



**Ain Shams University**  
**Faculty of Engineering**  
**Department of Computer And Systems Engineering**

**CSE441**  
**Software Project Management**

---

**Online Course Registration Portal**

---

**Group Members:**

| Member Name             | ID      |
|-------------------------|---------|
| Michael George Naem     | 2100709 |
| Mayar Walid Said        | 2100953 |
| Hana Shamel Mohamed     | 2100468 |
| Rodina Mohammed Mustafa | 2100754 |
| Kareem Ehab Ibrahim     | 2100913 |
| Moustafa Ahmed Hashem   | 2100467 |
| Karen Maurice William   | 2100748 |

**Academic Year: 2025-2026**  
**Semester: Fall 2025**

## **Abstract**

This document presents a project management plan for the development of an Online Course Registration Portal. The system seeks to simplify the course registration process by adding features like checking prerequisites and managing waitlists. The portal will enable students to efficiently browse available courses, register for classes while respecting academic constraints, and manage their academic schedules effectively.

The project addresses critical challenges in academic course management, including rule enforcement for course prerequisites, handling concurrent registration requests, and ensuring scalable system design to accommodate growing user demands.

This report summarizes key aspects of project management, including planning, scheduling, resource allocation, and risk management. It uses standard tools like Gantt charts, WBS and others to support effective project execution.

The intended audience for this document includes project stakeholders, development teams, academic administrators, and project evaluators who require a thorough understanding of the project scope, timeline, resources, and deliverables.

# Contents

|   |           |
|---|-----------|
| <b>Abstract</b>                                   | <b>i</b>  |
| <b>1 Introduction</b>                             | <b>1</b>  |
| 1.1 Purpose . . . . .                             | 1         |
| 1.2 List of Definitions . . . . .                 | 2         |
| 1.3 Overview . . . . .                            | 3         |
| 1.4 Assumptions . . . . .                         | 4         |
| 1.4.1 Technical Assumptions . . . . .             | 4         |
| 1.4.2 Resource Assumptions . . . . .              | 4         |
| 1.4.3 Schedule Assumptions . . . . .              | 4         |
| 1.4.4 Financial Assumptions . . . . .             | 4         |
| 1.4.5 Stakeholder Assumptions . . . . .           | 4         |
| 1.4.6 Quality Assumptions . . . . .               | 5         |
| <b>2 Project Charter</b>                          | <b>6</b>  |
| 2.1 Project Information . . . . .                 | 6         |
| 2.2 Business Case . . . . .                       | 6         |
| 2.3 Project Description . . . . .                 | 6         |
| 2.4 Project Objectives . . . . .                  | 7         |
| 2.5 Project Scope . . . . .                       | 7         |
| 2.6 Project Deliverables . . . . .                | 7         |
| 2.7 Key Stakeholders . . . . .                    | 8         |
| 2.8 High-Level Risks and Assumptions . . . . .    | 8         |
| 2.9 High-Level Requirements . . . . .             | 9         |
| 2.10 Budget and Resources . . . . .               | 9         |
| 2.11 High-Level Timeline and Milestones . . . . . | 10        |
| 2.12 Success Criteria and KPIs . . . . .          | 10        |
| 2.13 Approval Section . . . . .                   | 10        |
| <b>3 Project Constraints</b>                      | <b>12</b> |
| 3.1 Time Constraints . . . . .                    | 12        |
| 3.1.1 Project Duration . . . . .                  | 12        |
| 3.1.2 Critical Deadlines . . . . .                | 12        |
| 3.1.3 Time Constraint Factors . . . . .           | 12        |
| 3.1.4 Schedule Flexibility . . . . .              | 13        |
| 3.2 Scope Constraints . . . . .                   | 13        |
| 3.2.1 In-Scope Items . . . . .                    | 13        |
| 3.2.2 Out-of-Scope Items . . . . .                | 14        |
| 3.2.3 Scope Change Management . . . . .           | 14        |

|          |   |           |
|----------|---|-----------|
| 3.3      | Cost Constraints . . . . .                          | 15        |
| 3.3.1    | Total Project Budget . . . . .                      | 15        |
| 3.3.2    | Budget Constraints and Limitations . . . . .        | 15        |
| 3.3.3    | Cost Assumptions . . . . .                          | 15        |
| 3.3.4    | Budget Variance Tolerance . . . . .                 | 16        |
| 3.4      | Constraint Priorities and Trade-offs . . . . .      | 16        |
| 3.4.1    | Constraint Priority Matrix . . . . .                | 16        |
| 3.4.2    | Trade-off Scenarios . . . . .                       | 16        |
| 3.5      | Constraint Monitoring and Control . . . . .         | 17        |
| 3.5.1    | Performance Measurement . . . . .                   | 17        |
| 3.5.2    | Reporting Requirements . . . . .                    | 17        |
| 3.5.3    | Interdependency of Time, Scope, and Cost . . . . .  | 17        |
| 3.5.4    | Corrective Actions . . . . .                        | 17        |
| <b>4</b> | <b>Project Phases</b>                               | <b>18</b> |
| 4.1      | Project Lifecycle Overview . . . . .                | 18        |
| 4.2      | Phase 1: Project Initiation and Planning . . . . .  | 18        |
| 4.2.1    | Duration . . . . .                                  | 18        |
| 4.2.2    | Objectives . . . . .                                | 19        |
| 4.2.3    | Key Activities . . . . .                            | 19        |
| 4.2.4    | Deliverables . . . . .                              | 19        |
| 4.2.5    | Milestone . . . . .                                 | 20        |
| 4.3      | Phase 2: Requirements Analysis and Design . . . . . | 20        |
| 4.3.1    | Duration . . . . .                                  | 20        |
| 4.3.2    | Objectives . . . . .                                | 20        |
| 4.3.3    | Key Activities . . . . .                            | 20        |
| 4.3.4    | Deliverables . . . . .                              | 20        |
| 4.3.5    | Milestones . . . . .                                | 21        |
| 4.4      | Phase 3: Development and Implementation . . . . .   | 21        |
| 4.4.1    | Duration . . . . .                                  | 21        |
| 4.4.2    | Objectives . . . . .                                | 21        |
| 4.4.3    | Key Activities . . . . .                            | 21        |
| 4.4.4    | Deliverables . . . . .                              | 22        |
| 4.4.5    | Milestones . . . . .                                | 22        |
| 4.5      | Phase 4: Testing and Quality Assurance . . . . .    | 22        |
| 4.5.1    | Duration . . . . .                                  | 22        |
| 4.5.2    | Objectives . . . . .                                | 23        |
| 4.5.3    | Key Activities . . . . .                            | 23        |
| 4.5.4    | Deliverables . . . . .                              | 23        |
| 4.5.5    | Milestones . . . . .                                | 24        |
| 4.6      | Phase 5: Deployment and Training . . . . .          | 24        |
| 4.6.1    | Duration . . . . .                                  | 24        |
| 4.6.2    | Objectives . . . . .                                | 24        |
| 4.6.3    | Key Activities . . . . .                            | 24        |
| 4.6.4    | Deliverables . . . . .                              | 25        |
| 4.6.5    | Milestones . . . . .                                | 25        |
| 4.7      | Phase 6: Project Closure and Handover . . . . .     | 25        |
| 4.7.1    | Duration . . . . .                                  | 25        |

|          |  |           |
|----------|--|-----------|
| 4.7.2    | Objectives . . . . .   | 25        |
| 4.7.3    | Key Activities . . . . .   | 25        |
| 4.7.4    | Deliverables . . . . .   | 26        |
| 4.7.5    | Milestone . . . . .  | 26        |
| 4.8      | Phase Dependencies and Relationships . . . . .   | 26        |
| 4.9      | Phase Governance . . . . .   | 27        |
| <b>5</b> | <b>Work Breakdown Structure</b>  | <b>28</b> |
| 5.1      | WBS Overview . . . . .   | 28        |
| 5.2      | WBS Visual Representation . . . . .  | 28        |
| 5.3      | WBS Dictionary Sample . . . . .  | 28        |
| <b>6</b> | <b>Gantt Chart and Milestones</b>  | <b>32</b> |
| 6.1      | Project Schedule Overview . . . . .  | 32        |
| 6.2      | Gantt Chart . . . . .  | 32        |
| 6.3      | Project Milestones . . . . .   | 40        |
| 6.4      | Milestone Importance and Impact . . . . .  | 42        |
| 6.4.1    | Critical Milestones . . . . .  | 42        |
| 6.4.2    | Milestone Dependencies . . . . .   | 42        |
| 6.5      | Key Activity Durations Summary . . . . .   | 42        |
| 6.6      | Schedule Assumptions and Constraints . . . . .   | 43        |
| 6.6.1    | Assumptions . . . . .  | 43        |
| 6.6.2    | Constraints . . . . .  | 43        |
| 6.7      | Schedule Risk Management . . . . .   | 43        |
| 6.7.1    | Schedule Risks . . . . .   | 43        |
| 6.8      | Schedule Control Procedures . . . . .  | 43        |
| 6.8.1    | Monitoring and Reporting . . . . .   | 43        |
| 6.8.2    | Change Control . . . . .   | 43        |
| <b>7</b> | <b>AON Network</b>   | <b>44</b> |
| 7.1      | Introduction to AON and Critical Path Method . . . . .   | 44        |
| 7.1.1    | Critical Path Method (CPM) . . . . .   | 44        |
| 7.2      | High-Level Activity List with CPM Calculations (ES, EF, LS, LF, Free Slack, Total Slack) . . . . . | 44        |
| 7.3      | Forward Pass Calculation (ES and EF) . . . . .   | 46        |
| 7.3.1    | Calculation Method . . . . .   | 46        |
| 7.3.2    | Forward Pass Results . . . . .   | 46        |
| 7.4      | Backward Pass Calculation (LS and LF) . . . . .  | 47        |
| 7.4.1    | Calculation Method . . . . .   | 47        |
| 7.4.2    | Backward Pass Results . . . . .  | 47        |
| 7.5      | Slack Analysis . . . . .   | 47        |
| 7.5.1    | Types of Slack . . . . .   | 47        |
| 7.5.2    | Slack Calculations . . . . .   | 48        |
| 7.5.3    | Slack Calculations Explained . . . . .   | 48        |
| 7.5.4    | Slack Implications . . . . .   | 48        |
| 7.6      | Critical Path Identification . . . . .   | 48        |
| 7.6.1    | Critical Path . . . . .  | 48        |
| 7.6.2    | Critical Path Significance . . . . .   | 49        |
| 7.7      | Critical Path Management . . . . .   | 49        |

|           |   |           |
|-----------|---|-----------|
| 7.7.1     | Management Strategies for Critical Activities . . . . . | 49        |
| 7.7.2     | Managing Non-Critical Activities . . . . .              | 49        |
| 7.8       | Network Analysis Summary . . . . .                      | 50        |
| 7.8.1     | Key Findings . . . . .                                  | 50        |
| 7.8.2     | Schedule Risk Assessment . . . . .                      | 50        |
| 7.8.3     | Recommendations . . . . .                               | 50        |
| <b>8</b>  | <b>PERT Time Estimation</b>                             | <b>51</b> |
| 8.1       | What is PERT? . . . . .                                 | 51        |
| 8.2       | PERT Formula . . . . .                                  | 51        |
| 8.2.1     | Formula Explanation . . . . .                           | 51        |
| 8.3       | PERT Estimates for Project Activities . . . . .         | 51        |
| 8.4       | Comparison: Deterministic vs. PERT Estimates . . . . .  | 53        |
| 8.5       | PERT Analysis Interpretation . . . . .                  | 53        |
| 8.5.1     | Activities with High Uncertainty . . . . .              | 53        |
| 8.6       | Recommendations . . . . .                               | 53        |
| <b>9</b>  | <b>NPV Analysis</b>                                     | <b>54</b> |
| 9.1       | NPV Analysis Overview . . . . .                         | 54        |
| 9.1.1     | NPV Formula . . . . .                                   | 54        |
| 9.1.2     | Decision Criteria . . . . .                             | 54        |
| 9.2       | Project Cost Estimation . . . . .                       | 55        |
| 9.2.1     | Initial Investment (Year 0) . . . . .                   | 55        |
| 9.2.2     | Annual Operating Costs (Years 1-5) . . . . .            | 55        |
| 9.3       | Project Benefits Estimation . . . . .                   | 56        |
| 9.3.1     | Quantifiable Benefits . . . . .                         | 56        |
| 9.3.2     | Intangible Benefits (Not Quantified in NPV) . . . . .   | 56        |
| 9.4       | NPV Calculation . . . . .                               | 56        |
| 9.4.1     | Assumptions . . . . .                                   | 56        |
| 9.4.2     | Cash Flow Analysis . . . . .                            | 57        |
| 9.5       | Sensitivity Analysis . . . . .                          | 57        |
| 9.6       | Analysis and Recommendations . . . . .                  | 58        |
| 9.6.1     | Financial Viability Assessment . . . . .                | 58        |
| 9.6.2     | Strategic Considerations . . . . .                      | 58        |
| 9.6.3     | Recommendations . . . . .                               | 58        |
| <b>10</b> | <b>Estimated Cash Flow</b>                              | <b>59</b> |
| 10.1      | Cash Flow Analysis Overview . . . . .                   | 59        |
| 10.2      | Project Cash Outflows . . . . .                         | 59        |
| 10.2.1    | Development Phase Cash Outflows (Year 0) . . . . .      | 59        |
| 10.2.2    | Operating Phase Cash Outflows (Years 1-5) . . . . .     | 59        |
| 10.3      | Project Cash Inflows (Benefits) . . . . .               | 60        |
| 10.3.1    | Annual Cash Inflows from Benefits . . . . .             | 60        |
| 10.4      | Net Cash Flow Summary . . . . .                         | 60        |
| 10.5      | Cumulative Cash Flow . . . . .                          | 60        |
| 10.6      | Cash Flow Analysis Insights . . . . .                   | 60        |
| 10.6.1    | Key Observations . . . . .                              | 60        |
| 10.6.2    | Funding Requirements . . . . .                          | 61        |
| 10.7      | Cash Flow Management Strategies . . . . .               | 61        |

|   |           |
|---|-----------|
| 10.7.1 Funding Strategy . . . . .                                 | 61        |
| 10.7.2 Cash Flow Optimization . . . . .                           | 61        |
| <b>11 Payback Period and ROI</b>                                  | <b>62</b> |
| 11.1 Payback Period Analysis . . . . .                            | 62        |
| 11.1.1 Payback Period Calculation . . . . .                       | 62        |
| 11.1.2 Payback Period Interpretation . . . . .                    | 62        |
| 11.1.3 Discounted Payback Period . . . . .                        | 63        |
| 11.2 Return on Investment (ROI) Analysis . . . . .                | 63        |
| 11.2.1 ROI Formula . . . . .                                      | 63        |
| 11.2.2 Simple ROI Calculation (5-Year Period) . . . . .           | 63        |
| 11.2.3 Annualized ROI . . . . .                                   | 63        |
| 11.2.4 Alternative ROI: Initial Investment Only . . . . .         | 64        |
| 11.3 Financial Metrics Summary . . . . .                          | 64        |
| 11.4 Break-Even Analysis . . . . .                                | 64        |
| 11.4.1 Break-Even Point . . . . .                                 | 64        |
| 11.4.2 Break-Even Sensitivity . . . . .                           | 64        |
| 11.5 Financial Viability Assessment . . . . .                     | 65        |
| 11.5.1 Strengths . . . . .  | 65        |
| 11.5.2 Weaknesses . . . . .                                       | 65        |
| 11.5.3 Recommendations . . . . .                                  | 65        |
| 11.6 Comparative Analysis . . . . .                               | 65        |
| 11.6.1 Industry Benchmarks . . . . .                              | 65        |
| 11.6.2 Strategic Justification . . . . .                          | 65        |
| <b>12 RACI Chart</b>  | <b>67</b> |
| 12.1 RACI Matrix Overview . . . . .                               | 67        |
| 12.2 Project Roles . . . . .                                      | 67        |
| 12.3 RACI Matrix . . . . .  | 67        |
| 12.4 RACI Analysis and Validation . . . . .                       | 69        |
| 12.4.1 Key Accountability Assignments . . . . .                   | 69        |
| 12.5 Communication Implications . . . . .                         | 69        |
| 12.5.1 Consulted (C) Relationships . . . . .                      | 69        |
| 12.5.2 Informed (I) Relationships . . . . .                       | 69        |
| 12.6 Conflict Resolution . . . . .                                | 69        |
| <b>13 Risk Analysis</b>   | <b>71</b> |
| 13.1 Plan Risk Management . . . . .                               | 71        |
| 13.1.1 Risk Management Plan . . . . .                             | 71        |
| 13.2 Identify Risks . . . . .                                     | 71        |
| 13.2.1 Risk Identification Tools . . . . .                        | 71        |
| 13.2.2 Risk Register . . . . .                                    | 71        |
| 13.3 Perform Qualitative Risk Analysis . . . . .                  | 73        |
| 13.3.1 Probability and Impact Scales . . . . .                    | 73        |
| 13.3.2 Probability-Impact Matrix . . . . .                        | 73        |
| 13.4 Perform Quantitative Risk Analysis . . . . .                 | 73        |
| 13.4.1 Sensitivity Analysis . . . . .                             | 73        |
| 13.5 Plan Risk Responses . . . . .                                | 73        |
| 13.5.1 Response Strategies for Negative Risks (Threats) . . . . . | 74        |

|   |           |
|---|-----------|
| 13.5.2 Response Strategies for Positive Risks (Opportunities) . . . . . | 74        |
| 13.5.3 Risk Response Plan . . . . .                                     | 74        |
| 13.5.4 Contingency Reserves . . . . .                                   | 75        |
| 13.6 Monitor and Control Risks . . . . .                                | 75        |
| 13.6.1 Monitoring Activities . . . . .                                  | 75        |
| <b>14 Schedule Analysis</b>   | <b>76</b> |
| 14.1 Schedule Calculation Overview . . . . .                            | 76        |
| 14.1.1 Definitions . . . . .  | 76        |
| 14.1.2 Calculation Methods . . . . .                                    | 76        |
| 14.2 Complete Schedule Analysis Table . . . . .                         | 76        |
| 14.3 Critical Path Activities . . . . .                                 | 78        |
| 14.4 Float Analysis . . . . .   | 78        |
| 14.4.1 Activities with Total Float > 0 . . . . .                        | 78        |
| 14.4.2 Near-Critical Activities . . . . .                               | 78        |
| 14.5 Schedule Compression Opportunities . . . . .                       | 79        |
| 14.5.1 Fast-Tracking Opportunities . . . . .                            | 79        |
| 14.5.2 Crashing Opportunities . . . . .                                 | 79        |
| 14.6 Schedule Management Recommendations . . . . .                      | 79        |
| <b>15 Cost Estimation</b>   | <b>80</b> |
| 15.1 Cost Estimation Overview . . . . .                                 | 80        |
| 15.2 Method 1: Bottom-Up Estimation . . . . .                           | 80        |
| 15.2.1 Personnel Costs . . . . .  | 80        |
| 15.2.2 Equipment and Infrastructure . . . . .                           | 81        |
| 15.2.3 Software and Licenses . . . . .                                  | 81        |
| 15.2.4 Other Direct Costs . . . . .                                     | 81        |
| 15.2.5 Bottom-Up Total . . . . .  | 81        |
| 15.3 Method 2: Parametric Estimation . . . . .                          | 82        |
| 15.3.1 Parametric Model Basis . . . . .                                 | 82        |
| 15.3.2 Parametric Calculation . . . . .                                 | 82        |
| 15.3.3 Additional Costs (Parametric Method) . . . . .                   | 82        |
| 15.3.4 Parametric Total . . . . .                                       | 82        |
| 15.4 Method 3: Analogous (Comparative) Estimation . . . . .             | 82        |
| 15.4.1 Comparative Projects . . . . .                                   | 83        |
| 15.4.2 Adjustment Factors . . . . .                                     | 83        |
| 15.4.3 Analogous Calculation . . . . .                                  | 83        |
| 15.5 Comparison and Reconciliation . . . . .                            | 83        |
| 15.6 Analysis of Inconsistencies . . . . .                              | 84        |
| 15.6.1 Why Bottom-Up is Lower . . . . .                                 | 84        |
| 15.6.2 Why Parametric and Analogous are Higher . . . . .                | 84        |
| 15.6.3 Justification of Differences . . . . .                           | 84        |
| 15.7 Approved Project Budget . . . . .                                  | 84        |
| 15.7.1 Reconciliation with Detailed Estimates . . . . .                 | 85        |
| 15.8 Cost Control and Monitoring . . . . .                              | 85        |
| <b>16 References</b>  | <b>86</b> |
| 16.1 Textbook . . . . .   | 86        |
| 16.2 References . . . . .   | 86        |

|                               |           |
|-------------------------------|-----------|
| <b>17 Video Presentations</b> | <b>87</b> |
|-------------------------------|-----------|

# List of Figures

|      |   |    |
|------|---|----|
| 4.1  | Project Phases and Major Milestones Timeline . . . . .  | 18 |
| 5.1  | Work Breakdown Structure - Visual Hierarchy . . . . .   | 28 |
| 6.1  | Project Gantt Chart - tasks . . . . .   | 32 |
| 6.2  | Project Gantt Chart - tasks - continued . . . . .   | 33 |
| 6.3  | Project Gantt Chart - tasks - continued . . . . .   | 34 |
| 6.4  | Project Gantt Chart - tasks - continued . . . . .   | 35 |
| 6.5  | Project Gantt Chart - tasks - continued . . . . .   | 36 |
| 6.6  | Project Gantt Chart - tasks . . . . .   | 36 |
| 6.7  | Project Gantt Chart - timeline . . . . .  | 37 |
| 6.8  | Project Gantt Chart - timeline - continued . . . . .  | 37 |
| 6.9  | Project Gantt Chart - timeline - continued . . . . .  | 38 |
| 6.10 | Project Gantt Chart - timeline - continued . . . . .  | 38 |
| 6.11 | Project Gantt Chart - timeline - continued . . . . .  | 39 |
| 6.12 | Project Gantt Chart - timeline - continued . . . . .  | 39 |
| 7.1  | Activity on Node (AON) Network Diagram - critical path activities surrounded by red boxes . . . . . | 46 |

# List of Tables

|      |  |    |
|------|--|----|
| 3.1  | Project Budget Allocation . . . . .                                  | 15 |
| 3.2  | Constraint Priorities . . . . .                                      | 16 |
| 4.1  | Phase Dependencies . . . . .   | 26 |
| 5.1  | WBS Dictionary - Project Charter Development (1.1.1.1) . . . . .     | 28 |
| 5.2  | WBS Dictionary - Stakeholder Interviews (1.2.1.1) . . . . .          | 29 |
| 5.3  | WBS Dictionary - Database Schema Design (1.3.2.2) . . . . .          | 29 |
| 5.4  | WBS Dictionary - Prerequisite Validation Engine (1.4.1.4) . . . . .  | 29 |
| 5.5  | WBS Dictionary - Student Portal Interface (1.4.2.1) . . . . .        | 30 |
| 5.6  | WBS Dictionary - System Testing (1.5.2.3) . . . . .                  | 30 |
| 5.7  | WBS Dictionary - Application Deployment (1.6.2) . . . . .            | 30 |
| 5.8  | WBS Dictionary - Administrator Training Sessions (1.7.3.1) . . . . . | 31 |
| 5.9  | WBS Dictionary - Final Report and Sign-off (1.8.10) . . . . .        | 31 |
| 6.1  | Project Milestones and Justifications . . . . .                      | 41 |
| 6.2  | Milestone Dependencies . . . . .                                     | 42 |
| 6.3  | Major Phase Durations . . . . .                                      | 42 |
| 7.1  | High-Level Activity List with CPM Calculations . . . . .             | 45 |
| 7.2  | Critical Path Activities . . . . .                                   | 49 |
| 8.1  | PERT Three-Point Time Estimates for Project Activities . . . . .     | 52 |
| 8.2  | Deterministic vs. PERT Duration Comparison . . . . .                 | 53 |
| 9.1  | Initial Project Costs . . . . .                                      | 55 |
| 9.2  | Annual Operating and Maintenance Costs . . . . .                     | 55 |
| 9.3  | Annual Benefits . . . . .  | 56 |
| 9.4  | NPV Cash Flow Analysis . . . . .                                     | 57 |
| 9.5  | Present Value Calculations . . . . .                                 | 57 |
| 9.6  | NPV Sensitivity to Discount Rate . . . . .                           | 57 |
| 9.7  | NPV Sensitivity to Annual Benefits . . . . .                         | 57 |
| 10.1 | Development Phase Cash Outflows by Quarter . . . . .                 | 59 |
| 10.2 | Annual Operating Cash Outflows . . . . .                             | 59 |
| 10.3 | Annual Cash Inflows (Savings and Benefits) . . . . .                 | 60 |
| 10.4 | Net Cash Flow by Year . . . . .                                      | 60 |
| 10.5 | Cumulative Cash Flow Analysis . . . . .                              | 60 |
| 11.1 | Payback Period Calculation . . . . .                                 | 62 |
| 11.2 | Discounted Payback Period . . . . .                                  | 63 |
| 11.3 | Financial Performance Metrics Summary . . . . .                      | 64 |

|  |    |
|--|----|
| 11.4 Break-Even Sensitivity to Annual Benefits . . . . .                 | 64 |
| 11.5 Comparison with Industry Norms . . . . .                            | 65 |
| 12.1 Project Roles and Responsibilities . . . . .                        | 67 |
| 12.2 RACI Matrix for Project Activities . . . . .                        | 68 |
| 12.3 Primary Accountabilities by Role . . . . .                          | 69 |
| 13.1 Risk Management Plan Elements . . . . .                             | 71 |
| 13.2 Risk Identification Methods Used . . . . .                          | 71 |
| 13.3 Project Risk Register . . . . .                                     | 72 |
| 13.4 Probability and Impact Definitions . . . . .                        | 73 |
| 13.5 Risk Probability-Impact Matrix . . . . .                            | 73 |
| 13.6 Sensitivity Analysis Results . . . . .                              | 73 |
| 13.7 Threat Response Strategies . . . . .                                | 74 |
| 13.8 Opportunity Response Strategies . . . . .                           | 74 |
| 13.9 Risk Response Plan . . . . .  | 74 |
| 13.10 Contingency Reserve Allocation . . . . .                           | 75 |
| 14.1 Complete Schedule Analysis with ES, EF, LS, LF, and Float . . . . . | 77 |
| 14.2 Non-Critical Activities with Scheduling Flexibility . . . . .       | 78 |
| 14.3 Crashing Analysis for Critical Activities . . . . .                 | 79 |
| 15.1 Personnel Costs - Bottom-Up Estimation . . . . .                    | 80 |
| 15.2 Equipment and Infrastructure Costs . . . . .                        | 81 |
| 15.3 Software and License Costs . . . . .                                | 81 |
| 15.4 Other Direct Costs . . . . .  | 81 |
| 15.5 Bottom-Up Cost Estimation Summary . . . . .                         | 81 |
| 15.6 Additional Costs for Parametric Method . . . . .                    | 82 |
| 15.7 Parametric Cost Estimation Summary . . . . .                        | 82 |
| 15.8 Reference Projects for Analogous Estimation . . . . .               | 83 |
| 15.9 Analogous Cost Estimation Summary . . . . .                         | 83 |
| 15.10 Cost Estimation Method Comparison . . . . .                        | 83 |
| 15.11 Approved Project Budget . . . . .                                  | 84 |

# Chapter 1

## Introduction

This chapter introduces the Online Course Registration Portal project and sets the foundation for this project management plan.

### 1.1 Purpose

The purpose of this document is to present a project management plan for the development and implementation of an Online Course Registration Portal. This document serves as a complete reference guide for all stakeholders involved in the project, including project managers, development teams, quality assurance personnel, academic administrators, and project sponsors.

This project management plan aims to:

- Define the complete project scope, objectives, and deliverables
- Establish detailed project schedules, milestones, and timelines
- Identify and allocate necessary resources for project execution
- Analyze project costs and financial viability
- Assess potential risks and mitigation strategies
- Provide a structured framework for project monitoring and control

The intended readership of this document includes:

- **Project Sponsors and Stakeholders:** To understand project scope, timeline, and investment requirements
- **Project Managers:** To guide project planning, execution, and monitoring activities
- **Development Teams:** To understand their roles, responsibilities, and deliverables
- **Quality Assurance Teams:** To plan testing and validation activities
- **Academic Administrators:** To evaluate the system's alignment with institutional requirements
- **Academic Evaluators:** To assess the project management methodology and planning quality

## 1.2 List of Definitions

This section provides definitions for all technical terms, acronyms, and abbreviations used throughout this document to ensure clear understanding for all readers.

**AON** Activity-on-Node: A project management technique for scheduling activities where nodes represent activities and arrows show dependencies

**API** Application Programming Interface: A set of protocols and tools for building software applications

### Critical Path

The longest sequence of dependent activities that determines the minimum project duration

**EF** Early Finish: The earliest possible time an activity can finish

**EGP** Egyptian Pound: The official currency of Egypt

**ES** Early Start: The earliest possible time an activity can start

### Gantt Chart

A bar chart that illustrates a project schedule showing activities, durations, and dependencies

**LF** Late Finish: The latest time an activity can finish without delaying the project

**LS** Late Start: The latest time an activity can start without delaying the project

**NPV** Net Present Value: The difference between the present value of cash inflows and outflows over time

**PERT** Network analysis technique used to estimate project duration when there is a high degree of uncertainty about the individual activity duration estimates

**RACI** Responsible, Accountable, Consulted, Informed: A matrix describing roles and responsibilities

**ROI** Return on Investment: A financial performance metric used to evaluate the efficiency and profitability of an investment, calculated as the ratio of net profit to initial investment cost, expressed as a percentage

**WBS** Work Breakdown Structure: A hierarchical decomposition of project work into smaller manageable components

### Prerequisite

A course or requirement that must be completed before enrolling in another course

**Waitlist** A queue of students waiting for a spot in a course when it reaches maximum capacity

**Timetable Conflict**

A scheduling issue where two courses have overlapping class times

**Concurrency**

The ability to handle multiple simultaneous registration requests

**Milestone**

A significant point or event in the project timeline

**Float/Slack**

The amount of time an activity can be delayed without affecting the project completion date

**Free Slack/Free Float**

The amount of time an activity can be delayed without delaying the early start of any immediately following activities

**Total Slack/Total Float**

The amount of time an activity may be delayed from its early start without delaying the planned project finish date

**Resource Allocation**

The process of assigning available resources to project activities

**Risk Matrix**

A tool for assessing and prioritizing project risks based on probability and impact

**Project Charter**

The foundational document that formally authorizes a project and gives the project manager the authority to apply organizational resources. It ensures all stakeholders start with a shared understanding of the project's goals, scope, and constraints

### 1.3 Overview

This document is organized into seventeen chapters, each addressing specific aspects of project management for the Online Course Registration Portal. The structure covers all essential project management knowledge areas.

The document organization is as follows:

- **Chapters 1-2:** Provide introduction, project charter, and foundational project information
- **Chapters 3-5:** Define project constraints, phases, and work breakdown structure
- **Chapters 6-9:** Present project scheduling, network analysis, and time estimation
- **Chapters 10-12:** Cover financial analysis including NPV, cash flow, and ROI calculations
- **Chapters 13-14:** Address risk management and detailed schedule analysis

- **Chapters 15-17:** Discuss cost estimation methodologies and project references

Each chapter includes relevant diagrams, charts, tables, and calculations created using professional tools such as Microsoft Visio and Microsoft Project.

## 1.4 Assumptions

The following assumptions have been made in developing this project management plan. These assumptions are critical for project planning and should be validated during project execution.

### 1.4.1 Technical Assumptions

1. The university infrastructure supports web-based application deployment
2. Existing student information systems provide integration capabilities
3. Internet connectivity is available for both development and production environments

### 1.4.2 Resource Assumptions

1. Required personnel (developers, designers, testers) are available during project execution
2. Necessary hardware and software resources can be procured within budget
3. Key stakeholders will be accessible for decision-making

### 1.4.3 Schedule Assumptions

1. The project starts on the planned date
2. Requirements will be finalized within the allocated timeframe
3. User acceptance testing completes before the target registration period

### 1.4.4 Financial Assumptions

1. Project funding will be available according to the approved budget schedule
2. Foreign exchange rates will not significantly impact costs for external services
3. Third-party vendor costs will remain consistent with initial estimates

### 1.4.5 Stakeholder Assumptions

1. Stakeholders will provide timely feedback on deliverables
2. Key decision-makers will be available for project reviews and approvals
3. Requirements will remain stable throughout the planning phase
4. Stakeholder commitment to the project objectives remains constant

#### **1.4.6 Quality Assumptions**

1. Industry-standard coding practices will be followed
2. Security best practices will protect student data

## Chapter 2

### Project Charter

#### 2.1 Project Information

|                           |                                   |
|---------------------------|-----------------------------------|
| <b>Project Name</b>       | Online Course Registration Portal |
| <b>Project Start Date</b> | March 1, 2026                     |
| <b>Project End Date</b>   | September 15, 2026                |
| <b>Project Duration</b>   | 6.5 months (28 weeks)             |
| <b>Project Sponsor</b>    | Professor Gamal Ebrahim           |
| <b>Project Manager</b>    | Engineer Sally E. Shaker          |

#### 2.2 Business Case

|                          |  |
|--------------------------|--|
| <b>The Problem</b>       | Universities require reliable course registration systems to manage large numbers of students registering concurrently while enforcing academic rules. Manual or poorly designed systems often result in prerequisite violations, timetable conflicts, and unfair seat allocation. |
| <b>The Opportunity</b>   | Automation can significantly improve efficiency, accuracy, and student satisfaction  |
| <b>Purpose</b>           | The purpose of this project is to design and implement a scalable online course registration portal that enforces registration rules automatically, handles concurrent access correctly, and ensures data consistency under high load.   |
| <b>Strategic Context</b> | Academic institutions need scalable, reliable systems to handle growing student enrollment and complex registration rules while maintaining data integrity under concurrent access   |

#### 2.3 Project Description

The Online Course Registration Portal is a web-based system that automates the course registration process for university students. The system provides a user-friendly interface where students can browse available courses, check prerequisites, register for courses, and manage their academic schedules.

The portal enforces all academic policies automatically, including prerequisite validation, credit limit enforcement, and timetable conflict detection. It manages course capacity efficiently through a waitlist system that automatically allocates seats when they become available. The system is designed to handle high concurrent access during peak registration periods while maintaining data consistency and reliability.

This project delivers both the technical infrastructure and the functional capabilities required to support modern academic registration workflows in a distributed, scalable environment.

## 2.4 Project Objectives

| <b>SMART Objectives:</b>   |
|--|
| 1. Design and implement a student course registration system       |
| 2. Enforce academic rules such as prerequisites and credit limits  |
| 3. Detect and prevent timetable conflicts during registration      |
| 4. Support waitlists for full courses with fair promotion policies |
| 5. Handle concurrent registration requests safely and correctly    |
| 6. Demonstrate a scalable and maintainable system design           |

## 2.5 Project Scope

| <b>In Scope</b>   |
|---|
| - Student registration and course enrollment functionality                  |
| - Prerequisite validation before enrollment                                 |
| - Timetable conflict detection and prevention                               |
| - Course capacity management  |
| - Waitlist handling and automatic seat allocation                           |
| - Concurrent access handling (multiple students registering simultaneously) |
| - Backend logic and REST APIs   |
| - Database schema and design  |
| - System documentation and testing  |
| <b>Out of Scope</b>   |
| - Mobile application development  |
| - Payment or tuition processing   |
| - Learning management system (LMS) features                                 |
| - Advanced analytics or recommendation systems                              |

## 2.6 Project Deliverables

| <b>Major Deliverables:</b>  |
|---|
| 1. Functional online course registration portal                   |
| 2. Backend system implementing rule enforcement logic             |
| 3. REST API documentation   |
| 4. Database schema and design documentation                       |
| 5. Concurrency handling demonstration (simulations or test cases) |
| 6. System architecture and design documentation                   |
| 7. Quality assurance and testing documentation                    |
| 8. User documentation and guides                                  |
| 9. Final project report   |

## 2.7 Key Stakeholders

| Stakeholder             | Role and Interest   |
|-------------------------|---|
| Project Sponsor         | Provides funding and strategic oversight, champions the project and has ultimate accountability for its success.  |
| Project Manager         | Responsible for planning, coordination, task allocation and progress tracking, has the authority to assign tasks, manage technical decisions, and ensure adherence to project objectives and deadlines. |
| Development Team        | Designs, develops, and tests the system; needs clear requirements and adequate resources  |
| Students (End Users)    | Primary system users; interested in ease of use, reliability, and convenience   |
| Academic Affairs Office | Manages curriculum and courses; ensures prerequisite accuracy and data integrity  |
| IT Department           | Provides technical infrastructure; concerned with system reliability and security   |

## 2.8 High-Level Risks and Assumptions

| Key Risks  |
|--|
| <ul style="list-style-type: none"> <li>- Integration issues between distributed components</li> <li>- Race conditions during concurrent registration causing data inconsistencies</li> <li>- Complexity of prerequisite and conflict rules leading to implementation challenges</li> <li>- Performance bottlenecks under high user load</li> <li>- Time limitations affecting testing depth and quality assurance</li> <li>- Scope creep from stakeholder requests</li> <li>- Key resource unavailability</li> </ul> |
| Assumptions  |
| <ul style="list-style-type: none"> <li>- Team members are available throughout the semester</li> <li>- Required tools and frameworks are accessible</li> <li>- Student and course data are available or can be simulated</li> <li>- Team members have sufficient technical background</li> <li>- System will be deployed in a controlled academic environment</li> <li>- Subject matter experts will be available for consultation</li> <li>- Necessary computing resources will be provided</li> </ul>              |

## 2.9 High-Level Requirements

| <b>Functional Requirements</b>                               |  |
|--|--|
| 1. Students can view available courses                       |  |
| <b>Non-Functional Requirements</b>                           |  |
| 2. Students can register for courses                         |  |
| 3. System validates prerequisites before enrollment          |  |
| 4. System prevents timetable conflicts                       |  |
| 5. System manages course capacity and waitlists              |  |
| 6. System handles concurrent user registration requests      |  |
| 7. System provides enrollment confirmation and notifications |  |

## 2.10 Budget and Resources

| <b>Preliminary Budget Estimate</b> |                                     |                   |
|------------------------------------|-------------------------------------|-------------------|
| <b>Category</b>                    | <b>Description</b>                  | <b>Cost (EGP)</b> |
| Personnel Costs                    | Development Team                    | 800,000           |
| Project Management                 | Project coordination and oversight  | 150,000           |
| Software Licenses and Tools        | Development and collaboration tools | 100,000           |
| Hardware and Infrastructure        | Servers and hosting                 | 200,000           |
| Testing and QA                     | Quality assurance activities        | 120,000           |
| Training and Documentation         | User guides and training materials  | 80,000            |
| Contingency Reserve                | 10% buffer for unexpected costs     | 145,000           |
| <b>Total Estimated Budget</b>      |                                     | <b>1,595,000</b>  |

| <b>Resource Requirements:</b>   |
|---|
| <ul style="list-style-type: none"> <li>- Development team (backend, frontend, database developers)</li> <li>- Project manager</li> <li>- Quality assurance / testing resources</li> <li>- Development tools and software licenses</li> <li>- Cloud hosting or server infrastructure</li> <li>- Database management system</li> <li>- Version control and collaboration tools</li> </ul> |

## 2.11 High-Level Timeline and Milestones

| Phase                                 | Description and Activities  |
|---------------------------------------|---|
| Requirements Analysis & System Design | Define detailed requirements, design system architecture, and create technical specifications |
| Database Design                       | Design and implement database schema, create ER diagrams, and establish data models           |
| Backend and Frontend Development      | Implement core business logic, rule enforcement, API endpoints, and user interface components |
| Concurrency Handling & Testing        | Implement concurrent access controls, develop test cases, and perform load testing            |
| System Integration                    | Integrate all components, perform end-to-end testing, and resolve integration issues          |
| Final Testing & Documentation         | Complete comprehensive testing, finalize documentation, and prepare for deployment            |

## 2.12 Success Criteria and KPIs

| The project will be considered successful if:                |
|--|
| 1. Registration rules are correctly enforced                 |
| 2. No data inconsistencies occur under concurrent access     |
| 3. Timetable conflicts are accurately detected and prevented |
| 4. Waitlists function correctly with fair seat allocation    |
| 5. System meets academic evaluation criteria                 |
| 6. All functional requirements are implemented               |
| 7. System passes all test cases                              |
| 8. Documentation is complete and comprehensive               |
| 9. System demonstrates scalability under load testing        |

## 2.13 Approval Section

*This Charter formally authorizes the commencement of the Online Course Registration Portal project.*

**Project Sponsor**

Professor Gamal Ebrahim

Signature

Date

**Project Manager**

Engineer Sally E. Shaker

Signature

Date

## Chapter 3

# Project Time, Scope, and Cost Constraints

This chapter defines the triple constraints of project management—time, scope, and cost—which form the foundation for project planning and control. These constraints are interdependent and must be carefully balanced to ensure project success.

### 3.1 Time Constraints

The project timeline is governed by academic calendar requirements and institutional needs. The following temporal constraints apply to this project:

#### 3.1.1 Project Duration

- **Total Project Duration:** 6.5 months (28 weeks)
- **Project Start Date:** March 1, 2026
- **Project Completion Date:** September 15, 2026
- **Production Deployment Date:** August 31, 2026 (4 weeks before registration period)

#### 3.1.2 Critical Deadlines

1. **Requirements Finalization:** Week 3 (March 21, 2026)
2. **Design Approval:** Week 6 (April 11, 2026)
3. **Development Completion:** Week 16 (June 19, 2026)
4. **Testing Completion:** Week 21 (July 24, 2026)
5. **User Training:** Week 23 (August 7, 2026)
6. **System Go-Live (Production Deployment):** Week 26 (August 28, 2026)
7. **Post-Deployment Stabilization & Support:** Weeks 27-28 (August 29 - September 15, 2026)
8. **Project Closure:** Week 28 (September 15, 2026)
9. **Registration Period Start:** October 1, 2026

#### 3.1.3 Time Constraint Factors

- Registration period start date is non-negotiable and externally imposed
- Academic calendar dictates when the system must be operational

- Limited availability of stakeholders during semester breaks
- Peak registration periods require system to be fully functional
- Training must be completed before semester start

### 3.1.4 Schedule Flexibility

- **Buffer Time:** 2 weeks built into schedule for unforeseen delays
- **Fast-Tracking Opportunities:** Some testing activities can overlap with development
- **Resource Flexibility:** Additional resources can be allocated to critical path activities if needed
- **Non-Negotiable Deadline:** Final deployment must occur before registration period

## 3.2 Scope Constraints

The project scope is carefully defined to ensure deliverable quality while maintaining schedule and budget constraints.

### 3.2.1 In-Scope Items

The following features and capabilities are explicitly included in the project scope:

#### 1. Core Registration Functionality

- Student login and authentication
- Course browsing and search
- Course registration (add/drop/withdraw)
- Prerequisite validation
- Timetable conflict detection
- Waitlist management

#### 2. Student Features

- Personal schedule/timetable view
- Course details and descriptions
- Seat availability checking
- Registration history
- Email notifications

#### 3. Administrative Features

- Course setup and management

- Enrollment reports and analytics
- Waitlist monitoring
- System configuration
- User management

#### 4. System Integration

- Integration with student information system
- Integration with authentication system
- Integration with email notification system
- Integration limited to existing systems via documented APIs

#### 5. Documentation and Training

- User manuals (student and administrator)
- Technical documentation
- Training materials
- Online help system

##### 3.2.2 Out-of-Scope Items

The following features are explicitly excluded from the current project scope:

1. Mobile native applications (iOS/Android), Web-based interface only (no desktop application)
2. Grade reporting and transcript generation
3. Tuition payment processing
4. Course evaluation and feedback system
5. Social features (student forums, course reviews)
6. Advanced analytics and predictive modeling

##### 3.2.3 Scope Change Management

- All scope changes must go through formal change control process
- Impact analysis required for all change requests (time, cost, quality)
- Project sponsor approval required for scope changes
- Scope changes may result in timeline or budget adjustments
- Change request log maintained throughout project lifecycle

### 3.3 Cost Constraints

The project must be completed within the approved budget while maintaining quality standards and meeting scope requirements.

#### 3.3.1 Total Project Budget

**Table 3.1:** Project Budget Allocation

| Category                              | Budget (EGP)     | Percentage  |
|---------------------------------------|------------------|-------------|
| Personnel Costs                       | 800,000          | 50.2%       |
| - Project Manager                     | 150,000          | 9.4%        |
| - Development Team                    | 530,000          | 33.2%       |
| - QA and Testing Staff                | 120,000          | 7.5%        |
| Infrastructure and Hardware           | 200,000          | 12.5%       |
| - Server Infrastructure and Equipment | 120,000          | 7.5%        |
| - Development and Testing Environment | 80,000           | 5.0%        |
| Software and Licenses                 | 100,000          | 6.3%        |
| - Development Tools                   | 60,000           | 3.8%        |
| - Database and Server Licenses        | 40,000           | 2.5%        |
| Training and Documentation            | 80,000           | 5.0%        |
| - Training Materials and Sessions     | 50,000           | 3.1%        |
| - Documentation Development           | 30,000           | 1.9%        |
| Testing and Quality Assurance         | 120,000          | 7.5%        |
| Other Direct Costs                    | 150,000          | 9.4%        |
| Contingency Reserve (10%)             | 145,000          | 9.1%        |
| <b>Total Budget</b>                   | <b>1,595,000</b> | <b>100%</b> |

#### 3.3.2 Budget Constraints and Limitations

1. **Target Budget:** 1,595,000 EGP
2. **Maximum Budget Ceiling:** 1,674,750 EGP ( $1,595,000 + 5\% \text{ variance allowance}$ )
3. **Budget Approval:** Any expenditure exceeding 1,595,000 EGP requires project sponsor approval
4. **Funding Phases:** Budget released in quarterly installments
5. **Infrastructure:** Must utilize existing university infrastructure where possible
6. **Procurement:** All purchases must follow university procurement policies

#### 3.3.3 Cost Assumptions

- No significant market price fluctuations for technology components
- University provides existing infrastructure at no additional cost
- Open-source software utilized where appropriate to minimize licensing costs

### 3.3.4 Budget Variance Tolerance

- **Total Budget Variance:** Up to +5% (79,750 EGP) above target budget of 1,595,000 EGP with sponsor approval
- **Category Variance:** Individual budget categories may vary by ±10% through reallocation, provided total budget remains within approved limits
- **Category Reallocation:** Requires project manager approval for variances up to 50,000 EGP; sponsor approval for larger reallocations
- **Contingency Usage:** Requires project manager approval and must be documented with justification
- **Budget Overrun Prevention:** Regular monitoring and early escalation required when approaching variance thresholds

## 3.4 Constraint Priorities and Trade-offs

In case of conflicts between constraints, the following priority order applies:

### 3.4.1 Constraint Priority Matrix

**Table 3.2:** Constraint Priorities

| Priority | Constraint | Rationale   |
|----------|------------|---|
| 1        | Time       | Registration deadline is non-negotiable; missing it causes major institutional impact |
| 2        | Scope      | Core functionality must be delivered; system must be usable for students              |
| 3        | Cost       | Budget has some flexibility; additional funding may be available if justified         |

### 3.4.2 Trade-off Scenarios

#### 1. If timeline is at risk:

- Consider reducing scope (move non-critical features to Phase 2)
- Allocate additional resources (increase cost)

#### 2. If budget is at risk:

- Reduce scope to fit within budget
- Utilize more cost-effective resources or technologies

#### 3. If scope must expand:

- Request additional budget
- Evaluate impact on quality and prioritize features
- Consider phased implementation approach

### 3.5 Constraint Monitoring and Control

#### 3.5.1 Performance Measurement

- **Schedule Performance Index (SPI):** Monitored weekly
- **Cost Performance Index (CPI):** Monitored bi-weekly
- **Scope Completion:** Measured against WBS deliverables

#### 3.5.2 Reporting Requirements

- Weekly status reports on schedule adherence
- Bi-weekly budget variance reports
- Monthly comprehensive project status reports

#### 3.5.3 Interdependency of Time, Scope, and Cost

The three constraints are highly interdependent:

- **Time constraints** limit the extent of features that can be implemented and tested
- **Scope constraints** ensure focus on core system functionality and prevent schedule overruns
- **Cost constraints** influence technology choices and limit infrastructure capabilities

#### 3.5.4 Corrective Actions

When constraints are at risk of being violated:

1. Identify root cause of variance
2. Develop corrective action plan
3. Assess impact on other constraints
4. Obtain necessary approvals
5. Implement corrections and monitor results
6. Document lessons learned

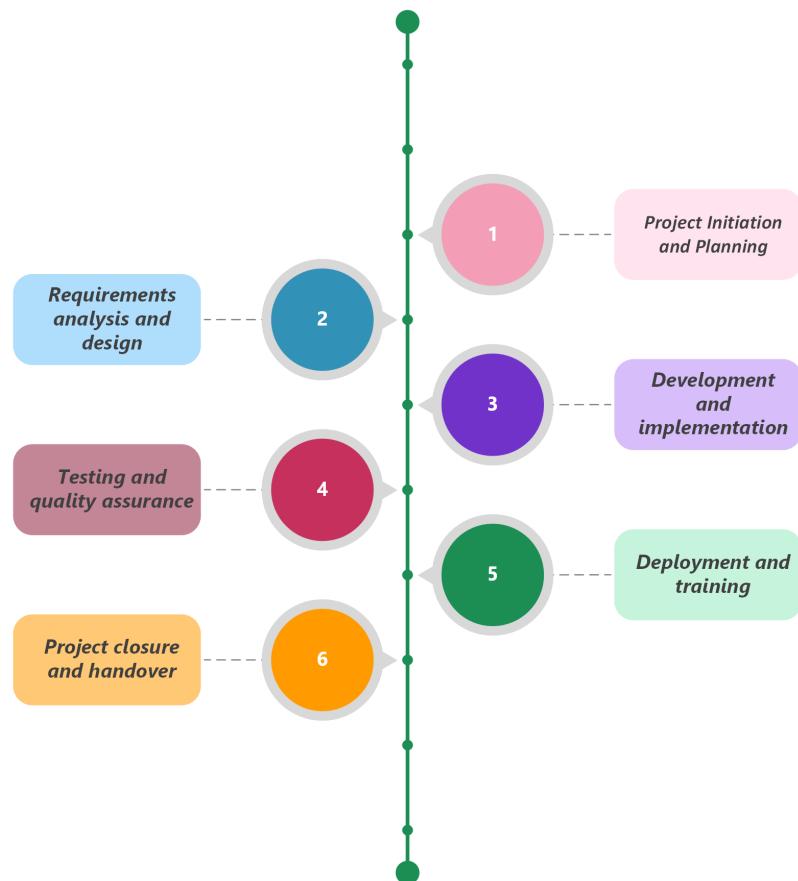
# Chapter 4

## Project Phases

This chapter describes the major phases of the Online Course Registration Portal project, outlining the key activities, deliverables, and milestones for each phase. The project follows a structured software development lifecycle approach.

### 4.1 Project Lifecycle Overview

The project is organized into six major phases, each with specific objectives, activities, and deliverables. The phases follow a sequential approach with some overlapping activities where appropriate.



**Figure 4.1:** Project Phases and Major Milestones Timeline

### 4.2 Phase 1: Project Initiation and Planning

#### 4.2.1 Duration

4 weeks (Weeks 1-4)

#### **4.2.2 Objectives**

- Establish project foundation and authorization
- Define project scope, objectives, and constraints
- Identify stakeholders and establish communication plans
- Develop project management plan
- Conduct feasibility analysis

#### **4.2.3 Key Activities**

1. Develop business case and justify project investment
2. Develop and approve project charter
3. Conduct stakeholder analysis and identification
4. Define project scope and requirements at high level
5. Establish project team and assign roles
6. Create Work Breakdown Structure (WBS)
7. Develop project schedule and timeline
8. Prepare budget and resource allocation plan
9. Identify initial risks and mitigation strategies
10. Establish project governance and communication protocols
11. Set up project infrastructure (tools, repositories, etc.)

#### **4.2.4 Deliverables**

- Business Case
- Approved Project Charter
- Project Management Plan
- Stakeholder Register
- Work Breakdown Structure (WBS)
- Project Schedule (Gantt Chart)
- Risk Register
- Communication Plan
- Resource Management Plan

#### 4.2.5 Milestone

**M1: Project Planning Approved** - Project management plan approved by sponsor

### 4.3 Phase 2: Requirements Analysis and Design

#### 4.3.1 Duration

4 weeks (Weeks 5-8)

#### 4.3.2 Objectives

- Gather and document detailed system requirements
- Design system architecture and technical solution
- Create database schema and data models
- Develop user interface prototypes
- Finalize technical specifications

#### 4.3.3 Key Activities

1. Conduct requirements gathering sessions with stakeholders
2. Document functional and non-functional requirements
3. Analyze prerequisite rules and business logic
4. Design system architecture and component interactions
5. Create database design and entity-relationship diagrams
6. Develop user interface mockups and prototypes
7. Design API specifications for system integration
8. Define security and access control mechanisms
9. Create detailed technical specifications
10. Conduct design reviews and obtain approvals
11. Prepare test plans and test cases

#### 4.3.4 Deliverables

- Requirements Specification Document
- System Architecture Document
- Database Design Document

- UI/UX Design Mockups and Prototypes
- API Specification Document
- Security Design Document
- Technical Specification Document
- Test Plan and Test Cases

#### 4.3.5 Milestones

- **M2: Requirements Approved** - Requirements specification signed off
- **M3: Design Approved** - System design approved by technical team

### 4.4 Phase 3: Development and Implementation

#### 4.4.1 Duration

12 weeks (Weeks 9-20)

#### 4.4.2 Objectives

- Implement system based on approved designs
- Develop all core functionality and features
- Integrate with existing university systems
- Conduct unit testing and code reviews
- Prepare for system testing

#### 4.4.3 Key Activities

1. Set up development environment and version control
2. Implement database schema and data access layer
3. Develop backend services and business logic
  - Prerequisite validation engine
  - Conflict detection algorithm
  - Waitlist management system
  - Course registration workflow
4. Develop frontend user interface
  - Student portal
  - Administrator dashboard

- Course catalog and search
  - Timetable builder
5. Implement system integrations
    - Student information system integration
    - Authentication system integration
    - Email notification system
  6. Conduct code reviews and quality checks
  7. Perform unit testing for all components
  8. Document code and system functionality
  9. Prepare deployment scripts and procedures

#### 4.4.4 Deliverables

- Fully Functional Web Application
- Source Code with Documentation
- Database with Test Data
- API Integration Layer
- Unit Test Results
- Code Review Reports
- Deployment Scripts and Procedures
- Developer Documentation

#### 4.4.5 Milestones

- **M4: Backend Development Complete** - Core services implemented
- **M5: Frontend Development Complete** - User interface implemented
- **M6: Integration Complete** - All system integrations functional
- **M7: Development Phase Complete** - System ready for testing

### 4.5 Phase 4: Testing and Quality Assurance

#### 4.5.1 Duration

6 weeks (Weeks 21-26)

#### 4.5.2 Objectives

- Verify system meets all requirements
- Identify and fix defects
- Validate system performance and scalability
- Ensure security and data integrity
- Obtain user acceptance

#### 4.5.3 Key Activities

1. Conduct system integration testing
2. Perform functional testing of all features
3. Execute performance and load testing
4. Conduct security testing and vulnerability assessment
5. Test prerequisite validation rules
6. Test concurrent registration scenarios
7. Validate conflict detection accuracy
8. Test waitlist automation
9. Conduct usability testing with sample users
10. Perform user acceptance testing (UAT) with stakeholders
11. Document and track defects
12. Fix critical and high-priority bugs
13. Retest fixed defects
14. Conduct regression testing
15. Prepare test summary reports

#### 4.5.4 Deliverables

- Test Execution Results
- Defect Reports and Resolution Log
- Performance Test Results
- Security Assessment Report
- User Acceptance Testing Sign-off
- Test Summary Report
- Updated System Documentation

#### 4.5.5 Milestones

- **M8: System Testing Complete** - All functional tests passed
- **M9: UAT Approved** - User acceptance testing successfully completed

### 4.6 Phase 5: Deployment and Training

#### 4.6.1 Duration

7 weeks (Weeks 27-33)

#### 4.6.2 Objectives

- Deploy system to production environment
- Train administrators and support staff
- Prepare students for system usage
- Establish support and maintenance procedures
- Ensure smooth system cutover

#### 4.6.3 Key Activities

1. Prepare production environment
2. Migrate data from legacy systems (if applicable)
3. Deploy application to production servers
4. Configure production settings and integrations
5. Conduct production verification testing
6. Develop training materials
  - Administrator training manuals
  - Student user guides
  - Video tutorials
  - Quick reference guides
7. Conduct administrator training sessions
8. Set up help desk and support procedures
9. Establish monitoring and alerting systems
10. Communicate go-live plans to all stakeholders
11. Prepare rollback procedures
12. Execute go-live activities

#### 4.6.4 Deliverables

- Production System Deployment
- Administrator Training Materials
- Student User Guides
- Video Tutorial Library
- Help Desk Procedures
- System Monitoring Dashboard
- Deployment Documentation
- Training Completion Reports

#### 4.6.5 Milestones

- **M10: System Deployed** - Production deployment successful
- **M11: Training Complete** - All users trained
- **M12: System Go-Live** - System operational for all users

### 4.7 Phase 6: Project Closure and Handover

#### 4.7.1 Duration

3 weeks (Weeks 34-36)

#### 4.7.2 Objectives

- Verify all project objectives achieved
- Document lessons learned
- Transition system to operations team
- Close project contracts and financials

#### 4.7.3 Key Activities

1. Monitor first registration period
2. Address post-deployment issues
3. Collect user feedback and satisfaction data
4. Conduct project performance review
5. Document lessons learned

6. Prepare final project report
7. Archive project documentation
8. Transfer knowledge to operations team
9. Close contracts with vendors
10. Release project resources
11. Conduct project closure meeting
12. Obtain final project sign-off

#### **4.7.4 Deliverables**

- Final Project Report
- Lessons Learned Document
- Post-Implementation Review Report
- Complete System Documentation
- Operations Handover Package
- Project Closure Report
- Final Financial Report

#### **4.7.5 Milestone**

**M13: Project Closed** - Project formally closed and accepted

### **4.8 Phase Dependencies and Relationships**

**Table 4.1:** Phase Dependencies

| <b>Phase</b>          | <b>Predecessor</b> | <b>Key Dependencies</b>                      |
|-----------------------|--------------------|--|
| Phase 1: Initiation   | None               | Project authorization, resource availability |
| Phase 2: Requirements | Phase 1            | Approved charter, stakeholder availability   |
| Phase 3: Development  | Phase 2            | Approved design, development resources       |
| Phase 4: Testing      | Phase 3            | Completed development, test environment      |
| Phase 5: Deployment   | Phase 4            | UAT approval, production environment         |
| Phase 6: Closure      | Phase 5            | Successful go-live, registration period      |

## 4.9 Phase Governance

### Phase Exits

At the end of each phase, a formal phase exit/kill point is conducted to:

- Verify phase objectives achieved
- Review deliverables for completeness and quality
- Assess risks and issues
- Evaluate schedule and budget performance
- Approve or reject progression to next phase
- Ensure that each phase gate requires meeting the following criteria:
  - All planned deliverables completed and approved
  - Quality standards met
  - No critical defects or unresolved issues
  - Budget variance within acceptable limits
  - Risks at acceptable levels
  - Stakeholder approval obtained

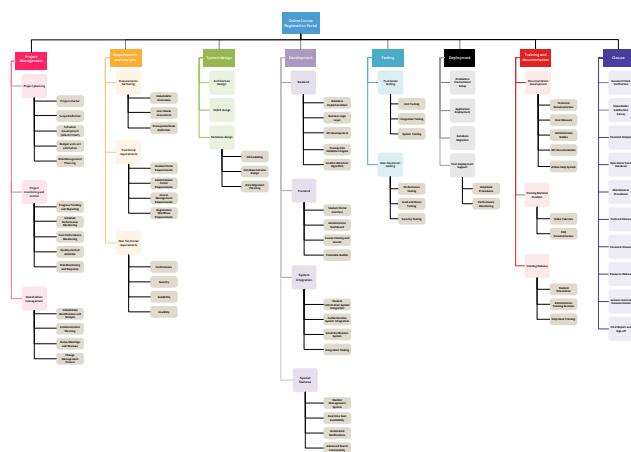
## Chapter 5

# Work Breakdown Structure (WBS)

### 5.1 WBS Overview

The Work Breakdown Structure organizes project deliverables and work into smaller, more manageable components. Each level provides increasing detail about the work required to complete the project successfully.

### 5.2 WBS Visual Representation



**Figure 5.1:** Work Breakdown Structure - Visual Hierarchy

### 5.3 WBS Dictionary Sample

The following tables provide WBS dictionary entries for selected work packages across different project phases:

**Table 5.1:** WBS Dictionary - Project Charter Development (1.1.1.1)

|                     |  |
|---------------------|--|
| <b>WBS Code</b>     | 1.1.1.1  |
| <b>Name</b>         | Project Charter Development  |
| <b>Description</b>  | Create the formal project charter document that authorizes the project and provides the project manager with authority to apply organizational resources |
| <b>Responsible</b>  | Project Manager  |
| <b>Duration</b>     | 3 days   |
| <b>Resources</b>    | Project Manager, Sponsor, Business Analyst   |
| <b>Dependencies</b> | None (project start activity)  |
| <b>Deliverables</b> | Signed project charter, stakeholder register, high-level requirements  |

**Table 5.2:** WBS Dictionary - Stakeholder Interviews (1.2.1.1)

|                     |  |
|---------------------|--|
| <b>WBS Code</b>     | 1.2.1.1  |
| <b>Name</b>         | Stakeholder Interviews   |
| <b>Description</b>  | Conduct structured interviews with key stakeholders to gather requirements, understand pain points, and identify system expectations |
| <b>Responsible</b>  | Business Analyst   |
| <b>Duration</b>     | 1 week   |
| <b>Resources</b>    | 2 Business Analysts, Stakeholder time  |
| <b>Dependencies</b> | Project charter approval (1.1.1.1), Stakeholder identification (1.1.3.1)   |
| <b>Deliverables</b> | Interview transcripts, requirements notes, stakeholder priorities document   |

**Table 5.3:** WBS Dictionary - Database Schema Design (1.3.2.2)

|                     |  |
|---------------------|--|
| <b>WBS Code</b>     | 1.3.2.2  |
| <b>Name</b>         | Database Schema Design   |
| <b>Description</b>  | Design the complete database schema including tables, relationships, constraints, indexes, and stored procedures for the registration system |
| <b>Responsible</b>  | Database Administrator   |
| <b>Duration</b>     | 1 week   |
| <b>Resources</b>    | Database Administrator, Data Architect, System Architect   |
| <b>Dependencies</b> | ER modeling (1.3.2.1), Requirements documentation (1.2.2, 1.2.3)   |
| <b>Deliverables</b> | Database schema document, table definitions, relationship diagrams, normalization documentation  |

**Table 5.4:** WBS Dictionary - Prerequisite Validation Engine (1.4.1.4)

|                     |  |
|---------------------|--|
| <b>WBS Code</b>     | 1.4.1.4  |
| <b>Name</b>         | Prerequisite Validation Engine   |
| <b>Description</b>  | Develop the core engine that validates student prerequisite requirements before allowing course registration |
| <b>Responsible</b>  | Backend Development Team   |
| <b>Duration</b>     | 2 weeks  |
| <b>Resources</b>    | 2 Senior Developers, 1 QA Engineer   |
| <b>Dependencies</b> | Prerequisite rules definition (1.2.1.5), Database implementation (1.4.1.1)                                   |
| <b>Deliverables</b> | Functional prerequisite validation module, unit tests, documentation   |

**Table 5.5:** WBS Dictionary - Student Portal Interface (1.4.2.1)

|                     |   |
|---------------------|---|
| <b>WBS Code</b>     | 1.4.2.1   |
| <b>Name</b>         | Student Portal Interface  |
| <b>Description</b>  | Develop the main student-facing interface for browsing courses, checking prerequisites, and registering for courses |
| <b>Responsible</b>  | Frontend Development Team   |
| <b>Duration</b>     | 3 weeks   |
| <b>Resources</b>    | 2 Frontend Developers, 1 UI/UX Designer   |
| <b>Dependencies</b> | UI design approval (1.3.3), API development (1.4.1.3)   |
| <b>Deliverables</b> | Responsive student portal, course search interface, registration workflow screens                                   |

**Table 5.6:** WBS Dictionary - System Testing (1.5.2.3)

|                     |  |
|---------------------|--|
| <b>WBS Code</b>     | 1.5.2.3  |
| <b>Name</b>         | System Testing   |
| <b>Description</b>  | Comprehensive end-to-end testing of the entire system to verify all components work together correctly and meet specified requirements |
| <b>Responsible</b>  | QA Team Lead   |
| <b>Duration</b>     | 2 weeks  |
| <b>Resources</b>    | 3 QA Engineers, Test Manager   |
| <b>Dependencies</b> | Integration testing complete (1.5.2.2), All features developed (1.4)   |
| <b>Deliverables</b> | System test plan, test cases, test results report, defect log  |

**Table 5.7:** WBS Dictionary - Application Deployment (1.6.2)

|                     |   |
|---------------------|---|
| <b>WBS Code</b>     | 1.6.2   |
| <b>Name</b>         | Application Deployment  |
| <b>Description</b>  | Deploy the registration system application to production environment including all configuration and verification steps |
| <b>Responsible</b>  | DevOps Team   |
| <b>Duration</b>     | 1 week  |
| <b>Resources</b>    | DevOps Engineer, System Administrator, DBA  |
| <b>Dependencies</b> | Production environment setup (1.6.1), UAT approval, Database migration (1.6.3)  |
| <b>Deliverables</b> | Deployed application, deployment checklist, rollback procedures, deployment report                                      |

**Table 5.8:** WBS Dictionary - Administrator Training Sessions (1.7.3.1)

|                     |   |
|---------------------|---|
| <b>WBS Code</b>     | 1.7.3.1   |
| <b>Name</b>         | Administrator Training Sessions   |
| <b>Description</b>  | Conduct comprehensive training sessions for system administrators covering system configuration, user management, and troubleshooting |
| <b>Responsible</b>  | Training Coordinator  |
| <b>Duration</b>     | 1 week  |
| <b>Resources</b>    | 2 Trainers, Technical Writer, Training Facility   |
| <b>Dependencies</b> | Training materials (1.7.2), System deployed (1.6.2)   |
| <b>Deliverables</b> | Training attendance records, administrator certifications, training feedback surveys  |

**Table 5.9:** WBS Dictionary - Final Report and Sign-off (1.8.10)

|                     |   |
|---------------------|---|
| <b>WBS Code</b>     | 1.8.10  |
| <b>Name</b>         | Final Report and Sign-off   |
| <b>Description</b>  | Prepare comprehensive final project report and obtain formal sign-off from project sponsor and key stakeholders |
| <b>Responsible</b>  | Project Manager   |
| <b>Duration</b>     | 4 days  |
| <b>Resources</b>    | Project Manager, Technical Writer   |
| <b>Dependencies</b> | All deliverables complete, Lessons learned (1.8.9), Financial closure (1.8.7)                                   |
| <b>Deliverables</b> | Final project report, formal acceptance document, signed project closure certificate                            |

# Chapter 6

## Gantt Chart and Milestones

This chapter presents the project schedule in the form of a Gantt chart, showing all project activities, their durations, dependencies, and relationships. Additionally, all project milestones are identified and justified.

### 6.1 Project Schedule Overview

The project schedule spans 28 weeks from initiation to closure. The Gantt chart provides a visual timeline of all activities, showing start and end dates, durations, and dependencies between tasks.

### 6.2 Gantt Chart

|    | (i) | Task ID | Task Name                          | Duration | Start       | Finish      |
|----|-----|---------|------------------------------------|----------|-------------|-------------|
| 1  |     | 1.1     | Project Planning and Initiation    | 3 wks    | Sun 3/1/26  | Thu 3/19/26 |
| 2  |     | 1.1.1   | Project planning                   | 2 wks    | Sun 3/1/26  | Thu 3/12/26 |
| 3  |     | 1.1.1.1 | Project Charter Development        | 3 days   | Sun 3/1/26  | Tue 3/3/26  |
| 4  |     | 1.1.1.2 | Scope Definition and WBS Creation  | 3 days   | Wed 3/4/26  | Fri 3/6/26  |
| 5  |     | 1.1.1.3 | Schedule development (gantt chart) | 3 days   | Mon 3/9/26  | Wed 3/11/26 |
| 6  |     | 1.1.1.4 | Budget cost estimation             | 2 days   | Mon 3/9/26  | Tue 3/10/26 |
| 7  |     | 1.1.1.5 | Risk Management Planning           | 2 days   | Wed 3/11/26 | Thu 3/12/26 |
| 8  |     | M1      | Project Planning Approved          | 0 days   | Thu 3/12/26 | Thu 3/12/26 |
| 9  |     | 1.1.2   | Stakeholder Management             | 1 wk     | Mon 3/16/26 | Fri 3/20/26 |
| 10 |     | 1.1.2.1 | Stakeholder Identification         | 2 days   | Mon 3/16/26 | Tue 3/17/26 |
| 11 |     | 1.1.2.2 | Communication Planning             | 2 days   | Wed 3/18/26 | Thu 3/19/26 |
| 12 |     | 1.1.2.3 | Status Meetings and reviews        | 1 day    | Fri 3/20/26 | Fri 3/20/26 |
| 13 |     | 1.1.2.4 | Change Management Process          | 2 days   | Thu 3/19/26 | Fri 3/20/26 |
| 14 |     | 1.2     | Requirements and analysis          | 3 wks    | Mon 3/16/26 | Fri 4/3/26  |
| 15 |     | 1.2.1   | Requirements gathering             | 1 wk     | Mon 3/16/26 | Fri 3/20/26 |
| 16 |     | 1.2.1.1 | Stakeholder Interview              | 3 days   | Mon 3/16/26 | Wed 3/18/26 |
| 17 |     | 1.2.1.2 | User needs assessments             | 2 days   | Mon 3/16/26 | Tue 3/17/26 |
| 18 |     | 1.2.1.3 | Prerequisite rules definition      | 2 days   | Thu 3/19/26 | Fri 3/20/26 |
| 19 |     | 1.2.2   | Functional requirements            | 1 wk     | Mon 3/23/26 | Fri 3/27/26 |
| 20 |     | 1.2.2.1 | Student Portal Requirements        | 2 days   | Mon 3/23/26 | Tue 3/24/26 |
| 21 |     | 1.2.2.2 | Administrator Portal Requirements  | 2 days   | Mon 3/23/26 | Tue 3/24/26 |
| 22 |     | 1.2.2.3 | Course Management Requirements     | 2 days   | Wed 3/25/26 | Thu 3/26/26 |

Figure 6.1: Project Gantt Chart - tasks

|    |  |         |                                |         |             |             |
|----|--|---------|--------------------------------|---------|-------------|-------------|
|    |  |         |                                |         |             |             |
| 23 |  | 1.2.2.4 | Registration Workflow Requirer | 3 days  | Wed 3/25/26 | Fri 3/27/26 |
| 24 |  | 1.2.3   | Non-functional Requirements    | 2 wks   | Mon 3/23/26 | Fri 4/3/26  |
| 25 |  | 1.2.3.1 | Performance Requirements       | 6 days  | Mon 3/23/26 | Sat 3/28/26 |
| 26 |  | 1.2.3.2 | Security Requirements          | 5 days  | Wed 3/25/26 | Tue 3/31/26 |
| 27 |  | 1.2.3.3 | Scalability Requirements       | 6 days  | Fri 3/27/26 | Fri 4/3/26  |
| 28 |  | 1.2.3.4 | Usability Requirements         | 6 days  | Fri 3/27/26 | Fri 4/3/26  |
| 29 |  | M2      | Requirements Approved          | 0 days  | Fri 4/3/26  | Fri 4/3/26  |
| 30 |  | 1.3     | System Design                  | 3 wks   | Mon 3/30/26 | Fri 4/17/26 |
| 31 |  | 1.3.1   | Architecture Design            | 2 wks   | Mon 3/30/26 | Fri 4/10/26 |
| 32 |  | 1.3.2   | Database Design                | 1 wk    | Mon 3/30/26 | Fri 4/3/26  |
| 33 |  | 1.3.2.1 | ER Modelling                   | 2 days  | Mon 3/30/26 | Tue 3/31/26 |
| 34 |  | 1.3.2.2 | Database schema design         | 2 days  | Wed 4/1/26  | Thu 4/2/26  |
| 35 |  | 1.3.2.3 | Data migration planning        | 1 day   | Fri 4/3/26  | Fri 4/3/26  |
| 36 |  | 1.3.3   | User Interface Design          | 1.2 wks | Fri 4/10/26 | Fri 4/17/26 |
| 37 |  | M3      | Design approved                | 0 days  | Fri 4/17/26 | Fri 4/17/26 |
| 38 |  | 1.4     | Development                    | 10 wks  | Tue 4/7/26  | Mon 6/15/26 |
| 39 |  | 1.4.1   | Backend                        | 8 wks   | Tue 4/7/26  | Mon 6/1/26  |
| 40 |  | 1.4.1.1 | Database implementation        | 2 wks   | Tue 4/7/26  | Mon 4/20/26 |
| 41 |  | 1.4.1.2 | Business logic layer           | 3.8 wks | Mon 4/20/26 | Thu 5/14/26 |
| 42 |  | 1.4.1.3 | API development                | 2.4 wks | Fri 5/15/26 | Mon 6/1/26  |
| 43 |  | 1.4.1.4 | Prerequisite validation engine | 2.8 wks | Mon 4/27/26 | Thu 5/14/26 |

**Figure 6.2:** Project Gantt Chart - tasks - continued

|    |  |         |                                     |         |             |             |
|----|--|---------|-------------------------------------|---------|-------------|-------------|
| 44 |  | 1.4.1.5 | Conflict detection algorithm        | 2.6 wks | Mon 5/4/26  | Wed 5/20/26 |
| 45 |  | M4      | Backend complete                    | 0 days  | Mon 6/1/26  | Mon 6/1/26  |
| 46 |  | 1.4.2   | Frontend                            | 5.8 wks | Tue 4/14/26 | Fri 5/22/26 |
| 47 |  | 1.4.2.1 | Student portal interface            | 2 wks   | Tue 4/14/26 | Mon 4/27/26 |
| 48 |  | 1.4.2.2 | Administrator Dashboard             | 2 wks   | Tue 4/14/26 | Mon 4/27/26 |
| 49 |  | 1.4.2.3 | Course catalog and search           | 2 wks   | Mon 4/27/26 | Fri 5/8/26  |
| 50 |  | 1.4.2.4 | Timetable builder                   | 2 wks   | Mon 5/11/26 | Fri 5/22/26 |
| 51 |  | M5      | Frontend complete                   | 0 days  | Fri 5/22/26 | Fri 5/22/26 |
| 52 |  | 1.4.3   | System integration                  | 2 wks   | Mon 6/1/26  | Fri 6/12/26 |
| 53 |  | 1.4.3.1 | Student information system integrat | 1 wk    | Mon 6/1/26  | Fri 6/5/26  |
| 54 |  | 1.4.3.2 | Authentication system integrati     | 1 wk    | Mon 6/1/26  | Fri 6/5/26  |
| 55 |  | 1.4.3.3 | Email notification system           | 1 wk    | Mon 6/1/26  | Fri 6/5/26  |
| 56 |  | 1.4.3.4 | Integration testing                 | 1 wk    | Mon 6/8/26  | Fri 6/12/26 |
| 57 |  | M6      | Integration complete                | 0 days  | Fri 6/12/26 | Fri 6/12/26 |
| 58 |  | 1.4.4   | Special features                    | 3 wks   | Mon 5/11/26 | Fri 5/29/26 |
| 59 |  | 1.4.4.1 | Waitlist management system          | 1 wk    | Mon 5/11/26 | Fri 5/15/26 |
| 60 |  | 1.4.4.2 | Real-time seat availability         | 1 wk    | Mon 5/11/26 | Fri 5/15/26 |
| 61 |  | 1.4.4.3 | Automated Notifications             | 1 wk    | Mon 5/18/26 | Fri 5/22/26 |
| 62 |  | 1.4.4.4 | Advanced search functionality       | 1 wk    | Mon 5/25/26 | Fri 5/29/26 |
| 63 |  | M7      | Development complete                | 0 days  | Mon 6/15/26 | Mon 6/15/26 |
| 64 |  | 1.5     | Testing                             | 5 wks   | Mon 6/1/26  | Fri 7/3/26  |

**Figure 6.3:** Project Gantt Chart - tasks - continued

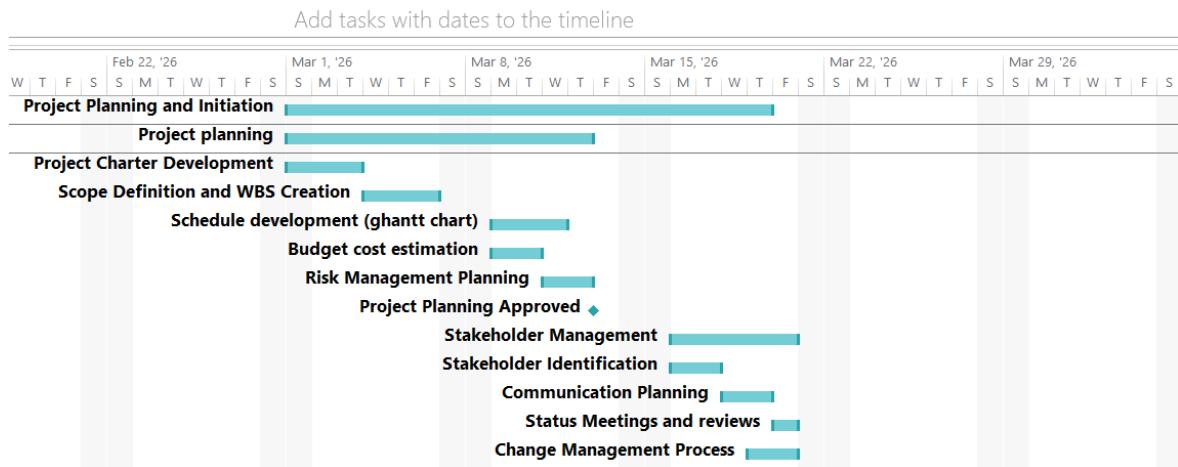
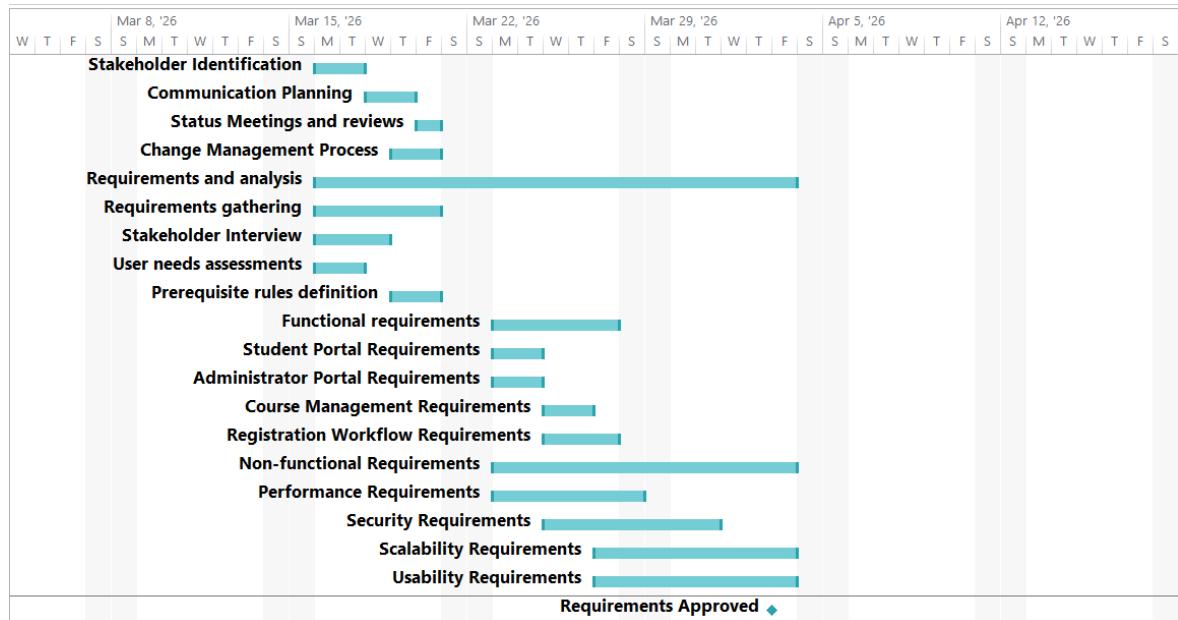
|    |  |         |                              |        |             |             |
|----|--|---------|------------------------------|--------|-------------|-------------|
| 65 |  | 1.5.1   | Functional testing           | 3 wks  | Mon 6/1/26  | Fri 6/19/26 |
| 66 |  | 1.5.1.1 | Unit testing                 | 1 wk   | Mon 6/1/26  | Fri 6/5/26  |
| 67 |  | 1.5.1.2 | Integration testing          | 1 wk   | Mon 6/8/26  | Fri 6/12/26 |
| 68 |  | 1.5.1.3 | System testing               | 2 wks  | Mon 6/15/26 | Fri 6/26/26 |
| 69 |  | 1.5.2   | Non-functional testing       | 2 wks  | Mon 6/22/26 | Fri 7/3/26  |
| 70 |  | 1.5.2.1 | Performance testing          | 1 wk   | Mon 6/22/26 | Fri 6/26/26 |
| 71 |  | 1.5.2.2 | Load and stress testing      | 1 wk   | Mon 6/22/26 | Fri 6/26/26 |
| 72 |  | 1.5.2.3 | Security testing             | 1 wk   | Mon 6/29/26 | Fri 7/3/26  |
| 73 |  | 1.5.2.4 | User acceptance testing      | 1 wk   | Mon 6/29/26 | Fri 7/3/26  |
| 74 |  | M8      | System testing complete      | 0 days | Fri 6/26/26 | Fri 6/26/26 |
| 75 |  | M9      | UAT approved                 | 0 days | Fri 7/3/26  | Fri 7/3/26  |
| 76 |  | 1.6     | Deployment                   | 6 wks  | Mon 7/6/26  | Fri 8/14/26 |
| 77 |  | 1.6.1   | Production environment setup | 1 wk   | Mon 7/6/26  | Fri 7/10/26 |
| 78 |  | 1.6.2   | Application Deployment       | 1 wk   | Mon 7/13/26 | Fri 7/17/26 |
| 79 |  | 1.6.3   | Database Migration           | 1 wk   | Mon 7/13/26 | Fri 7/17/26 |
| 80 |  | M10     | System deployed              | 0 days | Fri 7/24/26 | Fri 7/24/26 |
| 81 |  | 1.6.4   | Post-deployment support      | 3 wks  | Mon 7/27/26 | Fri 8/14/26 |
| 82 |  | 1.6.4.1 | Help desk procedures         | 1 wk   | Mon 7/27/26 | Fri 7/31/26 |
| 83 |  | 1.6.4.2 | Performance monitoring       | 2 wks  | Mon 8/3/26  | Fri 8/14/26 |
| 84 |  | 1.7     | Training and documentation   | 7 wks  | Mon 6/22/26 | Fri 8/7/26  |
| 85 |  | 1.7.1   | Documentation development    | 3 wks  | Mon 6/22/26 | Fri 7/10/26 |
| 86 |  | 1.7.1.1 | Technical documentation      | 1 wk   | Mon 6/22/26 | Fri 6/26/26 |

**Figure 6.4:** Project Gantt Chart - tasks - continued

| GANTT CHART |         |                                 |         |             |             |  |
|-------------|---------|---------------------------------|---------|-------------|-------------|--|
| 87          | 1.7.1.2 | User manuals                    | 1 wk    | Mon 6/29/26 | Fri 7/3/26  |  |
| 88          | 1.7.1.3 | Administrator guides            | 1 wk    | Mon 6/29/26 | Fri 7/3/26  |  |
| 89          | 1.7.1.4 | API documentation               | 1 wk    | Mon 6/22/26 | Fri 6/26/26 |  |
| 90          | 1.7.1.5 | Online help system              | 1 wk    | Mon 7/6/26  | Fri 7/10/26 |  |
| 91          | 1.7.2   | Training material creation      | 2 wks   | Mon 7/6/26  | Fri 7/17/26 |  |
| 92          | 1.7.2.1 | Video tutorials                 | 1 wk    | Mon 7/6/26  | Fri 7/10/26 |  |
| 93          | 1.7.2.2 | FAQ documentation               | 1 wk    | Mon 7/13/26 | Fri 7/17/26 |  |
| 94          | 1.7.3   | Training Delivery               | 2 wks   | Mon 7/27/26 | Fri 8/7/26  |  |
| 95          | 1.7.3.1 | Administrator training session  | 1 wk    | Mon 7/27/26 | Fri 7/31/26 |  |
| 96          | 1.7.3.2 | Student orientation             | 1 wk    | Mon 7/27/26 | Fri 7/31/26 |  |
| 97          | 1.7.3.3 | Help desk training              | 1 wk    | Mon 8/3/26  | Fri 8/7/26  |  |
| 98          | M11     | Training complete               | 0 days  | Fri 8/7/26  | Fri 8/7/26  |  |
| 99          | M12     | System Go-live                  | 0 days  | Mon 8/10/26 | Mon 8/10/26 |  |
| 100         | 1.8     | Closure                         | 4.4 wks | Mon 8/17/26 | Tue 9/15/26 |  |
| 101         | 1.8.1   | Success criteria verification   | 2 days  | Mon 8/17/26 | Tue 8/18/26 |  |
| 102         | 1.8.2   | Stakeholder satisfaction survey | 2 days  | Wed 8/19/26 | Thu 8/20/26 |  |
| 103         | 1.8.3   | Financial analysis              | 2 days  | Wed 8/19/26 | Thu 8/20/26 |  |
| 104         | M13     | First registration complete     | 0 days  | Thu 8/20/26 | Thu 8/20/26 |  |
| 105         | 1.8.4   | Operations team handover        | 4 days  | Fri 8/21/26 | Wed 8/26/26 |  |
| 106         | 1.8.5   | Maintenance procedures          | 2 days  | Wed 8/26/26 | Thu 8/27/26 |  |
| 107         | 1.8.6   | Contract closure                | 2 days  | Fri 8/28/26 | Mon 8/31/26 |  |

**Figure 6.5:** Project Gantt Chart - tasks - continued

**Figure 6.6:** Project Gantt Chart - tasks

**Figure 6.7:** Project Gantt Chart - timeline**Figure 6.8:** Project Gantt Chart - timeline - continued

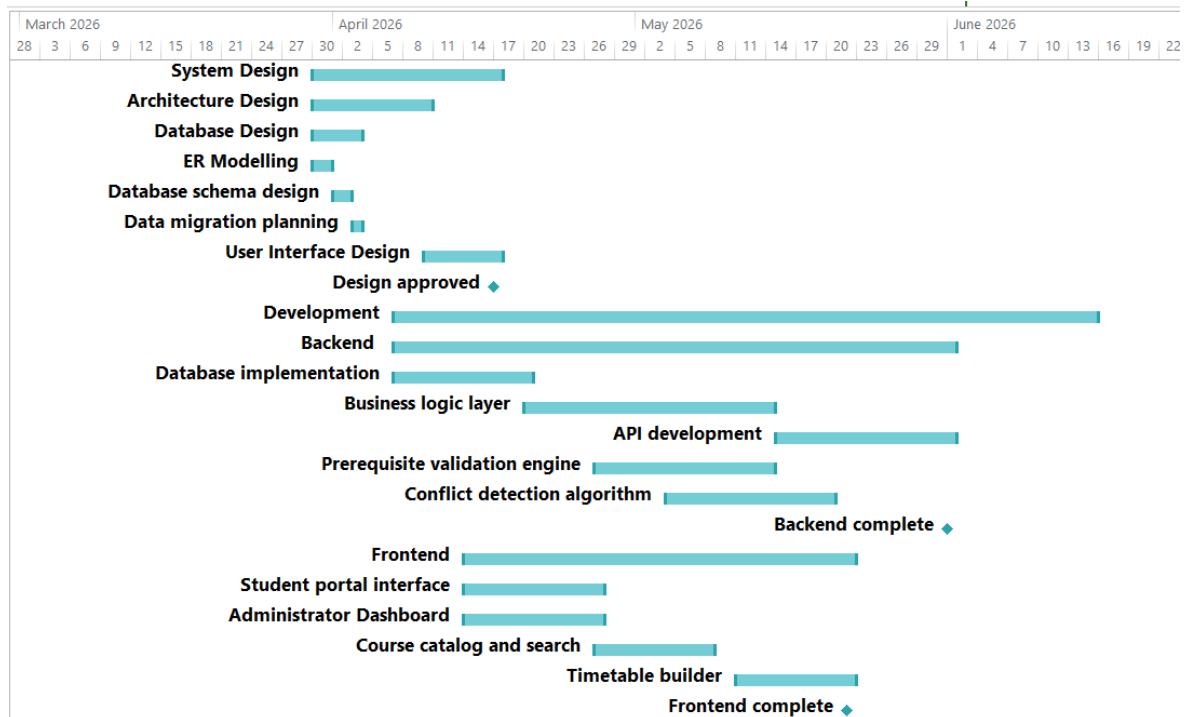


Figure 6.9: Project Gantt Chart - timeline - continued

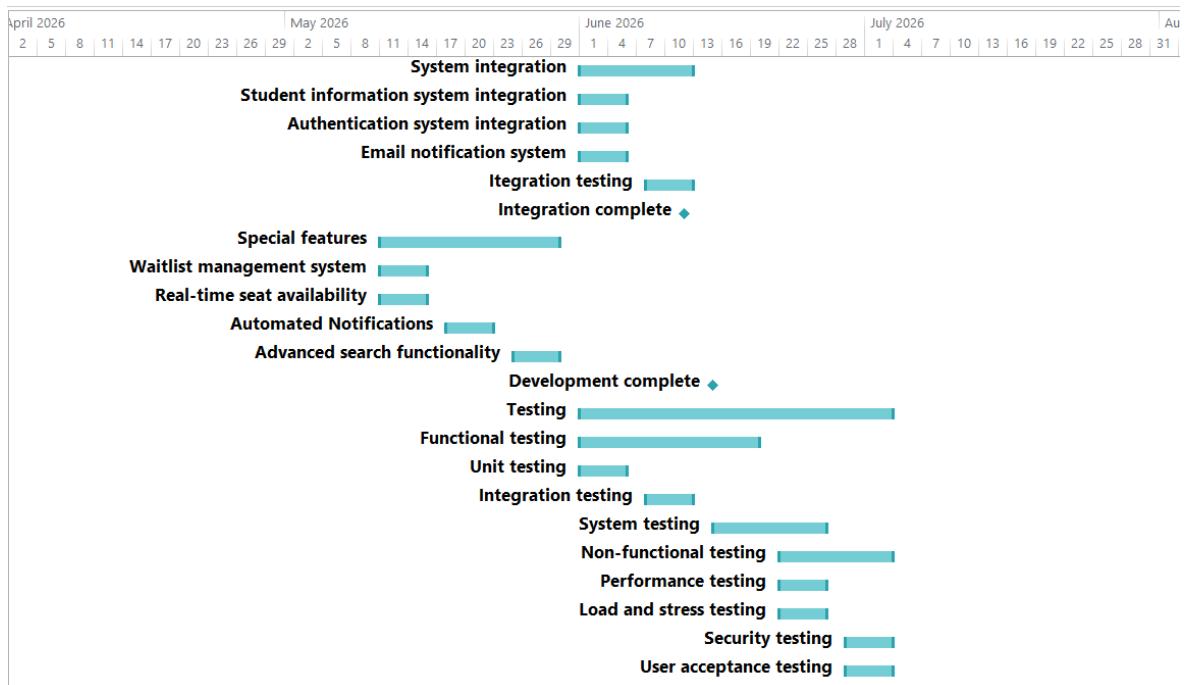


Figure 6.10: Project Gantt Chart - timeline - continued

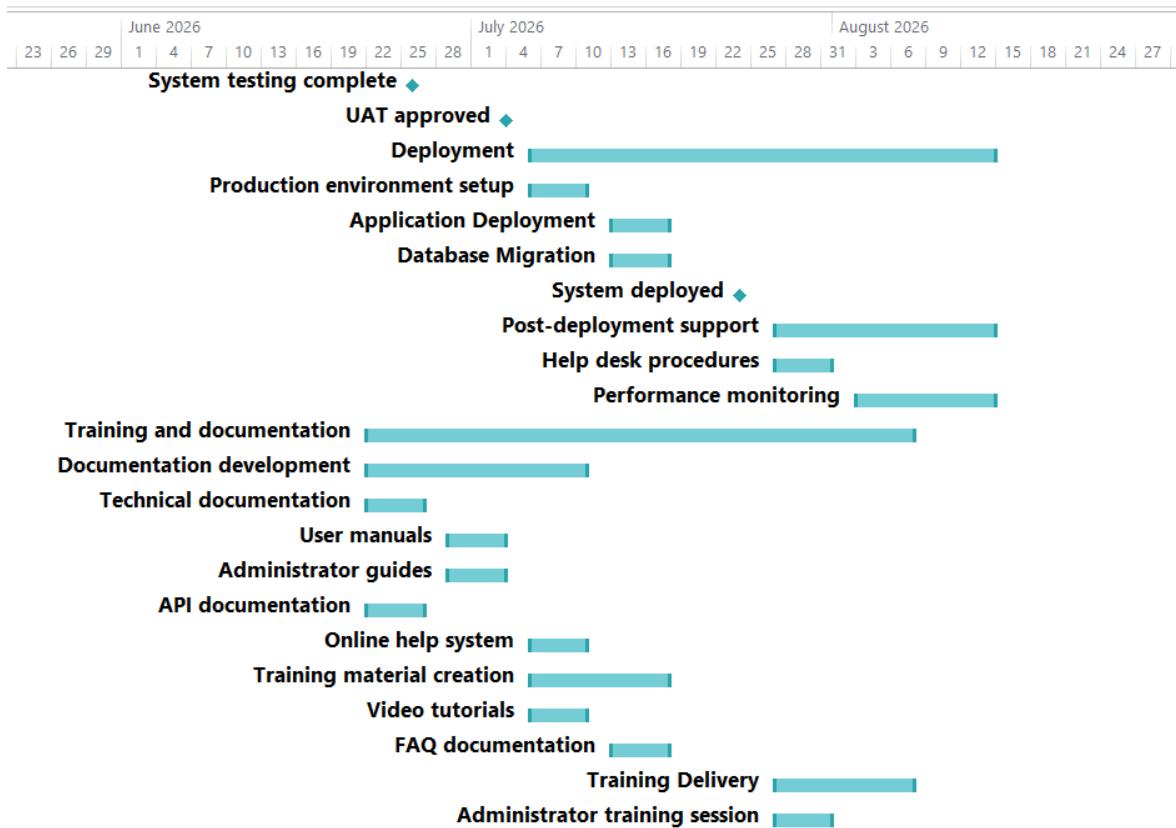


Figure 6.11: Project Gantt Chart - timeline - continued

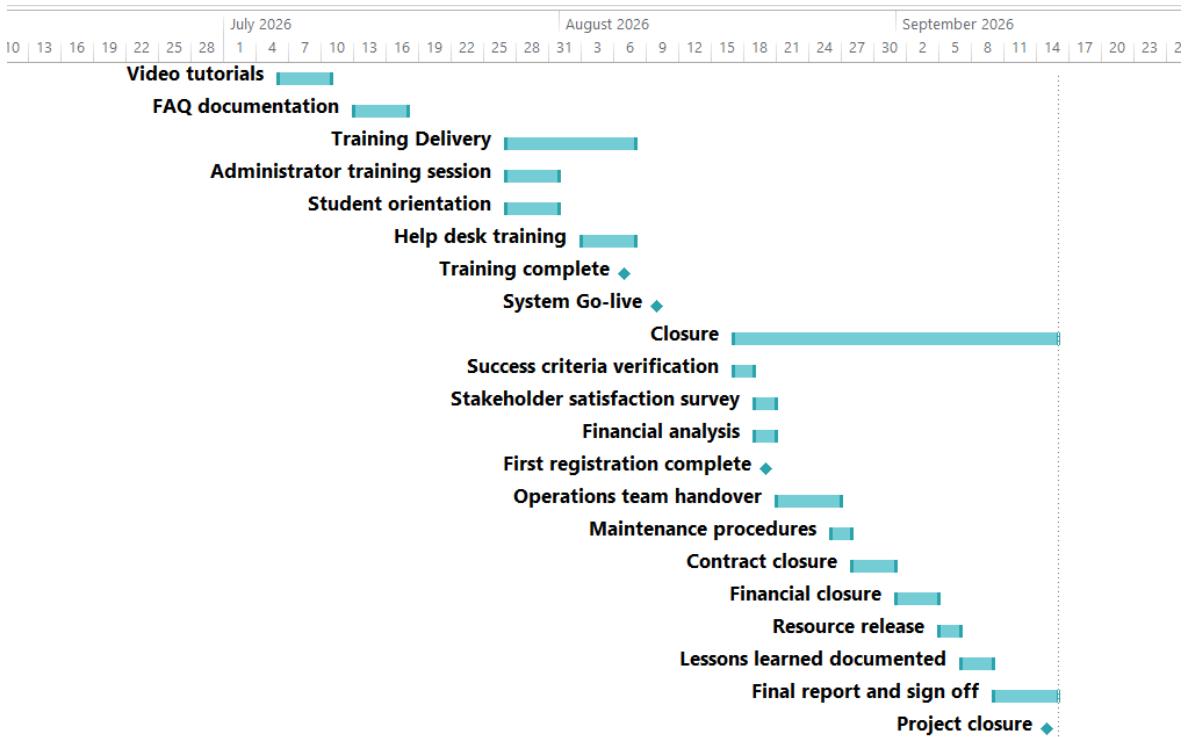


Figure 6.12: Project Gantt Chart - timeline - continued

[Click here to view the Project Gantt Chart as pdf exported from MS Project.](#)

### 6.3 Project Milestones

Milestones are significant events or decision points in the project that have zero duration but mark important achievements. The following table lists all project milestones with their justifications.

**Table 6.1:** Project Milestones and Justifications

| No. | Milestone                   | Target Date      | Justification   |
|-----|-----------------------------|------------------|---|
| M1  | Project Planning Approved   | Week 3 (Mar 15)  | Ensures all stakeholders agree on project scope, timeline, and budget before commencing work. |
| M2  | Requirements Approved       | Week 5 (Mar 29)  | Confirms all functional and non-functional requirements are documented and approved.          |
| M3  | Design Approved             | Week 6 (Apr 6)   | Validates system architecture and database design before development begins.                  |
| M4  | Backend Complete            | Week 11 (May 15) | Core backend services, API, and business logic implemented and ready for integration.         |
| M5  | Frontend Complete           | Week 12 (May 22) | User interface components complete including portals, dashboards, and timetable builder.      |
| M6  | Integration Complete        | Week 14 (Jun 12) | All system integrations functional including SIS, authentication, and notifications.          |
| M7  | Development Complete        | Week 14 (Jun 12) | All development work packages complete. Ready for formal testing phase.                       |
| M8  | System Testing Complete     | Week 18 (Jun 26) | Functional and non-functional testing passed. System meets quality standards.                 |
| M9  | UAT Approved                | Week 19 (Jul 10) | Stakeholder validation that system meets business needs. Authorization to deploy.             |
| M10 | System Deployed             | Week 21 (Jul 24) | Production deployment successful and verified. System operational in production.              |
| M11 | Training Complete           | Week 23 (Aug 7)  | All users trained including administrators, students, and help desk staff.                    |
| M12 | System Go-Live              | Week 23 (Aug 10) | System officially operational for all students. Ready for registration period.                |
| M13 | First Registration Complete | Week 25 (Aug 24) | Successful completion of first registration cycle. System validated under real load.          |
| M14 | Project Closed              | Week 28 (Sep 15) | All deliverables accepted, lessons learned documented, project formally closed.               |

## 6.4 Milestone Importance and Impact

### 6.4.1 Critical Milestones

The following milestones are critical to project success and have the highest priority:

1. **M3 - Design Approved:** Prevents costly rework during development by ensuring design correctness upfront
2. **M9 - UAT Approved:** Final validation before production deployment; failure requires major remediation
3. **M12 - System Go-Live:** Time-sensitive milestone aligned with academic calendar; delay impacts entire institution

### 6.4.2 Milestone Dependencies

Each milestone has specific dependencies and enables subsequent project work:

**Table 6.2:** Milestone Dependencies

| Milestone | Prerequisites                             | Enables                              |
|-----------|---|--------------------------------------|
| M1        | Stakeholder approval, resource allocation | Requirements gathering activities    |
| M2        | Complete requirements documentation       | Design activities, effort estimation |
| M3        | Approved requirements, design reviews     | Development work to commence         |
| M7        | All development work packages complete    | Formal testing phase                 |
| M9        | Successful testing, stakeholder UAT       | Production deployment authorization  |
| M12       | Deployment, training, verification        | Student registration activities      |

## 6.5 Key Activity Durations Summary

**Table 6.3:** Major Phase Durations

| Phase                            | Duration        | Start Week | End Week  |
|----------------------------------|-----------------|------------|-----------|
| Project Initiation and Planning  | 3 weeks         | 1          | 3         |
| Requirements Analysis and Design | 3 weeks         | 4          | 6         |
| Development and Implementation   | 10 weeks        | 7          | 16        |
| Testing and Quality Assurance    | 5 weeks         | 17         | 21        |
| Deployment and Training          | 5 weeks         | 22         | 26        |
| Project Closure and Handover     | 2 weeks         | 27         | 28        |
| <b>Total</b>                     | <b>28 weeks</b> | <b>1</b>   | <b>28</b> |

## 6.6 Schedule Assumptions and Constraints

### 6.6.1 Assumptions

- Resources available as planned throughout project duration
- No major holidays or institutional closures affecting schedule
- Stakeholder decisions made within agreed timeframes
- No significant scope changes after design approval
- Testing environment available when needed
- Production environment ready for deployment as scheduled

### 6.6.2 Constraints

- **Hard Deadline:** System must be operational before Week 30 (registration period)
- **Resource Constraints:** Limited number of developers and testers
- **External Dependencies:** Integration with existing systems requires coordination
- **Regulatory:** Must allow sufficient UAT time per university policies
- **Seasonal:** Summer months may have limited stakeholder availability

## 6.7 Schedule Risk Management

### 6.7.1 Schedule Risks

1. **Requirements Delay:** Late stakeholder approvals could delay design phase
2. **Integration Challenges:** Legacy system integration may take longer than estimated
3. **Resource Unavailability:** Key personnel absence could impact critical activities
4. **Testing Issues:** Discovery of major defects could extend testing phase

## 6.8 Schedule Control Procedures

### 6.8.1 Monitoring and Reporting

- Weekly schedule status updates
- Bi-weekly critical path analysis
- Monthly milestone tracking reports

### 6.8.2 Change Control

- All schedule changes require impact analysis
- Changes affecting milestones require sponsor approval
- Schedule baseline updates documented and communicated

## Chapter 7

# Activity on Node (AON) Network and Critical Path Analysis

This chapter presents the Activity-on-Node (AON) network diagram for the Online Course Registration Portal project. The AON technique is used to visualize project workflow, identify activity dependencies, and determine the critical path that defines the minimum project duration.

### 7.1 Introduction to AON and Critical Path Method

The Activity-on-Node (AON) network diagram represents project activities as nodes (boxes) and dependencies as arrows connecting them. This approach, also known as Precedence Diagramming Method (PDM), provides a clear visualization of:

- The sequence of project activities
- Dependencies and relationships between activities
- The critical path (longest path determining project duration)
- Activities with scheduling flexibility (slack)

#### 7.1.1 Critical Path Method (CPM)

The Critical Path Method uses forward and backward pass calculations to determine:

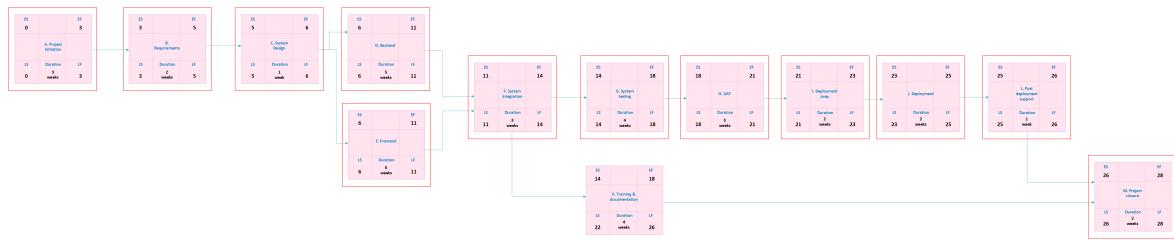
- **ES (Early Start):** Earliest time an activity can begin
- **EF (Early Finish):** Earliest time an activity can complete ( $ES + Duration$ )
- **LS (Late Start):** Latest time an activity can begin without delaying the project
- **LF (Late Finish):** Latest time an activity can complete without delaying the project
- **Total Slack:** Amount of time an activity can be delayed without delaying the project ( $LS - ES$ )
- **Free Slack:** Amount of time an activity can be delayed without delaying any successor ( $ES$  of successor -  $EF$ )

Activities with zero total slack form the **critical path** – any delay in these activities delays the entire project.

### 7.2 High-Level Activity List with CPM Calculations (ES, EF, LS, LF, Free Slack, Total Slack)

**Table 7.1:** High-Level Activity List with CPM Calculations

| ID | Activity Name (WBS)             | Duration | Pred. | ES | EF | LS | LF | Free Slack | Total Slack |
|----|---------------------------------|----------|-------|----|----|----|----|------------|-------------|
| A  | Project Initiation (1.1)        | 3 weeks  | -     | 0  | 3  | 0  | 3  | 0          | 0           |
| B  | Requirements Gathering (1.2)    | 2 weeks  | A     | 3  | 5  | 3  | 5  | 0          | 0           |
| C  | System Design (1.3)             | 1 week   | B     | 5  | 6  | 5  | 6  | 0          | 0           |
| D  | Backend Development (1.4.1)     | 5 weeks  | C     | 6  | 11 | 6  | 11 | 0          | 0           |
| E  | Frontend Development (1.4.2)    | 5 weeks  | C     | 6  | 11 | 6  | 11 | 0          | 0           |
| F  | System Integration (1.4.3)      | 3 weeks  | D, E  | 11 | 14 | 11 | 14 | 0          | 0           |
| G  | System Testing (1.5.2)          | 4 weeks  | F     | 14 | 18 | 14 | 18 | 0          | 0           |
| H  | User Acceptance Testing (1.5.3) | 3 weeks  | G     | 18 | 21 | 18 | 21 | 0          | 0           |
| I  | Deployment Preparation (1.6.1)  | 2 weeks  | H     | 21 | 23 | 21 | 23 | 0          | 0           |
| J  | System Deployment (1.6.2–1.6.3) | 2 weeks  | I     | 23 | 25 | 23 | 25 | 0          | 0           |
| K  | Training & Documentation (1.7)  | 4 weeks  | F     | 14 | 18 | 22 | 26 | 8 weeks    | 8 weeks     |
| L  | Post-Deployment Support (1.6.4) | 1 week   | J     | 25 | 26 | 25 | 26 | 0          | 0           |
| M  | Project Closure (1.8)           | 2 weeks  | L, K  | 26 | 28 | 26 | 28 | 0          | 0           |



**Figure 7.1:** Activity on Node (AON) Network Diagram - critical path activities surrounded by red boxes

### 7.3 Forward Pass Calculation (ES and EF)

The forward pass determines the earliest times activities can start and finish, working from project start to finish.

#### 7.3.1 Calculation Method

- **ES** = Maximum EF of all predecessor activities (or 0 for start activity)
- **EF** = ES + Duration

*Note: Activities with  $ES = LS$  and  $EF = LF$  have zero total slack and are on the critical path.*

#### 7.3.2 Forward Pass Results

1. **Activity A (Project Initiation):** ES = 0, EF =  $0 + 3 = 3$
2. **Activity B (Requirements):** ES = 3, EF =  $3 + 2 = 5$
3. **Activity C (Design):** ES = 5, EF =  $5 + 1 = 6$
4. **Activity D (Backend):** ES = 6, EF =  $6 + 5 = 11$
5. **Activity E (Frontend):** ES = 6, EF =  $6 + 5 = 11$  (must finish on time for F)
6. **Activity F (Integration):** ES =  $\max(11, 11) = 11$ , EF =  $11 + 3 = 14$
7. **Activity G (System Testing):** ES = 14, EF =  $14 + 4 = 18$
8. **Activity H (UAT):** ES = 18, EF =  $18 + 3 = 21$
9. **Activity I (Deployment Prep):** ES = 21, EF =  $21 + 2 = 23$
10. **Activity J (Deployment):** ES = 23, EF =  $23 + 2 = 25$
11. **Activity K (Training):** ES = 14, EF =  $14 + 4 = 18$
12. **Activity L (Support):** ES = 25, EF =  $25 + 1 = 26$
13. **Activity M (Closure):** ES =  $\max(26, 18) = 26$ , EF =  $26 + 2 = 28$

**Project Duration = 28 weeks** (from forward pass calculation)

## 7.4 Backward Pass Calculation (LS and LF)

The backward pass determines the latest times activities can start and finish without delaying the project, working from project finish to start.

### 7.4.1 Calculation Method

- **LF** = Minimum LS of all successor activities (or project duration for final activity)
- **LS** = LF - Duration

### 7.4.2 Backward Pass Results

1. **Activity M (Closure):** LF = 28, LS = 28 - 2 = 26
2. **Activity L (Support):** LF = 26, LS = 26 - 1 = 25
3. **Activity K (Training):** LF = 26, LS = 26 - 4 = 22
4. **Activity J (Deployment):** LF = 25, LS = 25 - 2 = 23
5. **Activity I (Deployment Prep):** LF = 23, LS = 23 - 2 = 21
6. **Activity H (UAT):** LF = 21, LS = 21 - 3 = 18
7. **Activity G (System Testing):** LF = 18, LS = 18 - 4 = 14
8. **Activity F (Integration):** LF = min(14, 22) = 14, LS = 14 - 3 = 11
9. **Activity E (Frontend):** LF = 11, LS = 11 - 5 = 6 (critical - required for F)
10. **Activity D (Backend):** LF = 11, LS = 11 - 5 = 6
11. **Activity C (Design):** LF = min(6, 7) = 6, LS = 6 - 1 = 5
12. **Activity B (Requirements):** LF = 5, LS = 5 - 2 = 3
13. **Activity A (Initiation):** LF = 3, LS = 3 - 3 = 0

## 7.5 Slack Analysis

Slack (or float) represents the amount of time an activity can be delayed. There are two types of slack:

### 7.5.1 Types of Slack

- **Total Slack:** Maximum time an activity can be delayed without delaying project completion
- **Free Slack:** Maximum time an activity can be delayed without delaying the early start of any successor activity

### 7.5.2 Slack Calculations

**Total Slack = LS - ES** (or equivalently, LF - EF)

**Free Slack = ES(successor) - EF(current)** (minimum if multiple successors)

### 7.5.3 Slack Calculations Explained

- **Activity K (Training):**

- Total Slack = LS - ES = 22 - 14 = 8 weeks
- Free Slack = ES(M) - EF(K) = 26 - 18 = 8 weeks
- Can be delayed 8 weeks without affecting project or successor M

- **All other activities:**

- Both Total Slack and Free Slack = 0
- Form the critical path
- Any delay immediately impacts project schedule

### 7.5.4 Slack Implications

- **Zero Slack Activities:** Must start and finish on schedule; form the critical path
- **Frontend Development (E):** Has 0 total and free slack (critical); must finish by week 11 as System Integration depends on it
- **Training (K):** Has 8 weeks of both total and free slack; can start anytime between weeks 14-22 without delaying project or successor
- **Equal Total and Free Slack:** When total slack equals free slack (as in Activity K), the activity can use all its slack without impacting any other activity

## 7.6 Critical Path Identification

The critical path is the sequence of activities with zero float, representing the longest path through the project network.

### 7.6.1 Critical Path

$$\text{A} \rightarrow \text{B} \rightarrow \text{C} \rightarrow \{\text{D}, \text{E}\} \rightarrow \text{F} \rightarrow \text{G} \rightarrow \text{H} \rightarrow \text{I} \rightarrow \text{J} \rightarrow \text{L} \rightarrow \text{M}$$

*Note: Activities D (Backend) and E (Frontend) are parallel critical activities that must both complete before F (Integration) can begin. The notation {D, E} indicates these activities run concurrently.*

**Table 7.2:** Critical Path Activities

| <b>Activity</b>                | <b>Activity Name</b>    | <b>Duration</b> | <b>Dates (2026)</b> |
|--------------------------------|-------------------------|-----------------|---------------------|
| A                              | Project Initiation      | 3 weeks         | Mar 1 – Mar 21      |
| B                              | Requirements Gathering  | 2 weeks         | Mar 22 – Apr 4      |
| C                              | System Design           | 1 week          | Apr 5 – Apr 11      |
| D                              | Backend Development     | 5 weeks         | Apr 12 – May 16     |
| E                              | Frontend Development    | 5 weeks         | Apr 12 – May 16     |
| F                              | System Integration      | 3 weeks         | May 17 – Jun 6      |
| G                              | System Testing          | 4 weeks         | Jun 7 – Jul 4       |
| H                              | User Acceptance Testing | 3 weeks         | Jul 5 – Jul 25      |
| I                              | Deployment Preparation  | 2 weeks         | Jul 26 – Aug 8      |
| J                              | System Deployment       | 2 weeks         | Aug 9 – Aug 22      |
| L                              | Post-Deployment Support | 1 week          | Aug 23 – Aug 29     |
| M                              | Project Closure         | 2 weeks         | Aug 30 – Sep 12     |
| <b>Total Project Duration:</b> |                         |                 | <b>28 weeks</b>     |

### 7.6.2 Critical Path Significance

- Any delay in critical path activities directly delays project completion
- Critical activities require closest monitoring and management attention
- **Parallel Critical Activities:** Both Backend (D) and Frontend (E) are critical activities running concurrently (weeks 6–11). System Integration (F) cannot start until BOTH are complete, making both paths critical
- Resources should be prioritized for critical path activities

## 7.7 Critical Path Management

### 7.7.1 Management Strategies for Critical Activities

1. **Resource Priority:** Allocate best resources to critical path activities (A, B, C, D, E, F, G, H, I, J, L, M)
2. **Close Monitoring:** Track progress daily on critical activities; weekly for non-critical
3. **Risk Mitigation:** Develop contingency plans for high-risk critical activities
4. **Early Warning System:** Implement alerts when critical activities fall behind schedule
5. **Buffer Management:** Consider adding time buffers before key milestones on critical path

### 7.7.2 Managing Non-Critical Activities

- **Training & Documentation (K):** 8 weeks float provides significant scheduling flexibility; can start anytime between weeks 14–22

- Use float in Training activities to balance workload and optimize resource utilization
- Training can be scheduled flexibly without impacting the critical deployment path

## 7.8 Network Analysis Summary

### 7.8.1 Key Findings

1. **Project Duration:** 28 weeks (Mar 1 – Sep 15, 2026)
2. **Critical Path:** 12 out of 13 activities are critical (92% of activities)
3. **Critical Path Length:** 28 weeks (no schedule reserve)
4. **Parallel Execution:** Backend (D) and Frontend (E) run in parallel but both are critical
5. **Slack Available:** Very limited (only Training with 8 weeks total and free slack)

### 7.8.2 Schedule Risk Assessment

- **High Risk:** 92% of activities are critical with zero total slack
- **Limited Flexibility:** Very little schedule buffer available
- **Risk Mitigation:** Focus on preventing delays in both Backend (D) and Frontend (E), Integration (F), Testing (G, H), and Deployment (I, J)

### 7.8.3 Recommendations

1. Assign most experienced resources to critical path activities
2. Implement daily progress tracking for activities D, E, F, G, H, I, J
3. Use 8 weeks of slack in Training (K) to balance resource allocation

# Chapter 8

## PERT Time Estimation

This chapter presents the Program Evaluation and Review Technique (PERT) time estimates for project activities. PERT uses three-point estimation to account for uncertainty in activity durations and provides a more realistic project completion estimate.

### 8.1 What is PERT?

**PERT** is a network analysis technique used to estimate project duration when there is a high degree of uncertainty about the individual activity duration estimates.

PERT uses **probabilistic time estimates** based on three scenarios for each activity:

- **Optimistic Time (O):** Best-case scenario – minimum time if everything goes perfectly
- **Most Likely Time (M):** Most realistic time estimate – what typically happens
- **Pessimistic Time (P):** Worst-case scenario – maximum time if significant problems occur

This three-point estimate approach provides a weighted average that accounts for both best and worst-case scenarios.

### 8.2 PERT Formula

The PERT weighted average is calculated as:

$$\text{PERT Weighted Average} = \frac{\text{Optimistic} + 4 \times \text{Most Likely} + \text{Pessimistic}}{6} \quad (8.1)$$

#### 8.2.1 Formula Explanation

The formula gives more weight to the most likely time (4 times) while still considering both optimistic and pessimistic scenarios. This provides a more realistic estimate than simply using the most likely time alone.

### 8.3 PERT Estimates for Project Activities

The following table shows three-point time estimates for all major project activities and their PERT weighted averages.

**Table 8.1:** PERT Three-Point Time Estimates for Project Activities

| ID                          | Activity Name            | Optimistic<br>(weeks) | Most Likely<br>(weeks) | Pessimistic<br>(weeks) | PERT Avg<br>(weeks) | Calculation             |
|-----------------------------|--------------------------|-----------------------|------------------------|------------------------|---------------------|-------------------------|
| A                           | Project Initiation       | 2.5                   | 3                      | 4                      | 3.1                 | $(2.5+4\times3+4)/6$    |
| B                           | Requirements Gathering   | 1.5                   | 2                      | 2.5                    | 2.0                 | $(1.5+4\times2+2.5)/6$  |
| C                           | System Design            | 0.75                  | 1                      | 1.5                    | 1.0                 | $(0.75+4\times1+1.5)/6$ |
| D                           | Backend Development      | 4.5                   | 5                      | 6.5                    | 5.2                 | $(4.5+4\times5+6.5)/6$  |
| E                           | Frontend Development     | 4.5                   | 5                      | 6                      | 5.1                 | $(4.5+4\times5+6)/6$    |
| F                           | System Integration       | 2.5                   | 3                      | 4                      | 3.1                 | $(2.5+4\times3+4)/6$    |
| G                           | System Testing           | 3.5                   | 4                      | 5                      | 4.1                 | $(3.5+4\times4+5)/6$    |
| H                           | User Acceptance Testing  | 2.5                   | 3                      | 4                      | 3.1                 | $(2.5+4\times3+4)/6$    |
| I                           | Deployment Preparation   | 1.5                   | 2                      | 2.5                    | 2.0                 | $(1.5+4\times2+2.5)/6$  |
| J                           | System Deployment        | 1.5                   | 2                      | 3                      | 2.1                 | $(1.5+4\times2+3)/6$    |
| K                           | Training & Documentation | 3.5                   | 4                      | 5                      | 4.1                 | $(3.5+4\times4+5)/6$    |
| L                           | Post-Deployment Support  | 0.75                  | 1                      | 1.5                    | 1.0                 | $(0.75+4\times1+1.5)/6$ |
| M                           | Project Closure          | 1.5                   | 2                      | 2.5                    | 2.0                 | $(1.5+4\times2+2.5)/6$  |
| <b>Total PERT Duration:</b> |                          |                       |                        |                        | <b>37.9 weeks</b>   |                         |

## 8.4 Comparison: Deterministic vs. PERT Estimates

Comparing the original deterministic (single-point) estimates from the AON analysis with the PERT weighted averages:

**Table 8.2:** Deterministic vs. PERT Duration Comparison

| Estimate Type         | Duration          | Basis                | Risk Coverage |
|-----------------------|-------------------|----------------------|---------------|
| Deterministic (AON)   | 28 weeks          | Single estimate      | Low           |
| PERT Weighted Average | 37.9 weeks        | Three-point estimate | High          |
| <b>Difference</b>     | <b>+9.9 weeks</b> | <b>Risk buffer</b>   |               |

## 8.5 PERT Analysis Interpretation

### 8.5.1 Activities with High Uncertainty

Activities where pessimistic time is higher than optimistic (showing uncertainty):

- **Backend Development (D):** 4.5-6.5 weeks range (moderate risk: complex prerequisite validation)
- **Frontend Development (E):** 4.5-6 weeks range (moderate risk: UI/UX iterations)
- **System Testing (G):** 3.5-5 weeks range (moderate risk: bug discovery and fixes)
- **User Acceptance Testing (H):** 2.5-4 weeks range (moderate risk: stakeholder availability)

## 8.6 Recommendations

Based on the PERT analysis, the following recommendations are made:

1. **Monitor High-Risk Activities:** Closely track Backend Development, Frontend Development, and Testing phases
2. **Early Risk Mitigation:** Start risk mitigation activities early for activities with wide time ranges
3. **Regular Re-estimation:** Update PERT estimates as work progresses and uncertainty decreases
4. **Communicate Realistically:** Inform stakeholders that 28 weeks is optimistic; 39=8 weeks is more realistic

**Final Recommendation:** Communicate a **38-40 week timeline** to stakeholders while internally targeting 28-30 weeks to maximize efficiency.

# Chapter 9

## NPV Analysis

This chapter presents the financial feasibility analysis of the Online Course Registration Portal project using Net Present Value (NPV) methodology. NPV analysis helps determine whether the project's benefits justify its costs over time.

### 9.1 NPV Analysis Overview

Net Present Value (NPV) is a capital budgeting technique that calculates the present value of future cash flows, both positive (benefits) and negative (costs), using a discount rate.

#### 9.1.1 NPV Formula

$$NPV = \sum_{t=0}^n \frac{CF_t}{(1+r)^t} \quad (9.1)$$

Where:

- $CF_t$  = Cash flow at time period t
- $r$  = Discount rate
- $t$  = Time period
- $n$  = Project lifetime

#### 9.1.2 Decision Criteria

- $NPV > 0$ : Project is financially viable (accept)
- $NPV = 0$ : Project breaks even (marginal)
- $NPV < 0$ : Project loses value (reject)

## 9.2 Project Cost Estimation

### 9.2.1 Initial Investment (Year 0)

**Table 9.1:** Initial Project Costs

| Cost Category                      | Amount (EGP)     |
|------------------------------------|------------------|
| Personnel Costs (Development Team) | 800,000          |
| Project Management                 | 150,000          |
| Software Licenses and Tools        | 100,000          |
| Hardware and Infrastructure        | 200,000          |
| Testing and QA                     | 120,000          |
| Training and Documentation         | 80,000           |
| Contingency Reserve (10%)          | 145,000          |
| <b>Total Initial Investment</b>    | <b>1,595,000</b> |

### 9.2.2 Annual Operating Costs (Years 1-5)

**Table 9.2:** Annual Operating and Maintenance Costs

| Cost Category                      | Annual Amount (EGP) |
|------------------------------------|---------------------|
| System Maintenance                 | 80,000              |
| Support Staff (2 FTE)              | 180,000             |
| Server and Infrastructure          | 60,000              |
| Software License Renewals          | 25,000              |
| System Updates and Enhancements    | 50,000              |
| <b>Total Annual Operating Cost</b> | <b>395,000</b>      |

### 9.3 Project Benefits Estimation

#### 9.3.1 Quantifiable Benefits

**Table 9.3:** Annual Benefits

| Benefit Category              | Annual Value (EGP) | Justification   |
|-------------------------------|--------------------|---|
| Administrative Labor Savings  | 450,000            | Reduction of 3 FTE in manual registration processing            |
| Reduced Registration Errors   | 80,000             | Fewer prerequisite violations and scheduling conflicts          |
| Improved Resource Utilization | 120,000            | Better course capacity management reduces under/over-enrollment |
| Reduced Paper and Printing    | 15,000             | Elimination of paper-based registration forms                   |
| IT Support Cost Reduction     | 35,000             | Fewer registration-related support tickets                      |
| Student Productivity Gains    | 50,000             | Faster registration saves student time                          |
| <b>Total Annual Benefits</b>  | <b>750,000</b>     |   |

#### 9.3.2 Intangible Benefits (Not Quantified in NPV)

- Improved student satisfaction and experience
- Enhanced institutional reputation
- Better data accuracy and reporting capabilities
- Increased competitive advantage
- Scalability for future growth
- Foundation for additional digital services

### 9.4 NPV Calculation

#### 9.4.1 Assumptions

- **Analysis Period:** 5 years
- **Discount Rate:** 12% (reflecting institutional cost of capital and risk)
- **Currency:** Egyptian Pounds (EGP)
- Benefits begin in Year 1 (first semester after deployment)
- Costs and benefits adjusted for inflation where appropriate

#### 9.4.2 Cash Flow Analysis

**Table 9.4:** NPV Cash Flow Analysis

| Year | Costs (EGP) | Benefits (EGP) | Net Cash Flow (EGP) | PV Factor (12%) |
|------|-------------|----------------|---------------------|-----------------|
| 0    | 1,595,000   | 0              | -1,595,000          | 1.0000          |
| 1    | 395,000     | 750,000        | 355,000             | 0.8929          |
| 2    | 395,000     | 750,000        | 355,000             | 0.7972          |
| 3    | 395,000     | 750,000        | 355,000             | 0.7118          |
| 4    | 395,000     | 750,000        | 355,000             | 0.6355          |
| 5    | 395,000     | 750,000        | 355,000             | 0.5674          |

**Table 9.5:** Present Value Calculations

| Year              | Net Cash Flow (EGP) | Present Value (EGP) |
|-------------------|---------------------|---------------------|
| 0                 | -1,595,000          | -1,595,000          |
| 1                 | 355,000             | 316,970             |
| 2                 | 355,000             | 283,009             |
| 3                 | 355,000             | 252,687             |
| 4                 | 355,000             | 225,613             |
| 5                 | 355,000             | 201,440             |
| <b>Total NPV:</b> |                     | <b>-315,281</b>     |

#### 9.5 Sensitivity Analysis

Testing NPV sensitivity to key variables:

**Table 9.6:** NPV Sensitivity to Discount Rate

| Discount Rate | NPV (EGP) | Decision |
|---------------|-----------|----------|
| 8%            | 120,450   | Accept   |
| 10%           | -85,250   | Marginal |
| 12%           | -315,281  | Reject   |
| 15%           | -595,100  | Reject   |

**Table 9.7:** NPV Sensitivity to Annual Benefits

| Annual Benefits (EGP) | NPV (EGP) | Decision |
|-----------------------|-----------|----------|
| 650,000               | -675,231  | Reject   |
| 750,000 (Base Case)   | -315,281  | Reject   |
| 850,000               | 44,669    | Accept   |
| 950,000               | 404,619   | Accept   |

## 9.6 Analysis and Recommendations

### 9.6.1 Financial Viability Assessment

Based on the NPV analysis with current assumptions:

- **NPV = -315,281 EGP** (at 12% discount rate)
- Project does not meet purely financial acceptance criteria
- Break-even requires either:
  - Discount rate below 9.2%, OR
  - Annual benefits above 850,000 EGP, OR
  - Initial costs reduced by 20%

### 9.6.2 Strategic Considerations

Despite negative NPV, the project may still be justified by:

1. **Intangible Benefits:** Student satisfaction and institutional reputation are valuable but not quantified
2. **Strategic Necessity:** Modern registration systems are expected by students and competitive institutions
3. **Future Capabilities:** Platform enables future digital services with additional value
4. **Risk Reduction:** Manual processes have higher error risk and compliance issues
5. **Conservative Estimates:** Benefits may be understated; productivity gains often exceed projections

### 9.6.3 Recommendations

1. **Re-evaluate Benefits:** Conduct more detailed analysis of productivity and efficiency gains
2. **Cost Optimization:** Explore cost reduction opportunities without compromising quality
3. **Phased Approach:** Consider phased implementation to spread costs over time
4. **Broader Analysis:** Include qualitative factors in decision-making
5. **Proceed with Project:** Recommend approval based on strategic value despite marginal financial returns

# Chapter 10

## Estimated Cash Flow

This chapter presents the detailed cash flow projections for the Online Course Registration Portal project over its lifecycle. Cash flow analysis is essential for budget planning and ensuring adequate funding availability.

### 10.1 Cash Flow Analysis Overview

Cash flow analysis tracks the timing and amount of actual cash inflows and outflows throughout the project lifecycle. Unlike NPV analysis, cash flow focuses on liquidity and funding requirements.

### 10.2 Project Cash Outflows

#### 10.2.1 Development Phase Cash Outflows (Year 0)

**Table 10.1:** Development Phase Cash Outflows by Quarter

| Category                | Q1 (EGP)       | Q2 (EGP)       | Q3 (EGP)       | Q4 (EGP)       | Total (EGP)      |
|-------------------------|----------------|----------------|----------------|----------------|------------------|
| Personnel               | 150,000        | 250,000        | 250,000        | 150,000        | 800,000          |
| Project Management      | 40,000         | 40,000         | 40,000         | 30,000         | 150,000          |
| Software/Licenses       | 80,000         | 10,000         | 5,000          | 5,000          | 100,000          |
| Hardware/Infrastructure | 150,000        | 30,000         | 15,000         | 5,000          | 200,000          |
| Testing/QA              | 10,000         | 20,000         | 40,000         | 50,000         | 120,000          |
| Training/Documentation  | 5,000          | 10,000         | 25,000         | 40,000         | 80,000           |
| Contingency             | 15,000         | 45,000         | 45,000         | 40,000         | 145,000          |
| <b>Quarterly Total</b>  | <b>450,000</b> | <b>405,000</b> | <b>420,000</b> | <b>320,000</b> | <b>1,595,000</b> |

#### 10.2.2 Operating Phase Cash Outflows (Years 1-5)

**Table 10.2:** Annual Operating Cash Outflows

| Category            | Year 1 (EGP)   | Year 2 (EGP)   | Year 3 (EGP)   | Year 4 (EGP)   | Year 5 (EGP)   |
|---------------------|----------------|----------------|----------------|----------------|----------------|
| Maintenance         | 80,000         | 85,000         | 90,000         | 95,000         | 100,000        |
| Support Staff       | 180,000        | 190,000        | 200,000        | 210,000        | 220,000        |
| Infrastructure      | 60,000         | 65,000         | 70,000         | 75,000         | 80,000         |
| Licenses            | 25,000         | 28,000         | 30,000         | 32,000         | 35,000         |
| Enhancements        | 50,000         | 55,000         | 60,000         | 65,000         | 70,000         |
| <b>Annual Total</b> | <b>395,000</b> | <b>423,000</b> | <b>450,000</b> | <b>477,000</b> | <b>505,000</b> |

### 10.3 Project Cash Inflows (Benefits)

#### 10.3.1 Annual Cash Inflows from Benefits

**Table 10.3:** Annual Cash Inflows (Savings and Benefits)

| Benefit Source         | Year 1<br>(EGP) | Year 2<br>(EGP) | Year 3<br>(EGP) | Year 4<br>(EGP) | Year 5<br>(EGP) |
|------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Labor Savings          | 450,000         | 470,000         | 490,000         | 510,000         | 530,000         |
| Error Reduction        | 80,000          | 85,000          | 90,000          | 95,000          | 100,000         |
| Resource Optimization  | 120,000         | 130,000         | 140,000         | 150,000         | 160,000         |
| Paper/Printing Savings | 15,000          | 16,000          | 17,000          | 18,000          | 19,000          |
| IT Support Reduction   | 35,000          | 38,000          | 40,000          | 42,000          | 45,000          |
| Productivity Gains     | 50,000          | 55,000          | 60,000          | 65,000          | 70,000          |
| <b>Annual Total</b>    | <b>750,000</b>  | <b>794,000</b>  | <b>837,000</b>  | <b>880,000</b>  | <b>924,000</b>  |

### 10.4 Net Cash Flow Summary

**Table 10.4:** Net Cash Flow by Year

| Year         | Cash Inflows<br>(EGP) | Cash Outflows<br>(EGP) | Net Cash Flow<br>(EGP) |
|--------------|-----------------------|------------------------|------------------------|
| 0            | 0                     | 1,595,000              | -1,595,000             |
| 1            | 750,000               | 395,000                | 355,000                |
| 2            | 794,000               | 423,000                | 371,000                |
| 3            | 837,000               | 450,000                | 387,000                |
| 4            | 880,000               | 477,000                | 403,000                |
| 5            | 924,000               | 505,000                | 419,000                |
| <b>Total</b> | <b>4,185,000</b>      | <b>3,845,000</b>       | <b>340,000</b>         |

### 10.5 Cumulative Cash Flow

**Table 10.5:** Cumulative Cash Flow Analysis

| Year | Net Cash Flow (EGP) | Cumulative Cash Flow (EGP) |
|------|---------------------|----------------------------|
| 0    | -1,595,000          | -1,595,000                 |
| 1    | 355,000             | -1,240,000                 |
| 2    | 371,000             | -869,000                   |
| 3    | 387,000             | -482,000                   |
| 4    | 403,000             | -79,000                    |
| 5    | 419,000             | 340,000                    |

### 10.6 Cash Flow Analysis Insights

#### 10.6.1 Key Observations

- Peak Negative Cash Flow: -1,595,000 EGP at end of Year 0

- **Positive Annual Flows:** All operating years generate positive net cash flow
- **Break-even Point:** Between Year 4 and Year 5 (approximately 4.2 years)
- **Total 5-Year Net Cash Flow:** +340,000 EGP (before discounting)
- **Increasing Benefits:** Annual net cash flow improves each year due to inflation-adjusted benefits

### 10.6.2 Funding Requirements

Maximum funding requirement occurs in Year 0:

- **Peak Funding Need:** 1,595,000 EGP
- **Recommended Reserve:** Additional 10% (159,500 EGP) for contingencies
- **Total Funding Required:** 1,754,500 EGP

## 10.7 Cash Flow Management Strategies

### 10.7.1 Funding Strategy

1. Secure full initial funding before project commencement
2. Establish contingency reserve for unforeseen expenses
3. Consider phased funding releases tied to milestone completion
4. Maintain cash flow monitoring dashboard

### 10.7.2 Cash Flow Optimization

- Negotiate payment terms with vendors to smooth cash outflows
- Front-load benefit realization where possible
- Consider early deployment of high-value features
- Monitor and manage working capital requirements

# Chapter 11

## Payback Period and ROI

This chapter calculates the Payback Period and Return on Investment (ROI) for the Online Course Registration Portal project, providing additional financial metrics for decision-making.

### 11.1 Payback Period Analysis

The payback period is the time required for cumulative cash inflows to equal the initial investment.

#### 11.1.1 Payback Period Calculation

From the cumulative cash flow analysis (Chapter 10):

**Table 11.1:** Payback Period Calculation

| Year | Net Cash Flow (EGP) | Cumulative Cash Flow (EGP) |
|------|---------------------|----------------------------|
| 0    | -1,595,000          | -1,595,000                 |
| 1    | 355,000             | -1,240,000                 |
| 2    | 371,000             | -869,000                   |
| 3    | 387,000             | -482,000                   |
| 4    | 403,000             | -79,000                    |
| 5    | 419,000             | 340,000                    |

The payback period occurs between Year 4 and Year 5.

#### Precise Payback Period Calculation:

$$\text{Payback Period} = 4 + \frac{79,000}{419,000} = 4 + 0.19 = 4.19 \text{ years} \quad (11.1)$$

**Payback Period = 4.19 years (approximately 4 years and 2 months)**

#### 11.1.2 Payback Period Interpretation

- The project will recover its initial investment in approximately 4.2 years
- This is within typical organizational acceptance criteria (usually 3-5 years for IT projects)
- After payback, all subsequent cash flows represent net gains
- Years 5 and beyond provide pure profit if the system continues operation

### 11.1.3 Discounted Payback Period

Accounting for time value of money at 12% discount rate:

**Table 11.2:** Discounted Payback Period

| Year          | Net Cash Flow (EGP) | Discounted CF (EGP) | Cumulative Discounted CF (EGP) |
|---------------|---------------------|---------------------|--------------------------------|
| 0             | -1,595,000          | -1,595,000          | -1,595,000                     |
| 1             | 355,000             | 316,970             | -1,278,030                     |
| 2             | 371,000             | 295,766             | -982,264                       |
| 3             | 387,000             | 275,465             | -706,799                       |
| 4             | 403,000             | 256,103             | -450,696                       |
| 5             | 419,000             | 237,675             | -213,021                       |
| 6 (projected) | 440,000             | 223,440             | 10,419                         |

**Discounted Payback Period  $\approx 6$  years**

The discounted payback period is significantly longer, reflecting the time value of money.

## 11.2 Return on Investment (ROI) Analysis

### 11.2.1 ROI Formula

$$ROI = \frac{\text{Total Benefits} - \text{Total Costs}}{\text{Total Costs}} \times 100\% \quad (11.2)$$

### 11.2.2 Simple ROI Calculation (5-Year Period)

From Chapter 10 cash flow data:

- **Total Investment (Costs):**  $1,595,000 + 2,250,000 = 3,845,000$  EGP
- **Total Benefits (5 years):** 4,185,000 EGP
- **Net Benefit:**  $4,185,000 - 3,845,000 = 340,000$  EGP

$$ROI = \frac{340,000}{3,845,000} \times 100\% = 8.8\% \quad (11.3)$$

**Simple ROI = 8.8% over 5 years**

### 11.2.3 Annualized ROI

$$\text{Annualized ROI} = \frac{8.8\%}{5} = 1.76\% \text{ per year} \quad (11.4)$$

### 11.2.4 Alternative ROI: Initial Investment Only

Calculating ROI based on initial investment recovery:

$$ROI_{initial} = \frac{340,000}{1,595,000} \times 100\% = 21.3\% \quad (11.5)$$

This represents the return on the initial development investment over the 5-year operational period.

## 11.3 Financial Metrics Summary

**Table 11.3:** Financial Performance Metrics Summary

| Metric                        | Value        | Assessment   |
|-------------------------------|--------------|--------------|
| Payback Period                | 4.19 years   | Acceptable   |
| Discounted Payback Period     | 6 years      | Marginal     |
| Simple ROI (5-year)           | 8.8%         | Low-Moderate |
| Annualized ROI                | 1.76%        | Low          |
| NPV (12% discount)            | -315,281 EGP | Negative     |
| Total Net Cash Flow (5 years) | +340,000 EGP | Positive     |

## 11.4 Break-Even Analysis

### 11.4.1 Break-Even Point

The project breaks even (cumulative cash flow = 0) at approximately:

- **Time:** 4.19 years from project start
- **Cumulative Students Served:** Approximately [calculate based on student numbers]
- **Registration Cycles:** Approximately 8-9 semesters

### 11.4.2 Break-Even Sensitivity

**Table 11.4:** Break-Even Sensitivity to Annual Benefits

| Annual Benefits (EGP) | Payback Period (years) |
|-----------------------|------------------------|
| 600,000               | 7.8                    |
| 700,000               | 5.2                    |
| 750,000 (base)        | 4.2                    |
| 800,000               | 3.5                    |
| 900,000               | 2.9                    |

## 11.5 Financial Viability Assessment

### 11.5.1 Strengths

- Achieves payback within acceptable organizational timeframe (< 5 years)
- Positive total cash flow after 5 years
- Consistent positive annual cash flows after Year 1
- Benefits increase over time, improving future returns

### 11.5.2 Weaknesses

- Low annualized ROI (1.76%)
- Negative NPV at 12% discount rate
- Long discounted payback period (6 years)
- Sensitive to benefit realization assumptions

### 11.5.3 Recommendations

- Proceed with Caution:** Financial metrics are marginal but acceptable for strategic projects
- Maximize Benefits:** Focus on realizing all quantified benefits and uncovering additional savings
- Monitor Performance:** Track actual costs and benefits closely to ensure projections are met
- Consider Intangibles:** Factor in non-financial benefits (student satisfaction, competitive position)
- Plan for Longevity:** System lifespan beyond 5 years significantly improves returns
- Cost Optimization:** Seek opportunities to reduce operational costs in Years 1-5

## 11.6 Comparative Analysis

### 11.6.1 Industry Benchmarks

**Table 11.5:** Comparison with Industry Norms

| Metric         | This Project | Industry Average | Status        |
|----------------|--------------|------------------|---------------|
| Payback Period | 4.2 years    | 3-4 years        | Slightly High |
| ROI (5-year)   | 8.8%         | 15-25%           | Below Average |
| NPV            | Negative     | Positive         | Below Norm    |

### 11.6.2 Strategic Justification

Despite below-average financial metrics, the project is strategically justified by:

- Essential infrastructure for modern educational institution
- Competitive necessity - peer institutions have similar systems
- Foundation for future digital transformation initiatives
- Risk mitigation - manual processes pose compliance and error risks
- Student expectations and satisfaction requirements
- Scalability for institutional growth

# Chapter 12

## RACI Chart

This chapter presents the RACI (Responsible, Accountable, Consulted, Informed) matrix for the Online Course Registration Portal project, clearly defining roles and responsibilities for all project activities and deliverables.

### 12.1 RACI Matrix Overview

The RACI matrix is a responsibility assignment matrix that clarifies roles for each task and deliverable:

#### R - Responsible

Person(s) who perform the work to complete the task

#### A - Accountable

Person who is ultimately answerable for the correct completion (only one A per task)

#### C - Consulted

People who provide input and with whom there is two-way communication

#### I - Informed

People who are kept up-to-date on progress (one-way communication)

### 12.2 Project Roles

**Table 12.1:** Project Roles and Responsibilities

| Role | Description   |
|------|---|
| PM   | Project Manager - Overall project leadership and coordination   |
| PS   | Project Sponsor - Executive oversight and funding approval      |
| BA   | Business Analyst - Requirements gathering and documentation     |
| SA   | System Architect - Technical architecture and design leadership |
| TL   | Technical Lead - Development team leadership                    |
| DEV  | Development Team - Software developers (backend and frontend)   |
| QA   | QA Team - Testing and quality assurance personnel               |
| DBA  | Database Administrator - Database design and management         |
| UX   | UX/UI Designer - User experience and interface design           |
| DO   | DevOps Engineer - Infrastructure and deployment                 |
| SM   | Security Manager - Security requirements and testing            |
| AA   | Academic Affairs - Business stakeholder representative          |
| IT   | IT Department - Infrastructure and support services             |
| TR   | Trainer - Training development and delivery                     |
| TW   | Technical Writer - Documentation specialist                     |

### 12.3 RACI Matrix

**Table 12.2:** RACI Matrix for Project Activities

| Activity / Deliverable            | PM | PS | BA | SA | TL | DEV | QA | DBA | UX | DO | SM | AA | IT | TR | TW |
|-----------------------------------|----|----|----|----|----|-----|----|-----|----|----|----|----|----|----|----|
| <b>Project Initiation</b>         |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |
| Project Charter                   | R  | A  | C  | C  | I  | I   | I  | I   | I  | I  | C  | I  | I  | I  | I  |
| Project Plan                      | A  | I  | C  | C  | C  | I   | C  | I   | I  | C  | I  | I  | I  | I  | I  |
| Stakeholder Analysis              | R  | C  | R  | I  | I  | I   | I  | I   | I  | I  | C  | C  | I  | I  | I  |
| <b>Requirements and Analysis</b>  |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |
| Requirements Gathering            | C  | I  | A  | C  | I  | I   | C  | I   | C  | I  | C  | R  | I  | I  | I  |
| Requirements Documentation        | C  | I  | A  | C  | I  | I   | C  | I   | I  | I  | C  | I  | I  | R  |    |
| Business Rules Definition         | I  | I  | A  | C  | I  | I   | I  | I   | I  | I  | R  | I  | I  | C  |    |
| <b>System Design</b>              |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |
| System Architecture               | C  | I  | C  | A  | R  | C   | I  | C   | C  | C  | C  | I  | C  | I  | I  |
| Database Design                   | C  | I  | C  | C  | C  | I   | I  | A   | I  | I  | I  | I  | C  | I  | I  |
| UI/UX Design                      | C  | I  | C  | I  | I  | I   | I  | A   | I  | I  | C  | I  | I  | I  | I  |
| Security Design                   | C  | I  | I  | C  | C  | I   | I  | I   | I  | A  | I  | C  | I  | I  | I  |
| <b>Development</b>                |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |
| Backend Development               | C  | I  | I  | C  | A  | R   | I  | C   | I  | I  | C  | I  | I  | I  | I  |
| Frontend Development              | C  | I  | I  | C  | A  | R   | I  | I   | C  | I  | I  | I  | I  | I  | I  |
| Database Implementation           | C  | I  | I  | C  | C  | C   | I  | A   | I  | I  | I  | I  | C  | I  | I  |
| System Integration                | C  | I  | C  | C  | A  | R   | C  | I   | I  | C  | C  | I  | C  | I  | I  |
| <b>Testing</b>                    |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |
| Test Planning                     | C  | I  | C  | I  | C  | C   | A  | I   | I  | I  | C  | I  | I  | I  | I  |
| Unit Testing                      | I  | I  | I  | I  | C  | R   | A  | I   | I  | I  | I  | I  | I  | I  | I  |
| Integration Testing               | C  | I  | C  | C  | C  | C   | A  | I   | I  | C  | C  | C  | C  | I  | I  |
| System Testing                    | C  | I  | C  | C  | C  | I   | A  | I   | I  | I  | C  | C  | I  | I  | I  |
| Performance Testing               | C  | I  | I  | C  | C  | I   | A  | I   | I  | C  | C  | I  | C  | I  | I  |
| Security Testing                  | C  | I  | I  | C  | C  | I   | R  | I   | I  | I  | A  | I  | C  | I  | I  |
| UAT                               | C  | I  | R  | I  | C  | C   | C  | I   | I  | I  | I  | A  | C  | I  | I  |
| <b>Deployment</b>                 |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |
| Deployment Planning               | A  | I  | I  | C  | C  | I   | I  | I   | I  | R  | C  | I  | C  | I  | I  |
| Production Deployment             | C  | I  | I  | I  | C  | C   | C  | R   | I  | A  | C  | I  | R  | I  | I  |
| Go-Live Support                   | A  | I  | C  | C  | R  | R   | R  | C   | I  | R  | C  | I  | R  | I  | I  |
| <b>Training and Documentation</b> |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |
| User Documentation                | C  | I  | C  | I  | I  | I   | I  | I   | C  | I  | I  | C  | I  | C  | A  |
| Technical Documentation           | C  | I  | I  | C  | R  | C   | I  | C   | I  | C  | C  | I  | I  | A  | A  |
| Training Materials                | C  | I  | C  | I  | I  | I   | I  | I   | I  | I  | C  | I  | A  | R  |    |
| Training Delivery                 | C  | I  | I  | I  | C  | C   | I  | I   | I  | I  | C  | C  | A  | C  |    |
| <b>Project Closure</b>            |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |
| Lessons Learned                   | A  | C  | R  | C  | C  | C   | C  | C   | C  | C  | C  | C  | C  | C  | C  |
| Final Report                      | A  | I  | C  | C  | C  | I   | I  | I   | I  | I  | I  | I  | I  | I  | R  |
| Project Sign-off                  | R  | A  | I  | I  | I  | I   | I  | I   | I  | I  | I  | C  | I  | I  | I  |

## 12.4 RACI Analysis and Validation

### 12.4.1 Key Accountability Assignments

**Table 12.3:** Primary Accountabilities by Role

| Role                   | Primary Accountabilities                         |
|------------------------|--|
| Project Sponsor        | Project charter, UAT approval, final sign-off    |
| Project Manager        | Project plan, schedule, budget, overall delivery |
| Business Analyst       | Requirements documentation and validation        |
| System Architect       | System architecture and technical design         |
| Technical Lead         | Development activities and code quality          |
| Database Administrator | Database design and implementation               |
| QA Team Lead           | Test planning and execution, quality assurance   |
| UX Designer            | User interface and experience design             |
| DevOps Engineer        | Infrastructure and deployment                    |
| Security Manager       | Security requirements and testing                |
| Academic Affairs       | Business rules, UAT leadership                   |
| Trainer                | Training materials and delivery                  |
| Technical Writer       | All documentation deliverables                   |

## 12.5 Communication Implications

The RACI matrix informs the communication plan:

### 12.5.1 Consulted (C) Relationships

- Require two-way communication and active collaboration
- Input needed before decisions or completion
- Regular meetings and reviews scheduled
- Timely feedback mechanisms established

### 12.5.2 Informed (I) Relationships

- One-way communication (status updates)
- Progress reports and notifications
- Inclusion in status meetings as observers
- Access to project documentation repository

## 12.6 Conflict Resolution

When responsibilities overlap or conflicts arise:

1. The "Accountable" person has final decision authority

2. Project Manager mediates conflicts between "Responsible" parties
3. Project Sponsor resolves escalated conflicts
4. RACI matrix updates require Project Manager and Sponsor approval

# Chapter 13

## Risk Analysis

This chapter presents Project Risk Management for the Online Course Registration Portal using the six core processes: Plan Risk Management, Identify Risks, Perform Qualitative Risk Analysis, Perform Quantitative Risk Analysis, Plan Risk Responses, and Monitor and Control Risks.

### 13.1 Plan Risk Management

This process defines how risk management will be performed for the project.

#### 13.1.1 Risk Management Plan

**Table 13.1:** Risk Management Plan Elements

| Element                  | Description   |
|--------------------------|---|
| Methodology              | Qualitative analysis for all risks; quantitative analysis for high-impact risks |
| Roles & Responsibilities | Project Manager owns risk register; Team Leads own technical risks              |
| Budget                   | 145,000 EGP contingency reserve (10% of 1,450,000 base)                         |
| Schedule                 | 2 weeks buffer included in 36-week timeline                                     |
| Risk Categories          | Technical, Organizational, Project Management, External                         |
| Probability & Impact     | Low/Medium/High scales with defined thresholds                                  |
| Documentation            | Risk Register updated weekly; reported in status meetings                       |

### 13.2 Identify Risks

This process determines which risks might affect the project and documents their characteristics.

#### 13.2.1 Risk Identification Tools

**Table 13.2:** Risk Identification Methods Used

| Tool          | Purpose  |
|---------------|--|
| Brainstorming | Generate a broad list of possible risks                |
| Interviews    | Learn from experienced stakeholders                    |
| SWOT Analysis | Identify strengths, weaknesses, opportunities, threats |

#### 13.2.2 Risk Register

**Table 13.3:** Project Risk Register

| ID  | Risk Description                                  | Category       | Root Cause                       | Trigger                 | Prob. | Impact | Owner  | Response | Status |
|-----|---|----------------|----------------------------------|-------------------------|-------|--------|--------|----------|--------|
| R1  | SIS integration more complex than anticipated     | Technical      | Legacy system documentation gaps | Integration tests fail  | H     | H      | SA     | Mitigate | Open   |
| R2  | System performance degrades under concurrent load | Technical      | Insufficient load testing        | Response time > 3s      | M     | H      | TL     | Mitigate | Open   |
| R3  | Prerequisite validation logic contains errors     | Technical      | Complex business rules           | Failed UAT tests        | M     | H      | Dev    | Mitigate | Open   |
| R4  | Key developer leaves mid-project                  | Organizational | Market competition               | Resignation notice      | M     | H      | PM     | Transfer | Open   |
| R5  | Requirements gathering delayed                    | PM             | Stakeholder unavailability       | Missed meetings         | H     | M      | BA     | Mitigate | Open   |
| R6  | Budget overrun exceeds 145,000 EGP reserve        | PM             | Scope creep                      | Cost variance > 10%     | M     | M      | PM     | Avoid    | Open   |
| R7  | Schedule slips beyond 36 weeks                    | PM             | Underestimated complexity        | SPI < 0.9               | M     | H      | PM     | Mitigate | Open   |
| R8  | Staff resistance to new system                    | Organizational | Change aversion                  | Training attendance low | H     | M      | PM     | Mitigate | Open   |
| R9  | Security vulnerabilities discovered               | Technical      | Inadequate security review       | Penetration test fails  | L     | H      | SecEng | Mitigate | Open   |
| R10 | Vendor discontinues critical component            | External       | Market changes                   | EOL announcement        | L     | H      | SA     | Transfer | Open   |

### 13.3 Perform Qualitative Risk Analysis

This process prioritizes risks based on probability of occurrence and impact on project objectives. The output is a ranked list of risks, helping the team focus on the most significant ones.

#### 13.3.1 Probability and Impact Scales

**Table 13.4:** Probability and Impact Definitions

| Level      | Probability | Impact on Schedule | Impact on Budget   |
|------------|-------------|--------------------|--------------------|
| Low (L)    | < 30%       | < 1 week delay     | < 50,000 EGP       |
| Medium (M) | 30-60%      | 1-3 weeks delay    | 50,000-150,000 EGP |
| High (H)   | > 60%       | > 3 weeks delay    | > 150,000 EGP      |

#### 13.3.2 Probability-Impact Matrix

**Table 13.5:** Risk Probability-Impact Matrix

| Probability \ Impact | Low         | Medium             | High                       |
|----------------------|-------------|--------------------|----------------------------|
| High                 | Medium Risk | High Risk (R5, R8) | Critical Risk (R1)         |
| Medium               | Low Risk    | Medium Risk (R6)   | High Risk (R2, R3, R4, R7) |
| Low                  | Low Risk    | Low Risk           | Medium Risk (R9, R10)      |

### 13.4 Perform Quantitative Risk Analysis

This process numerically estimates the effect of risks on project objectives. It is mainly used for large or complex projects.

#### 13.4.1 Sensitivity Analysis

Sensitivity analysis examines how changes in key variables affect project outcomes:

**Table 13.6:** Sensitivity Analysis Results

| Variable Changed            | Outcome Affected    | Result                             |
|-----------------------------|---------------------|------------------------------------|
| Development cost +20%       | Total budget        | Budget increases by 160,000 EGP    |
| Task duration +15%          | Project schedule    | Delay risk of 5 weeks              |
| Resource count -1 developer | Productivity        | Output decreases, schedule extends |
| Integration effort +50%     | Schedule and budget | Significant delay and cost overrun |

### 13.5 Plan Risk Responses

This process defines actions to address identified risks.

### 13.5.1 Response Strategies for Negative Risks (Threats)

**Table 13.7:** Threat Response Strategies

| Strategy     | Description                           |
|--------------|---------------------------------------|
| Acceptance   | Acknowledge the risk without action   |
| Avoidance    | Change the plan to eliminate the risk |
| Mitigation   | Reduce probability or impact          |
| Transference | Shift risk to a third party           |

### 13.5.2 Response Strategies for Positive Risks (Opportunities)

**Table 13.8:** Opportunity Response Strategies

| Strategy     | Description                                    |
|--------------|--|
| Acceptance   | Take advantage if it occurs                    |
| Exploitation | Ensure the opportunity happens                 |
| Enhancement  | Increase probability or impact                 |
| Sharing      | Partner with others to realize the opportunity |

### 13.5.3 Risk Response Plan

**Table 13.9:** Risk Response Plan

| ID  | Strategy | Response Action                                   | Contingency Plan                   |
|-----|----------|---|------------------------------------|
| R1  | Mitigate | Early prototype; dedicated integration specialist | Manual data procedures temporarily |
| R2  | Mitigate | Load testing in Phase 3; scalable architecture    | Add cloud resources on-demand      |
| R3  | Mitigate | Automated unit tests; peer code review            | Extended UAT cycle                 |
| R4  | Transfer | Cross-training; documentation                     | Contract backup developers         |
| R5  | Mitigate | Schedule interviews early; async reviews          | Use documented assumptions         |
| R6  | Avoid    | Strict change control; scope freeze               | Use contingency reserve            |
| R7  | Mitigate | Weekly schedule tracking; buffer time             | Fast-track critical path           |
| R8  | Mitigate | Change management program; early training         | Executive communication            |
| R9  | Mitigate | Security review in design; testing                | Emergency patch process            |
| R10 | Transfer | Contractual SLAs; alternatives identified         | Switch to backup vendor            |

### 13.5.4 Contingency Reserves

**Table 13.10:** Contingency Reserve Allocation

| Reserve Type     | Amount            | Purpose                   |
|------------------|-------------------|---------------------------|
| Budget Reserve   | 145,000 EGP (10%) | Cost overrun protection   |
| Schedule Reserve | 2 weeks           | Buffer for schedule risks |

## 13.6 Monitor and Control Risks

This is an ongoing process throughout the project lifecycle.

### 13.6.1 Monitoring Activities

- Track identified risks
- Identify new risks
- Execute response plans
- Evaluate response effectiveness

# Chapter 14

## Schedule Analysis

This chapter presents detailed schedule analysis calculations including Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), Free Float, and Total Float for all project activities.

### 14.1 Schedule Calculation Overview

#### 14.1.1 Definitions

##### **ES (Early Start)**

The earliest time an activity can start, considering all predecessors

##### **EF (Early Finish)**

$ES + Duration$  (earliest time an activity can finish)

##### **LS (Late Start)**

Latest time an activity can start without delaying the project

##### **LF (Late Finish)**

$LS + Duration$  (latest time an activity can finish without project delay)

**Total Float**       $LF - EF$  or  $LS - ES$  (maximum delay without delaying project)

**Free Float**      Minimum ES of successors - EF (delay without delaying successors)

#### 14.1.2 Calculation Methods

- **Forward Pass:** Calculate ES and EF from project start to finish
- **Backward Pass:** Calculate LS and LF from project finish to start
- **Float Calculations:** Determine scheduling flexibility for each activity

### 14.2 Complete Schedule Analysis Table

**Table 14.1:** Complete Schedule Analysis with ES, EF, LS, LF, and Float

| ID | Activity                     | Dur. (wks) | Predecessors | ES | EF | LS | LF | TF | FF | Critical? |
|----|------------------------------|------------|--------------|----|----|----|----|----|----|-----------|
| A  | Project Initiation           | 2          | -            | 0  | 2  | 0  | 2  | 0  | 0  | Yes       |
| B  | Requirements Gathering       | 4          | A            | 2  | 6  | 2  | 6  | 0  | 0  | Yes       |
| C  | Requirements Documentation   | 2          | B            | 6  | 8  | 6  | 8  | 0  | 0  | Yes       |
| D  | System Architecture Design   | 3          | C            | 8  | 11 | 8  | 11 | 0  | 0  | Yes       |
| E  | Database Design              | 2          | C            | 8  | 10 | 9  | 11 | 1  | 1  | No        |
| F  | UI/UX Design                 | 3          | C            | 8  | 11 | 9  | 12 | 1  | 1  | No        |
| G  | Backend Dev Setup            | 1          | D, E         | 11 | 12 | 11 | 12 | 0  | 0  | Yes       |
| H  | Prerequisite Engine Dev      | 3          | G            | 12 | 15 | 12 | 15 | 0  | 0  | Yes       |
| I  | Conflict Detection Module    | 2          | G            | 12 | 14 | 13 | 15 | 1  | 1  | No        |
| J  | Waitlist Management          | 2          | G            | 12 | 14 | 13 | 15 | 1  | 1  | No        |
| K  | Registration Workflow        | 3          | H, I, J      | 15 | 18 | 15 | 18 | 0  | 0  | Yes       |
| L  | Frontend Dev Setup           | 1          | F            | 11 | 12 | 12 | 13 | 1  | 1  | No        |
| M  | Student Portal Dev           | 4          | L            | 12 | 16 | 13 | 17 | 1  | 1  | No        |
| N  | Admin Dashboard Dev          | 3          | L            | 12 | 15 | 14 | 17 | 2  | 2  | No        |
| O  | Course Catalog Interface     | 2          | L            | 12 | 14 | 15 | 17 | 3  | 0  | No        |
| P  | Timetable Builder            | 3          | O            | 14 | 17 | 15 | 18 | 1  | 1  | No        |
| Q  | Frontend-Backend Integration | 2          | K, M, N, P   | 18 | 20 | 18 | 20 | 0  | 0  | Yes       |
| R  | SIS Integration              | 2          | Q            | 20 | 22 | 20 | 22 | 0  | 0  | Yes       |
| S  | Authentication Integration   | 1          | Q            | 20 | 21 | 21 | 22 | 1  | 1  | No        |
| T  | Email Notification System    | 1          | Q            | 20 | 21 | 21 | 22 | 1  | 1  | No        |
| U  | Unit Testing                 | 2          | K, Q         | 20 | 22 | 20 | 22 | 0  | 0  | No*       |
| V  | Integration Testing          | 2          | R, S, T, U   | 22 | 24 | 22 | 24 | 0  | 0  | Yes       |
| W  | System Testing               | 2          | V            | 24 | 26 | 24 | 26 | 0  | 0  | Yes       |
| X  | Performance Testing          | 1          | W            | 26 | 27 | 27 | 28 | 1  | 1  | No        |
| Y  | Security Testing             | 1          | W            | 26 | 27 | 27 | 28 | 1  | 1  | No        |
| Z  | User Acceptance Testing      | 2          | W, X, Y      | 27 | 29 | 27 | 29 | 0  | 0  | Yes       |
| AA | Deployment Planning          | 1          | Z            | 29 | 30 | 29 | 30 | 0  | 0  | Yes       |
| AB | Production Deployment        | 1          | AA           | 30 | 31 | 30 | 31 | 0  | 0  | Yes       |
| AC | Training Material Dev        | 2          | W            | 26 | 28 | 29 | 31 | 3  | 3  | No        |
| AD | Training Delivery            | 1          | AB, AC       | 31 | 32 | 31 | 32 | 0  | 0  | Yes       |
| AE | Go-Live Support              | 2          | AD           | 32 | 34 | 32 | 34 | 0  | 0  | Yes       |
| AF | Project Closure              | 1          | AE           | 34 | 35 | 34 | 35 | 0  | 0  | Yes       |

\*U is on a parallel critical path with zero float

### 14.3 Critical Path Activities

Activities with Total Float = 0 form the critical path(s):

**Primary Critical Path:**

A → B → C → D → G → H → K → Q → R → V → W → Z → AA → AB → AD  
→ AE → AF

**Parallel Critical Path:**

A → B → C → D → G → H → K → Q → U → V → W → Z → AA → AB → AD  
→ AE → AF

**Total Project Duration: 35 weeks**

### 14.4 Float Analysis

#### 14.4.1 Activities with Total Float > 0

**Table 14.2:** Non-Critical Activities with Scheduling Flexibility

| Activity ID | Activity Name                 | Total Float (weeks) | Implication                    |
|-------------|-------------------------------|---------------------|--------------------------------|
| AC          | Training Material Development | 3                   | Can be delayed without impact  |
| O           | Course Catalog Interface      | 3                   | Low priority for resources     |
| N           | Admin Dashboard Development   | 2                   | Some flexibility               |
| E           | Database Design               | 1                   | Near-critical, monitor closely |
| F           | UI/UX Design                  | 1                   | Near-critical, monitor closely |
| I           | Conflict Detection Module     | 1                   | Near-critical, monitor closely |
| J           | Waitlist Management           | 1                   | Near-critical, monitor closely |
| L           | Frontend Dev Setup            | 1                   | Near-critical, monitor closely |
| M           | Student Portal Development    | 1                   | Near-critical, monitor closely |
| P           | Timetable Builder             | 1                   | Near-critical, monitor closely |
| S           | Authentication Integration    | 1                   | Near-critical, monitor closely |
| T           | Email Notification System     | 1                   | Near-critical, monitor closely |
| X           | Performance Testing           | 1                   | Near-critical, monitor closely |
| Y           | Security Testing              | 1                   | Near-critical, monitor closely |

#### 14.4.2 Near-Critical Activities

Activities with Total Float  $\leq 1$  week are considered near-critical and require close monitoring:

- Any delay in these activities could make them critical
- Resource prioritization should favor these after critical activities
- Weekly status monitoring recommended
- Consider adding buffer resources if delays occur

## 14.5 Schedule Compression Opportunities

### 14.5.1 Fast-Tracking Opportunities

Activities that could potentially overlap:

1. Database Design (E) could start during Requirements Documentation (C) - saves 1 week
2. UI/UX Design (F) could start during Requirements Documentation (C) - saves 1 week
3. Training Material Development (AC) has 3 weeks float - can be done earlier

### 14.5.2 Crashing Opportunities

Critical activities that could be shortened by adding resources:

**Table 14.3:** Crashing Analysis for Critical Activities

| Activity                  | Normal Duration | Crashed Duration | Time Saved | Cost Increase |
|---------------------------|-----------------|------------------|------------|---------------|
| H (Prerequisite Engine)   | 3 weeks         | 2 weeks          | 1 week     | 50,000 EGP    |
| K (Registration Workflow) | 3 weeks         | 2 weeks          | 1 week     | 50,000 EGP    |
| M (Student Portal)        | 4 weeks         | 3 weeks          | 1 week     | 40,000 EGP    |
| W (System Testing)        | 2 weeks         | 1.5 weeks        | 0.5 week   | 30,000 EGP    |

## 14.6 Schedule Management Recommendations

1. **Focus on Critical Path:** Prioritize resources for critical activities ( $TF = 0$ )
2. **Monitor Near-Critical:** Weekly tracking of activities with  $TF \leq 1$  week
3. **Utilize Float Strategically:** Use non-critical activities as resource buffers
4. **Early Warning System:** Alert when critical activities show signs of delay
5. **Resource Leveling:** Use float to smooth resource demands
6. **Buffer Management:** Protect critical path with schedule buffers
7. **Fast-Track Carefully:** Only overlap activities when risks are acceptable

# Chapter 15

## Cost Estimation

This chapter presents cost estimation for the Online Course Registration Portal project using at least two different estimation methods. Any inconsistencies between methods are analyzed and justified.

### 15.1 Cost Estimation Overview

Accurate cost estimation is critical for project planning, budgeting, and financial management. This chapter applies multiple estimation techniques to ensure reliability and identify potential cost risks.

### 15.2 Method 1: Bottom-Up Estimation

Bottom-up estimation aggregates costs from the lowest level of the WBS up to the project total.

#### 15.2.1 Personnel Costs

**Table 15.1:** Personnel Costs - Bottom-Up Estimation

| Resource Role                 | Rate/Week (EGP) | Weeks | Quantity | Total Cost (EGP) |
|-------------------------------|-----------------|-------|----------|------------------|
| Project Manager               | 8,000           | 36    | 1        | 288,000          |
| Business Analyst              | 6,000           | 12    | 2        | 144,000          |
| System Architect              | 8,500           | 8     | 1        | 68,000           |
| Senior Developer              | 7,000           | 24    | 2        | 336,000          |
| Junior Developer              | 4,000           | 20    | 2        | 160,000          |
| Frontend Developer            | 6,000           | 18    | 2        | 216,000          |
| QA Engineer                   | 5,000           | 12    | 3        | 180,000          |
| Database Administrator        | 6,500           | 10    | 1        | 65,000           |
| UI/UX Designer                | 5,500           | 8     | 2        | 88,000           |
| DevOps Engineer               | 6,000           | 8     | 1        | 48,000           |
| Security Engineer             | 6,500           | 4     | 1        | 26,000           |
| Technical Writer              | 4,500           | 6     | 1        | 27,000           |
| Trainer                       | 5,000           | 4     | 2        | 40,000           |
| <b>Total Personnel Costs:</b> |                 |       |          | <b>1,686,000</b> |

### 15.2.2 Equipment and Infrastructure

**Table 15.2:** Equipment and Infrastructure Costs

| Item                          | Cost (EGP)     | Justification                       |
|-------------------------------|----------------|-------------------------------------|
| Development Workstations (10) | 120,000        | High-performance laptops/PCs        |
| Development Servers           | 150,000        | Staging and test environments       |
| Production Servers            | 200,000        | Redundant production infrastructure |
| Network Equipment             | 50,000         | Switches, load balancers            |
| <b>Total Equipment:</b>       | <b>520,000</b> |                                     |

### 15.2.3 Software and Licenses

**Table 15.3:** Software and License Costs

| Item                        | Cost (EGP)     |
|-----------------------------|----------------|
| Development Tools and IDEs  | 30,000         |
| Database Licenses           | 80,000         |
| Testing Tools               | 40,000         |
| Project Management Software | 15,000         |
| Security Tools              | 25,000         |
| Other Software Components   | 20,000         |
| <b>Total Software:</b>      | <b>210,000</b> |

### 15.2.4 Other Direct Costs

**Table 15.4:** Other Direct Costs

| Category                             | Cost (EGP)     |
|--------------------------------------|----------------|
| Training and Documentation Materials | 45,000         |
| Travel and Meetings                  | 25,000         |
| External Consultants (if needed)     | 50,000         |
| Miscellaneous Expenses               | 20,000         |
| <b>Total Other Costs:</b>            | <b>140,000</b> |

### 15.2.5 Bottom-Up Total

**Table 15.5:** Bottom-Up Cost Estimation Summary

| Category                         | Cost (EGP)       |
|----------------------------------|------------------|
| Personnel                        | 1,686,000        |
| Equipment and Infrastructure     | 520,000          |
| Software and Licenses            | 210,000          |
| Other Direct Costs               | 140,000          |
| <b>Subtotal:</b>                 | <b>2,556,000</b> |
| Contingency Reserve (10%)        | 255,600          |
| <b>Total Bottom-Up Estimate:</b> | <b>2,811,600</b> |

### 15.3 Method 2: Parametric Estimation

Parametric estimation uses statistical relationships between historical data and variables to calculate costs.

#### 15.3.1 Parametric Model Basis

Based on industry data for similar web-based student registration systems:

- **Parameter:** Function Points or Lines of Code
- **Estimated System Size:** 15,000 function points
- **Industry Cost per Function Point:** 150 EGP (for Egypt market, medium complexity)
- **Complexity Factor:** 1.2 (due to prerequisite validation and conflict detection)

#### 15.3.2 Parametric Calculation

$$\text{Base Development Cost} = 15,000 \times 150 = 2,250,000 \text{ EGP}$$

$$\text{Adjusted for Complexity} = 2,250,000 \times 1.2 = 2,700,000 \text{ EGP}$$

#### 15.3.3 Additional Costs (Parametric Method)

**Table 15.6:** Additional Costs for Parametric Method

| Category           | Cost (EGP)       | Basis                   |
|--------------------|------------------|-------------------------|
| Infrastructure     | 400,000          | 15% of development cost |
| Testing and QA     | 405,000          | 15% of development cost |
| Training           | 135,000          | 5% of development cost  |
| Project Management | 270,000          | 10% of development cost |
| <b>Subtotal:</b>   | <b>1,210,000</b> |                         |

#### 15.3.4 Parametric Total

**Table 15.7:** Parametric Cost Estimation Summary

| Category                          | Cost (EGP)       |
|-----------------------------------|------------------|
| Adjusted Development Cost         | 2,700,000        |
| Additional Costs                  | 1,210,000        |
| <b>Subtotal:</b>                  | <b>3,910,000</b> |
| Contingency Reserve (10%)         | 391,000          |
| <b>Total Parametric Estimate:</b> | <b>4,301,000</b> |

### 15.4 Method 3: Analogous (Comparative) Estimation

Analogous estimation uses actual costs from similar previous projects.

### 15.4.1 Comparative Projects

**Table 15.8:** Reference Projects for Analogous Estimation

| Project                   | Year | Cost (EGP) | Similarity                        |
|---------------------------|------|------------|-----------------------------------|
| University Library System | 2023 | 2,800,000  | High (similar size, less complex) |
| Alumni Portal             | 2022 | 1,600,000  | Medium (smaller scale)            |
| Faculty Management System | 2023 | 3,200,000  | High (similar complexity)         |

### 15.4.2 Adjustment Factors

- **Size Adjustment:** +15% (larger user base)
- **Complexity Adjustment:** +20% (prerequisite rules, conflict detection)
- **Technology Adjustment:** -5% (modern tech stack, better tools)
- **Team Experience:** -5% (experienced team)
- **Inflation Adjustment:** +8% (2024-2025)

### 15.4.3 Analogous Calculation

Average of comparable projects:  $(2,800,000 + 3,200,000)/2 = 3,000,000$  EGP

$$\begin{aligned}\text{Adjusted Cost} &= 3,000,000 \times 1.15 \times 1.20 \times 0.95 \times 0.95 \times 1.08 \\ &= 3,000,000 \times 1.26 \\ &= 3,780,000 \text{ EGP}\end{aligned}$$

**Table 15.9:** Analogous Cost Estimation Summary

| Category                         | Cost (EGP)       |
|----------------------------------|------------------|
| Base Analogous Estimate          | 3,000,000        |
| Adjustments Applied              | +780,000         |
| <b>Adjusted Estimate:</b>        | <b>3,780,000</b> |
| Contingency Reserve (10%)        | 378,000          |
| <b>Total Analogous Estimate:</b> | <b>4,158,000</b> |

## 15.5 Comparison and Reconciliation

**Table 15.10:** Cost Estimation Method Comparison

| Method                     | Total Estimate (EGP) | Variance from Mean |
|----------------------------|----------------------|--------------------|
| Bottom-Up Estimation       | 2,811,600            | -22.3%             |
| Parametric Estimation      | 4,301,000            | +18.9%             |
| Analogous Estimation       | 4,158,000            | +14.9%             |
| <b>Mean:</b>               | <b>3,756,867</b>     | -                  |
| <b>Recommended Budget:</b> | <b>3,800,000</b>     | -                  |

## 15.6 Analysis of Inconsistencies

### 15.6.1 Why Bottom-Up is Lower

1. May not account for all hidden costs and overhead
2. Based on optimistic duration and resource estimates
3. Could miss some integration complexities
4. Assumes high resource efficiency (may be unrealistic)
5. Does not fully account for rework and changes

### 15.6.2 Why Parametric and Analogous are Higher

1. Include industry averages which account for typical overruns
2. Factor in organizational overhead and inefficiencies
3. Based on actual outcomes (not optimistic plans)
4. Include buffer for unknowns and complexities
5. More conservative and risk-aware

### 15.6.3 Justification of Differences

The variance between methods is expected and valuable:

- Bottom-up provides detailed baseline but may be optimistic
- Parametric and analogous provide reality check based on actual experience
- Range indicates uncertainty level in estimation
- Higher estimates account for historical cost growth patterns

## 15.7 Approved Project Budget

After review by project stakeholders, the following budget has been approved for the project:

**Table 15.11:** Approved Project Budget

| Category                           | Amount (EGP)     | Notes                                   |
|------------------------------------|------------------|---|
| Personnel Costs (Development Team) | 800,000          | Optimized team structure                |
| Project Management                 | 150,000          | 36-week duration                        |
| Software Licenses and Tools        | 100,000          | Open-source alternatives where possible |
| Hardware and Infrastructure        | 200,000          | Cloud-based approach                    |
| Testing and QA                     | 120,000          | Integrated testing approach             |
| Training and Documentation         | 80,000           | In-house development                    |
| Contingency Reserve (10%)          | 145,000          | Risk buffer                             |
| <b>Total Approved Budget:</b>      | <b>1,595,000</b> |   |

### 15.7.1 Reconciliation with Detailed Estimates

The approved budget (1,595,000 EGP) is lower than the detailed estimates presented above. This difference is justified by:

1. **Scope Optimization:** Focus on core functionality with phased enhancement approach
2. **Resource Efficiency:** Experienced team with higher productivity
3. **Technology Choices:** Use of open-source tools and cloud infrastructure
4. **Lean Approach:** Agile methodology reducing rework and waste
5. **In-house Expertise:** Reduced external consulting needs

The detailed estimates (2.8M - 4.3M EGP) serve as reference points for risk assessment and represent maximum cost scenarios if significant changes occur.

## 15.8 Cost Control and Monitoring

- Track actual vs. estimated costs weekly
- Earned Value Management (EVM) for performance measurement
- Variance analysis and corrective actions
- Change control for scope changes affecting cost
- Reserve management and authorization procedures

# **Chapter 16**

## **References**

This chapter lists all sources and references consulted during the preparation of this project management plan.

### **16.1 Textbook**

1. Schwalbe, K. (2011). *Information Technology Project Management* (Revised 6th ed.). Course Technology.

### **16.2 References**

1. Kerzner, H. (2009). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* (10th ed.). John Wiley & Sons, Inc.
2. Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*.

## **Chapter 17**

### **Video Presentations**

#### **Team Member Video Explanations**

##### **Video Repository:**

<https://drive.google.com/drive/u/1/folders/1dr5xRk30AKyWjyDliWtCDawcQUmmHkCt>