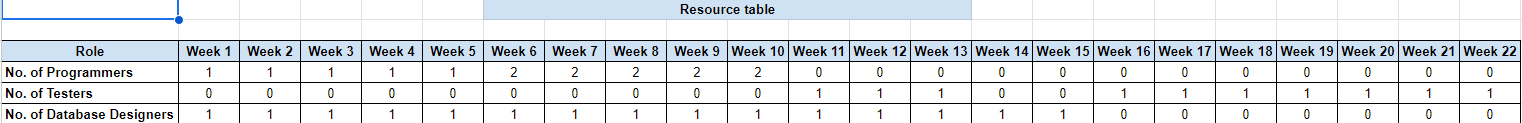
-chart and resource table

-identifying bottlenecks

-weekly number of developers needed, which weeks are busiest

A diagram of a project

AI-generated content may be incorrect.

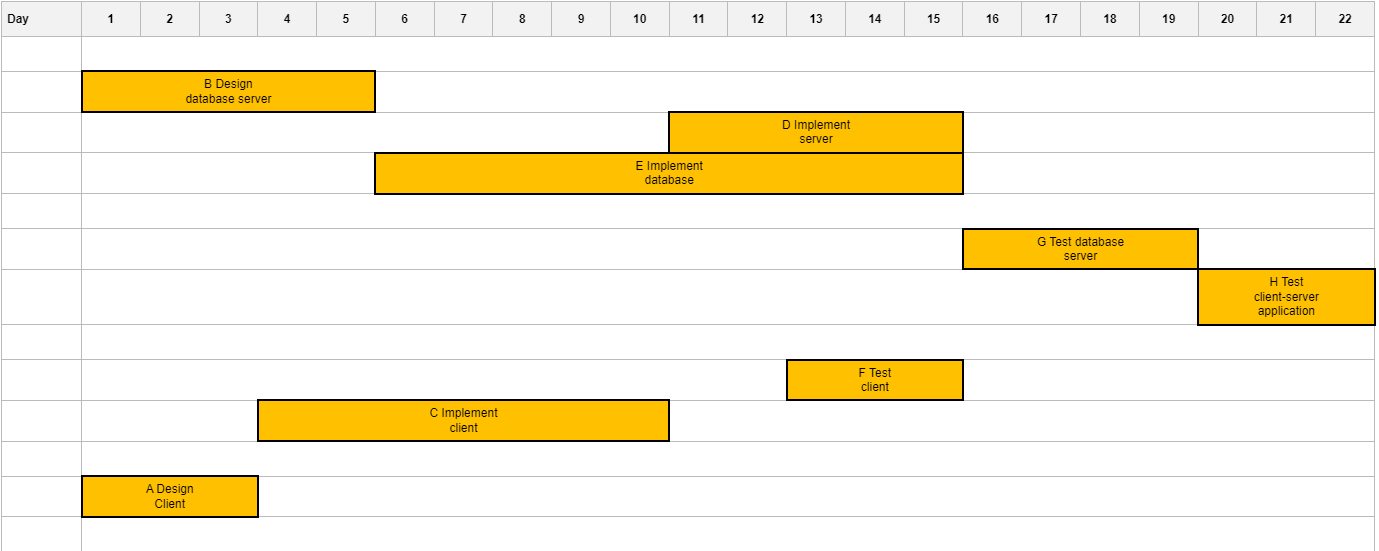


**Smoothing**

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**