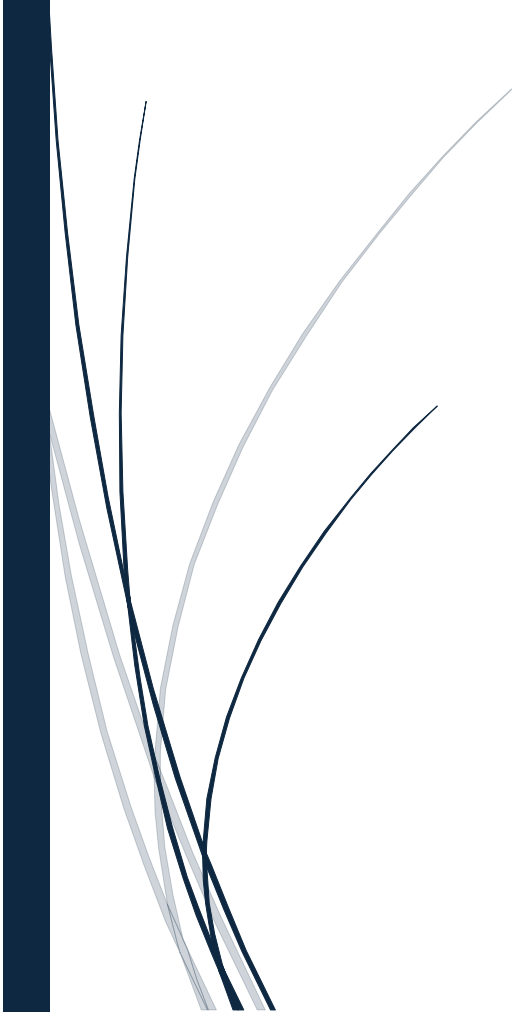
5/2/2025

MS Project Report

Computer System



Oisin Gibson - STUDENT
L00172671

Contents

Introduction

This assignment involved the planning and implementation of a project using Microsoft Project (MSP). The implementation phase included the creation of a Gantt Chart, a customized project calendar, assignment and leveling of resources, identification of the critical path, and progress tracking. The selected project for this assignment was the development of a computer system. The project is scheduled to begin on Monday the 5th of May.

Version 1

The Work Breakdown Structure was split into 7 major categories:

- Planning Phase
- System Design Phase
- Hardware and Software Acquestion
- System Development and Configuration
- Integration and Testing Phase
- Deployment Phase
- Training and Documentation Phase

Each of these phases areas based on modern Software Development Life Cycle (SDLC) that keeps the customers engaged throughput the duration of the project. An overview of the project can be seen in **Figure 1**.

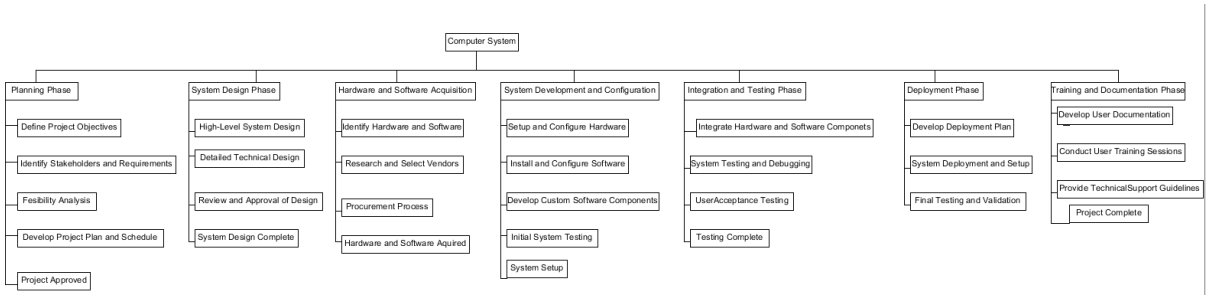


Figure 1: WBS Diagram

As shown in Figure 2 below, the project was created and structured in Microsoft Project (MSP), with all relevant tasks accurately defined. Summary tasks were used to represent the major phases of the project, offering a clear visual hierarchy and aiding in the tracking of each phase's duration.

Additionally, milestones were incorporated to highlight key events and decision points throughout the project. For instance, the milestone "Project Approved" was included to

signify the successful completion of the Planning Phase. While milestones have no duration, they serve as crucial markers of progress and phase transitions.

The project also demonstrates the use of parallel tasks, such as Tasks 2.1 and 2.2, which were scheduled to run concurrently due to their lack of dependency on one another. This parallel scheduling improves efficiency by enabling multiple project components to progress simultaneously.

Furthermore, both lag time and lead time were implemented where appropriate. Lag time is evident in Task 1.2, where a deliberate delay was introduced to allow for stakeholder identification before proceeding. Lead time is applied in Task 3.3, where the task begins one day before its predecessor is fully completed, allowing for slight task overlap and increased scheduling flexibility.

Most tasks in the project are connected using Finish-to-Start (FS) relationships, meaning that a successor task can only begin once its predecessor has been completed. This logical dependency structure helps maintain workflow order and ensures that each task progresses in a controlled, sequential manner.

Parallel Tasks

The concept of parallel tasks refers to activities within a project that can be executed simultaneously, as they are not dependent on one another for completion. In the context of this project, an example of parallel tasks can be found during the System Design Phase. Specifically, the High-Level System Design and the Detailed Technical Design were carried out in parallel. These two tasks were independent of each other, meaning that the initiation or completion of one did not require the other to be finished. Running these tasks concurrently allowed for more efficient use of time and resources, contributing to a shorter overall project timeline without compromising the quality of the design process.

Lag Time

There are instances in project scheduling when a delay is intentionally introduced between two tasks—this is known as lag time. Lag time allows for a necessary waiting period before the next task can begin. In this project, an example of lag time is observed in Task 1.2. To prevent the inefficient use of resources while awaiting stakeholder identification, a buffer of one day was introduced. This -1-day lag ensures that subsequent tasks do not begin prematurely, allowing sufficient time for the required stakeholder information to become available without causing idle time or misallocation of resources.

Lead Time

Lead time was introduced in the project schedule to allow a task to begin before its predecessor is fully completed. This technique helps to optimize workflow and reduce the overall project duration. An example of this can be seen in Task 3.3, where a +1-day lead time was applied. This means that Task 3.3 starts one day before the preceding task has fully finished, enabling both tasks to overlap slightly and thereby enhancing project efficiency without compromising task dependencies.

1		**1 Planning Phase**	**10 days**	**Mon 05/05/2**	**Thu 15/05/25**	
2		1.1 Define Project Objectives	2 days	Mon 05/05/25	Tue 06/05/25	
3		1.2 Identify Stakeholders and	3 days	Mon 05/05/25	Thu 08/05/25	2FS-1 day
4		1.3 Feasibility Ana	2 days	Fri 09/05/25	Tue 13/05/25	3FS+1 day
5		1.4 Develop Project Plan and	3 days	Tue 13/05/25	Thu 15/05/25	4
6		1.5 Project approv	0 days	Thu 15/05/2	Thu 15/05/25	5
7		**2 System Design Phase**	**14 days**	**Thu 15/05/25**	**Mon 02/06/25**	
8		2.1 High-Level System Design	5 days	Thu 15/05/25	Thu 22/05/25	6
9		2.2 Detailed Technical Design	7 days	Wed 21/05/25	Thu 29/05/25	8FS-1 day
10		2.3 Review and Approval of	3 days	Thu 29/05/25	Mon 02/06/25	9
11		2.4 System Design complete	0 days	Mon 02/06/25	Mon 02/06/25	10
12		**3 Hardware and Software Acquisition Phase**	**13 days**	**Tue 03/06/25**	**Wed 18/06/25**	
13		3.1 Identify Hardware and	3 days	Tue 03/06/25	Thu 05/06/25	11
14		3.2 Research and Select Vendors	4 days	Thu 05/06/25	Wed 11/06/25	13
15		3.3 Procurement P	5 days	Thu 12/06/2	Wed 18/06/25	14FS+1 day
16		3.4 Hardware and Software Aquired	0 days	Wed 18/06/25	Wed 18/06/25	15
17		**4 System Development and Configuration Phase**	**18 days**	**Wed 18/06/25**	**Thu 10/07/25**	
18		4.1 Setup and Configure	5 days	Wed 18/06/25	Tue 24/06/25	16
19		4.2 Install and Configure	6 days	Mon 23/06/25	Mon 30/06/25	18FS-2 days
20		4.3 Develop Custom Software	7 days	Mon 30/06/25	Tue 08/07/25	19
21		4.4 Initial System T	3 days	Mon 07/07/2	Thu 10/07/25	20FS-1 day
22		4.5 System Setup	0 days	Thu 10/07/2	Thu 10/07/25	21

23		**5 Integration and Testing Phase**	**13 days**	**Thu 10/07/25**	**Fri 25/07/25**	
24		5.1 Integrate Hardware and Software	4 days	Thu 10/07/25	Tue 15/07/25	22
25		5.2 System Testing and	5 days	Wed 16/07/25	Tue 22/07/25	24FS+1 day
26		5.3 User Acceptance	3 days	Tue 22/07/25	Fri 25/07/25	25
27		5.4 Testing Comple	0 days	Fri 25/07/25	Fri 25/07/25	26
28		**6 Deployment Phase**	**7 days**	**Fri 25/07/25**	**Mon 04/08/25**	
29		6.1 Develop Deployment Plan	2 days	Fri 25/07/25	Tue 29/07/25	27
30		6.2 System Deployment and	4 days	Mon 28/07/25	Thu 31/07/25	29FS-1 day
31		6.3 Final Testing and Validation	2 days	Thu 31/07/25	Mon 04/08/25	30
32		**7 Training and Documentation**	**10 days**	**Thu 31/07/25**	**Wed 13/08/25**	
33		7.1 Develop User Documentation	4 days	Thu 31/07/25	Wed 06/08/25	31FS-2 days
34		7.2 Conduct User Training Sessions	4 days	Wed 06/08/25	Mon 11/08/25	33
35		7.3 Provide Technical Support	2 days	Mon 11/08/25	Wed 13/08/25	34
36		7.4 Project Comple	0 days	Wed 13/08/2	Wed 13/08/25	35

Figure 2: V1

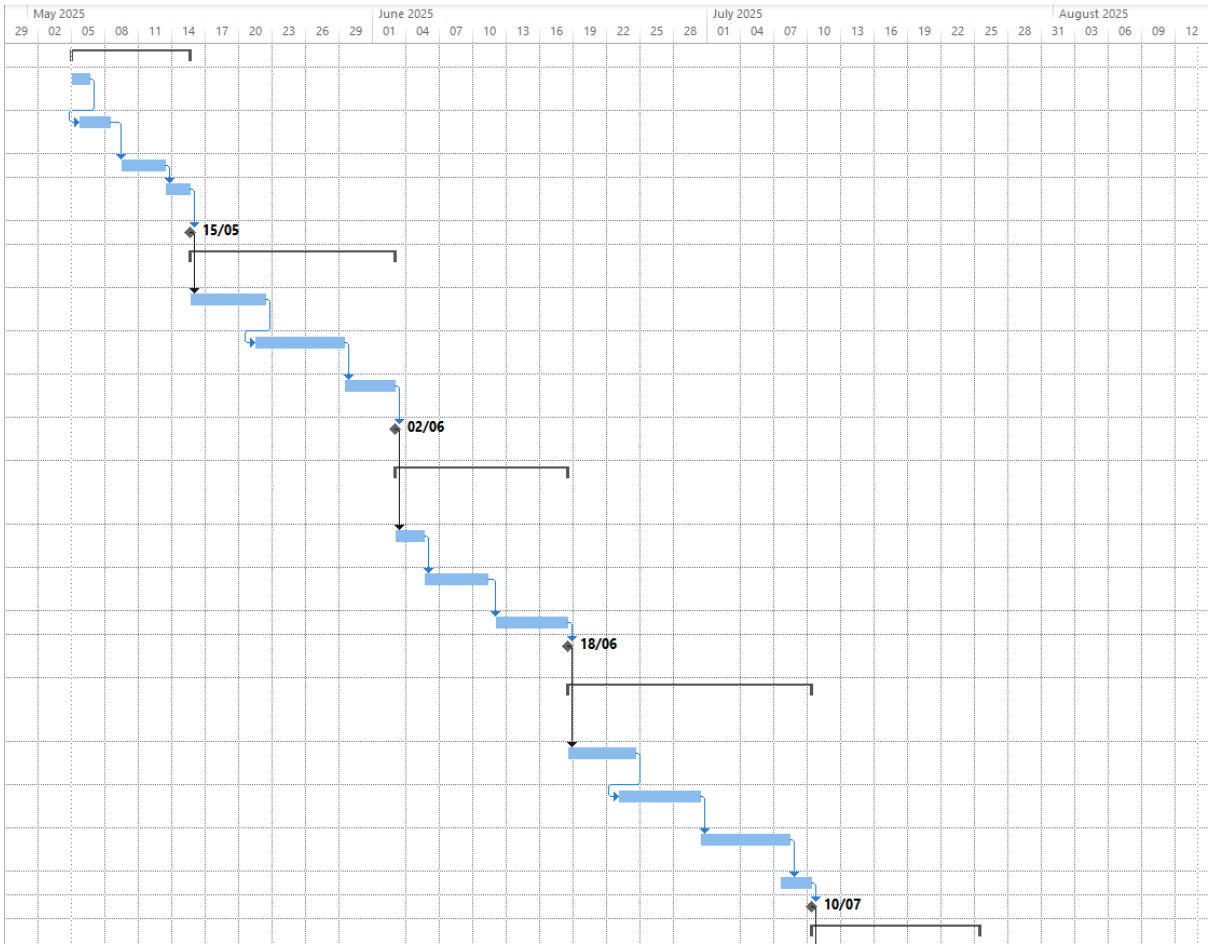


Figure 3: V1 Calander

Version 2

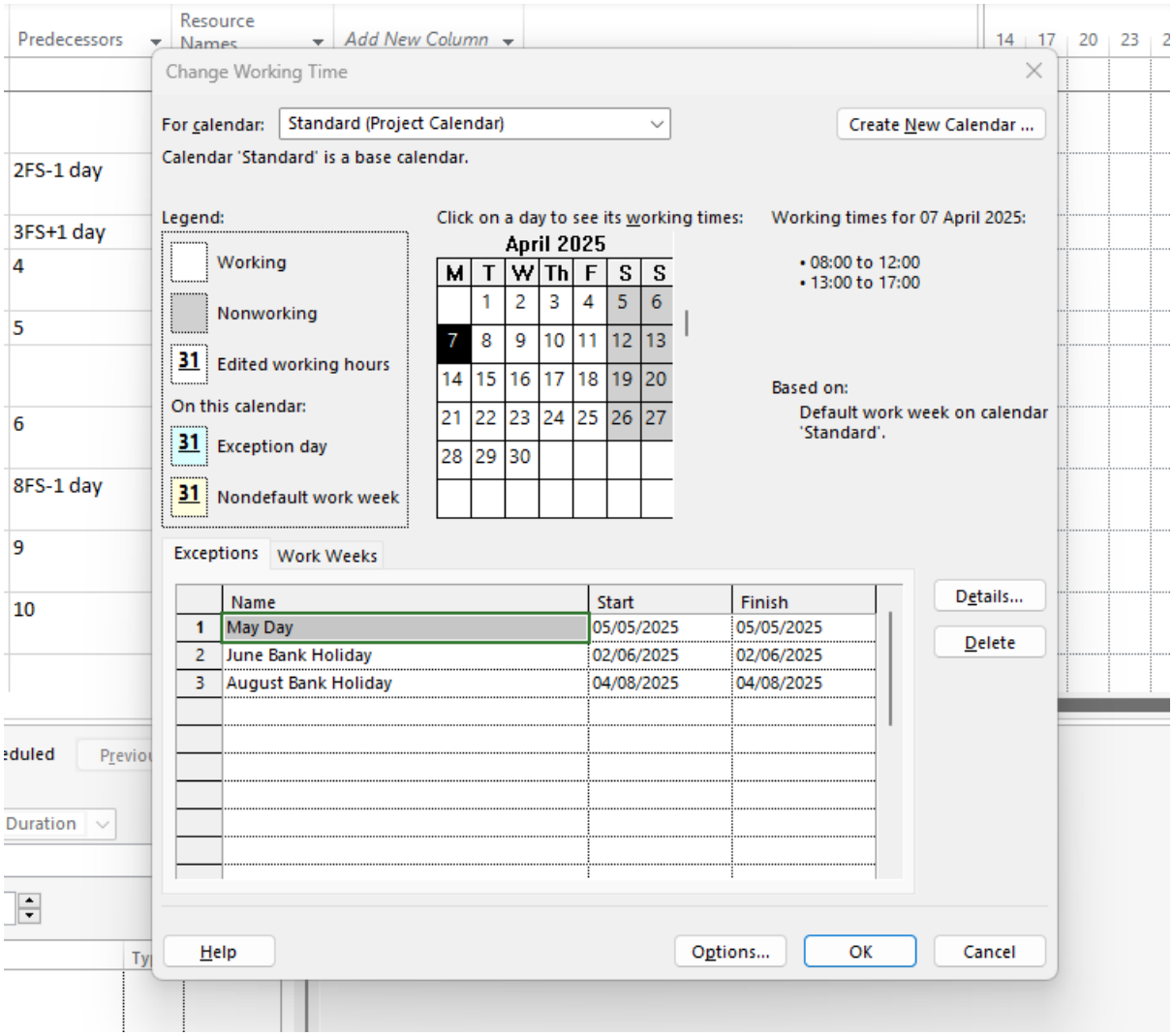


Figure 4: Calander

The project calendar in Microsoft Project (MSP) plays a crucial role in scheduling and execution. It sets working days, hours, weekends, and holidays, directly impacting task timelines, project duration, and resource availability. If non-working days or reduced hours are included, tasks may take longer, potentially extending the entire schedule.

Milestones and deadlines must align with the calendar; if a milestone falls on a non-working day, MSP adjusts the date, which could lead to missed deadlines if not managed carefully. Additionally, progress tracking relies on defined working hours, meaning work done outside of those times may not be accurately reflected in reports.

In this case, adding the project calendar shifted task dates, even requiring a change in the start date due to the May Bank Holiday.

Version 3

	Resource Name	Type	Material	Initials	Group	Max.	Std. Rate	Ovt.	Cost/Use	Accrue	Base	Code	Add New Column
	Project Manager	Work		PM	Management	100%	€65.00/hr	€97.50/hr	€0.00	Prorated	Standard	101-200	
	Business Analyst	Work		BA	Management	100%	€50.00/hr	€75.00/hr	€0.00	Prorated	Standard	101-200	
	Financial Analyst	Work		FA	Management	100%	€55.00/hr	€82.50/hr	€0.00	Prorated	Standard	101-200	
	System Architect	Work		SA	Design Team	100%	€60.00/hr	€90.00/hr	€0.00	Prorated	Standard	201-300	
	Technical Writer	Work		TW	Design Team	100%	€40.00/hr	€60.00/hr	€0.00	Prorated	Standard	201-300	
	Software Developers	Work		SD	Development	100%	€55.00/hr	€82.50/hr	€0.00	Prorated	Standard	301-400	
	Hardware Engine	Work		HE	Development	100%	€50.00/hr	€75.00/hr	€0.00	Prorated	Standard	301-400	
	IT Specialist	Work		ITS	Development	100%	€45.00/hr	€67.50/hr	€0.00	Prorated	Standard	301-400	
	QA Engineers	Work		QAE	Testing Team	100%	€50.00/hr	€75.00/hr	€0.00	Prorated	Standard	401-500	
	Deployment Specialist	Work		DS	Implementation	100%	€48.00/hr	€72.00/hr	€0.00	Prorated	Standard	501-600	
	Trainer	Work		T	Implementation	100%	€42.00/hr	€63.00/hr	€0.00	Prorated	Standard	501-600	
	IT Support Engine	Work		ITSE	Documentation	100%	€42.00/hr	€63.00/hr	€0.00	Prorated	Standard	601-700	
	Procurement Officer	Work		PO	Development	100%	€45.00/hr	€67.50/hr	€0.00	Prorated	Standard	301-400	

Figure 5: Resource Sheet Before Levelling

		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Add New Column
1			**1 Planning Phase**	**9 days**	**Tue 06/05/25**	**Thu 15/05/25**			
2			1.1 Define Project Objectives	1 day	Tue 06/05/25	Tue 06/05/25		Business Analyst	
3			1.2 Identify Stakeholders and	3 days	Tue 06/05/25	Thu 08/05/25	2FS-1 day	Business Analyst	
4			1.3 Feasibility Ana	2 days	Fri 09/05/25	Tue 13/05/25	3FS+1 day	Business Analyst	
5			1.4 Develop Project Plan and	3 days	Tue 13/05/25	Thu 15/05/25	4	Project Manager	
6			1.5 Project approv	0 days	Thu 15/05/25	Thu 15/05/25	5	Project Manager	
7			**2 System Design Pha**	**14 days**	**Thu 15/05/25**	**Wed 04/06/25**			
8			2.1 High-Level System Design	5 days	Thu 15/05/25	Thu 22/05/25	6	System Architect	
9			2.2 Detailed Technical Design	7 days	Wed 21/05/25	Thu 29/05/25	8FS-1 day	System Architect	

Figure 6: Resource Overallocation P1

27			5.4 Testing Comple	0 days	Wed 23/07/2	Wed 23/07/25	26	QA Engineers
28			**6 Deployment Phase**	**7 days**	**Wed 23/07/2**	**Thu 31/07/25**		
29			6.1 Develop Deployment Plan	2 days	Wed 23/07/25	Fri 25/07/25	27	Deployment Specialist
30			6.2 System Deployment and	4 days	Thu 24/07/25	Tue 29/07/25	29FS-1 day	Deployment Specialist
31			6.3 Final Testing and Validation	2 days	Tue 29/07/25	Thu 31/07/25	30	Deployment Specialist

Figure 7: Resource Overallocation P2

In the next phase of the project, additional resources were added. For the purpose of this assignment, some of these resources were intentionally over-allocated. An over-allocated resource refers to a team member who is scheduled to work on more than one task simultaneously creating a scheduling conflict. As shown in Figures 5–7, these over-allocations are indicated by red icons next to the tasks involved.

The following resources have been assigned to this project, each with distinct roles and responsibilities:

- Project Manager: Oversees the entire project, ensuring timelines, budgets, and scope are adhered to, while coordinating efforts across all teams
- Business Analyst: Works to understand business needs and translates them into technical requirements, serving as a bridge between stakeholders and the development team.
- Financial Analyst: Manages the budgeting and financial forecasting for the project, ensuring resources are used efficiently and cost-effectively.
- System Administrator: Responsible for maintaining and configuring the IT infrastructure needed for the project, including servers, networks, and security settings.
- Technical Writer: Creates user manuals, system documentation, and other instructional materials to support system use and training.
- Software Developers: Design, build, and test the software components of the project, working closely with analysts and QA engineers.
- Hardware Engineer: Manages the selection, installation, and configuration of any physical hardware required for the solution.
- IT Specialist: Provides support for both software and hardware components, ensuring the technical aspects of the project are implemented smoothly.
- QA Engineer: Tests the system to identify bugs and ensure the solution meets quality standards and user requirements.
- Deployment Specialist: Handles the transition of the completed system into a live environment, managing installations and rollouts.
- Trainer: Prepares and delivers training to end users and support staff, ensuring they are equipped to use the system effectively.
- IT Support Engineer: Provides ongoing technical support during and after deployment, helping resolve any operational issues.
- Procurement Officer: Manages the acquisition of hardware, software, and services necessary for the project, ensuring procurement aligns with budget and timelines.

Due to the complexity and overlapping nature of some tasks, over-allocation was necessary to simulate real-world project challenges. These instances help illustrate how resource constraints can impact project timelines and require careful management.

Version 4

	Project Manager	Work		PM	Manageme	2	€65.00/hr	€97.50/hr	€0.00	Prorated	Standard	101-200
	Business Analyst	Work		BA	Manageme	1	€50.00/hr	€75.00/hr	€0.00	Prorated	Standard	101-200
	Financial Analyst	Work		FA	Manageme	1	€55.00/hr	€82.50/hr	€0.00	Prorated	Standard	101-200
	System Architect	Work		SA	Design Tea	1	€60.00/hr	€90.00/hr	€0.00	Prorated	Standard	201-300
	Technical Writer	Work		TW	Design Tea	1	€40.00/hr	€60.00/hr	€0.00	Prorated	Standard	201-300
	Software Developers	Work		SD	Developme	1	€55.00/hr	€82.50/hr	€0.00	Prorated	Standard	301-400
	Hardware Engineer	Work		HE	Developme	1	€50.00/hr	€75.00/hr	€0.00	Prorated	Standard	301-400
	IT Specialist	Work		ITS	Developme	1	€45.00/hr	€67.50/hr	€0.00	Prorated	Standard	301-400
	QA Engineers	Work		QAE	Testing Tea	1	€50.00/hr	€75.00/hr	€0.00	Prorated	Standard	401-500
	Deployment Specialist	Work		DS	Implement	1	€48.00/hr	€72.00/hr	€0.00	Prorated	Standard	501-600
	Trainer	Work		T	Implement	1	€42.00/hr	€63.00/hr	€0.00	Prorated	Standard	501-600
	IT Support Engineer	Work		ITSE	Documenta	1	€42.00/hr	€63.00/hr	€0.00	Prorated	Standard	601-700
	Procurement Officer	Work		PO	Developme	1	€45.00/hr	€67.50/hr	€0.00	Prorated	Standard	301-400

Figure 8: Resource Sheet after Levelling

Resource Levelling

Resource levelling is a project management technique used to resolve resource conflicts by adjusting the project schedule. It aims to ensure that no resource is over-allocated meaning no one is assigned more work than they can handle at a given time.

In simple terms, resource levelling delays or reschedules tasks so that resources are used more evenly and realistically, even if it means extending the project timeline. The goal is to balance workload across the team without exceeding resource limits.

1. Adjust Resource Units

Sometimes, a resource may be over-allocated because they're set to work at 100% capacity (unit = 1), but they can take on more. By increasing the Project Manager's unit from 1 to 2, we effectively indicate that they can handle double the workload, either due to part-time availability becoming full-time or increased efficiency. This helps reduce overallocation without changing task assignments.

2. Reassign Tasks to Other Resources

If certain team members are overloaded while others are underutilized, tasks can be reassigned to those with more availability. For instance, if a Software Developer is over-allocated, some of their tasks might be reassigned to an IT Specialist or another developer with similar skills. This balances the workload and ensures all resources are used effectively.

3. Shorten the Critical Path

The critical path represents the longest sequence of dependent tasks that determine the project's total duration. Shortening it can reduce the overall timeline. This is done by targeting tasks on the critical path for optimization—like reducing durations,

overlapping activities, or assigning additional resources. It's a strategic way to improve

		1 Planning Phase	**10 days**	**Tue 06/05/25**	**Fri 16/05/25**		
		1.1 Define Project Objectives	2 days	Tue 06/05/25	Wed 07/05/25		Project Manager
		1.2 Identify Stakeholders and	3 days	Tue 06/05/25	Fri 09/05/25	2FS-1 day	Business Analyst, Project Manager
		1.3 Feasibility Analysis	2 days	Mon 12/05/25	Wed 14/05/25	3FS+1 day	Business Analyst,F
		1.4 Develop Project Plan and Schedule	3 days	Wed 14/05/25	Fri 16/05/25	4	Project Manager
		1.5 Project approved	0 days	Fri 16/05/25	Fri 16/05/25	5	Project Manager
		2 System Design Phase	**14 days**	**Fri 16/05/25**	**Wed 04/06/25**		
		2.1 High-Level System Design	5 days	Fri 16/05/25	Fri 23/05/25	6	System Architect
		2.2 Detailed Technical Design	7 days	Thu 22/05/25	Fri 30/05/25	8FS-1 day	Technical Writer
		2.3 Review and Approval of Design	3 days	Fri 30/05/25	Wed 04/06/25	9	Project Manager, System Architect
		2.4 System Design complete	0 days	Wed 04/06/25	Wed 04/06/25	10	System Architect
		3 Hardware and Software Acquisition	**13 days**	**Thu 05/06/25**	**Fri 20/06/25**		
		3.1 Identify Hardware and Software	3 days	Thu 05/06/25	Mon 09/06/25	11	Hardware Engineer

Figure 9: Resources Levelled

14		3.2 Research and Select Vendors	4 days	Mon 09/06/25	Fri 13/06/25	13	Procurement Officer
15		3.3 Procurement Proce	5 days	Mon 16/06/25	Fri 20/06/25	14FS+1 day	Procurement Offic
16		3.4 Hardware and Software Aquired	0 days	Fri 20/06/25	Fri 20/06/25	15	
17		**4 System Development and Configuration Phase**	**18 days**	**Wed 18/06/25**	**Thu 10/07/25**		
18		4.1 Setup and Configure Hardware	5 days	Fri 20/06/25	Thu 26/06/25	16	Hardware Engineer
19		4.2 Install and Configure Software	6 days	Wed 25/06/25	Wed 02/07/25	18FS-2 days	Software Developers
20		4.3 Develop Custom Software Components	7 days	Wed 02/07/25	Thu 10/07/25	19	Software Developers
21		4.4 Initial System Testir	3 days	Wed 09/07/25	Mon 14/07/25	20FS-1 day	QA Engineers
22		4.5 System Setup	0 days	Mon 14/07/25	Mon 14/07/25	21	IT Specialist
23		**5 Integration and Testing Phase**	**9.5 days**	**Mon 14/07/25**	**Thu 24/07/25**		
24		5.1 Integrate Hardware and Software Components	2 days	Mon 14/07/25	Tue 15/07/25	22	Hardware Engineer Software

Figure 10: Resources Levelled

25		5.2 System Testing and Debugging	5 days	Wed 16/07/25	Wed 23/07/25	24FS+1 day	QA Engineers
26		5.3 User Acceptance Testing	1.5 days	Wed 23/07/25	Thu 24/07/25	25	QA Engineers, Business Analyst
27		5.4 Testing Complete	0 days	Thu 24/07/25	Thu 24/07/25	26	QA Engineers
28		**6 Deployment Phase**	**6 days**	**Thu 24/07/25**	**Thu 31/07/25**		
29		6.1 Develop Deployment Plan	1 day	Thu 24/07/25	Fri 25/07/25	27	Business Analyst, Project Manager
30		6.2 System Deployment and	4 days	Thu 24/07/25	Tue 29/07/25	29FS-1 day	Deployment Specialist
31		6.3 Final Testing and Validation	2 days	Tue 29/07/25	Thu 31/07/25	30	Deployment Specialist
32		**7 Training and Documentation Phase**	**10 days**	**Tue 29/07/25**	**Tue 12/08/25**		
33		7.1 Develop User Documentation	4 days	Tue 29/07/25	Tue 05/08/25	31FS-2 days	Technical Writer
34		7.2 Conduct User Training Sessions	4 days	Tue 05/08/25	Fri 08/08/25	33	Trainer
35		7.3 Provide Technical Support Guidelines	2 days	Fri 08/08/25	Tue 12/08/25	34	IT Support Engineer
36		7.4 Project Complete	0 days	Tue 12/08/25	Tue 12/08/25	35	Project Manager

Figure 11: Resources Levelled

Version 5

Critical Path

The critical path is the longest sequence of dependent tasks in a project that determines the shortest possible project duration. In other words, it's the path of tasks that must be completed on time for the entire project to finish on schedule.

If any task on the critical path is delayed, the whole project will be delayed, unless adjustments are made. Tasks not on the critical path usually have some flexibility, called float or slack, while critical path tasks have zero float.

By identifying and optimizing the critical path, project managers can focus on the tasks that have the most impact on the project's completion date.

To shorten the critical path by approximately one week, a combination of three key project management techniques was applied: crashing, fast tracking, and reducing task durations. Each method was used strategically on tasks that were part of the critical path, ensuring that any changes made would have a direct impact on the overall project timeline.

Allocating Additional Resources

First, crashing was used by adding an additional IT Specialist to the critical task “Develop Custom Software.” This task originally had a long duration and significant influence on the project’s completion date. By increasing the number of resources working on it, the task could be completed more quickly without affecting quality.

Crashing is particularly effective when tasks are effort-driven and can benefit from parallel work done by multiple team members. After the resource was added, the updated Gantt chart and project statistics confirmed a reduction in the task's duration and a positive impact on the overall timeline.

Fast Tracking

Next, fast tracking was applied to optimize task sequencing. In this case, the relationship between "Software Integration" and "System Testing" was modified from a Finish-to-Start (FS) dependency to a Start-to-Start (SS) dependency with a slight lag. This allowed system testing to begin halfway through the integration phase, rather than waiting for it to be completed. By overlapping these two critical tasks, idle time was reduced and the overall schedule compressed. While this method introduces a small amount of risk—such as rework if testing begins too early—it was implemented in a controlled way to maintain project quality.

Reduce Task Duration

Lastly, task duration reduction was applied where feasible. The task "Research and Select Vendors" was originally scheduled for four days but was realistically reduced to two days by improving team focus and eliminating delays. This change was made without additional resources and only where it was certain that the shorter timeframe was achievable. Since this task was on the critical path, the reduction contributed directly to shortening the overall project duration.

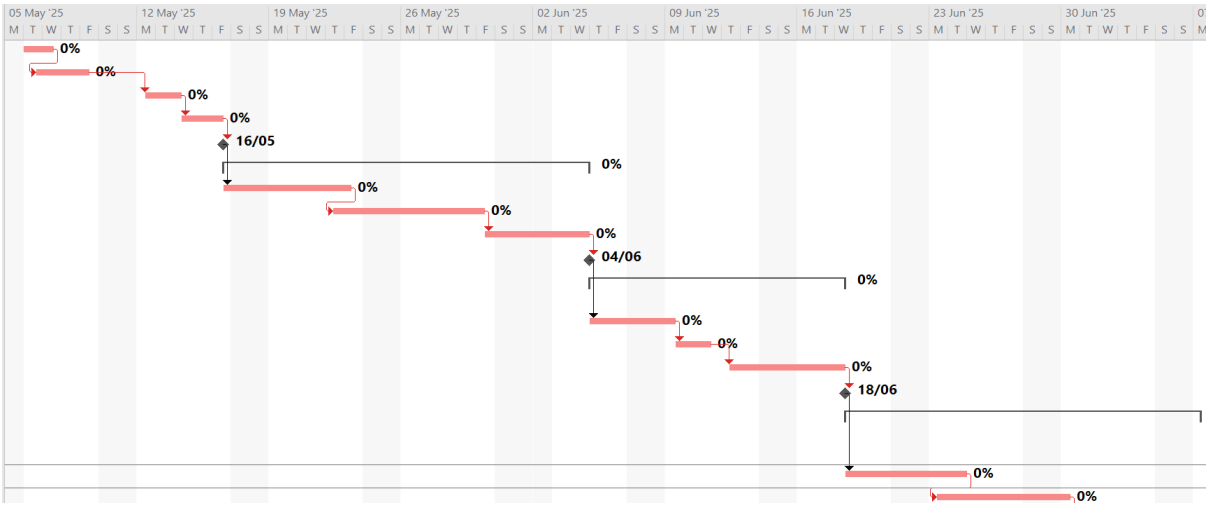
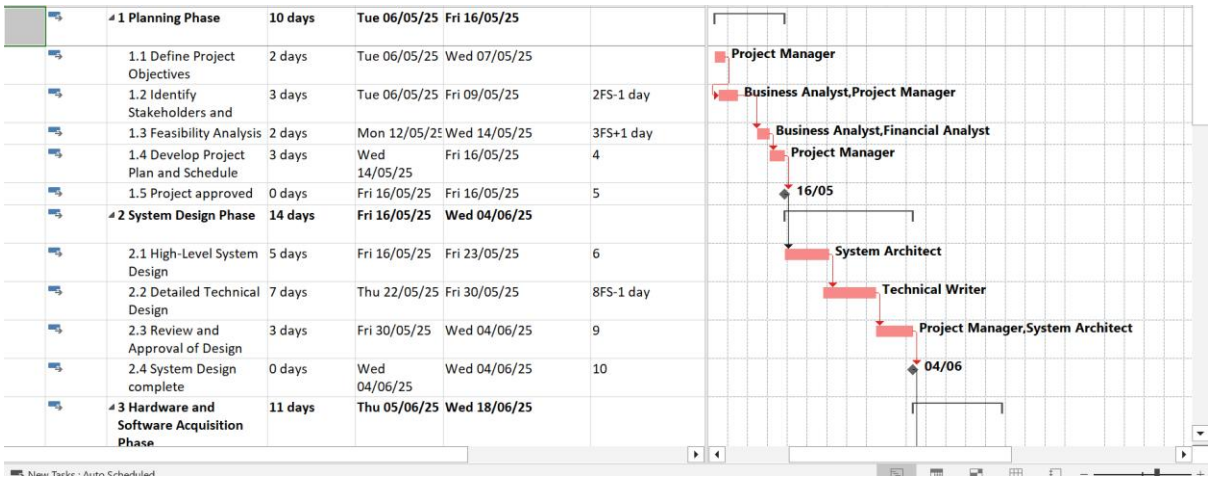


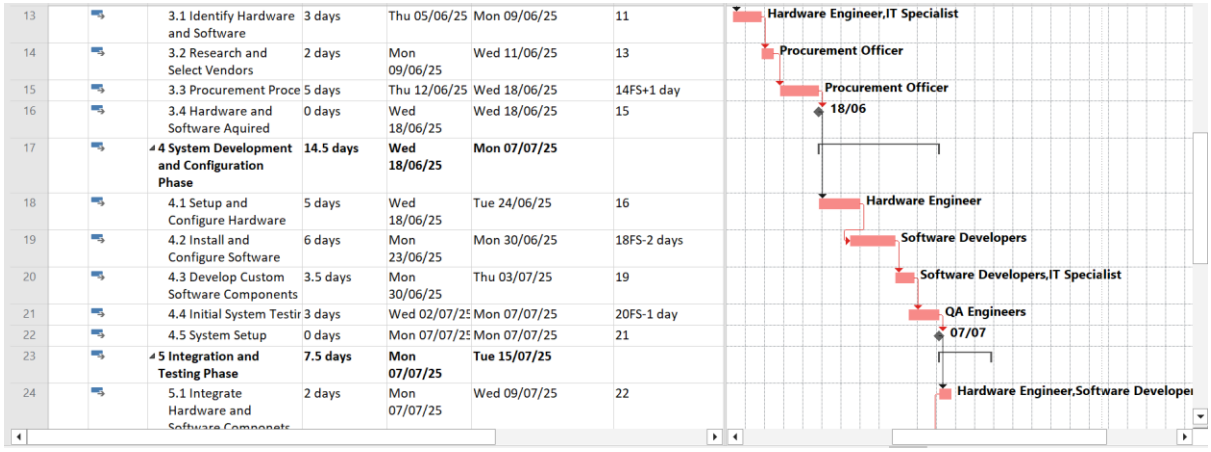
Figure 12: Sample of Tasks on the Critical Path

As seen in Figure 12, all tasks are on the critical path.



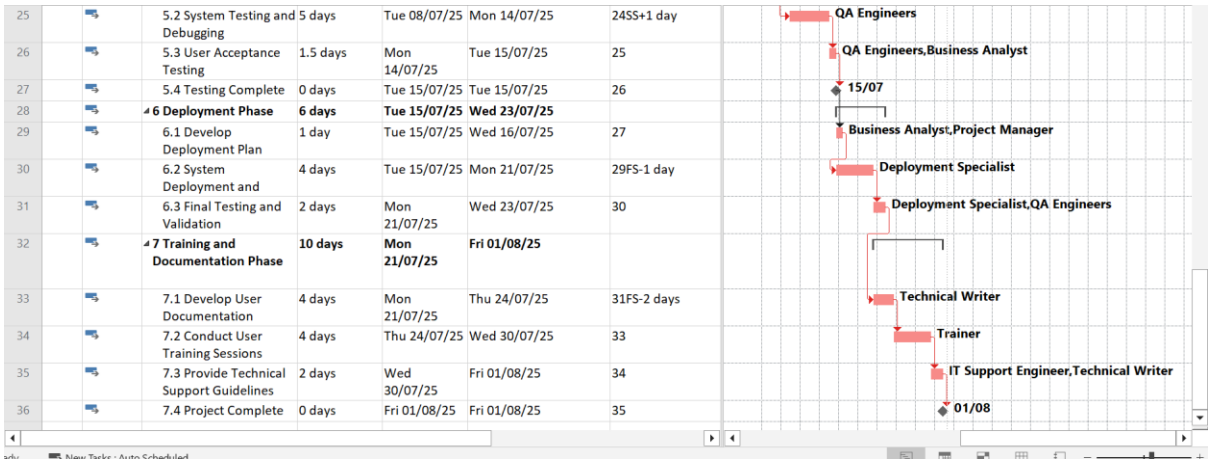
	1 Planning Phase	**10 days**	**Tue 06/05/25**	**Fri 16/05/25**	
	1.1 Define Project Objectives	2 days	Tue 06/05/25	Wed 07/05/25	
	1.2 Identify Stakeholders and	3 days	Tue 06/05/25	Fri 09/05/25	2FS-1 day
	1.3 Feasibility Analysis	2 days	Mon 12/05/25	Wed 14/05/25	3FS+1 day
	1.4 Develop Project Plan and Schedule	3 days	Wed 14/05/25	Fri 16/05/25	4
	1.5 Project approved	0 days	Fri 16/05/25	Fri 16/05/25	5
	2 System Design Phase	**14 days**	**Fri 16/05/25**	**Wed 04/06/25**	
	2.1 High-Level System Design	5 days	Fri 16/05/25	Fri 23/05/25	6
	2.2 Detailed Technical Design	7 days	Thu 22/05/25	Fri 30/05/25	8FS-1 day
	2.3 Review and Approval of Design	3 days	Fri 30/05/25	Wed 04/06/25	9
	2.4 System Design complete	0 days	Wed 04/06/25	Wed 04/06/25	10
	3 Hardware and Software Acquisition Phase	**11 days**	**Thu 05/06/25**	**Wed 18/06/25**	

Figure 13: Critical Path after Crashing



13	3.1 Identify Hardware and Software	3 days	Thu 05/06/25	Mon 09/06/25	11
14	3.2 Research and Select Vendors	2 days	Mon 09/06/25	Wed 11/06/25	13
15	3.3 Procurement Proce	5 days	Thu 12/06/25	Wed 18/06/25	14FS+1 day
16	3.4 Hardware and Software Aquired	0 days	Wed 18/06/25	Wed 18/06/25	15
17	**4 System Development and Configuration Phase**	**14.5 days**	**Wed 18/06/25**	**Mon 07/07/25**	
18	4.1 Setup and Configure Hardware	5 days	Wed 18/06/25	Tue 24/06/25	16
19	4.2 Install and Configure Software	6 days	Mon 23/06/25	Mon 30/06/25	18FS-2 days
20	4.3 Develop Custom Software Components	3.5 days	Mon 30/06/25	Thu 03/07/25	19
21	4.4 Initial System Testir	3 days	Wed 02/07/25	Mon 07/07/25	20FS-1 day
22	4.5 System Setup	0 days	Mon 07/07/25	Mon 07/07/25	21
23	**5 Integration and Testing Phase**	**7.5 days**	**Mon 07/07/25**	**Tue 15/07/25**	
24	5.1 Integrate Hardware and Software Components	2 days	Mon 07/07/25	Wed 09/07/25	22

Figure 14: Critical Path after Crashing



25	5.2 System Testing and Debugging	5 days	Tue 08/07/25	Mon 14/07/25	24SS+1 day
26	5.3 User Acceptance Testing	1.5 days	Mon 14/07/25	Tue 15/07/25	25
27	5.4 Testing Complete	0 days	Tue 15/07/25	Tue 15/07/25	26
28	**6 Deployment Phase**	**6 days**	**Tue 15/07/25**	**Wed 23/07/25**	
29	6.1 Develop Deployment Plan	1 day	Tue 15/07/25	Wed 16/07/25	27
30	6.2 System Deployment and	4 days	Tue 15/07/25	Mon 21/07/25	29FS-1 day
31	6.3 Final Testing and Validation	2 days	Mon 21/07/25	Wed 23/07/25	30
32	**7 Training and Documentation Phase**	**10 days**	**Mon 21/07/25**	**Fri 01/08/25**	
33	7.1 Develop User Documentation	4 days	Mon 21/07/25	Thu 24/07/25	31FS-2 days
34	7.2 Conduct User Training Sessions	4 days	Thu 24/07/25	Wed 30/07/25	33
35	7.3 Provide Technical Support Guidelines	2 days	Wed 30/07/25	Fri 01/08/25	34
36	7.4 Project Complete	0 days	Fri 01/08/25	Fri 01/08/25	35

Figure 15: Critical Path after Crashing

Version 6

Tracking Progress

As with any project, the plan will evolve over time as the project progresses. Once a baseline (the planned timeline) is established, it becomes possible to track the project's progress and adjust accordingly. In the case shown in Figure 17, Task 5.2 was delayed, resulting in a bottleneck. This delay caused the actual timeline (represented by the red line) to surpass the baseline (represented by the grey bar), indicating that the task is now behind schedule. Adjusting the project plan based on these updates allows for better management of timelines and resource allocation moving forward.

1	✓		**1 Planning Phase**	**10 days**	**Tue 06/05/25**	**Fri 16/05/25**		**100%**
2	✓		1.1 Define Project Objectives	2 days	Tue 06/05/25	Wed 07/05/25	Project Manager	100%
3	✓		1.2 Identify Stakeholders and Requirements	3 days	Tue 06/05/25	Fri 09/05/25	2FS-1 day	Business Analyst, Project Manager	100%
4	✓		1.3 Feasibility Analysis	2 days	Mon 12/05/25	Wed 14/05/25	3FS+1 day	Business Analyst, Financial Analyst	100%
5	✓		1.4 Develop Project Plan and Schedule	3 days	Wed 14/05/25	Fri 16/05/25	4	Project Manager	100%
6	✓		1.5 Project approved	0 days	Fri 16/05/25	Fri 16/05/25	5	Project Manager	100%
7	✓		**2 System Design Phase**	**14 days**	**Fri 16/05/25**	**Wed 04/06/25**			**100%**
8	✓		2.1 High-Level System Design	5 days	Fri 16/05/25	Fri 23/05/25	6	System Architect	100%
9	✓		2.2 Detailed Technical	7 days	Thu 22/05/25	Fri 30/05/25	8FS-1 day	Technical Writer	100%

Figure 16: Baseline on Target

22			4.5 System Setup	0 days	Mon 07/07/25	Mon 07/07/25	21	IT Specialist	0%
23			**5 Integration and Testing Phase**	**7.5 days**	**Mon 07/07/25**	**Tue 15/07/25**			**50%**
24	✓		5.1 Integrate Hardware and Software Componets	2 days	Mon 07/07/25	Wed 09/07/25	22	Hardware Engineer Software Developers	100%
25			5.2 System Testing and Debugging	5 days	Tue 08/07/25	Mon 14/07/25	24SS+1 day	QA Engineers	45%
26			5.3 User Acceptance Testing	1.5 days	Mon 14/07/25	Tue 15/07/25	25	QA Engineers, Business Analyst	0%
27			5.4 Testing Complete	0 days	Tue 15/07/25	Tue 15/07/25	26	QA Engineers	0%
28			**6 Deployment Phase**	**6 days**	**Tue 15/07/25**	**Wed 23/07/25**			**40%**
29	✓		6.1 Develop Deployment Plan	1 day	Tue 15/07/25	Wed 16/07/25	27	Business Analyst, Project Manager	100%
30			6.2 System Deployment and	4 days	Tue 15/07/25	Mon 21/07/25	29FS-1 day	Deployment Specialist	45%

Figure 17: Baseline being of Target

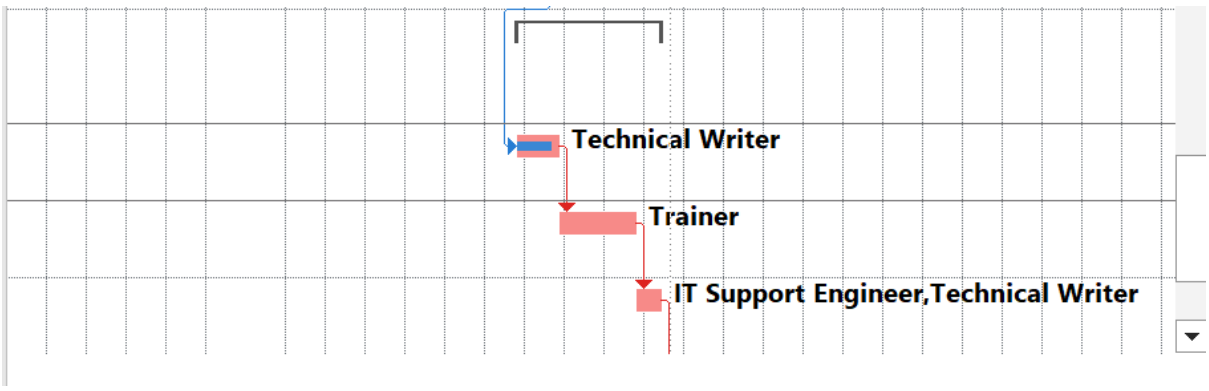


Figure 18: Task Overrun

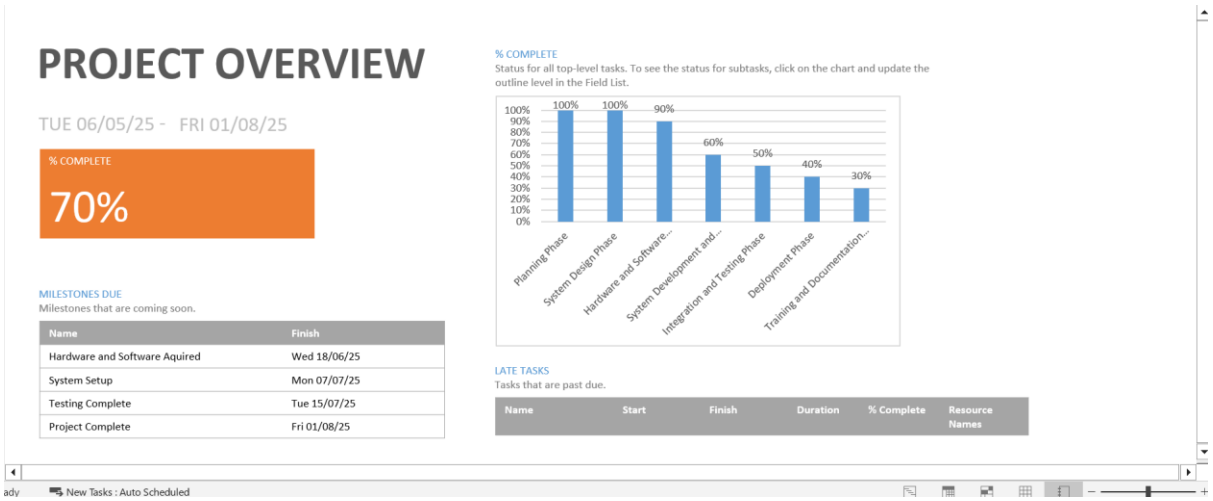


Figure 19: Final Project Overview