

Smart Select

Work Breakdown Structure and Project Schedule

Raheim Burkett 2210198 (Team Member 1)

Garrett Grant 2205586 (Team Member 2)

T'Yondre Leslie 2210176 (Team Member 3)

Iyana Taylor 2209566 (Team Member 4)

School of Computing and Information Technology (SCIT),

Faculty of Engineering and Computing (FENC),

University of Technology, Jamaica (UTECH)

Information Technology Project Management (UN2)

Mrs. Sophia McNamarah

October 18, 2024

Table of Contents

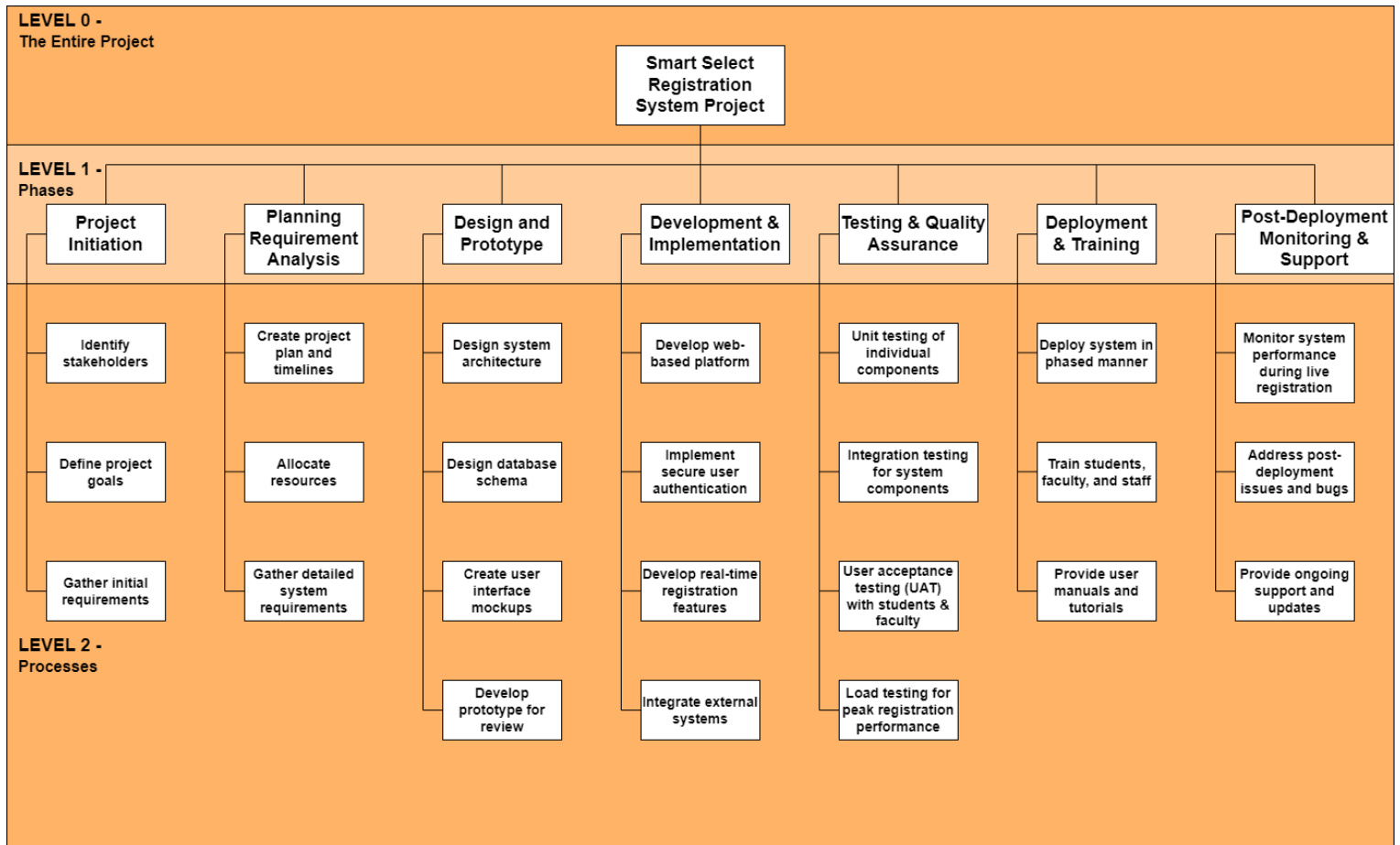
Scope Statement	3
Work Breakdown Structure	3
WBS Dictionary	3
Activity List	6
Project Schedule	7
Critical Path	13
Two (2) activities on the critical path	13
Importance of identifying activities on the critical path	13
Report related to Human Resource Management	14
Team Contributions	17

Scope Statement

This project seeks to create a modern web-based system that will improve the student registration process at UTECH. The scope includes improving load management and performance, secure authentication, and a staggered registration strategy based on faculty schedules.

Work Breakdown Structure

Smart Select - 3 level *Process* Work Breakdown Structure, in graphical format:



WBS Dictionary

Level	Code	Name	Description of Work	Deliverable(s)
Level 0	0.0	Smart Select Registration System Project	The entire project involving the development and implementation of the new registration system for UTech	Fully developed, tested, and deployed registration system

Level 1	1.0	Project Initiation	Initial project setup involving stakeholder identification, goal definition, and requirements gathering	Identified stakeholders and clear project goals
Level 2	1.1	Identify stakeholders	Identify key project stakeholders and define their roles	List of key stakeholders
Level 2	1.2	Define project goals	Establish project goals and objectives	Documented project goals
Level 2	1.3	Gather initial requirements	Gather initial system requirements from stakeholders	Initial requirement list
Level 1	2.0	Planning & Requirement Analysis	Detailed project planning, resource allocation, and requirement analysis	Project plan and detailed requirements document
Level 2	2.1	Create project plan and timelines	Develop the project schedule and allocate resources	Project plan with detailed timeline
Level 2	2.2	Allocate resources	Allocate necessary human and technical resources for the project	Resource allocation chart
Level 2	2.3	Gather detailed system requirements	Analyze technical and user needs for the system	Software Requirement Specification (SRS) Document
Level 1	3.0	Design & Prototype	Create system design and develop a prototype	System architecture and prototype
Level 2	3.1	Design system architecture	Design the high-level structure of the system, including the client, server, and database components	System architecture document
Level 2	3.2	Design database schema	Design the organization and relationships within the database	Database schema
Level 2	3.3	Create user interface mockups	Create wireframes/mockups for the user interface	User interface wireframes
Level 2	3.4	Develop prototype for review	Build a functional prototype based on the design	Prototype ready for stakeholder review
Level 1	4.0	Development & Implementation	Develop the system iteratively and implement key functionalities	Fully functional web-based system

Level 2	4.1	Develop web-based platform	Develop the main components of the system to support web-based registration	First version of the web-based platform
Level 2	4.2	Implement secure user authentication	Implement user authentication mechanisms, including role-based access control	Secure authentication system
Level 2	4.3	Develop real-time registration features	Implement the system's ability to handle live registration and updates	Real-time registration module
Level 2	4.4	Integrate external systems	Connect the new system with the university's existing systems like Student Information System, module catalog, and payment systems	Integrated external systems
Level 1	5.0	Testing & Quality Assurance	Perform testing to ensure system functionality and performance	Tested and quality-assured system
Level 2	5.1	Unit testing of individual components	Test each component of the system separately	Unit test results
Level 2	5.2	Integration testing for system components	Test how different components of the system work together	Integration test results
Level 2	5.3	User Acceptance Testing (UAT)	Allow end users (students and faculty) to test the system for usability and functionality	UAT results
Level 2	5.4	Load testing for peak registration performance	Simulate high traffic registration to ensure system stability	Load testing results
Level 1	6.0	Deployment & Training	Deploy the system and train users	Deployed system and trained users
Level 2	6.1	Deploy system in phased manner	Deploy the system in stages to ensure minimal disruptions	Fully deployed system
Level 2	6.2	Train students, faculty, and staff	Provide training to system users, including tutorials and workshops	Trained system users
Level 2	6.3	Provide user manuals and tutorials	Create and distribute user manuals and video tutorials	User manuals and tutorials
Level 1	7.0	Post-Deployment	Monitor system performance and provide ongoing support	Post-deployment report and support system

		Monitoring & Support		
Level 2	7.1	Monitor system performance	Ensure the system is functioning as expected during live registration	Performance report
Level 2	7.2	Address post-deployment issues and bugs	Fix any bugs and handle issues as they arise during the post-deployment period	Post-deployment bug fixes
Level 2	7.3	Provide ongoing support and updates	Continue monitoring and providing updates if needed	Final support report

Activity List

Activity Code	Activity Name	Predecessor(s)	Duration (in days)
1.1	Identify Stakeholders	None	3
1.2	Define project goals	1.1	3
1.3	Gather initial requirements	1.2	4
2.1	Create project plan and timelines	1.3	4
2.2	Allocate Resources	2.1	3
2.3	Gather detailed system requirements	2.2	7
3.1	Design system architecture	2.3	8
3.2	Design database schema	3.1	4
3.3	Create user interface mockups	3.2	5
3.4	Develop prototype for review	3.3	5
4.1	Develop web-based platform	3.4	30
4.2	Implement secure user authentication	4.1	7
4.3	Develop real-time registration features	4.2	17
4.4	Integrate external systems	4.3	7
5.1	Unit testing of individual components	4.4	6
5.2	Integration testing for system components	5.1	5
5.3	User Acceptance Testing (UAT)	5.2	5
5.4	Load testing for peak registration performance	5.3	1
6.1	Deploy system in phased manner	5.4	9
6.2	Train students, faculty, and staff	6.1	10
6.3	Provide user manuals and tutorials	6.1	10
7.1	Monitor system performance	6.2; 6.3	5

Project Schedule

1. Start-to-Start (SS):

- **Activity:** "2.2 Allocate resources" has an SS (Start-to-Start) dependency with "2.1 Create project plan and timelines."
 - **Reason:** Allocating resources can begin as soon as the project planning has started, rather than waiting for the entire planning phase to finish.

2. Finish-to-Finish (FF):

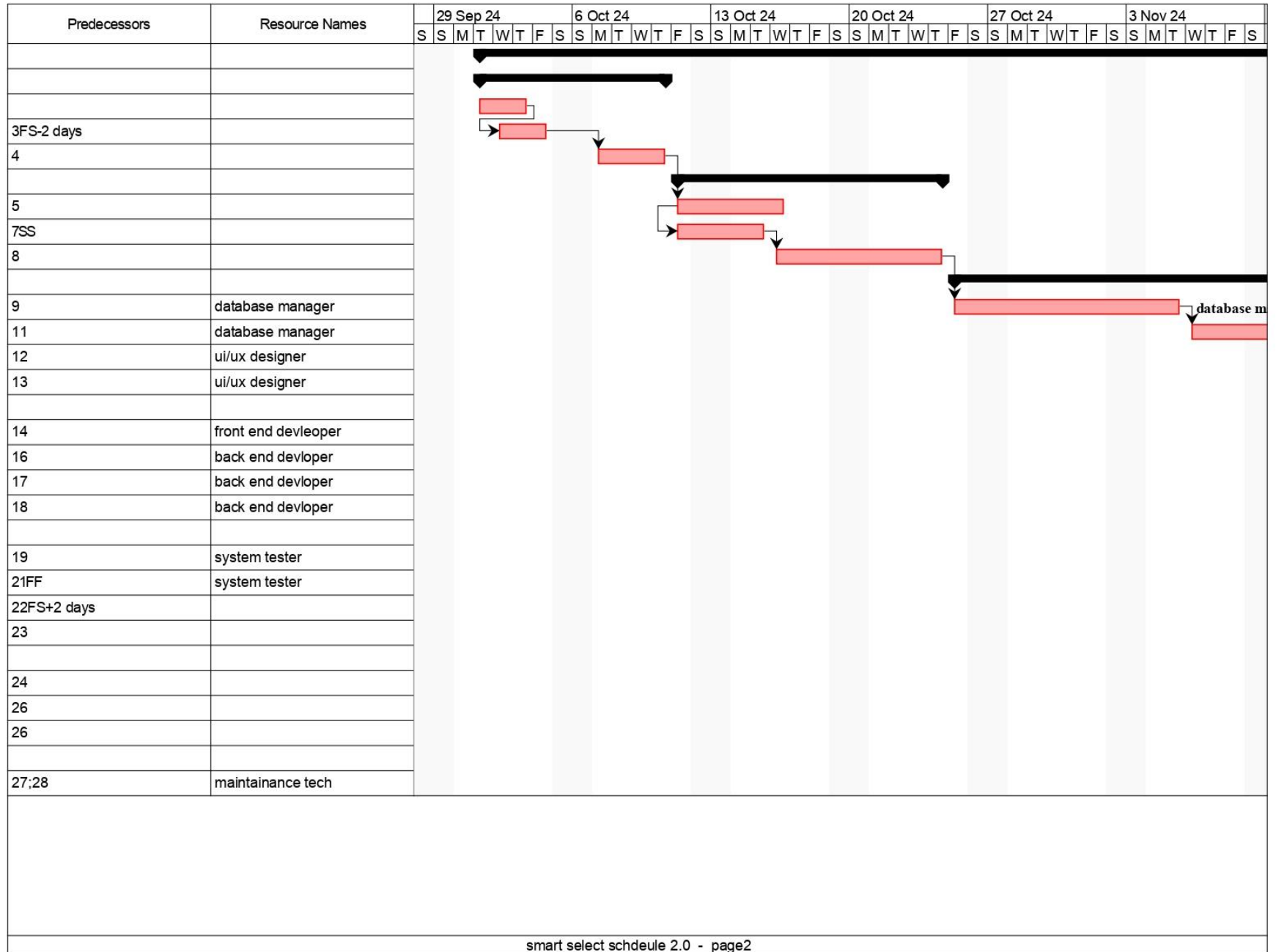
- **Activity:** "5.2 Integration testing" has an FF (Finish-to-Finish) dependency with "5.1 Unit testing of components."
 - **Reason:** Both testing activities can finish around the same time, as integration testing relies on most unit testing to be completed, but some parts can run in parallel.

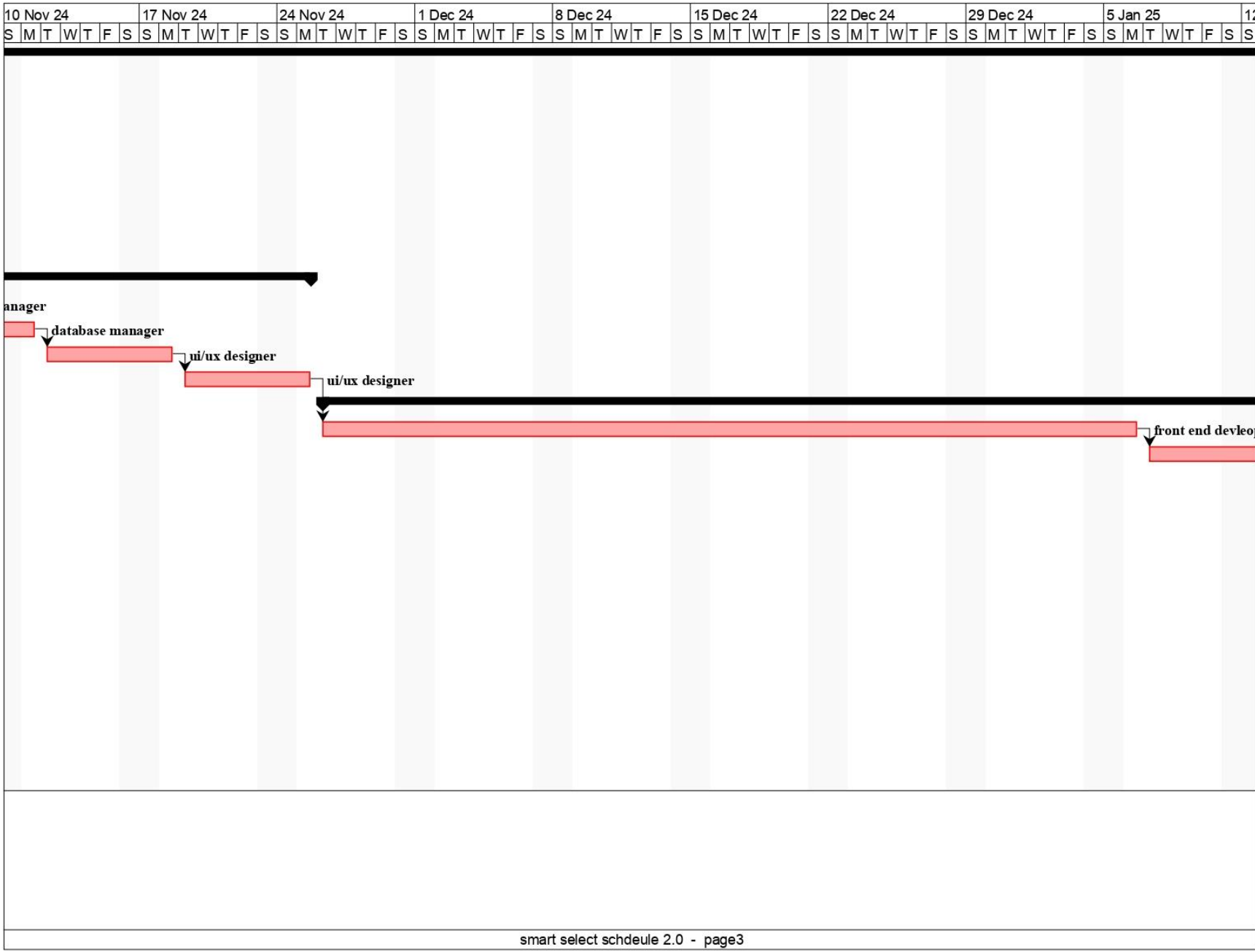
3. **Activity** "1.2 Define project goals" has a **Lead** of 2 days.

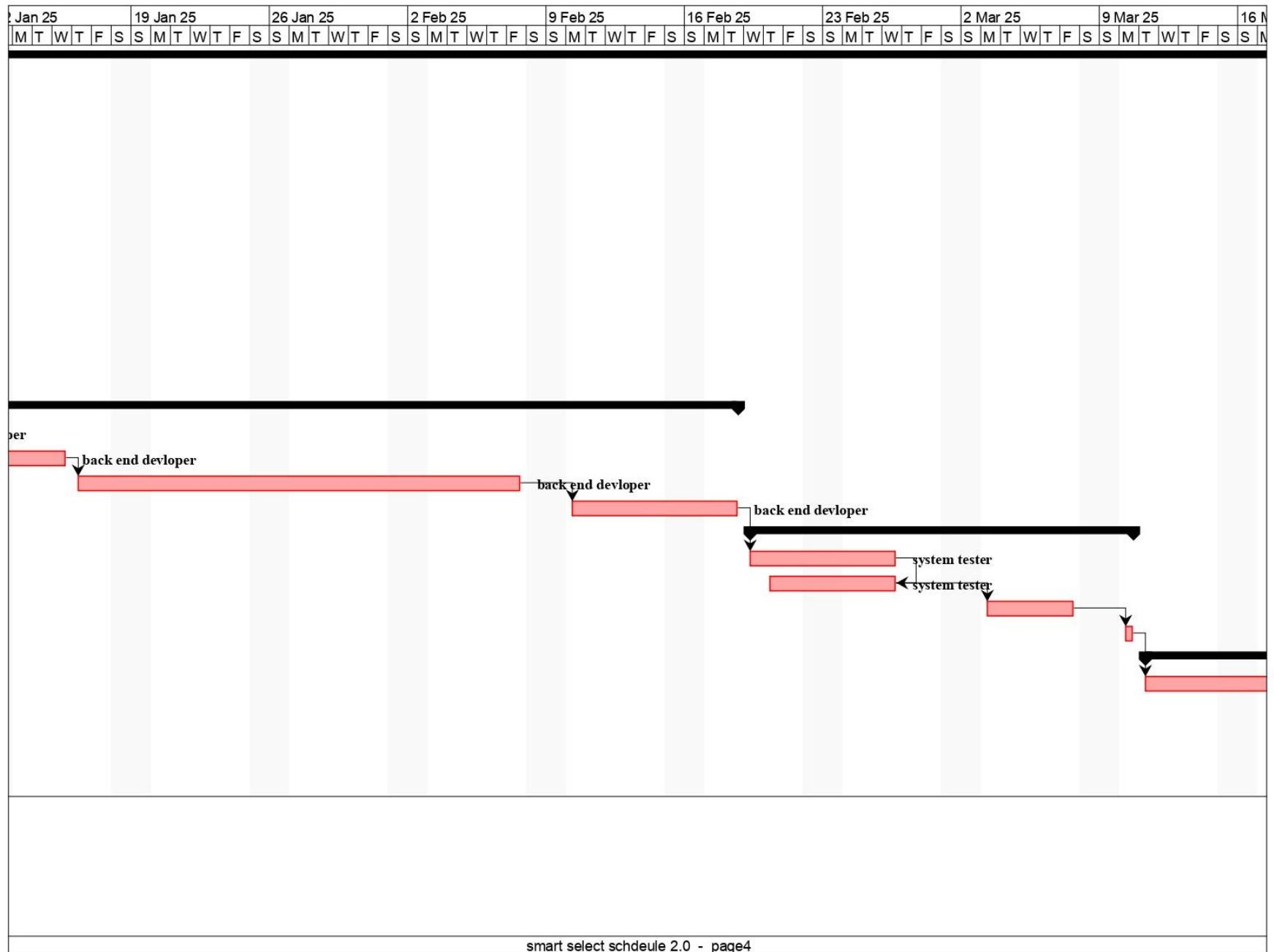
4. **Activity** "5.3 User Acceptance Testing (UAT)" has a **Lag** of 2 days.

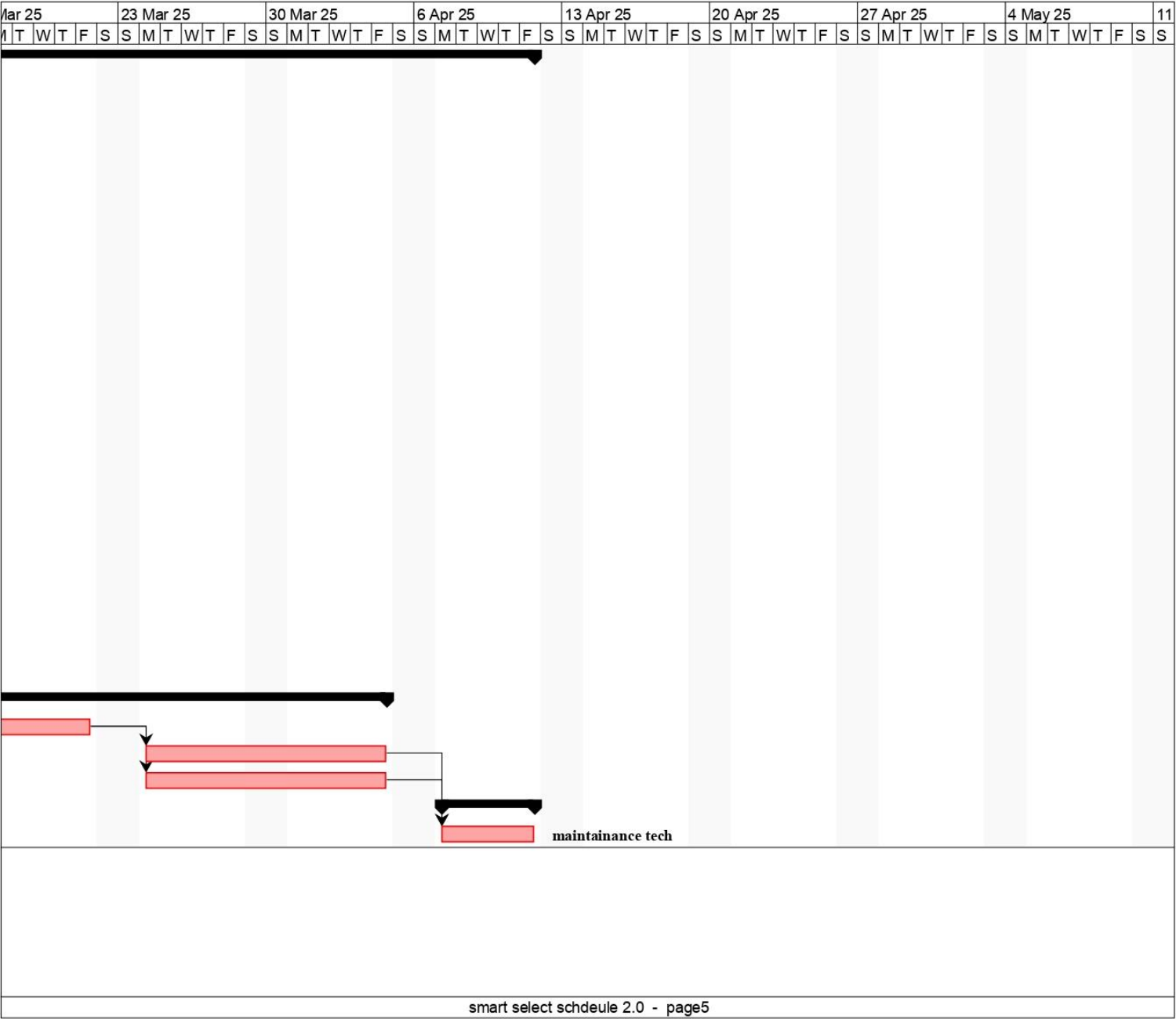
The following schedule was created using Project Libre, *see below*:

		Name	Duration	Start	Finish	Total Slack
1		0.0 Smart Select Registration System Project	139 days	10/1/24 8:00 AM	4/11/25 5:00 PM	0 days
2		1.0 Project Initiation	8 days	10/1/24 8:00 AM	10/10/24 5:00 PM	0 days
3		1.1 Identify stakeholders	3 days	10/1/24 8:00 AM	10/3/24 5:00 PM	0 days
4		1.2 Define project goals	3 days	10/2/24 8:00 AM	10/4/24 5:00 PM	0 days
5		1.3 Gather initial requirements	4 days	10/7/24 8:00 AM	10/10/24 5:00 PM	0 days
6		2.0 Planning & Requirement Analysis	10 days	10/11/24 8:00 AM	10/24/24 5:00 PM	0 days
7		2.1 Create project plan and timelines	4 days	10/11/24 8:00 AM	10/16/24 5:00 PM	0 days
8		2.2 Allocate resources	3 days	10/11/24 8:00 AM	10/15/24 5:00 PM	0 days
9		2.3 Gather detailed system requirements	7 days	10/16/24 8:00 AM	10/24/24 5:00 PM	0 days
10		3.0 Design & Prototype	22 days	10/25/24 8:00 AM	11/25/24 5:00 PM	0 days
11		3.1 Design system architecture	8 days	10/25/24 8:00 AM	11/5/24 5:00 PM	0 days
12		3.2 Design database schema	4 days	11/6/24 8:00 AM	11/11/24 5:00 PM	0 days
13		3.3 Create user interface mockups	5 days	11/12/24 8:00 AM	11/18/24 5:00 PM	0 days
14		3.4 Develop prototype for review	5 days	11/19/24 8:00 AM	11/25/24 5:00 PM	0 days
15		4.0 Development & Implementation	61 days	11/26/24 8:00 AM	2/18/25 5:00 PM	0 days
16		4.1 Develop web-based platform	30 days	11/26/24 8:00 AM	1/6/25 5:00 PM	0 days
17		4.2 Implement secure user authentication	7 days	1/7/25 8:00 AM	1/15/25 5:00 PM	0 days
18		4.3 Develop real-time registration features	17 days	1/16/25 8:00 AM	2/7/25 5:00 PM	0 days
19		4.4 Integrate external systems	7 days	2/10/25 8:00 AM	2/18/25 5:00 PM	0 days
20		5.0 Testing & Quality Assurance	14 days	2/19/25 8:00 AM	3/10/25 5:00 PM	0 days
21		5.1 Unit testing of individual components	6 days	2/19/25 8:00 AM	2/26/25 5:00 PM	0 days
22		5.2 Integration testing for system components	5 days	2/20/25 8:00 AM	2/26/25 5:00 PM	0 days
23		5.3 User Acceptance Testing (UAT)	5 days	3/3/25 8:00 AM	3/7/25 5:00 PM	0 days
24		5.4 Load testing for peak registration performance	1 day	3/10/25 8:00 AM	3/10/25 5:00 PM	0 days
25		6.0 Deployment & Training	19 days	3/11/25 8:00 AM	4/4/25 5:00 PM	0 days
26		6.1 Deploy system in phased manner	9 days	3/11/25 8:00 AM	3/21/25 5:00 PM	0 days
27		6.2 Train students, faculty, and staff	10 days	3/24/25 8:00 AM	4/4/25 5:00 PM	0 days
28		6.3 Provide user manuals and tutorials	10 days	3/24/25 8:00 AM	4/4/25 5:00 PM	0 days
29		7.0 Post-Deployment Monitoring & Support	5 days	4/7/25 8:00 AM	4/11/25 5:00 PM	0 days
30		7.1 Monitor system performance	5 days	4/7/25 8:00 AM	4/11/25 5:00 PM	0 days









Critical Path

Two (2) activities on the critical path

For the Smart Select Registration System Project, two activities on the critical path are:

1. Activity **2.3 Gather detailed system requirement**
2. Activity **4.1 Web-based platform development**

2.3 Collect detailed system requirements: This stage will involve stipulating what the system needs to realize, including user needs, technical specifications, and functionality. Delays in this activity will affect subsequent stages of the project, such as the system design and development stages, since they will not have the required information to proceed with the work, thus creating a bottleneck.

4.1 Web-based platform development: This is where the actual development of the system, based on requirements and designs into a functional platform, takes place. Any slippages in this phase will have a trickledown effect on testing, deployment, and training, thus pretty much impacting the project timeline.

Importance of identifying activities on the critical path

In project management, the **Critical Path** means the longest set of dependent activities and determines the entire duration of the project. If a task on the critical path is delayed, then the whole project will be late. Therefore, the identification of such activities and their close monitoring becomes very crucial for the project manager.

Why is it Important for Project Managers to Identify Critical Path Activities?

1. **Resource Prioritization:** Knowing which tasks are on the critical path allows the project manager to prioritize resources time, money, and labor to ensure these tasks are completed on time. For instance, in this case, the project manager can make sure key stakeholders are readily available to approve the requirements gathered in 2.3 and allocate adequate developer hours for 4.1.

2. **Risk Management:** Activities included in the critical path, such as Gathering System Requirements and Developing the Platform, are highly risky because any delay will prolong the

entire project duration. The project manager can plan strategies for mitigation of risk that includes early identification of probable roadblocks and contingency planning.

3. Project Monitoring and Control: It also allows the project manager, through its focus on the critical path, to better track the completion of key activities to meet a schedule. The project manager should also be checking on a regular basis, the status of 4.1, for instance, to rapidly find problems well in advance that might delay downstream activities such as testing or deployment.

Identification of such critical activities ensures that delays are at a minimum and that the project will be delivered on time and scope.

Report related to Human Resource Management

The following report was generated using Project Libre, *see below*:

Who Does What

Resource ID		Resource					
1		ui/ux designer					
Task ID	Task	Work	Assignment Units	Assignment	Start	Finish	
13	3.3 Create user interface	40 hours	100%	0 days	11/12/24 8:00 AM	11/18/24 5:00 PM	
14	3.4 Develop prototype for review	40 hours	100%	0 days	11/19/24 8:00 AM	11/25/24 5:00 PM	
		80 hours					
2		front end devleoper					
Task ID	Task	Work	Assignment Units	Assignment	Start	Finish	
16	4.1 Develop web-based platform	240 hours	100%	0 days	11/26/24 8:00 AM	1/6/25 5:00 PM	
		240 hours					
3		database manager					
Task ID	Task	Work	Assignment Units	Assignment	Start	Finish	
11	3.1 Design system architecture	64 hours	100%	0 days	10/25/24 8:00 AM	11/5/24 5:00 PM	
12	3.2 Design database schema	32 hours	100%	0 days	11/6/24 8:00 AM	11/11/24 5:00 PM	
		96 hours					
4		back end developer					
Task ID	Task	Work	Assignment Units	Assignment	Start	Finish	
18	4.3 Develop real-time registration	136 hours	100%	0 days	1/16/25 8:00 AM	2/7/25 5:00 PM	
19	4.4 Integrate external systems	56 hours	100%	0 days	2/10/25 8:00 AM	2/18/25 5:00 PM	
17	4.2 Implement secure user	56 hours	100%	0 days	1/7/25 8:00 AM	1/15/25 5:00 PM	
		248 hours					

Resource ID		Resource				
5		maintainance tech				
Task ID	Task	Work	Assignment Units	Assignment	Start	Finish
30	7.1 Monitor system performance	40 hours	100%	0 days	4/7/25 8:00 AM	4/11/25 5:00 PM
		40 hours				
6		system tester				
Task ID	Task	Work	Assignment Units	Assignment	Start	Finish
21	5.1 Unit testing of individual	48 hours	100%	0 days	2/19/25 8:00 AM	2/26/25 5:00 PM
22	5.2 Integration testing for system	40 hours	100%	0 days	2/20/25 8:00 AM	2/26/25 5:00 PM
		88 hours				

Team Contributions

<i>Team Member Number</i>	<i>Name</i>	<i>Contribution(s)</i>
1	Raheim Burkett	Work Breakdown Structure WBS Dictionary Activity List
2	Garrett Grant	Work Breakdown Structure Activity List Project Schedule
3	T'Yondre Leslie	Scope Statement Work Breakdown Structure Project Schedule
4	Iyana Taylor	Work Breakdown Structure Project Schedule Critical Path Report