

System Analysis & Design

Higher Nationals

Internal verification of assessment decisions – BTEC (RQF)

INTERNAL VERIFICATION – ASSESSMENT DECISIONS			
Programme title	HND in Computing		
Assessor		Internal Verifier	
Unit(s)	Unit 34: System Analysis & Design		
Assignment title	Automated system for E-Solutions Private Limited		
Student's name	Ryan Wickramaratne		
List which assessment criteria the Assessor has awarded.	Pass	Merit	Distinction
INTERNAL VERIFIER CHECKLIST			
Do the assessment criteria awarded match those shown in the assignment brief?	Y/N		
Is the Pass/Merit/Distinction grade awarded justified by the assessor's comments on the student work?	Y/N		
Has the work been assessed accurately?	Y/N		
Is the feedback to the student: Give details: • Constructive? • Linked to relevant assessment criteria? • Identifying opportunities for improved performance? • Agreeing actions?	Y/N Y/N Y/N Y/N		
Does the assessment decision need amending?	Y/N		
Assessor signature		Date	
Internal Verifier signature		Date	
Programme Leader signature (if required)		Date	

Confirm action completed			
Remedial action taken Give details:			
Assessor signature		Date	
Internal Verifier signature		Date	



Programme Leader signature (if required)		Date	
--	--	------	--

Higher Nationals - Summative Assignment Feedback Form

Student Name/ID	Ryan Wickramaratne (COL 00081762)		
Unit Title	Unit 34: System Analysis & Design		
Assignment Number	1	Assessor	
Submission Date	2/2/2023	Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	

Assessor Feedback:

LO1 Evaluate the strengths and weaknesses of the traditional and agile systems analysis methodologies

Pass, Merit & Distinction P1 M1 D1
 Descripts

LO2 Produce a feasibility study for a system for a business-related problem

Pass, Merit & Distinction P2 M2
 Descripts

LO3 Analyse their system using a suitable methodology.

Pass, Merit & Distinction P3 M3 D2
 Descripts

LO4 Design the system to meet user and system requirements.

Pass, Merit & Distinction P4 M4
 Descripts

Grade:	Assessor Signature:	Date:
Resubmission Feedback:		
Grade:	Assessor Signature:	Date:
Internal Verifier's Comments:		
Signature & Date:		

* Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment board.

Pearson Higher Nationals in Computing

Unit 34: Systems Analysis & Design
Assignment 01

General Guidelines

1. A Cover page or title page – You should always attach a title page to your assignment. Use previous page as your cover sheet and make sure all the details are accurately filled.
2. Attach this brief as the first section of your assignment.
3. All the assignments should be prepared using a word processing software.
4. All the assignments should be printed on A4 sized papers. Use single side printing.
5. Allow 1" for top, bottom , right margins and 1.25" for the left margin of each page.

Word Processing Rules

1. The font size should be **12 point**, and should be in the style of **Time New Roman**.
2. **Use 1.5 line spacing**. Left justify all paragraphs.
3. Ensure that all the headings are consistent in terms of the font size and font style.
4. Use **footer function in the word processor to insert Your Name, Subject, Assignment No, and Page Number on each page**. This is useful if individual sheets become detached for any reason.
5. Use word processing application spell check and grammar check function to help editing your assignment.

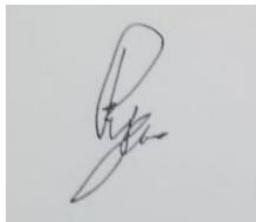
Important Points:

1. It is strictly prohibited to use textboxes to add texts in the assignments, except for the compulsory information. eg: Figures, tables of comparison etc. Adding text boxes in the body except for the before mentioned compulsory information will result in rejection of your work.
2. Avoid using page borders in your assignment body.
3. Carefully check the hand in date and the instructions given in the assignment. Late submissions will not be accepted.
4. Ensure that you give yourself enough time to complete the assignment by the due date.
5. Excuses of any nature will not be accepted for failure to hand in the work on time.
6. You must take responsibility for managing your own time effectively.
7. If you are unable to hand in your assignment on time and have valid reasons such as illness, you may apply (in writing) for an extension.
8. Failure to achieve at least PASS criteria will result in a REFERRAL grade .
9. Non-submission of work without valid reasons will lead to an automatic RE FERRAL. You will then be asked to complete an alternative assignment.
10. If you use other people's work or ideas in your assignment, reference them properly using HARVARD referencing system to avoid plagiarism. You have to provide both in-text citation and a reference list.
11. If you are proven to be guilty of plagiarism or any academic misconduct, your grade could be reduced to A REFERRAL or at worst you could be expelled from the course

Student Declaration

I hereby, declare that I know what plagiarism entails, namely to use another's work and to present it as my own without attributing the sources in the correct form. I further understand what it means to copy another's work.

1. I know that plagiarism is a punishable offence because it constitutes theft.
2. I understand the plagiarism and copying policy of Edexcel UK.
3. I know what the consequences will be if I plagiarise or copy another's work in any of the assignments for this program.
4. I declare therefore that all work presented by me for every aspect of my program, will be my own, and where I have made use of another's work, I will attribute the source in the correct way.
5. I acknowledge that the attachment of this document signed or not, constitutes a binding agreement between myself and Pearson , UK.
6. I understand that my assignment will not be considered as submitted if this document is not attached to the assignment.

ryandilthusha@gmail.com**Student's Signature:**
*(Provide E-mail ID)***2/2/2023****Date:**
(Provide Submission Date)

Higher National Diploma in Computing

Assignment Brief

Student Name /ID Number	Ryan Wickramaratne (COL 00081762)
Unit Number and Title	Unit 4: Systems Analysis & Design
Academic Year	2021/22
Unit Tutor	Ms. Prasadini
Assignment Title	Automated system for E-Solutions Private Limited
Issue Date	
Submission Date	2/2/2023
IV Name & Date	

Submission format

The submission should be in the form of an individual written report written in a concise, formal business style using single spacing and font size 12. You are required to make use of headings, paragraphs, and subsections as appropriate, and all work must be supported with research and referenced. Please provide in-text citations, reference list and bibliography using Harvard referencing system. Please also provide a bibliography using the Harvard referencing system.

The recommended word limit is not less than 5000 words, although you will not be penalised for exceeding the total word limit.

Unit Learning Outcomes:

LO1 Evaluate the strengths and weaknesses of the traditional and agile systems analysis methodologies.

LO2 Produce a feasibility study for a system for a business-related problem.

LO3 Analyse their system using a suitable methodology.

LO4 Design the system to meet user and system requirements.

Assignment Brief and Guidance:

*Please note that assignment guidance is for reference only and should be more specific in detail to meet customized needs.

Assignment brief

Case study

The new automated system is designed to replace the current, manual, error-prone process of E-Solutions private Limited. The automation of existing process is to reduce the company's expenses and enhance the productivity significantly. This transformation also would support for:

- 1) Successful teams working
- 2) Completing projects on time and within budget due to a better understanding of system requirements and tasks to be completed
- 3) Starting projects on time through automated project scheduling system.

In the proposed system, the Project director creates a project and a “project profile” for each project. The creation of the project profile includes identification of project employee costs, the assignment of tasks to the project, and the assignment of a project manager. The project profile is consisted of project id, project personnel cost, a list of tasks assigned, and the project manager. The Project director also creates the teams for a given project, assigns employees to the teams, and assigns a team leader. The Project manager is responsible for assigning tasks to various teams working on the projects(s). The Team Leader assigns tasks to the team members.

Additional functionality includes:

- a. Produce and update information about different software projects, project teams, specific team member assignments and team skills.
- Perform function point analysis to identify the personnel cost of the project and provide information to generate invoices upon completion of project phases.
- Monitor projects and identify completed tasks and ongoing tasks of each project.

Activity 01

Discuss traditional and agile system analysis methodologies used in the industry by comparing and contrasting the strengths and weaknesses of them. Critically evaluate two methodologies by referring to the examples to support your answer.

Activity 2

Produce a feasibility report for the scenario given above and assess the importance of feasibility criteria used for the system investigation. Critically evaluate the strengths and weaknesses of feasibility study with relevant to the proposed solution.

Activity 3

Analyse and review the system requirements of the proposed solution given in the scenario using a suitable methodology. Functional and non-functional requirements of the system should be clearly mentioned. Assessment of the effectiveness and suitability of the chosen methodology should be provided with proper justifications.

Activity 4

Produce a system design specification for the above scenario and assess the effectiveness of your design and the methodology used with reference to how it meets the user requirements.

Your system design specification should include architectural design, interface design, database design, and program design.

--	--

Grading Criteria	Achieved	Feedback
LO1 Evaluate the strengths and weaknesses of the traditional and agile systems analysis methodologies.		
P1 Discuss the strengths and weaknesses of the traditional and agile systems analysis methodologies.		
M1 Compare and contrast the strengths and weaknesses of the traditional and agile systems analysis methodologies.		
LO2 Produce a feasibility study for a system for a business-related problem.		
P2 Produce a feasibility study for a system for a business related problem.		
M2 Evaluate the relevance of the feasibility criteria on the systems investigation for the business related problem.		
LO1 & LO2 D1 Critically evaluate the strengths and weaknesses of the traditional and agile methodologies and feasibility study.		
LO3 Analyse their system using a suitable Methodology		

P3 Review a system using a suitable methodology for a business-related problem.		
M3 Analyse the effectiveness of the methodology used in providing a solution for a given business context.		
LO4 Design the system to meet user and system Requirements		
P4 Design a fully functional system to meet user and system requirements for the business related problem.		
M4 Assess the effectiveness of the system design with reference to the methodology used and how the design meets user and system requirements.		
LO3 & LO4 D2 Justify the choice of the analysis methodology used in the context of the business problem.		

Acknowledgement

I would like to express my special thanks of gratitude to my Systems Analysis & Design lecturer Ms. Prasadini for providing invaluable guidance and giving immense amount of knowledge to work on this assignment perfectly. I specially thanks her because she helped us in doing a lot of research and I came to know about so many new things about the Systems Analysis & Design.

Secondly, I would like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Executive Summary

This entire assignment is based on the implementation of a Project Management system design for the selected company (E-Solutions Pvt Ltd). The purpose of this assignment is to improve Systems Analysis skills & Designing skills.

This report includes basic system analysis and designing theories with the System Development Life Cycle details relating to the given scenario. System designing methodologies, business case and feasibility study, investigation techniques and tools to investigate, user and system requirements, documentation for different phases, design elements and features, and designing techniques and tools have been discussed in this assignment.

Abbreviations

Systems Analysis and Design (SAD)
System Development Life Cycle (SDLC)
Project Life Cycle (PLC)
Software Requirement Specification (SRS)
Static Application Security Testing (SAST)
Software Requirement Specification (SRS)
Software Design Document (SDD)
WBS (Work Breakdown Structure)
Business Requirements Document (BRD)
User Requirements Document (URD)
Software Requirements Specification (SRS)
Joint Application Development (JAD)
Entity Relationship Diagram (ERD)
Database Management Systems (DBMS)
Unified Modeling Language (UML)
Data Flow Diagram (DFD)
American Society of Mechanical Engineers (ASME)
Unified Modeling Language (UML)
User experience (UX)

List of figures

FIGURE 1. 1 SEVEN PHASES OF THE SYSTEM DEVELOPMENT LIFE CYCLE.....	25
FIGURE 1. 2 WATERFALL MODEL	33
FIGURE 1. 3 WHAT IS PHOTOTYPE	38
FIGURE 1. 4 PHOTOTYPE MODEL.....	40
FIGURE 1. 5 SPIRAL MODEL	45
FIGURE 1. 6 THE 4 VALUES OF THE AGILE MANIFESTO	53
FIGURE 1. 7 THE 12 PRINCIPLES OF THE AGILE MANIFESTO	54
FIGURE 1. 8 AGILE DEVELOPMENT METHOD'S CHARACTERISTICS	55
FIGURE 1. 9 SCRUM PROCESS	60
FIGURE 1. 10 XP (EXTREME PROGRAMMING) PROCESS.....	68
FIGURE 1. 11 FEASIBILITY STUDY.....	93
FIGURE 2. 1 BUSINESS CASE	76
FIGURE 2. 2 EXAMPLE BUSINESS CASE.....	86
FIGURE 2. 3 BUSINESS CASE TEMPLATE 1	87
FIGURE 2. 4 BUSINESS CASE TEMPLATE 2	88
FIGURE 2. 5 DESIRABILITY, VIABILITY AND FEASIBILITY DIAGRAM.....	89
FIGURE 2. 6 DESIRABILITY	90
FIGURE 2. 7 VIABILITY	91
FIGURE 2. 8 FEASIBILITY	92
FIGURE 2. 9 WORK BREAKDOWN STRUCTURE FOR THE PROJECT	120
FIGURE 2. 10 GNATT CHART FOR THE PROJECT 1.....	121
FIGURE 2. 11 GNATT CHART FOR THE PROJECT 2	122
FIGURE 2. 12 GNATT CHART FOR THE PROJECT 3.....	122
FIGURE 2. 13 GNATT CHART FOR THE PROJECT 1.....	123
FIGURE 2. 14 PROJECT BUDGET PLAN	124
FIGURE 4. 1 ER DIAGRAMS SYMBOLS & NOTATIONS	175
FIGURE 4. 2 STUDENT ENTITY	177
FIGURE 4. 3 RELEVANT ATTRIBUTES FOR STUDENT ENTITY.....	177
FIGURE 4. 4 ENTITY RELATIONSHIP DIAGRAM FOR E-SOLUTIONS PVT LTD	179
FIGURE 4. 5 LOGICAL DATABASE DESIGN FOR E-SOLUTIONS PVT LTD	182
FIGURE 4. 6 CLASS DIAGRAM FOR E-SOLUTIONS PVT LTD	184
FIGURE 4. 7 USE CASE DIAGRAM FOR E-SOLUTIONS PVT LTD	187
FIGURE 4. 8 DATA FLOW DIAGRAM NOTATIONS	190
FIGURE 4. 9 EXAMPLE CONTEXT-LEVEL DIAGRAM (LEVEL 0).....	191
FIGURE 4. 10 EXAMPLE LEVEL 1 DFD	192
FIGURE 4. 11 EXAMPLE LEVEL 2 DFD	193
FIGURE 4. 12 LOGICAL DATA FLOW DIAGRAM (GROCERY STORE).....	195
FIGURE 4. 13 PHYSICAL DATA FLOW DIAGRAM (GROCERY STORE)	196
FIGURE 4. 14 DATA FLOW DIAGRAM FOR E-SOLUTIONS PVT LTD (LEVEL 0)	197
FIGURE 4. 15 DATA FLOW DIAGRAM FOR E-SOLUTIONS PVT LTD (LEVEL 1)	198
FIGURE 4. 16 FLOW CHART SYMBOLS & NOTATIONS	200
FIGURE 4. 17 FLOW CHART DIAGRAM FOR E-SOLUTIONS PVT LTD	201
FIGURE 4. 18 ACTIVITY DIAGRAM SYMBOLS & NOTATIONS	203

FIGURE 4. 19 ACTIVITY DIAGRAM FOR E-SOLUTIONS PVT LTD (FOR THE STAFF)	204
FIGURE 4. 20 ACTIVITY DIAGRAM FOR E-SOLUTIONS PVT LTD (FOR THE PROJECT MANAGER)	205
FIGURE 4. 21 ACTIVITY DIAGRAM FOR E-SOLUTIONS PVT LTD (FOR THE PROJECT DIRECTOR)	206
FIGURE 4. 22 SEQUENCE DIAGRAM SYMBOLS & NOTATIONS WITH AN EXAMPLE	208
FIGURE 4. 23 SEQUENCE DIAGRAM FOR E-SOLUTIONS PVT LTD (FOR THE STAFF).....	209
FIGURE 4. 24 SEQUENCE DIAGRAM FOR E-SOLUTIONS PVT LTD (FOR THE PROJECT MANAGER)	210
FIGURE 4. 25 SEQUENCE DIAGRAM FOR E-SOLUTIONS PVT LTD (FOR THE PROJECT DIRECTOR).....	211
FIGURE 4. 26 WIREFRAME DIAGRAM BASIC COMPONENTS.....	212
FIGURE 4. 27 WIREFRAME DIAGRAM EXAMPLE.....	213
FIGURE 4. 28 WIREFRAME DIAGRAMS FOR E-SOLUTIONS PVT LTD - LOGIN PAGE.....	214
FIGURE 4. 29 WIREFRAME DIAGRAMS FOR E-SOLUTIONS PVT LTD - HOME PAGE	214
FIGURE 4. 30 WIREFRAME DIAGRAMS FOR E-SOLUTIONS PVT LTD - EMPLOYEE DEATILS PAGE	215
FIGURE 4. 31 WIREFRAME DIAGRAMS FOR E-SOLUTIONS PVT LTD - PROJECT PROFILE PAGE	215
FIGURE 4. 32 WIREFRAME DIAGRAMS FOR E-SOLUTIONS PVT LTD - TEAM DETAILS PAGE.....	216
FIGURE 4. 33 WIREFRAME DIAGRAMS FOR E-SOLUTIONS PVT LTD - INVOICING PAGE	216
FIGURE 4. 34 WIREFRAME DIAGRAMS FOR E-SOLUTIONS PVT LTD - PROJECT PROFILES PAGE	217
FIGURE 4. 35 WIREFRAME DIAGRAMS FOR E-SOLUTIONS PVT LTD - PROJECTS STATUS PAGE	217
FIGURE 4. 36 WIREFRAME DIAGRAMS FOR E-SOLUTIONS PVT LTD - REVENUE PAGE.....	218
FIGURE 4. 37 PROTOTYPE DIAGRAMS FOR E-SOLUTIONS PVT LTD - LOGIN PAGE.....	220
FIGURE 4. 38 PROTOTYPE DIAGRAMS FOR E-SOLUTIONS PVT LTD - HOME PAGE.....	220
FIGURE 4. 39 PROTOTYPE DIAGRAMS FOR E-SOLUTIONS PVT LTD - EMPLOYEE DETAILS PAGE	221
FIGURE 4. 40 PROTOTYPE DIAGRAMS FOR E-SOLUTIONS PVT LTD - PROJECT PROFILE PAGE	221
FIGURE 4. 41 PROTOTYPE DIAGRAMS FOR E-SOLUTIONS PVT LTD - INVOICING PAGE	222
FIGURE 4. 42 PROTOTYPE DIAGRAMS FOR E-SOLUTIONS PVT LTD - TEAM DETAILS PAGE.....	222
FIGURE 4. 43 PROTOTYPE DIAGRAMS FOR E-SOLUTIONS PVT LTD - PROJECTS PROFILES PAGE	223
FIGURE 4. 44 PROTOTYPE DIAGRAMS FOR E-SOLUTIONS PVT LTD - PROJECTS STATUS PAGE	223
FIGURE 4. 45 PROTOTYPE DIAGRAMS FOR E-SOLUTIONS PVT LTD - REVENUE PAGE.....	224
FIGURE 4. 46 FEEDBACK FORM PART 1	232
FIGURE 4. 47 FEEDBACK FORM PART 2	233
FIGURE 4. 48 FEEDBACK FORM PART 3	234
FIGURE 4. 49 FEEDBACK FORM PART 4	235
FIGURE 4. 50 LOGIN PAGE INTERFACE.....	238
FIGURE 4. 51 HOME PAGE INTERFACE.....	239
FIGURE 4. 52 EMPLOYEE DETAILS PAGE INTERFACE	240
FIGURE 4. 53 PROJECT PROFILE PAGE INTERFACE	241
FIGURE 4. 54 INVOICING PAGE INTERFACE	242
FIGURE 4. 55 TEAM DETAILS PAGE INTERFACE	243
FIGURE 4. 56 PROJECT PROFILES PAGE INTERFACE.....	244
FIGURE 4. 57 PROJECTS STATUS PAGE INTERFACE	245
FIGURE 4. 58 REVENUE PAGE INTERFACE.....	246

List of Tables

TABLE 1. 1 WATERFALL MODEL VS PHOTOTYPE MODEL	49
TABLE 1. 2 WATERFALL MODEL VS SPIRAL MODEL	50
TABLE 1. 3 PHOTOTYPE MODEL VS SPIRAL MODEL	51
TABLE 1. 4 COMPARISON BETWEEN TRADITIONAL METHODOLOGY AND AGILE METHODOLOGY.....	71
TABLE 2. 1 PROJECT GOAL AND OBJECTIVES	110
TABLE 2. 2 SCOPE OF THE PROJECT - E-SOLUTIONS SOFTWARE TECHNOLOGIES	115
TABLE 2. 3 COMPARISON OF CURRENT AND PROPOSED SYSTEM.....	128
TABLE 3. 1 DIFFERENCES BETWEEN STRUCTURED AND UNSTRUCTURED INTERVIEWS.....	152
TABLE 3. 2 OPEN-ENDED QUESTIONNAIRES VS CLOSE-ENDED QUESTIONNAIRES	156
TABLE 3. 3 TRADITIONAL VS AGILE METHODOLOGY	166
TABLE 3. 4 SCRUM VS XP	169

TABLE OF CONTENTS

ACTIVITY 1	22
1.1 SYSTEMS ANALYSIS AND DESIGN (SAD)	22
1.1.1 <i>Introduction to Systems Analysis and Design (SAD)</i>	22
1.1.2 <i>Importance of System Analysis and Design</i>	23
1.2 SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)	24
1.2.1 <i>Introduction to System Development Life Cycle (SDLC)</i>	24
1.2.2 <i>Seven phases of the System Development Life Cycle (SDLC)</i>	25
1.3 SOFTWARE DEVELOPMENT METHODOLOGIES	29
1.3.1 <i>Introduction to software development methodologies</i>	29
1.4 TRADITIONAL METHODOLOGY	30
1.4.1 <i>Introduction to Traditional Methodology</i>	30
1.5 STRENGTHS AND WEAKNESSES OF TRADITIONAL METHODOLOGY	31
1.5.1 <i>Strengths (Advantages) of Traditional Methodology</i>	31
1.5.2 <i>Weaknesses (Disadvantages) of Traditional Methodology</i>	31
1.6 TYPES OF TRADITIONAL METHODOLOGIES.....	32
1.6.1 <i>Waterfall Model</i>	32
1.6.2 <i>Phototype Model</i>	38
1.6.3 <i>Spiral Model</i>	44
1.7 TRADITIONAL METHODOLOGIES COMPARISON.....	49
1.7.1 <i>Waterfall Model vs Phototype Model</i>	49
1.7.2 <i>Waterfall Model vs Spiral Model</i>	50
1.7.3 <i>Phototype Model vs Spiral Model</i>	51
1.8 AGILE METHODOLOGY	52
1.8.1 <i>Introduction to Agile Methodology with the history</i>	52
1.8.2 <i>The Agile Development Method's characteristics</i>	55
1.9 STRENGTHS AND WEAKNESSES OF AGILE METHODOLOGY	58
1.9.1 <i>Strengths (Advantages) of Agile Methodology</i>	58
1.9.2 <i>Weaknesses (Disadvantages) of Agile Methodology</i>	58
1.10 TYPES OF AGILE METHODOLOGIES.....	59
1.10.1 <i>Scrum Model</i>	59
1.10.2 <i>XP (eXtreme Programming) Model</i>	67
1.11 COMPARISON BETWEEN TRADITIONAL METHODOLOGY AND AGILE METHODOLOGY.....	71
1.12 BEST SOFTWARE DEVELOPMENT METHODOLOGY FOR A PROJECT AS MY PERSPECTIVE	73
1.13 HOW TO CHOOSE BEST SOFTWARE DEVELOPMENT METHODOLOGY FOR A PROJECT	74
1.14 CONCLUSION TO ACTIVITY 1.....	75
ACTIVITY 2	76
2.1 BUSINESS CASE.....	76
2.1.1 <i>Introduction to Business Case</i>	76
2.1.2 <i>Why use Business Case</i>	77
2.1.3 <i>Steps required to make a Business Case</i>	78
2.1.4 <i>Elements of Business Case</i>	80
2.1.5 <i>Advantages of using Business Case</i>	84
2.1.6 <i>Disadvantages of using Business Case</i>	85
2.1.7 <i>Business Case example</i>	86

1.1.7 Business Case templates	87
2.2 DESIRABILITY, VIABILITY AND FEASIBILITY OF SYSTEMS	89
2.2.1 Introduction to the topic	89
2.2.2 Introduction to Desirability	90
2.2.3 Introduction to Viability	91
2.2.4 Introduction to Feasibility	92
2.3 FEASIBILITY STUDY	93
2.3.1 Introduction to Feasibility Study	93
2.3.2 Why use Feasibility Study.....	93
2.3.3 Steps to conduct feasibility study.....	94
2.3.4 Elements of Feasibility Study.....	96
2.3.5 Types of Feasibility Study	98
2.3.6 Advantages of Feasibility Study	100
2.3.7 Disadvantages of Feasibility Study.....	100
2.4 INVESTIGATION TECHNIQUES.....	101
2.4.1 Introduction to Investigations	101
2.4.2 Introduction to Interviews.....	102
2.4.3 Introduction to Record Inspection.....	103
2.4.3 Introduction to Observations	105
2.5 FEASIBILITY STUDY REPORT	107
ACTIVITY 3	135
3.1 REQUIREMENTS	135
3.1.1 Introduction to Requirements in Systems Analysis & Design	135
3.1.2 Types of Requirements in Systems Analysis & Design.....	136
3.2 SYSTEM REQUIREMENTS.....	139
3.2.1 Introduction to System Requirements in Systems Analysis & Design.....	139
3.2.2 Introduction to Functional Requirements	140
3.2.3 Examples of Functional Requirements	140
3.2.4 Functional Requirements for the E-Solutions Private Limited New System	141
3.2.5 Introduction to Non-Functional Requirements.....	143
3.2.6 Examples of Non-Functional Requirements	144
3.2.7 Non-Functional Requirements for the E-Solutions Private Limited	145
3.3 REQUIREMENTS GATHERING TECHNIQUES	147
3.3.1 Review of Procedural Forms.....	147
3.3.2 Interviews.....	149
3.3.3 Questionnaires	154
3.3.4 Observations	157
3.3.5 Focus Groups.....	160
3.3.6 Discovery Prototyping	162
3.3.7 Joint requirements planning (JRP).....	164
3.4 METHODOLOGY AND MODEL TO DEVELOP A SYSTEM FOR E-SOLUTIONS PRIVATE LIMITED	166
3.4.1 Choosing Methodology for the proposed system.....	166
3.4.2 Choosing Model from selected methodology for the proposed system	169
3.5 CONCLUSION TO ACTIVITY 3.....	172
ACTIVITY 4	173
4.1 ENTITY RELATIONSHIP DIAGRAM (ERD)	173
4.1.1 Introduction to Entity Relationship Diagram	173
4.1.2 Why use Entity Relationship Diagram (Benefits).....	174
4.1.3 ER Diagrams Symbols & Notations	175

4.1.4 Basic 3 components of the ER Diagram	176
4.1.5 Entity Relationship Diagram for E-Solutions Pvt Ltd	179
4.2 LOGICAL DATABASE DESIGN	180
4.2.1 Introduction to Logical Database Design	180
4.2.2 Why use Logical Database Design (Benefits)	181
4.2.3 Logical Database Design for E-Solutions Pvt Ltd.....	182
4.3 CLASS DIAGRAM	183
4.3.1 Introduction to Class Diagram	183
4.3.2 Why use Class Diagram (Benefits)	183
4.3.3 Class Diagram for E-Solutions Pvt Ltd	184
4.4 USE CASE DIAGRAM	185
4.4.1 Introduction to Use Case Diagram	185
4.4.2 Use Case Diagram main components	185
4.4.3 Why using the Use Case Diagram (Benefits)	186
4.4.4 Use Case Diagram for E-Solutions Pvt Ltd	187
4.5 DATA FLOW DIAGRAM (DFD)	188
4.5.1 Introduction to Data Flow Diagram (DFD)	188
4.5.2 Why use Logical Data Flow Diagram (Benefits)	188
4.5.3 Data Flow Diagram Symbols & Notations.....	189
4.5.4 Data Flow Diagram Levels.....	191
4.5.5 Difference between a logical DFD and physical DFD.....	194
4.5.6 Logical Data Flow Diagram benefits with an example.....	194
4.5.7 Physical Data Flow Diagram benefits with an example	195
4.5.8 Data Flow Diagram for E-Solutions Pvt Ltd (Level 0).....	197
4.5.9 Data Flow Diagram for E-Solutions Pvt Ltd (Level 1).....	198
4.6 FLOW CHART DIAGRAM	199
4.6.1 Introduction to Flow Chart Diagram	199
4.6.2 Why use Flow Chart Diagram (Benefits)	200
4.6.3 Flow Chart Symbols & Notations.....	200
4.6.4 Flow Chart Diagram for E-Solutions Pvt Ltd.....	201
4.7 ACTIVITY DIAGRAM	202
4.7.1 Introduction to Activity Diagram.....	202
4.7.2 Activity Diagram Basic Components	202
4.7.3 Why use Activity Diagram (Benefits).....	203
4.7.4 Activity Diagram Symbols & Notations.....	203
4.7.5 Activity Diagram for E-Solutions Pvt Ltd (For the Staff)	204
4.7.6 Activity Diagram for E-Solutions Pvt Ltd (For the Project Manager).....	205
4.7.7 Activity Diagram for E-Solutions Pvt Ltd (For the Project Director).....	206
4.8 SEQUENCE DIAGRAM.....	207
4.8.1 Introduction to Sequence Diagram	207
4.8.2 Sequence Diagram Basic Components	207
4.8.3 Why use Sequence Diagram (Benefits)	208
4.8.4 Sequence Diagram Symbols & Notations.....	208
4.8.5 Sequence Diagram for E-Solutions Pvt Ltd (For the Staff)	209
4.8.6 Sequence Diagram for E-Solutions Pvt Ltd (For the Project Manager)	210
4.8.7 Sequence Diagram for E-Solutions Pvt Ltd (For the Project Director)	211
4.9 WIREFRAME DIAGRAM	212
4.9.1 Introduction to Wireframe Diagram	212
4.9.2 Wireframe Diagram Basic Components.....	212
4.9.3 Why use Wireframe Diagram (Benefits)	213
4.9.4 Wireframe Diagrams for E-Solutions Pvt Ltd	214

4.10 PROTOTYPE DIAGRAM	219
4.10.1 <i>Introduction to Prototype Diagram</i>	219
4.10.2 <i>Why use Prototype Diagram (Benefits)</i>	219
4.10.3 <i>Prototype Diagrams for E-Solutions Pvt Ltd</i>	220
4.11 TOOLS AND TECHNIQUES	225
4.11.1 <i>Tools and Techniques that can use to create ERD Diagram</i>	225
4.11.2 <i>Tools and Techniques I used to create ERD Diagram</i>	225
4.11.3 <i>Tools and Techniques that can use to create Logical Database Diagram</i>	225
4.11.4 <i>Tools and Techniques I used to create Logical Database Diagram</i>	226
4.11.5 <i>Tools and Techniques that can use to create Class Diagram</i>	226
4.11.6 <i>Tools and Techniques I used to create Class Diagram</i>	226
4.11.7 <i>Tools and Techniques that can use to create Use Case Diagram</i>	227
4.11.8 <i>Tools and Techniques I used to create Use Case Diagram</i>	227
4.11.9 <i>Tools and Techniques that can use to create Data Flow Diagram</i>	227
4.11.10 <i>Tools and Techniques I used to create Data Flow Diagram</i>	228
4.11.11 <i>Tools and Techniques that can use to create Flow Chart Diagram</i>	228
4.11.12 <i>Tools and Techniques I used to create Flow Chart Diagram</i>	228
4.11.13 <i>Tools and Techniques that can use to create Activity Diagram</i>	229
4.11.14 <i>Tools and Techniques I used to create Activity Diagram</i>	229
4.11.15 <i>Tools and Techniques that can use to create Sequence Diagram</i>	229
4.11.16 <i>Tools and Techniques I used to create Sequence Diagram</i>	230
4.11.17 <i>Tools and Techniques that can use to create Wireframe Diagram</i>	230
4.11.18 <i>Tools and Techniques I used to create Wireframe Diagram</i>	230
4.11.19 <i>Tools and Techniques that can use to create Prototype Diagram</i>	231
4.11.20 <i>Tools and Techniques I used to create Prototype Diagram</i>	231
4.12 SYSTEM FEEDBACK	232
4.13 USER DOCUMENTATION FOR E-SOLUTIONS PVT LTD PROJECT MANAGEMENT SYSTEM.....	236
4.13 CONCLUSION TO ACTIVITY 4.....	247
CONCLUSION	248
REFERENCES.....	249

Activity 1

1.1 Systems Analysis and Design (SAD)

1.1.1 Introduction to Systems Analysis and Design (SAD)

The capacity to obtain, arrange, and evaluate information is essential for business success. Systems analysis and design is indeed a proven methodology that enables both large and small enterprises to fully utilize information and benefit from it. Systems Analysis and Design (SAD) is a fascinating, dynamic field where analysts are constantly learning new methods and strategies to create systems that are more productive and effective. Despite the strategy or methodology chosen, there are some fundamental abilities that all analysts must possess.

Planning, assessment, designing, and implementation are the four stages that any information systems project goes through. Every project needs analysts to gather requirements, model the company's needs, and draw out plans for the system's construction. and every project has to comprehend ideas in organizational behaviour like change management and team development.

Many businesses use the system analysis and design process to assess specific company challenges and create solutions using more effective techniques. This procedure can be used by businesses to restructure their operations or to achieve their growth and profitability targets. Additionally, system analysis and design frequently focus on how systems behave, their connections to other subsystems, and their capacity to achieve a certain objective. This frequently includes analysing a system's output quality and performance.

([What Is System Analysis and Design, 2022](#))

1.1.2 Importance of System Analysis and Design

The most typical advantage of system analysis and design is the enhancement of an existing system and greater operational effectiveness. Additional advantages that an organization may get from this technique are listed below.

- Making it possible to understand complex structures.
- Making it possible to manage any business changes better.
- Coordinating the organization's actions with its surroundings and key objectives.
- Decreasing IT problems and lowering the pressure on IT staff.
- Lowering expenses in some areas, conserving the company's money and allowing it to employ those resources in other areas.
- The early detection of potential risks and dangers to the processes.
- Enhancing the system's overall quality.
- Enhancing the system's usability for staff.
- Both productivity and customer happiness are increasing.

1.2 System Development Life Cycle (SDLC)

1.2.1 Introduction to System Development Life Cycle (SDLC)

A multi-step, iterative approach with a methodical structure is the System Development Life Cycle (SDLC). In order to produce a quality system that meets or exceeds a business's expectations or to manage the progression of decision-making, this process is used to model or offer a framework for both technical and non-technical operations. The systems-development life cycle previously included five stages. There are now seven phases in all. Systems analysts were able to design clearer activities to accomplish specified goals by raising the number of stages.

The System Development Life Cycle (SDLC) employs a systems approach to define a process, much like a project life cycle (PLC). When an IT project is being developed, it is frequently used and followed. Different phases (or processes) of the development process are highlighted by the SDLC. Users may observe and comprehend what actions are involved inside a given stage by using the life cycle approach. Users are also informed that whatever time the system has to be changed or improved, steps can be repeated, or an earlier step can be changed.

(The Seven Phases of the System-Development Life Cycle, n.d.)

12.2 Seven phases of the System Development Life Cycle (SDLC)



Figure 1. 1 Seven phases of the System Development Life Cycle

1) Preliminary Phase

To accomplish a business's strategic goals, the team decides in the first phase whether or not a new system is required. This is a basic business plan to see if the business can find the resources to enhance a service or develop a certain infrastructure. Finding out the magnitude of the problem and developing potential solutions is the major goal of this step. Costs, rewards, time, resources, and others are some of the factors to think about in this circumstance. The success of the project as a whole depends on the outcome of this step, making it the most important. Before proceeding to the next stage, much investigation is necessary.

2) Analysis Phase

Teams will focus on the cause of their issue or the need for a change during the second phase. In the event that an issue needs to be solved, potential solutions are proposed and evaluated to determine which one best fits the project's main goal. Teams here think about the needs for the solution's functionality. Collecting all the specific information needed for a new system and coming up with the initial prototype concepts are both tasks of the analysis stage.

- Any need for a prototype system could be defined by developers.
- Alternatives to the current prototypes could be assessed by developers.
- The needs of end users could be determined by developers through analysis and research.
- A software requirement specification, often known as an SRS document, is also commonly produced by developers.

All the requirements for the system they intend to develop are listed here, including the software, hardware, and network specifications. By doing this, they will avoid using up too many funds or resources while collaborating with other development teams in the same location.

3) Design Phase

Before moving on to the major developer stage, the design phase is required. The developers will first describe the specifics of the entire program in detail, such as its User interfaces, System interfaces, Network and the requirements of the network, and Databases. The SRS document they produced will normally be transformed into a more logical structure that may then be implemented in a computer language. Plans for maintenance, operation, and training will all be created so that moving forward, developers are aware of what they must accomplish at each point of the cycle. Development managers will create a design document once everything is finished, which will be used as a guide throughout the System Development Life Cycle's later steps.

4) Development Phase

The stage of development is when programmers actually write the application's code in accordance with the preceding design papers and laid-out specifications. Tools for Static Application Security Testing (SAST) are useful in this situation. The design document's specifications are followed when creating the product's computer code. The actual development process ought to be rather simple given the extensive planning and outlining that has already taken place. The organization's coding standards will be followed by developers, who will also make use of various tools like interpreters, debuggers, and compilers. Standard programming languages include C++, PHP, and others. Depending on the requirements and specifications of the project, developers will decide which programming language to employ.

5) Testing Phase

The creation of software is not the end. To ensure that there are no issues and that the end-user experiences won't be compromised at any stage, it should now be tested. Developers thoroughly review their software throughout the testing phase, identifying any flaws or defects that must be identified, corrected, and then retested. It's crucial that the final version of the software adheres to the quality requirements outlined in the SRS document. Testing can be a very quick step or take a very long time, depending on the expertise of the developers, the complexity of the software, and the requirements of the end user.

6) Implementation Phase

The full software design will be completed after testing. Developers will work to include various modules or designs into the main source code, frequently using training environments to find further problems or flaws. Eventually, the information system will be created and integrated into its surroundings. The software is technically ready for market after clearing this stage and may be made available to any end customers.

7) Maintenance Phase

When software is sold on the market, the SDLC is not over. The time has come for developers to enter maintenance mode and start practising any tasks needed to address concerns reported by end users. Developers are also in charge of making any modifications that the software may require after it has been deployed. This can involve fixing remaining flaws that could not be fixed prior to launch or resolving brand-new problems that emerge as a result of user complaints. In comparison to smaller systems, larger systems could need more extensive maintenance phases.

1.3 Software development methodologies

1.3.1 Introduction to software development methodologies

Since the 1950s, there have been various techniques for developing software. It also constantly improves with new developments and techniques. Software engineers utilize different software development approaches to address issues in each generation. Increasing the workflow's structure is one of the main goals of software development engineers. Additionally, having the appropriate techniques helps with this by providing the workflow with a well-organized and integrated image.

A set of procedures used in the development of software is known as a software development methodology. They function as a type of regulated communication that enables us to establish rules for how the information will be distributed among a group of individuals. In the lifecycle of the entire software, these phases are defined. Consequently, we may argue that these techniques are software development procedures that also carry their underlying ideologies.

Because of this, software engineers have access to a wide variety of software development methodologies. Every one of these software development methodologies has its own procedures, traditions, and set of rules. It could be challenging for us software developers to choose the best methodology for our project. In this assignment, we will thus talk about how to pick the appropriate software development approaches for our projects.

The 2 main methodologies used in software development are as follows.

- 1) Traditional Methodology.
- 2) Agile Methodology.

(Types of Software Development Methodologies and Their Pros and Cons, 2021)

1.4 Traditional Methodology

1.4.1 Introduction to Traditional Methodology

The basis of traditional software development approaches is the software development lifecycle's pre-organized phases and stages. In this case, development proceeds in a single direction from requirements to design to development to testing to maintenance. Each phase in traditional methodologies, such as the Waterfall model, includes distinct deliverables and complete documentation that has undergone a thorough procedure.

When requirements are clearly recognized, traditional methods work well. For instance, in fields like construction, in which everyone is aware of the desired outcome. Traditional development techniques, on the other hand, may fail to accomplish project goals in fast-changing industries like IT.

Therefore, the problem statement or business need must be defined well in advance when employing the traditional methodology. Additionally, the answer must be decided upon beforehand and cannot be altered. And there is no room for modification or change after project development has begun; the whole set of requirements must be provided in the initial phase.

Only when the requirements are clear-cut are traditional development approaches appropriate. To put it another way, when the client is certain of what they want and is able to guarantee that there won't be any significant scope major changes throughout the project's progress. Large projects like maintenance projects, whose requirements are modest and there is a lot of room for ongoing adjustment, are not appropriate for it.

There are 3 major models that belong to the Traditional Methodology

- 1) Waterfall Model
- 2) Prototype Model
- 3) Spiral Model

(KPI-Partners-News-Team, 2018)

1.5 Strengths and Weaknesses of Traditional Methodology

1.5.1 Strengths (Advantages) of Traditional Methodology

- The likelihood of running into flaws, errors, or complicated issues during the project's planning, designing, testing, and final execution is extremely low.
- For smaller, sequential initiatives, the companies gladly choose the Traditional Methodology. They take less time to finish, require fewer developers, and don't require the usage of sophisticated equipment.
- Remote projects are very suitable to traditional projects because they are simpler and smaller. As a result, these projects may be created by a smaller team and the challenges can be overcome with little communication.
- Since it doesn't require any prior knowledge or training to start a project, traditional project management is a flexible option.
- When employing this methodology, the project manager has a clear understanding of the project.
- This approach is useful for risk management.
- There is thorough documentation for this procedure.

1.5.2 Weaknesses (Disadvantages) of Traditional Methodology

- Large and sophisticated projects should not be handled using this methodology.
- The customer's feedback doesn't always have a lot of room in the development process. Until the product is finished, the clients are not involved or accepting of the full development process.
- Due to a lack of collaborative teamwork, unity, and progress, the time management problem might occur in traditional methodologies.
- Team members must exhibit a high degree of discipline when using this strategy.
- The documentation for this approach can be quite extensive.

1.6 Types of Traditional Methodologies

1.6.1 Waterfall Model

Introduction →

The core life cycle model for software development is called the waterfall model. It is straightforward but idealistic. This model was quite popular in the past but is no longer in use. However, it is crucial since the waterfall model serves as the foundation for all other software development life cycle models.

The Software Development Life Cycle is divided into several phases according to the waterfall approach. According to this model, each phase may begin following the completion of an earlier phase. In other words, the output of one phase serves as the input for the previous step. As a result, the development process can be viewed as a waterfall development process in a linear.

When to use →

When the requirements are precise and fixed and may not alter frequently, this model is applicable. Additionally, this model can be applied if there are no requirements that are unclear and cause confusion. With a solid understanding of the technology, this model is beneficial to use. When the projects are small and the workforce is small, this model is usually employed. This model can be applied if the risk is 0% or very low.

Diagram and explanation of phases →

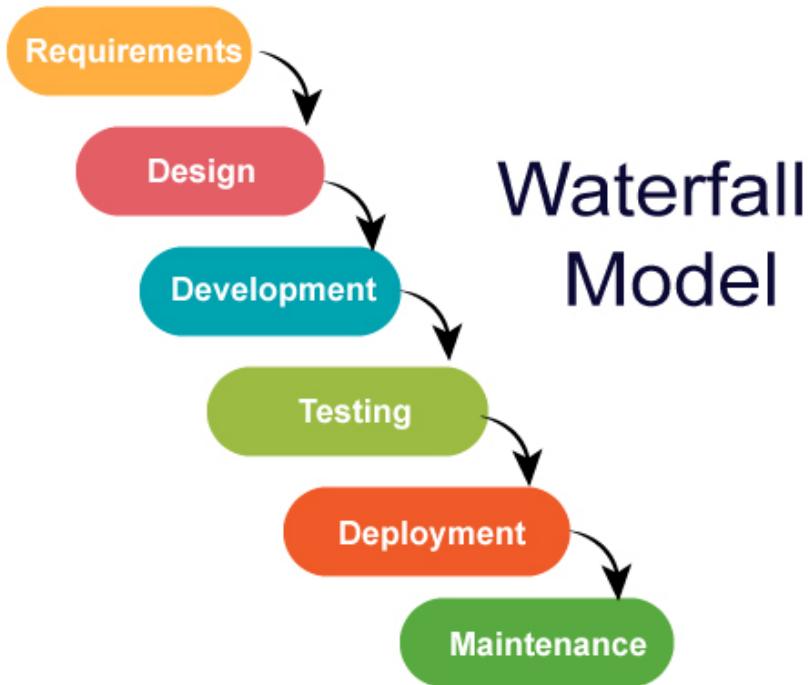


Figure 1.2 Waterfall Model

In the image above, the different waterfall model phases are represented sequentially. The graphic above shows that the phases do not cross over. Now let's take a quick look at each of these phases.

1. Requirement gathering and analysis

Understanding the exact needs of the client and effectively capturing them are the goals of the requirement analysis and specification phase. There are two distinct activities in this phase.

First, all of the customer's software needs are obtained, and then those collected requirements are evaluated. The analysis section's objective is to eliminate incompleteness. This means when portions of the real criteria are missing, the requirement is said to be incomplete. Eliminating inconsistencies is another objective of the analysis phase. When one aspect of a requirement conflicts with another, it is said to be inconsistent.

Secondly, Software Requirement Specification (SRS) document contains these evaluated requirements. A relationship between the development team and clients is established in the software requirement specification document. The SRS document can be used to resolve any potential conflicts between clients and developers.

2. Design

In this phase, the required specifications from the initial phase are examined, and the system design is created. This phase's objective is to translate the requirements gathered in the Software Requirement Specification (SRS) into a format that can be expressed through coding. The general software architecture is included, together with both high-level and detailed design. All of this work is documented using a Software Design Document (SDD).

3. Deployment / Coding

The system is initially built as small programs known as units, which are then incorporated into the following phase, using inputs from the design phase. In a process known as unit testing, each unit is created and then tested for functionality. Software design is converted into source code during this phase using any appropriate programming language. Each designed component is therefore coded. The goal of the unit testing step is to determine whether or not each module is functioning correctly.

4. Integration and Testing

Different modules are integrated as soon as they have completed unit testing and coding. Over a number of steps, the integration of various units is done incrementally. The system is tested after each integration phase, which involves adding previously designed modules to the partially integrated system. System testing is then done on the completed, completely functional system that has been obtained after all the modules have been successfully integrated and tested.

There are three different categories of testing activities that make up system testing. Alpha testing is the first one. The system testing carried out by the development team is known as

alpha testing. Beta testing is the second. The system testing performed by a friendly group of clients is known as beta testing. Acceptance testing comes in third. The customer carried out acceptance testing to decide whether to accept or reject the provided software once it had been delivered.

5. Deployment

After the software has been thoroughly tested in the testing phase, the software is made live in the production / real environment. Once the product has undergone functional and non-functional testing, it is either published to the market or deployed in the customer's environment.

6. Maintenance

The part of a software life cycle that requires maintenance is the most crucial. 60% of the total work required to produce complete software is spent on maintenance. Maintenance can be broadly divided into three categories. Corrective maintenance comes first. Errors that were not found during the product development stage are fixed by this type of maintenance. Perfective Maintenance is the next. Based on the customer's request, this form of maintenance is done to improve the system's functions. Adaptive Maintenance is the third option. When transferring software to a new environment, such as working on a brand-new computer platform or with a new operating system, adaptive maintenance is typically necessary.

This stage assists with a few problems that may arise in the client environment. Patches are published to address certain problems. Additionally, improved versions of the product are issued.

Advantages →

In terms of software development, the waterfall method is idealistic. It can be used as the foundation for various software development life cycle models due to its simplicity. basic. Some of the main benefits of this model are listed below.

- The simplicity and readability of this approach are outstanding.
- In this model, each phase is processed separately.
- The model's stages are each precisely defined.
- The milestones in this model are highly distinct and understandable.
- The process, the activities, and the outcomes are very clearly documented.
- This methodology encourages good practices like defining before designing and designing before coding.
- For smaller projects and those with clearly defined requirements, this model performs effectively.

Disadvantages →

Since we can't utilize the waterfall model in big projects due to its many flaws, we instead employ other software development lifecycle models that are based on the traditional waterfall model. Some of the main drawbacks of this model are listed below.

- There is no feedback path in this model. The evolution of software from one phase to the next is similar to a waterfall in the traditional waterfall concept. It is based on the concept that developers never make mistakes at any stage. Because of this, it lacks error correction.
- It's challenging for this model to respond to change requests. In order for this model to work, it must be assumed that all customer requirements can be accurately and completely established at the start of the project. however, the needs of clients truly change with time. Any requests for changes after the requirements specification phase is over are challenging to fulfil.

- There is no phase overlap in this model. According to this approach, a new phase should only begin after the preceding one has ended. This, however, cannot be maintained in actual projects. Phases may overlap for cost- and efficiency-saving reasons.

1.6.2 Phototype Model

Introduction →

The process of creating a functional replica of a system or product that needs to be engineered is referred to as prototyping. It provides a miniature version of the final product and is employed for gathering client feedback as presented below.

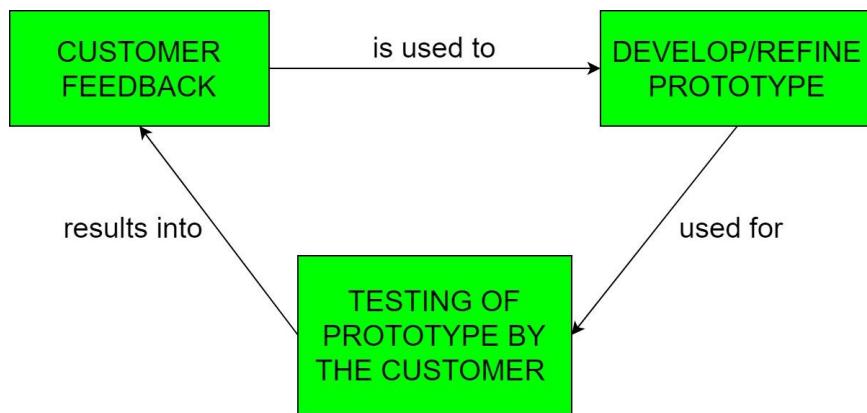


Figure 1. 3 What is phototype

When customers are unsure about the precise project requirements in advance, companies employ this paradigm. This process starts with creating a prototype of the finished product, which is then constantly tested and improved based on consumer response, until a final, acceptable prototype is produced, which serves as a foundation for creating the finished product.

The system is partially developed before or during the analysis phase in this process model, allowing clients to look at the product upfront in its project lifecycle. Customer interviews and the creation of an imperfect high-level paper model are the first steps in the process. Building the initial prototype using only the key features requested by the customer is performed using this document. The prototype is even further improved to address the issues once the customer has identified them. Until the user accepts the prototype and determines that the functioning model is suitable, the process is carried out.

When to use →

When the clients are unaware of the precise project specifications beforehand, the prototyping model is used. This process starts with creating a prototype of the finished product, which is then constantly tested and improved based on consumer input, until a final, acceptable prototype is produced, which serves as the basis for creating the finished product.

When the proposed system requires a lot of engagement with the end users, a prototype model should then be adopted. Prototype models are typically best suited for online systems and web interfaces that interact heavily with end users. Building a system that allows for simplicity of use and requires little training for the end user may take some time. Prototyping makes ensuring that users are constantly interacting with the software and offering comments that are incorporated into the prototype to produce a usable solution. For creating effective systems for human-computer interfaces, Prototype Model is excellent.

Diagram and explanation of phases →

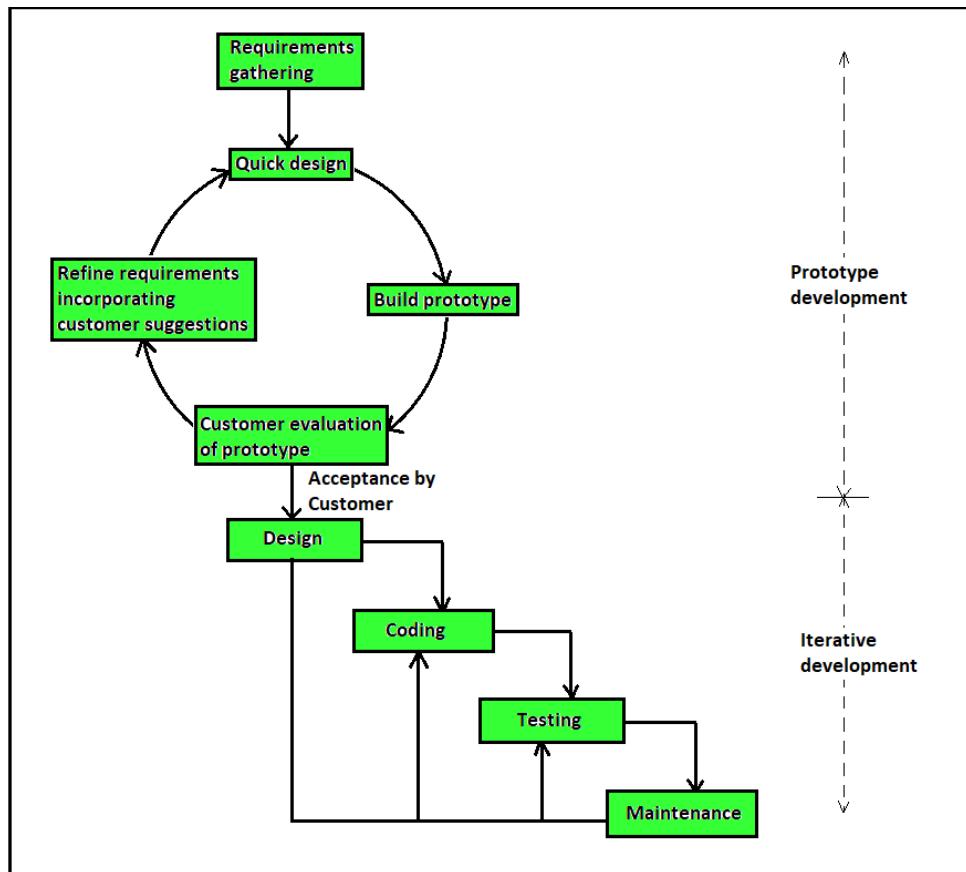


Figure 1. 4 Phototype Model

In the image above, the software development prototyping model is represented graphically. The creation of prototypes and iterative waterfall-based software development are the two main processes used to create the software.

1. Prototype Development

Initially obtaining requirements is the first step in prototype creation. A prototype is constructed after a rapid design is completed. The created prototype is given to the client for evaluation. The prototype is appropriately modified based on client feedback as well as enhanced specifications. Up until the customer approves the prototype, this cycle of getting customer feedback and making changes to it continues.

2. Iterative Development

Iterative waterfall development is used to create the final software after the client approves the prototype. Even when a functioning prototype is present, the Software Requirements Specification (SRS) document is typically required because it is essential for carrying out design structure analysis, verification, and test case design in later stages.

Normally, the prototype's source code is discarded. But when it comes to creating the actual software, the experience gained from creating the prototype is quite helpful. Many customer requirements are accurately specified by building the prototype and submitting it for user review, while technical problems are overcome by experimenting with the prototype. By doing this, the costs of further client change requests and redesigns are reduced to a minimum.

Advantages →

- An active role is played by this model. It is simpler to design the model in this case since consumers are actively participating in the development process, making it easier to include their preferences. Due to the users' active participation, problems are found early on, simplifying the procedure.
- Missing functionality in this model is easily detectable. It is simple to identify the prototype model's missing features. The risks of failure are decreased as a result.
- In this approach, prompt feedback is a favourable characteristic. Feedback from customers is provided much more quickly because they can interact directly with the prototype model. Since they are taken into account when developing the final system, these responses are important. The project can be immediately modified by the developers once the clients quickly share their ideas and report any adjustments that are necessary.
- The major positive aspect of this model is customer satisfaction. In terms of client satisfaction, the prototype model performs better. Early on, the consumer has the opportunity to actually experience and feel the product, which helps them better

understand its requirements. Additionally, customers love the pleasure of helping to create the operational version of their idea.

- The model of the prototype is flexible. It is easily adaptable to the tastes and requirements of the client or the developer. The prototype can also be reused by the developer for use in later, more challenging projects.
- Money can be saved with this model. In the early stages of the project, a prototype model can help with error detection. The project's total cost and duration are therefore decreased. The ability to create prototype models helps the developer anticipate costs in areas that weren't initially planned for. Before the adjusting process gets expensive, it addresses the modifications that must be made to the project.

Disadvantages →

- The creation of the prototype model takes a lot of time. Multiple prototypes are tested before the final product is developed, which takes a lot of time.
- There may be misunderstandings about how the finished product will arrive. Early on, the consumer has the opportunity to interact directly with the prototype. Because of this, the buyer might believe that the actual product will likewise come earlier than expected, which could result in confusion. This might not always be the fact.
- Making bad decisions can have an impact on the result. A high-quality product is what the developer always strives for. However, they could make poor choices about the prototype's quality while rushing to create it, which could have an impact on the final product.
- It is possible for the final version to be misunderstood. Customers may become disappointed and frustrated with the prototype model and lose all interest in the final product. Customers may believe that the final version will have the same flaws even though it is enhanced and optimized. Another significant issue is that a feature that was included in the prototype model's early stages may be appreciated by the client.

It is possible, nevertheless, that the specific component will be eliminated later. In both situations, the client misunderstands the end product.

- It's possible that the developer will focus on a particular prototype and neglect to do a thorough evaluation of the entire project. Because of this, the developer may ignore more suitable alternatives, and forget about important details, and the project as a whole might be poorly engineered, requiring complicated maintenance.

1.6.3 Spiral Model

Introduction →

One of the most significant models for the Software Development Life Cycle that supports risk handling is the spiral model. It resembles a spiral with numerous loops when represented diagrammatically. The spiral's precise number of loops is unclear and varies from project to project. In the software development process, each spiral loop is referred to as a Phase. Depending on the project risks, the project manager might alter the precise number of phases required to build the product.

The project manager plays a crucial role in the spiral model of product development since they dynamically set the number of phases. The spiral's radius at any given point symbolizes the project's overall costs, while the angular dimension symbolizes the current phase's progress to date.

When to use →

When working on a huge project, as was previously indicated, software developers employ the spiral model. Large projects may necessitate revisions from time to time. So when the production of a prototype is appropriate, this model is used. The project's risks increase in level as it expands in size. Spiral approach, as opposed to other traditional software development methodologies, can be effective for medium- to high-risk projects. Additionally, the project's scope may result in ambiguous or complex requirements, which calls for the use of the spiral model.

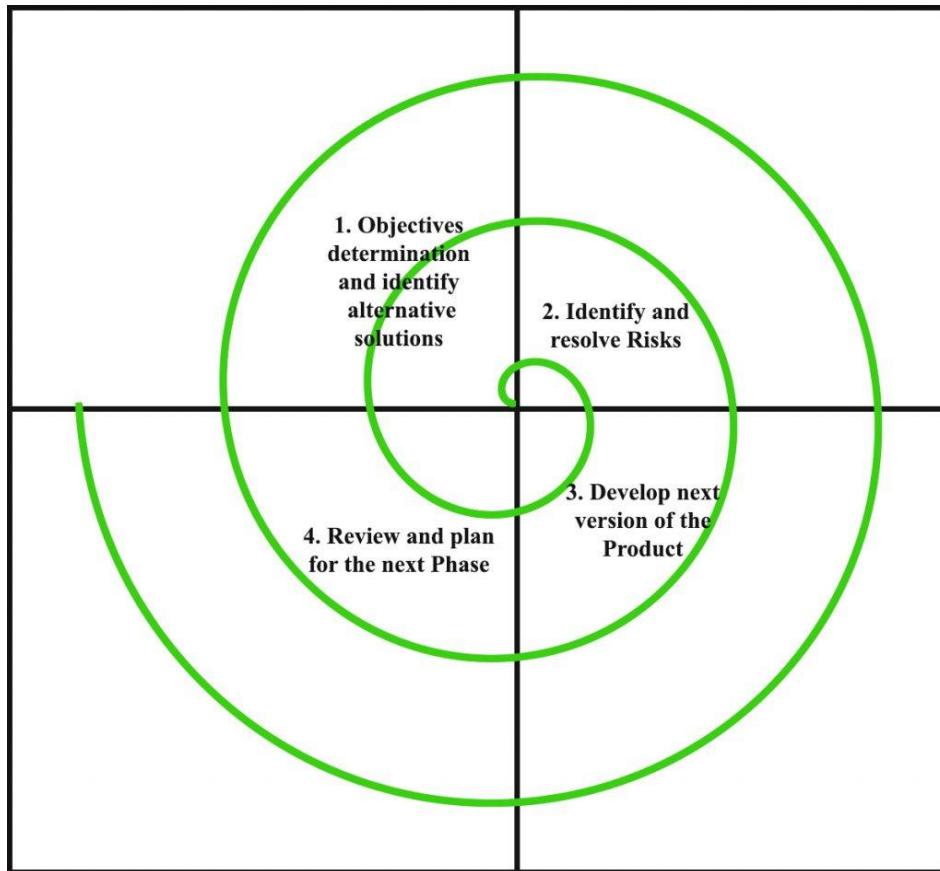
Diagram and explanation of phases →

Figure 1. 5 Spiral Model

According to the above diagram, the four quadrants represent each phase of the spiral model. These four quadrants' functions are detailed below.

1. Objectives determination and identify alternative solutions

At the beginning of each step, the objectives are specified, developed upon, and analyzed while requirements are acquired from the clients. Then, in this quadrant, other solutions that might be appropriate for the phase are offered.

2. Identify and resolve Risks

All potential solutions are assessed in the second quadrant in order to choose the best option. The risks connected to that solution are then determined, and the risks are dealt with in the best way possible. The Prototype is constructed for the ideal outcome at the end of this quadrant.

3. Develop the next version of the Product

The third quadrant is when the specified features are created and tested. The next software version is accessible at the completion of the third quadrant.

4. Review and plan for the next Phase

The customers assess the software's currently created version in the fourth quadrant. Planning for the following phase is then begun.

The phases are divided into quadrants, and each quadrant can be further divided into the steps that take place within it. The spiral model's steps can be broadly extended as follows.

1. We provide as much information as we can on the new system requirements. A number of users who represent all the internal and external users as well as various aspects of the current system must typically be interviewed in order to do this.
2. For the new system, an initial design is made.
3. The initial design of the new system is used to build the first prototype. This system is typically scaled-down and serves as an approximation of the final product's qualities.
4. Four steps are used to create a second prototype. The first step is to assess the original prototype's strengths, flaws, and problems. The second is then outlining what the second prototype will need to be. Afterwards, the second prototype is planned and created. Creating and evaluating the second prototype is the last step.

5. If the risk is assessed to be too high, the entire project may be abandoned. Risk elements could include operating cost misjudgments, development cost overruns, and other things that could lead to a poor final product.
6. A new prototype may be created from the current one using the four-step process described above, once the present one has been evaluated similarly to how the previous prototype was evaluated.
7. The previous processes are repeated until the customer is convinced that the improved prototype accurately reflects the desired final product.
8. On the basis of the improved prototype, the finalized system is built.
9. Tests and evaluations of the finished system are performed in-depth. To avoid catastrophic failures and save downtime, routine maintenance is continuously performed.

Advantages →

- The spiral model is an excellent choice for significant, complicated projects, as was previously stated. Given the model's progressive nature, developers may divide a large project into manageable chunks and work on one feature at a time, guaranteeing that nothing is missing.
- Additionally, since this prototype is constructed piece by piece, it is sometimes simpler to estimate the whole project cost.
- Since this approach is more adaptable, modifications to the requirements made after work has begun can be quickly adopted and implemented.
- Every phase of the spiral model includes risk analysis and management, increasing security and increasing the likelihood that threats and malfunctions won't occur. The iterative development method makes risk management easier.
- In this model, customer satisfaction is higher. Customer feedback is facilitated via the spiral model. Customers will be able to observe and assess their product at every stage if the software is being developed for them. This saves the development team time and money by allowing them to express their dissatisfaction or request adjustments before the product is entirely constructed.

Disadvantages →

- Since the spiral model is expensive, it cannot be used for smaller projects.
- This model's weakness is that it is dependent on risk analysis. Personnel working on the project must be skilled in risk assessment because the successful completion of the project depends on appropriate risk management.
- When compared to other SDLC alternatives, the spiral model is much more complex. It requires strict adherence to protocols in order to function properly. Since the model includes intermediate phases, there is also more documentation.
- It's challenging to manage time using this model. The number of phases that must be completed before the project can be considered complete makes time management nearly difficult. As a result, there is always a chance of running over budget or falling behind schedule.

1.7 Traditional Methodologies Comparison

1.7.1 Waterfall Model vs Prototype Model

Table 1. 1 Waterfall Model vs Prototype Model

Waterfall Model	Prototype Model
Software development models like the waterfall model follow a sequential process.	A prototype is produced, tested, and then improved according to customer needs in the prototype model of software development.
Risk analysis is highlighted.	The importance of risk analysis is rarely highlighted.
The waterfall model has a significant amount of risk.	It is appropriate for projects with a high degree of risk.
This allows for quick initial reviews.	It is impossible to do quick initial reviews in this.
It works best when the client's needs are perfectly clear.	It works best when the client's requirements are unclear and have to be adjusted.
User involvement in this process is still in its early stages.	High user involvement is shown in this.
There is less code writing because it facilitates automatic code generation.	The automatic generation of code is not supported.
Given the nature of the model and how each phase follows the other, errors get more complex.	The prototype makes it possible for the developer to find any flaws early in the process, which reduces the complexity of an error.
The waterfall model has limited flexibility.	Prototype models are highly flexible.

1.7.2 Waterfall Model vs Spiral Model

Table 1. 2 Waterfall Model vs Spiral Model

Waterfall Model	Spiral Model
The Waterfall model is straightforward and easy.	The spiral model is far more complicated.
This functions in a sequential manner.	The evolutionary approach makes this work.
After the various processes are complete, any flaws or risks in this model are found and fixed.	Errors or dangers in this model are recognized and fixed earlier.
Small projects can use this model.	Large projects use this model.
Early planning and model requirements are essential.	If necessary, this model's requirements and early planning are essential.
The waterfall concept has limited flexibility.	It's not difficult to be flexible with the spiral model.
The waterfall model has a significant amount of risk.	In the spiral model, the risk is minimized.
It is relatively cheap to use a waterfall model.	The spiral model, however, is exceedingly expensive.
In the waterfall model, there is minimal customer involvement.	Customer involvement is quite high in the spiral model.
It takes the least amount of maintenance.	Regular maintenance is necessary.
The framework type is linear.	It is based on a framework that is both linear and iterative.
Reusability is highly possible.	Reusability is possible in some circumstances.
The customer has relatively limited influence over the administrator.	Unlike a waterfall paradigm, customers have influence over the administrator.

1.7.3 Phototype Model vs Spiral Model

Table 1. 3 Phototype Model vs Spiral Model

Phototype Model	Spiral Model
This method of software development entails the creation, testing, and further modification of a prototype in accordance with client requirements.	This paradigm, which is based on incremental, waterfall, or evolutionary prototyping techniques, is one for risk-driven software development.
A rapid or closed-ended prototype is another name for this model.	The term "meta model" is also used to describe this model.
The following phases are used - <ol style="list-style-type: none"> 1. Requirements 2. Quick Design 3. Build a Prototype 4. User Evaluation 5. Refining Prototype 6. Implement and Maintain 	The following phases are used – <ol style="list-style-type: none"> 1. Planning Phase 2. Risk Analysis Phase 3. Engineering Phase 4. Evaluation Phase
Risk analysis is not highlighted.	Risk analysis is seriously evaluated, and a different course of action is selected.
Customers interact continuously with the prototype model up till the approved final version.	There is no ongoing consumer interaction in the spiral model.
It works best when the client's requirements are unclear and require to be altered.	It works best when the client's needs are perfectly clear.
It is highly probable that quality can be improved at a low cost.	It is not possible to improve quality at a low cost.
The price of the product does not go up in the Prototype model when the quality of the product is improved.	The price of the product may go up as quality improves in the Spiral model.
Maintaining a large-scale project	Maintain small to medium-sized projects.

1.8 Agile Methodology

1.8.1 Introduction to Agile Methodology with the history

Development teams have been concentrating on reducing the time it takes for new features and products to reach the market in order to meet requirements immediately. The majority of the time, development teams have accomplished this goal in really amazing ways. And Agile Methodology is largely to credit for that performance. Development teams used to identify issues and make plans for solutions prior to the advent of the Agile Methodology. Then they would set to work creating that solution and bringing it entirely to the market. In fact, most teams employed the Waterfall approach at that time.

This method may appear to be acceptable, but Waterfall requires that teams adhere to the specifications and work scope that were established at the very beginning of the project and avoid making any changes or additions later. It could be difficult to stick to the predetermined plan because Waterfall prioritized releasing a finished product to market, which meant it might be years before teams completed the task at hand.

All of this began in the spring of 2000. 17 software developers, namely Martin Fowler, Jim Highsmith, Jon Kern, Jeff Sutherland, Ken Schwaber, and Bob Martin, were involved in its creation. They gathered together in Oregon to talk about ways to accelerate software development and get new products to market more quickly. They identified two significant prospects that would be made possible by achieving this goal:

1. Reducing the time it takes to deliver the final product to users in order to fix development issues.
2. Obtaining immediate user input allows developers to assess new software's functionality and make necessary improvements.

Despite the fact that this conference did not lead to the development of the Agile approach as we know it today, it was still a significant turning point in the history of Agile because the Agile methodology's core principles include speed to market, quick feedback, and continuous improvement.

The Waterfall approach of software development was employed by developers prior to the invention of Agile. This methodology can be described as a top-down, sequential procedure for creating systems or software. This model was simple and linear. However, this model had certain issues, such as the following.

- It took too much time to use this model. We cannot move on to the following stages until we have finished a particular stage.
- Only projects with stable requirements could use this paradigm.
- Only after the model's final step is completed is the working software delivered.
- Making modifications that we hadn't considered in the initial phase by going back to earlier phases is really difficult.

The leaders of the various development techniques made the decision to get together in order to find a solution to the challenges caused by the aforementioned shortcomings of the Waterfall model. These leaders were able to come to an arrangement on a lightweight development approach and provide the wording for it in the form of a manifesto. "The Manifesto for Agile Software Development" was its name. 4 Values and 12 Principles make up this manifesto as follows.



Figure 1. 6 The 4 values of the Agile Manifesto

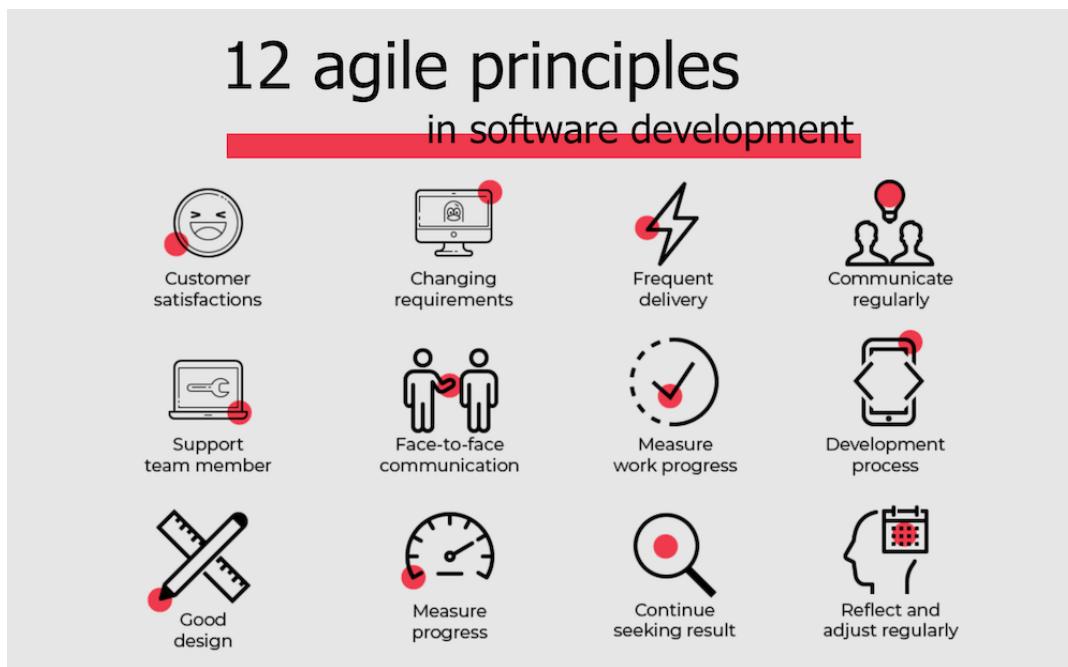


Figure 1. 7 The 12 principles of the Agile Manifesto

Therefore, Agile Software development is a method where requirements and solutions are developed in collaboration by self-organizing, cross-functional team members and their clients. It encourages quick and adaptable responses to change and supports adaptive planning, developmental growth, early delivery, and continuous improvement. Complex projects can be developed by the team more quickly and effectively thanks to agile software development. It is made up of strategies that are incremental and iterative, simple to use and produce excellent outcomes.

There are 2 major models that belong to the Agile Methodology

- 1) Scrum Model
- 2) XP (Extreme Programming) Model

(Lynn, n.d.)

1.8.2 The Agile Development Method's characteristics

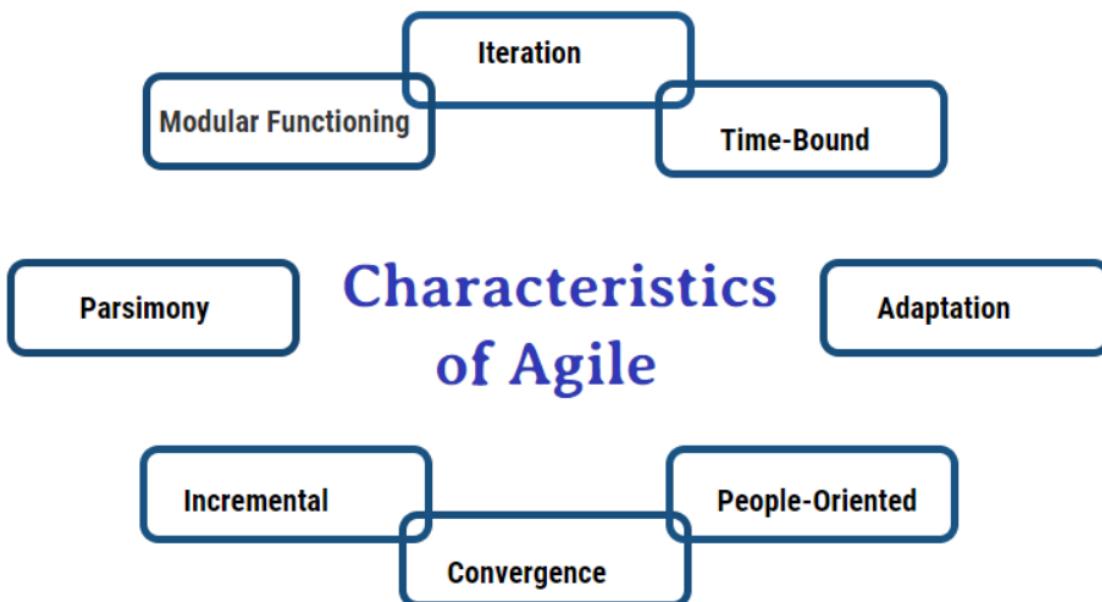


Figure 1. 8 Agile Development Method's characteristics

1. Modular Functioning

One of the most important components of a good process is modularity. Activities are the broken component, and modularity is the element that makes this possible. Simply said, the software development process is a collection of operations designed to frame or manifest the software system's conceptual design. The agile software development method uses quality tools that are handled by skilled developers who are known for using them when they should be.

2. Iteration

The agile software development method accepts that mistakes are made while trying to do something necessary. The agile procedure therefore, emphasizes short cycles. Each cycle has a set of tasks that must be carried out correctly. From the moment the cycle begins to the time it ends, each task has approximately a week or 2 to be accomplished. A 100% accurate element may or may not be obtained during the iteration or single cycle. This is why a single short cycle is repeated numerous times until the desired outcome is obtained.

3. Time-Bound

To keep things on track, software development is subject to time limitations, or the development team must provide a delivery date to the client. The benefits of iterations include keeping each iteration's time limit between one and six weeks so that it can be scheduled appropriately. In order to meet the objectives that were specified at the beginning of the iteration, it is much more possible than not that not all activities will be scheduled in a single iteration. To accomplish the project on schedule and within the allowed time, rescheduling or functionality reductions may be necessary.

4. Parsimony

The agile Method is seen as an improved version of the Traditional Method with the addition of time constraints. Each stage of development is taken into consideration rather than trying to meet impossible schedules for quick delivery, as this could compromise the product's quality. The agile approach concentrates on parsimony, on the other hand, reduces activities to a minimum and only those that are essential to reducing risks and achieving their objective.

5. Adaptation

There is a higher probability that unforeseen risks will emerge during development or iterations. The agile methodology is equipped to handle these unexpected risks. New activities or functionality can be introduced to the functionality if there are variations in the results in order to achieve the goal.

6. Incremental

Not everything in an agile system is created at once. The system is divided into chunks that can be developed simultaneously but at separate times and speeds. Iterations are independently tested. If everything is deemed to be in order, it is all integrated into the final system.

7. Convergence

Convergence indicates that risks are actively pursued because it is important to be aware of them. By doing this, the system gets closer to the outcomes. One of the major techniques that result in a successful iteration is risk-solving at each iteration.

8. People-Oriented

Customers are prioritized over process and technology in the agile approach. The customer is naturally involved in the process. The ability to adapt and modify allows developers to advance their performance and productivity. The alterations in the system at every iteration are something these developers are highly aware of.

9. Collaboration

Direct communication, whether it be with the client or the team members themselves, is a very useful strategy used in the agile process. When it comes to software development projects, effective communication is crucial for their success. When the system is developed in chunks, the likelihood of misunderstanding increases. Every team member is required to comprehend how the various components come together to form the finished result. When each iteration is finished, the process is more focused on integration than communication. Working together with the teams is essential to combine the smaller integrations into the larger parts generated simultaneously so that the final product is fixed appropriately in the system.

1.9 Strengths and Weaknesses of Agile Methodology

1.9.1 Strengths (Advantages) of Agile Methodology

- We can deliver software more quickly, allowing our customers to benefit sooner rather than later.
- Since you always work on current activities, we squander less resources.
- We can respond more quickly and more effectively to changes.
- Faster response time.
- Faster faults and flaws can be found and corrected.
- Busywork takes up less of our time.
- We can share information with a huge network of Agile practitioners.
- It also boosts team morale since we may receive rapid feedback.
- Based on Question & Answer comments, developers can enhance their abilities.
- We don't have to be concerned about early optimization.
- Due to the low costs involved, we can experiment and test ideas.

1.9.2 Weaknesses (Disadvantages) of Agile Methodology

- It can be challenging for new users to catch up with the documentation because it frequently gets sidetracked.
- Progress occurs throughout a number of cycles, making it more challenging to evaluate than in a Waterfall project.
- Agile requires more time and effort from everyone because consumers and developers must communicate regularly.
- When developers are out of work, they are unable to work on another project because they will soon be required again.
- Because there is no obvious outcome, projects may become endless.
- Clients working within a predetermined budget or timeline would be unable to know the final cost of the project.

1.10 Types of Agile Methodologies

1.10.1 Scrum Model

Introduction →

With the help of the Scrum framework, individuals can manage difficult dynamic problems and deliver solutions of the highest quality in a productive and innovative manner. It offers a straightforward foundation for productive teamwork on complicated projects. The most well-known and extensively used Agile methodology is the Scrum Model.

Two major software development pain points are addressed by the Scrum framework. The first is how quickly software is developed, and the second is how frequently the client needs revision. With this method, the software development project is carried out in phases, each of which is referred to as a Sprint. Small teams of 5–6 people are also formed, and they all work together to achieve the intended results.

Every stage of the Scrum model allows for active customer participation, ensuring that any necessary adjustments are addressed right away and implemented. As a result, the project will be delivered on schedule and will successfully satisfy the client's requirements.

Diagram and explanation of Scrum Requirements →

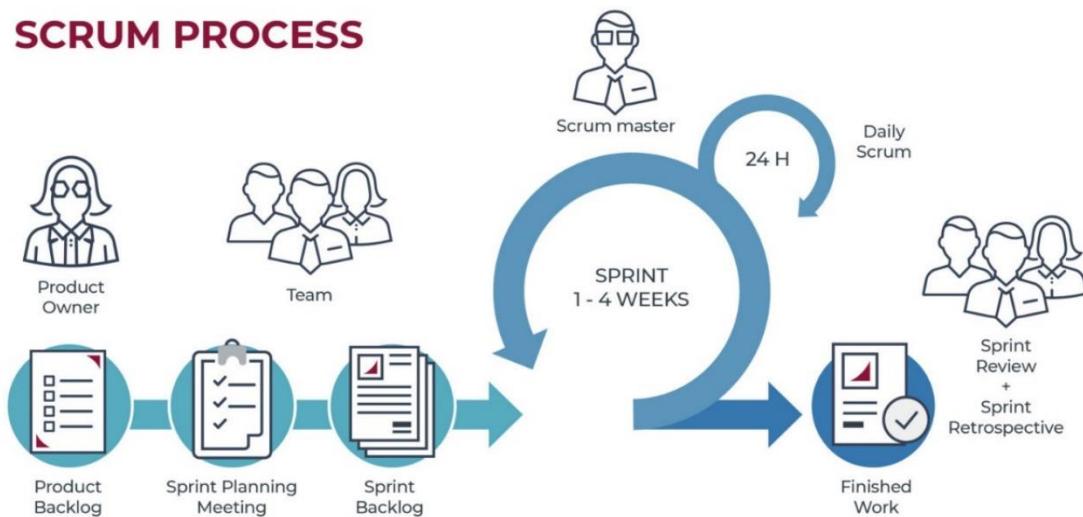


Figure 1. 9 Scrum Process

The following 3 major components set the backbone of Scrum.

- 1) Scrum Roles
 - Product Owner
 - Scrum Master
 - Team
- 2) Scrum Artifacts
 - User Stories
 - Product Backlog
 - Sprint Backlog
 - Burn Down Charts
- 3) Scrum Processes (Ceremonies)
 - Sprint Planning
 - Sprint Review
 - Sprint Retrospective
 - Daily scrum meeting

Scrum Roles →

1) Product Owner

The activity of the development team is guided by the product owner. The person in charge of requirements is the product owner, who is also an expert in the product. In other words, stakeholder management is also their responsibility. Given this information, it also makes sense that the product owner directs the Scrum teams by way of a backlog and release plan on which product features to develop and when. The Product Owner's primary responsibility is to serve as a link between the project's business and technical components, to put it simply. So they serve as the bridge between the client and the development team. The product vision in Scrum is carried by the Product Owner. They are fully a part of the Scrum team and are in charge of creating user stories and updating the Product Backlog.

2) Scrum Master

The Scrum Master is a leader for the team and thus is accountable for maintaining proper applicability of the model, according to the Scrum Master guideline. He is not, however, a project manager. Actually, it is the Scrum Master's job to support and promote Scrum as it is set out in the Scrum Guide. The Scrum Master accomplishes this by assisting everyone in comprehending the principles, procedures, guidelines, and values of Scrum. To put it another way, the Scrum Master's job is to encourage better communication among all team members in order to maximize the value the Scrum team produces.

3) Development Team

The majority of the Scrum team is made up of the development team. Although the size of a scrum team might vary, 5 to 9 persons from the development team are usually suitable. The completion of the job is the development team's main priority. It's crucial to remember that not all members of development teams are programmers or developers.

Scrum Artifacts →

1) User Stories

The Product Owner is in charge of writing User Stories, which are often handled by them. Although the structure is not standardized, it often includes a name, some informative text, references to other papers (like screenshots), and details on how the application will be evaluated. Of course, this is done to direct the creation of test cases for the story. The Team will require this data to figure out how much effort is necessary to accomplish the story.

2) Product Backlog

The Product Backlog is an ongoing, prioritized list of what is required to improve the product, as stated in the Scrum Guide. It is the Scrum Team's sole source of work. Items from the Product Backlog that can be completed by the Scrum Team in one Sprint are considered ready for selection in a Sprint Planning event. Usually, refinement actions provide them with this level of transparency. Refining the product backlog involves dividing and further defining the things on it into smaller, more exact ones. Adding information, such as a description, order, and size is a continuous process. Depending on the area of work, attributes frequently change.

3) Burndown Chart

A project management chart called a burndown chart demonstrates how rapidly a team is moving through a client's user stories. With the help of this tool, we can record a feature's description from the viewpoint of the end user. This displays the overall effort in relation to the amount of work in each agile sprint or iteration. Agile project management teams can keep track of what has been completed, what still has to be done, and how much time is left in the project by using a burndown chart. Although a burndown chart is typically a visual aid, it can also serve as a list of the work that needs to be done and the proportion of it that has already been done.

Scrum Processes (Ceremonies) →

1) Sprint Planning

One of the most significant activities in a Scrum development project is the Sprint Planning meeting. The development team chooses the Product Backlog items that are of highest priority that they believe they can complete during the sprint during this meeting. A sprint planning is produced as a result of the entire Scrum Team working together. For each story in the Sprint Backlog, the Testers estimate the testing effort.

2) Sprint Review/ Retrospective

Undoubtedly, the Sprint Retrospective is a crucial component of continuous improvement. The entire development team comes together for this reunion, which happens at the end of a sprint. They examine graphs like the Velocity graph and the Burnup and Burndown charts. The team wants improvement and considers how to enhance interpersonal relationships even more in order to increase motivation and well-being, but they also openly discuss and step back from the most recent sprint. The goal of this is to increase.

3) Daily Scrum Meeting

The Daily Scrum's goal, as stated in the Scrum Guide, is to evaluate the Sprint Goal's progress and modify the Sprint Backlog as necessary to change the next planned work. The Scrum Team's developers meet for 15 minutes each day for the Daily Scrum. It takes place every working day of the Sprint at the same time and location to reduce complexity. The Product Owner and Scrum Master engage as Developers if they are actively working on items in the Sprint Backlog. As long as their Daily Scrum concentrates on progress toward the Sprint Goal and generates an actionable plan for the following day of work, the Developers are free to choose whatever structure and methodologies they wish. Focus and self-management are enhanced as a result.

Scrum Methodology Process →

The foundation of the Scrum approach is a set of extremely specific responsibilities and procedures that must be followed throughout the software development process. Sprints are the transient, recurring periods of time used in Scrum. The timeline for feedback and reflection is typically 2 to 4 weeks. Each Sprint stands alone as a distinct entity. Upon request, it must be able to be supplied to the client with the least amount of work necessary. It provides a comprehensive outcome.

A set of requirements that makes up the project plan serves as the process's starting point. It is up to the project's client to prioritize these requirements while keeping a balance of value and cost in mind. Consequently, that is how the iterations and next delivery are decided. On the one hand, the market demands high-quality products delivered quickly and affordably. To meet these demands, a company must be very agile and flexible in the product development process. This allows for quick development cycles that can satisfy customer demand without compromising the quality of the final product. Because of the quick results it produces, it is a very popular methodology to use.

Below is a brief list of the steps that make up the Scrum framework.

- A product backlog, or list of all the activities necessary to complete the project, is made by the product owner.
- All significant tasks are broken down into smaller, more manageable ones by the Scrum team after evaluating the product backlog.
- After that, the team chooses how it will be implemented and builds a sprint backlog.
- Each sprint is given a specific period, which ranges between two and four weeks on average.
- Throughout the course of a sprint, the team typically meets daily, with each member providing an update and the project manager reviewing the project's development in regard to the most recent information.
- The product owner and the stakeholders of the organization evaluate the outcomes after each sprint.

Advantages →

- Team members may finish project deliverables more quickly and effectively with Scrum.
- Effective time and money management is ensured by scrum.
- Scrums are used to break large projects into easily controlled phases.
- During the sprint review, development is coded and tested.
- Effective for quickly moving development projects.
- With the help of scrum meetings, the team has a better overview.
- Scrum incorporates feedback from consumers and stakeholders since it is agile.
- Changes made in response to feedback are easier to implement during short sprints.
- During daily scrum meetings, each team member's individual effort is visible.

Disadvantages →

- The lack of a set completion date in Scrum frequently causes scope creep.
- If staff are not extremely dedicated or cooperative, there is a great likelihood that the project will fail.
- It can be difficult to implement the Scrum framework in larger teams.
- Only an experienced team can implement the framework successfully.
- Team members can become frustrated by daily meetings.
- A project may suffer significantly if a team member leaves in the middle of it.
- Before the team goes through a thorough testing process, quality is challenging to implement.

When to use Scrum →

Scrum can be used to organize any project. It extends beyond a team of developers using the Agile methodology. The Scrum project management approach is incredibly customizable and may be used for a wide range of projects. Teams with a variety of skill sets can easily operate well together since it promotes a great deal of collaboration amongst team members. Scrum may be the project management method to use if we're seeking a structure that keeps things organized but also flexible. The following are some great examples of projects that can be managed using Scrum.

- Redesigning a website is being worked on by a group of different web experts. Each team member could collaborate using Scrum to launch the new website in 2-4 weeks.
- A team of journalists collaborating on a blog post or article series. Utilizing Scrum, each journalist can work toward the objective of releasing a fresh article each week so that their audience has wonderful, new stuff to read.
- Working on UI/UX projects as a group of designers. To produce a polished and useful end result, each person's experience might be utilized.

Scrum Method is applicable in the following circumstances.

- If a continuous modification will be most advantageous to the project's requirements.
- If each team member is entirely cooperative and driven by their own interests.
- If the project encourages inventiveness and creative design.
- If the project would benefit the most from some level of process control.

1.10.2 XP (eXtreme Programming) Model

Introduction →

One of the most significant Agile methodologies for software development is extreme programming (XP). It is applied to enhance the quality of software and responsiveness to consumer needs. The extreme programming approach suggests taking best practices that have previously performed successfully in program development projects and pushing them up to extreme levels. Extreme programming requires the application of good practices.

The XP methodology is built on the frequent iterations that developers use to put User Stories into practice. User stories are straightforward, casual statements made by the client regarding the functionalities required. A User Story is a typical description of a feature of the necessary system provided by the user. Finer details, such as the various possible scenarios, are not mentioned. Metaphors are suggested by the project team based on user stories. Metaphors are a popular way to picture how the system would operate.

In order to implement specific functionality, the development team could choose to create a Spike. A Spike is a relatively straightforward software designed to investigate the feasibility of a proposed solution. It may be compared to a prototype. Coding, Testing, Listening, Designing, Feedback, and Simplicity are some of the fundamental processes used in software development utilizing the XP methodology.

This method is known as Extreme because it takes effective principles and practices to extreme levels.

- Since the code is always being reviewed, code reviews are effective.
- Since testing and regression are ongoing processes, testing is effective.
- Everyone must perform daily refactoring, so design is effective.
- As we integrate and test multiple times each day, integration testing is crucial.

Diagram and explanation of XP (eXtreme Programming) Process →

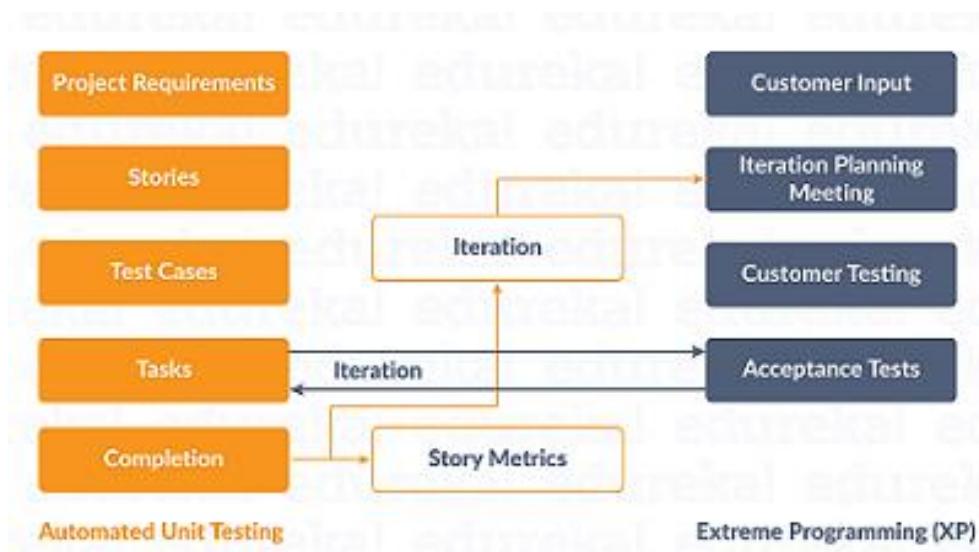


Figure 1. 10 XP (eXtreme Programming) Process

Because of its emphasis on customer satisfaction, XP is the most successful method of building agile software. The development of XP demands as much client involvement as possible. The complete software development life cycle is split up into a large number of short development cycles. At each point in the development life cycle, it accepts and implements customer changes or requirements. The first step in extreme programming is gathering user requirements. The entire development process is separated into multiple smaller numbers of cycles based on these requirements. The next step is cycle planning, which involves selecting the number of cycles, prioritizing the requirements, and estimating the work needed to complete each cycle.

Each iteration is now developed via pair programming. New user requirements could emerge during the development phase, and the iteration plan should be modified as a result. The latest version must then be tested for bugs if any are found. In the following iteration, the bugs will be fixed. There should be project tracing performed following each acceptance test, in which input based on the number of completed tasks is obtained from the project. Developers now have access to numerous additional features owing to XP, including pair programming, thorough code reviews, code refactoring, and open workspaces.

Advantages →

- Extreme Programming's key benefit is that it enables software development organizations to reduce project execution costs and timelines. Owing to XP's emphasis on the rapid delivery of final products, time savings are possible. Teams that employ extreme programming make significant financial savings by not creating a lot of documentation. Problems are typically resolved through team conversations.
- Another benefit of projects using extreme programming is simplicity. This approach produces incredibly straightforward code that can always be enhanced by the developers who use it.
- In XP, the entire process is transparent and accountable. As they work toward their goals, developers exhibit their progress.
- Another plus is the constant feedback. It's important to listen and adapt as necessary.
- Thanks to frequent testing during the development phase, XP helps to produce software more quickly.
- Increased employee retention and satisfaction are both benefits of extreme programming.

Disadvantages →

- According to some experts, Extreme Programming puts more emphasis on the code than the design. Since software applications depend so greatly on good design, this could be an issue. It promotes their sales in the software industry. Furthermore, defect documentation in XP projects is not always adequate. The absence of defect documentation may cause similar defects to appear in the future.
- Another drawback of XP is that it doesn't track code quality assurance. It might result in bugs in the original code.
- If programmers are geographically dispersed, XP is not the best choice.

When to use XP (eXtreme Programming) Model →

It's necessary to ensure that a company's size, structure, and knowledge, as well as the experience of its people, allow for the use of XP methods. The following are some circumstances when projects can be managed using XP.

- Many systems require frequent changes as some do not have continual functionality features. XP was created to make it easier for development teams to adjust to rapidly changing requirements.
- XP can be used in risky projects. When a client sets severe timelines for a project, teams who use XP methods are more likely to avoid issues related to working on a new system. A high level of consumer participation also lowers the possibility that they won't like the final product.
- For teams with no more than 12 members, XP practices are effective. In such groups, management is frequently simpler, communication is more effective, and meetings and brainstorming sessions are usually quicker.
- If the company or team is prepared to adopt new cultures and knowledge, XP can be applied. Because XP differs from traditional methods for developing software, it may be difficult to know how to put some of its practices into effect. Therefore, it's crucial that the team and company are prepared to accept change. If the company hasn't worked with XP before, it's also worthwhile to invite an experienced coach.

1.11 Comparison between Traditional Methodology and Agile Methodology

The order of project phases, which include requirements gathering, planning, design, implementation, testing, and maintenance, is the primary distinction between traditional and agile methodologies. Traditional development methodologies follow a linear phase-by-phase process, whereas Agile uses an iterative approach.

Table 1. 4 Comparison between Traditional Methodology and Agile Methodology

Traditional Methodology	Agile Methodology
Simple software development is done with it.	Complex software development uses it.
After the development phase is complete, testing is carried out according to this process.	The testing and development stages are carried out simultaneously in this methodology.
It is organized in a linear manner.	It is organized in an iterative manner.
Less security is offered.	It offers a great level of security.
In comparison to Agile development, there is less client participation.	In contrast to traditional software development, there is a lot of client interaction.
The development model is fixed.	It encourages a flexible model of development.
The software has less features available.	All the features that users require are offered by it.
Using this process leads to lower development costs.	The expense of employing this process for development is significant.

There are 5 main phases to it.	There are mainly 3 stages to it.
When it comes to customer communication, traditional software development methodologies are formal.	The approaches used in agile software development are informal. This means, Customers are more willing to interact with companies that use agile software development methodologies.
To begin with, traditional methods for software development use a predictive strategy. Because the product is created through rigorous and precise planning, the software development processes are fully specified and predicted. This method for project development has defined times and costs, thus changes are not allowed.	A flexible approach is adopted because software development methodologies are based on the concept of continuous design refinement and testing is based on team and customer response.
Software development companies utilize it less frequently.	It is commonly used by software development companies.
Most of the time beginners utilize it.	Most of the time experts use it.

1.12 Best Software Development Methodology for a Project as my perspective

Given the project's nature and organizational priority, selecting a methodology can be difficult. The following are some important criteria to take into account before selecting one of the major software development methodologies.

Agile methodology is far more adaptable than waterfall methodology when it comes to making modifications to a product or a process. The agile methodology makes it simple for team members to experiment with and try something that isn't planned while they're working. The best feature about this methodology is that it prioritizes the product over adhering to a predetermined structure. Agile methodology isn't top-down or linear, in contrast to the conventional way. In this manner, all alterations made at the last minute can be taken into account without compromising the end result or delaying the project timeline.

With the agile methodology, everything is transparent and accessible. The launch, design, review, and testing include active participation from the clients and decision-makers. In contrast, the project manager controls the traditional approach, preventing others from participating in the decision-making process. Team members may observe progress from beginning to end owing to the agile methodology. A productive workplace is significantly influenced by this degree of transparency.

The traditional method involves outlining and planning each and every step of the procedure from the very beginning of the project. Both the expected time and budget for the project must be met. In order to avoid pushing the deadline, any significant modification or feedback is forfeit. Agile method, on the other hand, enables continuous feedback that enhances production. Many project managers and software developers now prefer the agile methodology because of how well-liked feedback is in it. As customers validate each iteration, they may respond to requests from customers and offer a high-quality good or service within the delivery window.

1.13 How to choose best Software Development Methodology for a Project

The truth is that there isn't a single methodology that works for all projects and organizations. The project's nature, scale, and resource requirements are the main determinants of which method will be used. Smart project managers typically choose the methodology to use at the start or initiation of the project. In accordance with other project sponsors and those participating in the project planning process, he makes the final decision. Some things to think about when selecting an appropriate methodology for our project are listed below.

- Look at the project specifications. The requirements must be clear. Select the agile methodology if the project requirements are unclear or subject to change. And when the conditions are firmly established and known from the start, the Traditional approach works well.
- Think about the project's technological requirements. If no new tools or technology are used, the traditional project management methodology is more suitable. Agile methodologies are more flexible than the Traditional approach, giving more room for experimenting with cutting-edge technologies.
- Are there any unwelcome hazards or threats to the project? It's not a good idea to use the traditional methodology because of how rigid it is. Though the agile method appears to be a better alternative in terms of risk management, risks can be addressed more quickly in this manner.
- The availability of resources is another important element. With large and complicated teams and projects, the Traditional method performs well. An agile team, on the other hand, often consists of a small number of professional team members.
- The nature of the project management methodology adopted has a significant impact on how important the final product is. Compared to the agile project management methodology, the traditional method is significantly more appropriate for crucial projects because it incorporates documentation.

1.14 Conclusion to Activity 1

It is important to understand the background of the topic before answering the activity question, which is why this activity began with the introduction of System Analysis Design. The task then goes on to explain the various software development approaches used throughout the life cycle of a system. There are 2 main methodologies for developing software. Traditional and Agile are those. This activity provided a thorough explanation of the two techniques along with their various methods and approaches. Then, the contrasts between each of them as well as their strengths and weaknesses were critically evaluated. I shared my all-time favorite software methodology towards the conclusion. But I made it very obvious in the conclusion that there isn't always one optimal methodology to use; it depends on the circumstances.

Activity 2

2.1 Business Case

2.1.1 Introduction to Business Case

The why, what, how, and who are outlined in a business case, which is created in the early phases of a project and helps determine whether it is worthwhile to continue. In order to inform and persuade a decision maker to take further action, a business case is a value proposition that is presented verbally or in writing. The term "business case" refers to the written

A business case serves just as a tool for communication. It is written in an understandable manner and contains enough information to let the intended audience make decisions. When it comes to the size of a business case, there is no secret formula. Size doesn't matter. What matters is that the business case offers the decision maker all the data they require to accomplish their goal. In reality, a business case doesn't even need to be a written document. Although it might be delivered verbally, the structure and content are the same as if it had been written down.

The main purpose of a business case is to justify an investment. At the project level, it helps with investment decisions. An organization can use this document to decide whether to support or abandon a project by summarizing the risks, advantages, and costs. Based on actual data, the company will be able to make a wise decision.



Figure 2. 1 Business Case

2.1.2 Why use Business Case

An informed decision can be made by seeing at officially detailing the project's justification and then comparing the scope to the strategy. Companies want their project managers, as well as their team members, to have a better understanding of business. As a result, this business case justifies the suggested business change or plan. It details the possible capital allocation you're looking for as well as the resources needed to put it into action. A plan of action is then possible. Additionally, it gives the decision-makers a wide range of choices.

Both money and employees are already in limited supply in the majority of organizations. In order to ensure that projects are successful, it is important that they are in line with the organization's strategy or goals and that there is a strong business case for investing resources in them. Members of senior management commonly have personal or professional projects that are very important to them. Unfortunately, this might not always be in accordance with the organization's goals and objectives. An informed decision can be made by seeing at officially detailing the project's justification and then comparing the scope to the strategy.

The business case's main objective is to persuade decision-makers to approve the project concept, but it also supports the organization in prioritizing projects that are consistent with the company's overall strategy. Using business cases, it is possible to compare ideas that are extremely different from one another using the same standards, resulting in an open process and a fair playing field for all potential projects.

The business case can assist in managing the project's scope at the initial planning phase if the project is chosen. The business case is then used to measure how well the company did with its planning and implementation when the project is finished.

2.1.3 Steps required to make a Business Case

When there is no strong business case to support a project, it fails. The project charter and project plan are based on this document, which is required to launch the project. However, if a project's business case is not based in reality and fails to fulfill a need that is relevant with the organization's overall business objectives, it is useless.

The why, what, how, and who of our project are the subjects of the investigation we need to produce a solid business case. This needs to be explained in detail. The components of our business case will go into deeper detail about the why. During the project initiation stage, the business case as a document is developed. But will serve as a guide throughout the course of the project. How to write a business case is demonstrated in the steps below.

- 1) Finding the business problem is the first step.

Every project has a purpose. They are typically started in order to address a particular business issue or develop a business opportunity. Identifying the issue or opportunity, describing it, and determining its origin is the first order of business. Next, we must talk about how long it will take to resolve. This can be a straightforward statement, but it is better expressed with research on the market and the status of the competition to support the project's timing.

- 2) The second step is to determine the alternatives.

We are unable to determine whether the project being performed is the best option for solving the previously mentioned issue. It's challenging to choose the best course of action, and the road to success is not built with proven presumptions. After conducting the necessary research, the procedures listed below can be used to eliminate some alternatives and make the best choice apparent.

- Take note of the alternative options.
- The advantages of each solution should be measured.
- Forecast the costs associated with each solution as well.
- Then determine whether it is feasible.

- Recognize the risks and problems related to each option.
- Lastly, include a business case with all of this information.

3) Recommending a Preferred Solution is step three.

The solutions need to be ranked next. However, it's best to establish criteria first. Have a rating system, if possible, to assist us in ranking the options so that we can select the best one.

4) The fourth step is to describe the implementation strategy.

The fourth step is to describe the implementation strategy. We now know what the business opportunity or problem is and how to solve it. In order to accomplish our aims, we must now persuade our stakeholders that we are right and have the best approach in place. Documentation is essential because of this. The fundamental issue we noticed can be resolved practically due to documentation. We can convince our sponsors or the organizational leadership to support us when we work hard to build a solid business case. Additionally, we can see clearly how to guarantee that the expected commercial benefits are delivered.

2.1.4 Elements of Business Case

Creating a business case checklist is one of the essential first tasks. A thorough plan to follow when creating our business case is provided below. To add to our business case template, we could select the components that our project stakeholders will find most useful.

1) Executive Summary

A condensed version of each section of our business case can be found in the executive summary. It serves as a brief summary of our project for stakeholders.

2) Project Definition

This section is intended to provide basic details about our projects. Such as the project plan's framework and the expected business objectives.

3) Vision, Goals and Objectives

Identifying our goals and the issue we hope to resolve is the first step. Our project's vision, objectives, and goals must be defined. We can use this to define the project's scope and determine its deliverables.

4) Project Scope

All of the tasks and deliverables that will be carried out as part of our project to accomplish our business objectives are determined by the project scope. The project's limits as well as all associated tasks, resources, deadlines, and deliverables might be included in this.

5) Background Information

The context for our project can be provided here. This can clarify the issue it's aimed at addressing and show how it fits with the mission and strategic goal of our company.

6) Stakeholder requirements and success criteria

The quality requirements will vary depending on the type of project we are working on. They are, nonetheless, essential to the project's success. Collect all of the quality requirements, identify what decides whether we have successfully met them, and report on the outcomes.

7) Project Plan

Determine the actions we must do to complete the project. For the purpose of ensuring completion, we can utilize a template for a work breakdown structure. Estimate the time it will take to finish each task once we have gathered all of the activities. To build a project timeline, we can use a Gantt chart view or a Kanban board view.

8) Project Budget

All of the costs in our project plan are estimated in our budget. Additionally, it illustrates the cost of extending the project's completion date.

9) Project Schedule

Estimate how long it will take for each activity to be finished and use that information to create a timeline for the project. Utilize a tool to create a Gantt chart, then print it out to use as part of a more convincing project schedule. This would provide excellent data visualization and skill, which Excel sheets cannot supply.

10) Project Governance

All the project management policies and practices that are relevant to the project are referred to as project governance. It specifies things like the structure for decision-making and the roles and responsibilities of the project team members.

11) Communication Plan

The communication plan should include check-in and status update milestones and specify how stakeholders will be kept up to date on the project's progress.

12) Progress Reports

In order to compare planned and actual progress, we can set up a plan to track and monitor our progress throughout the project. Tools for task monitoring are available to assist us keep track of development and output.

13) Financial Appraisal

This part of our business case is important because it's where we discuss how the financial advantages exceed the disadvantages. Here, we evaluate our project's financial costs and benefits. A sensitivity analysis and a cost-benefit analysis can be used to accomplish this.

14) Market Assessment

To identify opportunities and risks, research the market, the competition, and the industry.

15) Competitor Analysis

Find business competitors, both direct and indirect, and evaluate their strategies, products, and competitive advantages.

16) SWOT Analysis

Our organization's strengths, weaknesses, opportunities, and threats can be determined via a SWOT analysis. The opportunities and threats are external, whereas the strengths and weaknesses are internal.

17) Marketing Strategy

At here describe the target market, pricing, distribution routes, and other elements of our marketing approach.

18) Risk Assessment

Our project could be impacted by a wide range of risk categories. Finding and evaluating the risks related to our project's operations is the first step in minimizing them.

1.1.5 Advantages of using Business Case

- We can forecast an idea's potential success using a business case to see whether it is worth considering. If a project is only going to cost us money, there is no reason to move through with its implementation, but if we dive in blindly without considering the consequences, that is exactly what happens. We will still get a picture of where our business should be thanks to the business case, even though the future may appear uncertain.
- With a business case, we can gain insight into the inventory and budget we now hold. There will be limited supplies of some materials that our company needs. We can modify our journey once we have a clear understanding of what our potential financial future will be. By doing so, we can avoid the roadblocks that stand in our way of achieving success.
- We'll have a variety of perspectives on what will lead to the most success when we're working with different individuals. People with different opinions often go rogue and just do their own thing when there isn't any structure associated with a business. We can direct those creative energies into ideas that provide our business with a better chance of success by making sure that everyone is on the same page with a business case.
- Making a business case for an idea indicates that we are considering the business case seriously. Others can see that we are confident in the business case's value by looking at it, and that we're prepared to support it as well. We may express our ideas more clearly, defend the value of our proposal, and explain how its growth will benefit others by using a business case.
- The business case enables us to determine how we can use our current goods or services to expand into new markets. Additionally, we will be able to improve our value proposition, providing our brand with a stronger presence across all demographics.

1.1.6 Disadvantages of using Business Case

- Involving the appropriate individuals in the business case process is crucial. These are the individuals who will have an impact on our company's long-term goals. Sometimes small business owners believe that by developing the business case alone, they may avoid this drawback. However, for that to succeed, we need to be expert in several other areas. The ideal business case typically requires a wide range of perspectives and participation because otherwise, inaccuracies might have many unforeseen effects.
- It is challenging to hold one person accountable for the process because a business case is often created by one person. Their perspective of the company and desired level of success is influenced by the business case. Additionally, it means that instead of what is best for the company, the business case is developed on their schedule, which makes it challenging to hold them accountable for completing the task.
- Business cases determine what we should do and how to do it. Sometimes a strong company needs to give its most creative employees the flexibility to develop original new concepts. However, Business Case frequently creates an atmosphere in which the company's executives set everyone else's objectives and missions. In the long run, this disadvantages a business.
- Business cases are simply forecasts based on ideas and information that are now available. Everything in our ever-changing world is uncertain. A business may not be able to adjust to the changes that the outside world is forcing upon it if the business case that has been established contains too much confidence.
- Failure is more likely to occur than success, despite the highest-quality research, the best employees, and a thorough business plan all working in our direction. 95% new businesses that are founded today would fail within the next five years, yet many of these businesses had sophisticated business cases and strategies.

1.1.7 Business Case example



www.knowledgetrain.co.uk

Handheld device project – business case

Executive summary

Three options were considered to address the current working problems. The best option is to develop a software and hardware solution for engineers - achieving benefits of £280k in cost savings per year (p/a).

Reasons

The company faces several problems which incur overheads of £280,000 p/a. These are:

- a) A high volume of paperwork involved when ordering spare parts
- b) A high number of costly errors made when ordering spare parts by mistake
- c) Engineers often arrive at customer's premises with the wrong parts.

Business Options

1. Do nothing. This option has overheads of £280,000 p/a. due to the current problems.
2. Train backroom staff to more reliably order spare parts. This option would reduce the number of wrongly ordered parts and achieve cost savings of £10,000 p/a. It would reduce the number of rescheduled appointments saving £60,000 p/a. Initial training costs would be £20,000 plus a further £5,000 p/a. This option will deliver fewer benefits than option 3 and the company will still incur significant overheads each year.
3. Let engineers use handheld devices to order parts. This option would reduce paperwork, reduce errors when ordering, reduce rescheduled appointments and reduce customer contract cancellations. It would also reduce overheads by £280,000 p/a. The costs of developing the software and purchasing hardware would be £190,000. Introducing new working practices would cost £30,000. Hardware and software support costs would be £20,000 p/a. This option gives the best return on investment.

Expected benefits

- Reduced errors when ordering, reduced rescheduled appointments and reduced customer contract cancellations would lead to reduced overheads of £280k per year.

Expected dis-benefits

- Lower morale amongst the back-office staff because overtime payments will be reduced.

Timescale

- Project time: 12 months
- The benefits (reduced overheads) are to be measured annually (for two years), starting one year after the solution is delivered.

Costs

- Project costs: Hardware £40k; Software £150k; New working practices £30k
- Operational costs (per year): Support £20k

Major risks

Staff lack experience in specifying requirements on a software project. This could lead to the wrong software solution being delivered. This is likely to reduce the benefits realized from the project.

Investment appraisal

	Year 1	Year 2	Year 3
Project costs	-£220k	£0k	£0k
Operational costs	£0k	-£20k	-£20k
Benefits	£0k	£280k	£280k
Net benefits	-£220k	+£260k	£260k

Figure 2. 2 Example Business Case

1.1.7 Business Case templates

Project Name		Project Manager
Client		Duration
Executive Summary	<ul style="list-style-type: none"> Write a short version of each of the following sections in your business case. 	
Mission Statement	<ul style="list-style-type: none"> Define the vision, goals and objectives of the project. 	
Product/Service	<ul style="list-style-type: none"> Explain what the product or service and how it fits a niche or serves a need. 	
Project Definition	<ul style="list-style-type: none"> Provide general information about the project, such as a project plan outline. 	
Project Organization	<ul style="list-style-type: none"> What is the structure of the project, such as functional, matrix, projectized or composite 	
Financial Appraisal	<ul style="list-style-type: none"> Estimate the cost of executing the project plan over the schedule of the project. 	
Market Assessment	<ul style="list-style-type: none"> Research the market opportunities and threats, including competitors. 	
Marketing Strategy	<ul style="list-style-type: none"> Show how your product or service will be distributed, what its pricing will be, the target audience, etc. 	
Risk Assessment	<ul style="list-style-type: none"> Figure out risks to your project and work on how to identify and mitigate them. 	

Figure 2. 3 Business Case template 1

Knowledge **TRAIN®** ●●●

www.knowledgetrain.co.uk

Business case template

Executive summary

Describe the key points from the business case, including the most important benefits and the return on investment.

Reasons

Describe the reasons for doing the project e.g. the business problems which you are trying to solve. It might also be useful to describe how the project helps to achieve corporate objectives or strategies.

Business options

Describe the different options which have been considered and recommend an option. One option must always be the 'Do nothing' option. This option does not result in a project.

Other options might be 'Do the minimum we can', 'Do something else' etc. The difference between these options are the benefits that the investment will buy.

You should try to provide enough information that the approvers of the business case will be able to judge which option gives the best value for money.

Expected benefits

Describe the benefits of the recommended option in measurable terms. They can be both qualitative and quantitative.

Expected dis-benefits

Describe any dis-benefits. These are actual consequences of the project which are negative. If possible, quantify the dis-benefits and include them in the investment appraisal.

Timescale

Describe 2 timescales: the project timescale and the timescale over which the benefits will be realised.

Costs

Describe 2 costs: the project costs, and the ongoing operations and maintenance costs which will be incurred over the benefits realisation timescale. State which corporate budget(s) will fund both these costs.

Investment appraisal

Weigh up on the one hand the costs (both project costs and ongoing operational and maintenance costs) and the dis-benefits versus the benefits. There are lots of techniques which can be used e.g. return on investment, net present value, cash-flow statement, internal rate of return and payback period.

You should be trying to quantify the value of a project as an investment.

Major risks

Describe the main project risks, their likely impact should they occur, and any proposed responses.

Knowledge Train® is a registered trademark of Knowledge Train Limited | Copyright © 2018 Knowledge Train®

Figure 2. 4 Business Case template 2

2.2 Desirability, Viability and Feasibility of systems

2.2.1 Introduction to the topic

Every firm aims to provide new products that delight their current clients while attracting new ones. A positive user experience does not always lead to innovation. What is desirable from a human perspective is combined with what is technically possible and commercially viable using design thinking. Following these three criteria are centered around successful products.

1. Desirability : a product that consumers require or demand.
2. Viability : a revenue-generating product
3. Feasibility : a product that can be manufactured using either new or current technologies

These three criteria work best when combined to encourage innovation. But if a product doesn't tick one or more of these boxes, creating, producing, and marketing it becomes riskier. We must conduct research and evaluate our concepts early in the design process to see whether our product idea satisfies each of the three criteria for being desirable, viable, and feasible.

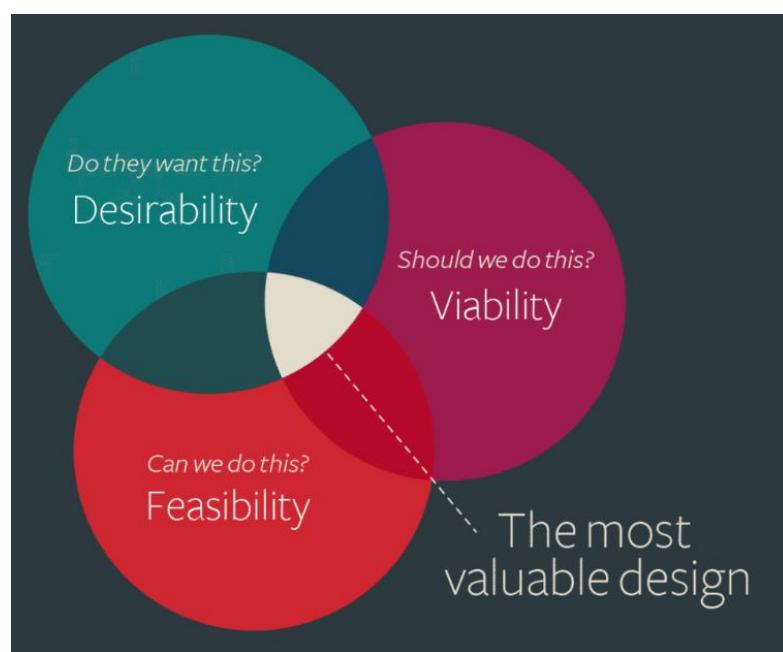


Figure 2. 5 Desirability, viability and feasibility diagram

2.2.2 Introduction to Desirability

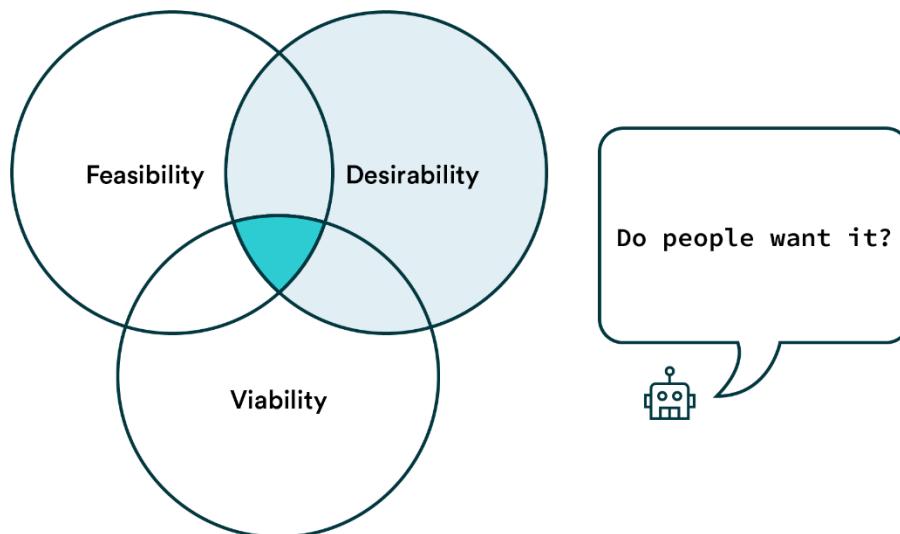


Figure 2. 6 Desirability

Customers or users desire or require products that match the criteria for attractiveness. We won't earn any money off of it if we haven't created a product that people want to buy, after all. Our product must provide a user-friendly, attractive solution to a problem in order for it to be desirable. People that think like designers are rare in many workplaces. So they frequently ignore a product's popularity in favor of concentrating on a product's viability and feasibility. This is a serious mistake because it can discourage customers from using the company's newest product.

Instagram, for example, has got a reputation as a good site to post still photographs. However, Instagram recently added Reels, a feature that prioritizes video over still photographs. To compete with the TikTok-like video app, the business added these functionalities. Because there was a strong business case for the features, they were introduced and were therefore viable. They were designed to lure consumers away from TikTok and grow Instagram's number of users, which would be profitable for the business. Due to Instagram's ability to add features fast and without incurring significant costs, the video features became practical.

2.2.3 Introduction to Viability

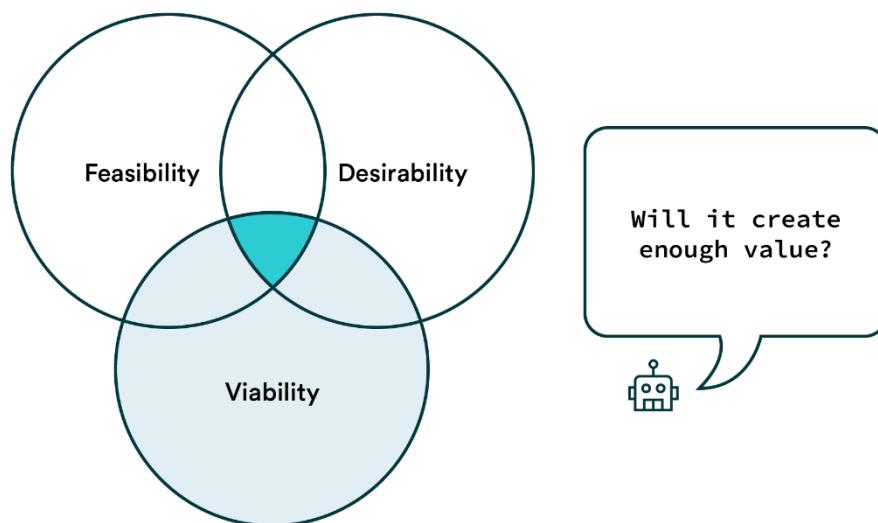


Figure 2. 7 Viability

Products that satisfy viability requirements present a strong business case. Immediately after their release and throughout time, these products will bring in money or save money for a company. Companies must look at who will be willing to buy the product, how they could pay for it, and whether all of it will lead to profitability to determine whether it is viable. A product is not viable if consumers want it, but it is too expensive.

As an instance, there could be a commercial case for starting a new streaming platform that solely offers holiday movies. In December, people enjoy watching Christmas movies, and the streaming service claims to have them all in one spot. In the weeks following its launch, the streamer seemed to be on the correct path to turn a profit. But the business quickly realizes it has two issues: The first issue is that some of the production corporations that own the films charge outrageous prices. Second, many of the service's current users terminate their subscriptions when the holidays are over.

Market research should be done by companies early in the design process, and all elements of their business model should be examined, in order to avoid this consequence. There is a chance that they won't find the customers they anticipate buying their new product if they neglect to consider whether or not it can make money.

2.2.4 Introduction to Feasibility

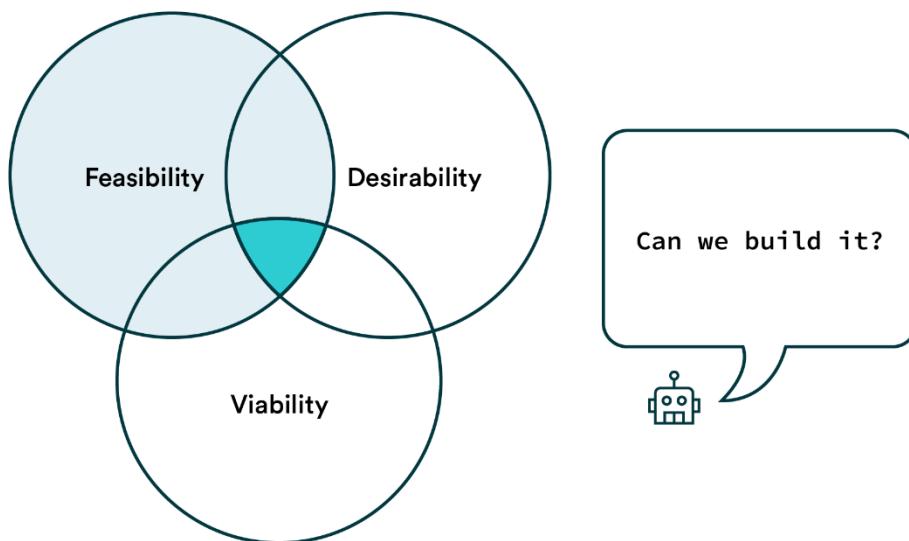


Figure 2. 8 Feasibility

Feasibility refers to a product's manufacturability. This could be either by the business that developed the idea or by a third party with the necessary technological capabilities. To be a feasible product, a corporation must be certain that the technologies required to produce the product can be achieved. Additionally, they need to have confidence that it can be accomplished in a manageable amount of time and at a cost that will keep the product viable.

For example, an auto manufacturer hopes to develop a charger that can charge electric vehicles in the same amount of time as gas-powered vehicles do now. This device would undoubtedly be desired by those who own electric cars because it takes several hours to charge an electric car. It would also be financially feasible if the price was fair. The product may not be feasible since the technology for it does not yet exist.

The business must evaluate the technology required for the product, the time required to produce it, its cost, and distribution options before deciding whether it makes sense to design it. The product will not be feasible if the answers to any of those questions don't benefit the business and the long-term profitability of the product.

2.3 Feasibility Study

2.3.1 Introduction to Feasibility Study

Simply put, a feasibility study evaluates how possible the project idea or approach is. By examining technical, economic, legal, operational, and time feasibility factors, this is performed. For instance, we need to think about whether the technology we suggest using is already available or if it can be developed. Or, we need to consider if we have the required personnel, equipment, and resources. After the business case is finished, project managers conduct a feasibility study at that stage of the project management life cycle. Therefore, a feasibility study is quite crucial because it identifies the variables that affect project viability.

2.3.2 Why use Feasibility Study

Based on the organization's goal to get things correctly before allocating business resources, time, and money, a feasibility study is important. An analysis of the project's feasibility may produce fresh insights that entirely change the project's goals. It's important to think carefully before deciding to do a feasibility study. This process is time-consuming and money-consuming. But skipping the feasibility studies can cost business even more in terms of the bad decisions that can end up making as a consequence.



Figure 1. 11 Feasibility Study

2.3.3 Steps to conduct feasibility study

1) Make a preliminary analysis

Our project plan needs to be outlined first. In a market where there is a gap between supply and demand, we should concentrate on an unmet need. Thus, our product would have a clear benefit. Next, we must assess if the feasibility requirements are too high to meet.

2) Make a projected income statement

We have to go backwards for this phase. Starting with the projected income, we can determine the amount of project investment required to reach that objective. An income statement is built on this. The necessary services and their associated costs, as well as any adjustments to revenues, such as reimbursements, should all be considered in this circumstance.

3) Perform market research or a survey of the market

Our feasibility study's success depends on this stage. Therefore, we must do a thorough market analysis. The results of the market research will offer us the clearest image possible of the project's potential earnings and return on investment.

4) Plan Business Organization and Operations

It's time to set up the organization and operations of the intended project after the fundamental work of the earlier steps has been completed. This supports its ability to satisfy its technological, operational, economic, and legal requirements. It's important to understand that this is a serious project. Start-up expenses, fixed investments, and operating expenses should all be included in a comprehensive plan. These expenses cover items like machinery, marketing strategies, real estate, staff and supplier availability.

5) An Opening Day Balance Sheet should be prepared

An estimate of the assets and liabilities should be included in this, and it should be as accurate as feasible. We must make a list of the supplies and equipment, suppliers, expenses, and funding options in order to do this. The liabilities that we need to take into account include things like renting or buying property, buildings, machinery and financing assets.

6) Review and analyze all the data

The review and analysis are extremely vital, even if all of these procedures are significant. This verifies that everything is in order and that nothing needs to be adjusted or changed. Therefore, we must pause to review our work one last time. Our earlier actions, like the revenue statement, need to be reviewed. We must then contrast it with our liabilities and expenses. Additionally, now is the time to consider risks, analyze and manage them, and create any necessary backup plans.

7) Make a go or no-go decision

It's time to decide whether or not the project is feasible at this point. Although it appears straightforward, all the preceding processes have led to this choice-making stage. Before making that binary decision, you should also think about whether the commitment is worthwhile in terms of time, money, and effort. Additionally, we must think about whether it is consistent with the organization's long-term objectives and strategic goals.

2.3.4 Elements of Feasibility Study

Companies may use different formats for their feasibility reports, and we may receive particular guidelines from our company on how to format our own. There are a few crucial sections that almost all feasibility reports usually have as listed below.

1. Executive summary
2. Introduction
3. Background information
4. Evaluation criteria
5. Evaluation of solutions
6. Conclusion
7. Final recommendation

1) Executive summary

The executive summary is one of the first elements of a feasibility report. The main points of our study can be understood by our readers and can be read as an overview in an executive summary. The executive summary should be written simply and clearly, but it can be short. The following are some things to think about mentioning in the executive summary.

- A good summation of the information in our report, including the issue we are addressing or the project we are working on
- Opinions on the key points from our research or significant details from our study.
- Clearly state how the project or issue ties to our company's overall mission.

2) Introduction

The introduction is a key component of a feasibility report. We can write an introduction that describes the nature of the issue or project, as well as the suggested solutions, after the executive summary. Our introduction can be simple and short, similar to our executive summary, since we will go into greater detail later in the report.

3) Background information

Background information and environment should be included in a feasibility study. The purpose of this part is to clarify important contextual information for readers of the report. The project's history and objectives could be included in the background and context section if we're discussing various ways. When comparing several solutions to a problem, we could discuss how the issue came to be and how it affects our business. This will help our audience grasp the feasibility of possible methods.

4) Evaluation criteria

An explanation of our evaluation criteria can be provided in the report section. The purpose of this section is to explain to the readers of our report how we evaluated the feasibility of multiple possibilities and how we arrived at our suggestion. Financial costs, tax impacts, public perception, environmental effects and resources information could be part of our evaluation criteria.

5) Evaluation of solutions

The evaluation of solutions section is a vital part of a feasibility study. The goal of a feasibility report is to evaluate the feasibility of potential solutions and project approaches, and this section achieves in doing so. Comparing potential strategies using the evaluation criteria is what the evaluation part is for. We make your recommendation for the optimal strategy based on the evaluation procedure.

6) Conclusion

In a conclusion section, we can repeat our main points and summarize our report. The advantages and disadvantages of each of the strategies outlined could be briefly discussed in this section. This section's goal is to remind our audience how we assessed each strategy before we offer the final recommendation.

7) Final recommendation

Our specific recommendation for the best way to proceed is provided in the final part of a feasibility report. We can briefly discuss the solution in this section and discuss its feasibility as well as our reasons for selecting it.

([indeed.com, 2021](#))

2.3.5 Types of Feasibility Study

There are five distinct types of feasibility studies, which are divided into the following categories.

1) Technical Feasibility

The technical resources that the organization has access to are the main focus of this evaluation. Organizations can use it to assess whether technical resources are sufficient. And this helps to figure out whether the technical team can transform the concepts into reality. Evaluation of the proposed system's technical specifications, including its hardware, software, and other needs, is also a part of the technical feasibility process. Here, a number of elements need to be considered carefully, including the need for labor, availability, and technological expertise.

2) Economic Feasibility

An analysis of the project's costs or advantages is usually part of determining economic feasibility. Before allocating financial resources to a project, this helps businesses in evaluating its viability, costs, and advantages. This feasibility study strengthens project credibility by acting as an unbiased project evaluation. In the end, this assists decision-makers in determining the favorable economic benefits that the proposed project would bring to the business.

3) Legal Feasibility

Legal Feasibility assists in determining whether any elements of the planned project break any laws. Such as social media regulations, data protection legislation, or zoning laws. This kind of feasibility study may show the company its chosen location's legal suitability for zoning for that kind of business if it wants to establish a new office building in a particular location. By identifying whether or not their project was legally feasible from the start, that company can save a great deal of time and work.

4) Operational Feasibility

The structure of the organization, including its legal framework and management team's competence, are the main areas of focus for organizational feasibility. With the help of this study, we can decide if the project can be finished while still adhering to the organization's structure. A project plan's ability to satisfy the requirements is also examined through an operational feasibility assessment.

Operational Feasibility takes into account a variety of aspects, including the opinions of customers about recent products. We can manage the tasks of managers, stakeholders, and end users with the help of this study. By doing this, we can assess and address management, team, and individual resistance. And this takes into account potential changes to the workplace. Thus, we can provide a solution on how to get through these obstacles.

5) Schedule Feasibility

The schedule feasibility analysis reveals whether or not the project can be finished on time. The purpose of this study is to prevent the project from missing its deadline once it has been completed. The business, customers, and clients may choose the timeline. The business could lose clients and its employees could lose their jobs if the deadline is not met appropriately. The feasibility study also assists in determining how quickly the development process will go. If it takes too long, we can add more team members if necessary. However, if we can't afford to hire more people, we can train the employees we already have to work faster and more effectively.

2.3.6 Advantages of Feasibility Study

- With the use of a feasibility study, we may examine the market to determine whether the company's intended clients are in need of the product or service being offered.
- With the use of this study, we can examine the competitive positions and compare businesses.
- We can determine whether or not all the resources needed to start working are easily accessible.
- We are able to create a logistical or strategic plan outlining how our company will manufacture, store, transport, and monitor its products and services.
- We can calculate the sum of money required to begin managing the project, its costs, income, and the future cash flows.
- We are able to determine the company's legal framework.
- We can get to know how many and what kind of human resources are needed.

2.3.7 Disadvantages of Feasibility Study

- The study is initially only done on paper, therefore it won't identify any actual practical issues that could cause the business case to completely fail.
- Stronger simulations and iterations should be made to reduce any gap to solve issue which cost highly.
- The analysis could require some time and work, which is another drawback.
- According to the industry,

2.4 Investigation Techniques

2.4.1 Introduction to Investigations

Businesses that allocate a variety of resources to gather great knowledge can help people achieve their goals. Investigations are typically done to assess all of the possible dangers and frauds that can occur. In order to prevent these risks from occurring in the future and to reduce the loss caused by fraudulent actions, the company should conduct a investigation to gather more information. This information is acquired through an investigation using a carefully constructed questionnaire. After conducting a brainstorming session and keeping the goal of the initial survey in mind, the questionnaire is created.

The investigator (project manager or someone) manages various types of investigations depending on how we are doing our own independent investigations. Research Investigation, Corruption Investigation, Financial Investigation and Electronic Investigation are some types of investigations that businesses can conduct prior to the feasibility study.

Advantages →

- It is simple to tabulate the findings of a questionnaire-based investigation.
- This technique allows for the sampling of the respondent.
- This technique follows the same structure as the questionnaire.
- This is a quick way to collect data.
- Although it is an expensive approach, the expense compared to the advantages is minimal.
- It is simple to analyze questionnaires and tabulate the results.
- Due to its extensive structure, this approach reduces the likelihood of failure.
- The time needed to perform this procedure is minimal.

Disadvantages →

- This technique prevents the respondent from fully expressing his genuine emotions because there are few options for answers (to the questions).
- This technique is made extremely rigid and unflexible by the use of questionnaires because the responder can only respond to the questions that are included in the questionnaire. They can't move on from there.
- There is no return guarantee with this technique.
- This type of data collection is not 100% accurate. The validity of the results acquired using this method is therefore not guaranteed.
- There is no chance to expand the questions because the form is rigid.
- There is no room for volunteer information, thus if a respondent wants to provide any information outside the answer to a question on the questionnaire, he is unable to do so.
- As a result of this technique, responses to the questionnaire's questions cannot be supported with feelings or emotions.

2.4.2 Introduction to Interviews

Probably the most popular method for gathering data for investigations is interviewing people. The backstory underlying a participant's experiences can be learned through interviews. To learn more about an area of study, researchers can ask more questions and further study the answers or results. Prior to gathering data, however, the researcher must prepare and select the interview's structure.

Data on a wide range of issues are gathered from a small group of individuals during interviews. Interviews can be either structured or unstructured. Structured interviews are similar to questionnaires in that they ask the same questions about each topic in the same order and provide multiple choice answers. There is no fixed list of acceptable responses for unstructured interviews because questions might vary depending on the subject and on the responses to previous questions.

Advantages →

- This technique of data collection is informal.
- Interviews may be conducted whenever the process is in action.
- Nonverbal communication is also permitted during interviews.
- The respondent's quick feedback might be recorded during the interviews.
- Using this technique, the interviewer and interviewee converse face-to-face, allowing the interviewer to assess the participant's emotions and facial expressions.

Disadvantages →

- Since this technique uses an informal approach, an interviewer may occasionally deviate from the planned question on topic.
- The distance between the interviewer and the interviewee frequently makes the interviewing procedure impractical.
- This technique requires a lot of time.
- Since there is a structured questionnaire, this technique heavily rely on the interviewer's skills.
- The success of this technique depends on the interviewer's capacity for dealing with issues.

2.4.3 Introduction to Record Inspection

We must maintain accurate records while conducting an investigation. The main goal of an investigation is to gather evidence. Unless that evidence wasn't recorded at the time, nobody would know how trustworthy our findings might be. Humans are typically unreliable, so unless we record our findings, we might not even be able to trust them. Each measurement we take can be recorded in a lab notebook, where we can also make sketches of the equipment and make notes about anything we saw but might later forget.

Investigation records typically refer to any kind of records or materials that show the results of our investigation. These can be digital or hard copy, including different kinds of logs,

notebooks, letters, videos, computer databases, audio or digital records, or even the actual results of research. All records related to the investigation are crucial to maintaining accurate and comprehensive investigation records for data analysis. We may refer back to those records once our investigations are over and arrange the data into a format that is more suitable to sharing with others.

Advantages →

- It is not necessary to obtain the subject's permission to use their information for study if they have already given their permission. This can make data collection easier and allow for more complex study than would otherwise be feasible.
- Information can be gleaned more easily from numerous cases.
- This technique is faster and less expensive.
- Information included in records occasionally cannot be obtained in any other way.
- It is usually possible to collect random samples from record systems since they commonly provide a sampling frame.

Disadvantages →

- Records have a limited amount of information that can be gathered from them, and there is always more missing information when using records.
- Certain biases in information exist in records.
- Incomplete, unorganized, or missing records are all frequent.
- The researcher has no control over the quality of the records.
- The people who are in charge of keeping the records must have access, even though they might have a reason to oppose our request.
- Erroneous information may be recorded.
- Workers in the field may make mistakes when recording information on data sheets taken from documents.

2.4.3 Introduction to Observations

As the name suggests, observation is a technique of gathering data by observing. This technique of gathering data is categorized as a participatory study. Due to the fact that the researcher must take notes while simultaneously being fully immersed in the environment where her respondents are. A variety of techniques, including listening, studying, sensing, and recording behavior and phenomenon features, can be used to acquire data through observation.

Both structured and unstructured observation can be used to collect data. Data gathering in structured observation is done using predefined variables and on a set timetable. On the other hand, unstructured observation is carried out transparent and unrestricted transparent and unrestricted way because there aren't any predetermined variables or goals.

It is also possible to categorize this data collection technique as overt or covert. In overt observation, participants in the study are aware of being watched. On the other hand, with covert observation, the observer is hidden and the members of the sample group are not aware that they are being watched. Because individuals of the sample group in this situation are more likely to behave naturally. Covert observation is seen to be more effective because this has favourable effects on the reliability of research findings.

Advantages →

- With this technique, facts and opinions can both be addressed.
- The observation technique can be used in a realistic work environment where the investigation is performed.
- It is possible to perform measurements using this method.
- An understanding of the process is provided by this technique.

Disadvantages →

- Productivity may be interrupted frequently with this strategy due to observation.
- People do not like to be watched, and this strategy involves watching people's movements, reactions, and emotions.
- People hate being judged.
- It is necessary to allocate some more time for observation. As a result, the outcomes could differ from reality.

2.5 Feasibility Study Report

FEASIBILITY STUDY REPORT FOR E-Solutions Software Technologies

Ryan Wickramaratne

*Project Consultant,
Star Software Solutions.*



Table of Contents

1. Introduction

- 1.1 Client Company Overview**
- 1.2 Project Overview**
- 1.3 Project Goal and Objectives**
- 1.4 Significance of this Project**
- 1.5 Overview of the existing systems and technologies**
- 1.6 Scope of the project for E-Solutions Software Technologies**
- 1.7 Main Actors for the system**

2. Feasibility Study

- 2.1 Technical Feasibility**
- 2.2 Time/ Schedule Feasibility**
- 2.3 Economic Feasibility**
- 2.4 Legal Feasibility**
- 2.5 Operational Feasibility**

3. Comparison of Current and Proposed System

4. Conclusion for the Feasibility Study

1. Introduction

1.1 Client Company Overview

E-Solutions Software Technologies was founded on February 25, 2012, as an information and technology company that provides technical and programming knowledge to both local and international businesses. During its ten years in the sector, it was able to establish a monopoly in the creation and provision of information security technology software, as well as Web development and other services. Since its beginning, E-Solutions Software Technologies has strived to attain the greatest level of service quality.

1.2 Project Overview

The project management system used by the E-Solutions Software Technologies organization is manual and prone to error. As a result, they require a new automated system to replace the current, manual, error-prone system. The automation system should primarily cut the company's expenses while also greatly increasing productivity.

Along with encouraging teamwork, this transformation must also include an excellent project planner. The new automated system they require must be user-friendly and free of bugs or defects. Furthermore, the new system must be deployed on time and on budget, as well as fulfil client requirements and expectations.

1.3 Project Goal and Objectives

Table 2. 1 Project Goal and Objectives

Project Goal and Objectives	
Project Goal	Objectives
To enhance productivity and reduce expenses of the E-Solutions Software Technologies company a new automated system should introduce to replace the current, manual, error-prone system	Client requirements gathering and analysis.
	Make an efficient system.
	Create a system that is less expensive.
	Add features to the system that allow it to work as a group.
	Add analyzing tools to the system to keep projects on budget.
	Enhance the system with an automated project scheduling system.
	Enhance the system with features that identify personnel costs and information.
	Add features to the system that will create invoices when project phases are completed.
	Add features to the system to track projects and identify completed and ongoing tasks for each project.
	Provide system documentation.

1.4 Significance of this Project

E-Solutions Software Technologies is a significant client of ours. E-Solutions Software Technologies has a large client base that works with them, and they are an expanding good company in the IT area. Our collaborative work resulted in a positive relationship with our organization. E-Solutions Software Technologies has an old system that requires most operations to be completed manually, which takes longer and costs extra labour. Due to the size of E-Solutions Software Technologies' projects, they do not have the time to fix and improve the current system's capabilities. As a result, they require our assistance in order to improve their system while continuing their operations.

Since E-Solutions Software Technologies has a large client base, supporting them would allow us to extend our business to their customers as well. They would collaborate with us more in the future to handle the massive loads of the project, and this project would strengthen the relationship between our two companies, which will be beneficial to us in the future. And, as a favour, E-Solutions Software Technologies is willing to spend a significant amount of money on this job. So, with this project, we may earn a big amount of money while expanding our business and strengthening our company's future.

1.5 Overview of the existing systems and technologies

Most computers nowadays run Windows 10 or Windows 11. However, E-Solutions Software Technologies' existing system is compatible with older Windows versions such as Windows 8.1, Windows 8, Windows 7, Windows Vista, Windows XP Professional x64 Edition, and Windows XP. Most businesses and individuals no longer utilize these older operating systems. Furthermore, modern computer hardware performance is wasted on older versions of these operating systems.

Providing IT solutions to a large customer base necessitates extensive data analysis. This E-Solutions Software Technologies company was using an old mainframe from the 1990s, and due to the age of the hardware, it was only operational 60% of the time.

When speaking about the software system E-Solutions Software Technologies currently uses is a manual, error-prone system. The following flaws can be identified when studying this system.

- Staff Members, Project Managers, and Project Directors do not have separate logins.
- There are no different dashboards for the three sorts of users listed above.
- Project Directors cannot access real-time information about the number of employees, who are now active/offline, and who are unproductive.
- When entering, deleting, or updating data, the system generates problems on various pages.
- The system lacks essential pages such as the Invoicing page, the Time and Attendance page, the Running Projects page, the Project Profile page, and the Teams page.
- The system does not allow to designate Project Managers for existing projects.
- There is no stopwatch to keep track of time. of every project
- There is no Kanban Board View system in place to monitor project progress.
- There is no effective way to create tasks and display them in the system's ongoing tasks.
- There is no efficient way to update the status of a task.

- There is no effective way in the system to reveal further details about the task.
- There is no efficient way to allocate tasks to team members in the system.
- With the current system, the Project Director cannot create teams for a given project, assign personnel to the teams, or assign a team leader.
- Like a project manager, the team leader is unable to delegate work to the team members.
- Users cannot create checklists to display mini-tasks and progress toward completion.
- There are no deadline alerts or notifications in the system.
- There are no automated Email notifications in the system.
- The Dashboard view displays less team progress and provides fewer insights into team performance.
- There is no Calendar view to provide a monthly view of project management.
- Users are unable to attach files, notes, or comments to running tasks.
- The system lacks a discussion function that allows users to communicate one another, tag members in comments, and organize their work.
- Project managers are unable to track team members' time spent on projects and tasks.
- The system does not include a function that alerts employees when they are idle for an extended period of time.
- The system does not have an automatic payroll calculation feature.
- Payroll and invoicing activities cannot be automated by the system.
- When users push the submit button, the confirmation screen takes a long time to load.
- The home page takes a long time to load.
- Sensitive data is not securely stored.
- The system does not have enough storage capacity.
- The existing system makes extensive use of memory.
- Customer information is not encrypted in accordance with encryption standards.
- The system does not support the most recent versions of Windows and iOS.

- The system is not portable. This means that switching from one operating system to another causes issues.
- Most of the time, the system is unavailable to end users due to maintenance.
- In web-based applications, the system response to user activity takes a long time.
- Every failed attempt by a user is not recorded.
- The system cannot accommodate 100 users without degrading its performance.
- In fewer than 24 hours, the system cannot recover from any operational disruption.
- If the automated email services go down, they will be inaccessible for about 3 days.
- If a big problem occurs on the system, the company cannot be completely functioning within two days.
- Every user has access to the employee's personal information.
- When editing essential pages that require assistance from other windows, the remainder of the windows do not stay open.
- The system does not ensure data integrity by retaining backups of all database updates for each record transaction.
- The system interface is extremely complicated.
- All security configuration changes are not logged by the system.
- All audit log entries that are not associated with the user who performed the action.
- The system is unable to track and erase test transactions.
- The system is incapable of providing a responsive user interface. (From smartphones to monitors).
- The system does not have an easy installation or upgrade process.
- There is no way to export all data to machine-readable format from the system.
- In the event of a system failure, the system is unable to immediately switch to a redundant environment.
- The system is not compatible with normal backup/recovery software.
- The system does not support on-screen contextual prompts that instruct users on how to operate the software.

1.6 Scope of the project for E-Solutions Software Technologies

Table 2. 2 Scope of the Project - E-Solutions Software Technologies

Scope of the Project - E-Solutions Software Technologies	
Project Title : "E-Solution Projects" software.	Date Prepared : 02/01/2023
<p>Project Scope Description :</p> <p>Due to the global technological revolution, the majority of business systems are becoming outdated. With their current workload, those businesses lack the time and resources to improve their systems. E-Solutions Software Technologies also has old hardware and a highly outdated project management system. They frequently experience technical difficulties, and as a result, their productivity decreases. As a result, we are here to improve their system with IT consulting so that they can keep up with new technologies and increase their efficiency.</p>	
<p>Project Deliverables :</p> <p>The following are the main deliverables of the project. We'll create a more efficient and cost-effective system. The system will include features that enable it to function as a group. In addition, the system will provide analysis capabilities to help keep projects on track. We improve the system by adding an automated project scheduling system as well as capabilities that identify personnel costs and information.</p> <p>In addition, the system will have features that will generate bills when project phases are completed. The system will be capable of tracking projects and identifying completed and continuing tasks for each one. We'll offer a thorough system documentation following all of this.</p>	

Success (Acceptance) criteria:

Putting in place founded solutions from this project to vulnerabilities in cyber security concerns at Lions Restorations Company.

Inclusions :

- This project will deliver improved project management software along with proper documentation.
- This project will provide recommendations for upgrading hardware and other non-related software systems.

Exclusions :

- Hardware components won't be offered as part of this project.
- Other unrelated software upgrades will not be offered by this project.

Constraints:

The project has a strict deadline of December 31st, 2023, and a budget of \$50,000 dollars.

1.7 Main Actors for the system

Project Director: The E-Solutions Software Technologies company's projects are all managed by a project director. At this company, they are executive leaders who supervise and direct project managers. They are also in charge of tactical or strategic projects. They must therefore be able to monitor the overall team performance as well as all the project insights in their dashboard. Each project in this organization has a "project profile" that the project director creates. This profile details the project's staff costs, the activities that are assigned to it, and the project manager who will be in charge of overseeing it. The Project Director can form teams for a given project, assign people to the teams, and appoint a team leader.

Project Manager: The project manager works for E-Solutions Software Technologies as a mid-level manager. They supervise a project's daily progress. In this organization, project managers collaborate closely with team members to assign tasks, monitor their activities, and generate status reports. When planning projects, project managers and project directors may occasionally collaborate on decisions such as deciding which team members to assign to specific tasks. Their job makes sure that projects go off without a problem, and they can make adjustments to the project plan as necessary to achieve the project's objectives.

Team Lead: This company's team lead gives advice and direction to a working group over project activities. They are accountable for assigning small tasks to team members, monitoring the progress of tasks, and providing coaching as necessary. The amount of information they need to see on the dashboard should be restricted because they are primarily responsible for managing project tasks.

Team Members: Team members are members of the team that is directed by the project manager and team lead. They are unable to add Projects or edit or remove a Team's profile, however. As soon as they are scheduled to tasks, they need to carry out the given activities. Therefore, team members have very limited access to the dashboard.

2. Feasibility Study

2.1 Technical Feasibility

Good computers are used by E-Solutions Software Technologies to run their tasks. Every Project Manager and Project Director has a good PC that can operate our proposed system. They have 2GB of external VGA with intel 960 GTX graphics, 8GB of RAM, and an Intel i5 10th Generation CPU, which is more than enough to execute our proposed system.

And all Team Leads and Developers (Staff) have excellent computers that are powerful enough to execute our proposed system while carrying out their project activities. They have an 8GB external VGA with Intel 1060 GTX graphics, 16GB RAM, and an Intel i7 10th Generation CPU, which is more than enough to handle our proposed system as well as a variety of software applications.

With cable connections, this E-Solutions Software Technologies organization gets a moderate internet speed of 250 Mbps. However, we advised that they be upgraded to Fiber optics with a minimum download speed of 1000 Mbps in order to execute our proposed system with other software applications at optimum speed. Streaming conference calls, executing projects, and generating multimedia projects are all possible with practically parallel upload and download rates (940/880). Meanwhile, clients can reduce their dependence on hard drives because our service is fast enough to allow them to run our system from the cloud.

E-Solutions Software Technologies does have an old mainframe from the 1990s, and because of the age of the hardware, it was only operational 60% of the time. We suggested replacing them with new servers and purchasing Amazon Web Services (AWS) to operate web-based applications. Among the several tiers that Amazon Web Services (AWS) offers is a 12-month free usage tier for new customers of both its Simple Storage Service and Elastic Block Storage up to 30GB. Additionally, the website offers Amazon EC2 customers a free shared file storage service of 5GB for 12 months and 100GB of free cloud storage per account with optimized data transfer. So, with this, E-Solutions Software Technologies will be feasible to run their existing projects along with our proposed system.

Although E-Solutions Software Technologies' cyber security system is not satisfactory, our proposed system has excellent security features. Their expertise and procedures for cyber security are questionable for running our system. Therefore, we suggested that they run a campaign to raise awareness of cyber security among their employees and requested that they install firewalls on the firm network infrastructure. Furthermore, they were advised to use SolarWinds Security Event Manager since it offers Threat Intelligence, SIEM Security & Monitoring, Log Correlation & Analysis, and Network & Host Intrusion Detection. As a result, our system could be operated with the best security measure by combining network security features with built-in security features of the software.

Conclusion:

It is not enough to provide only sufficient technological resources to a proposed system because that is not the primary goal of the "E-Solutions Software Technologies" organization. They must have sufficient technological resources to perform their main operations while our system runs in the background.

Therefore, it is evident from this Technical Feasibility analysis that the E-Solutions Software Technologies are not currently Technically Feasible of operating the proposed system ("E-Solution Projects" software.).

But if the client company follows our advice prior to deploying the system to their network, it will be technically feasible for them to run our proposed system alongside the existing projects they already have.

2.2 Time/ Schedule Feasibility

Breaking work down into smaller pieces is a typical productivity method intended to make work more manageable and accessible. The Work Breakdown Structure (WBS) is a tool that uses this technique for projects and is one of the most significant project management papers. It incorporates scope, cost, and schedule baselines all on its own, ensuring that project plans are in alignment. A good WBS simply makes the project easier to manage. Every project is unique, just as every project manager and WBS are unique. The WBS that best addresses the question of "What structure makes the project more manageable?" is thus the optimal WBS.

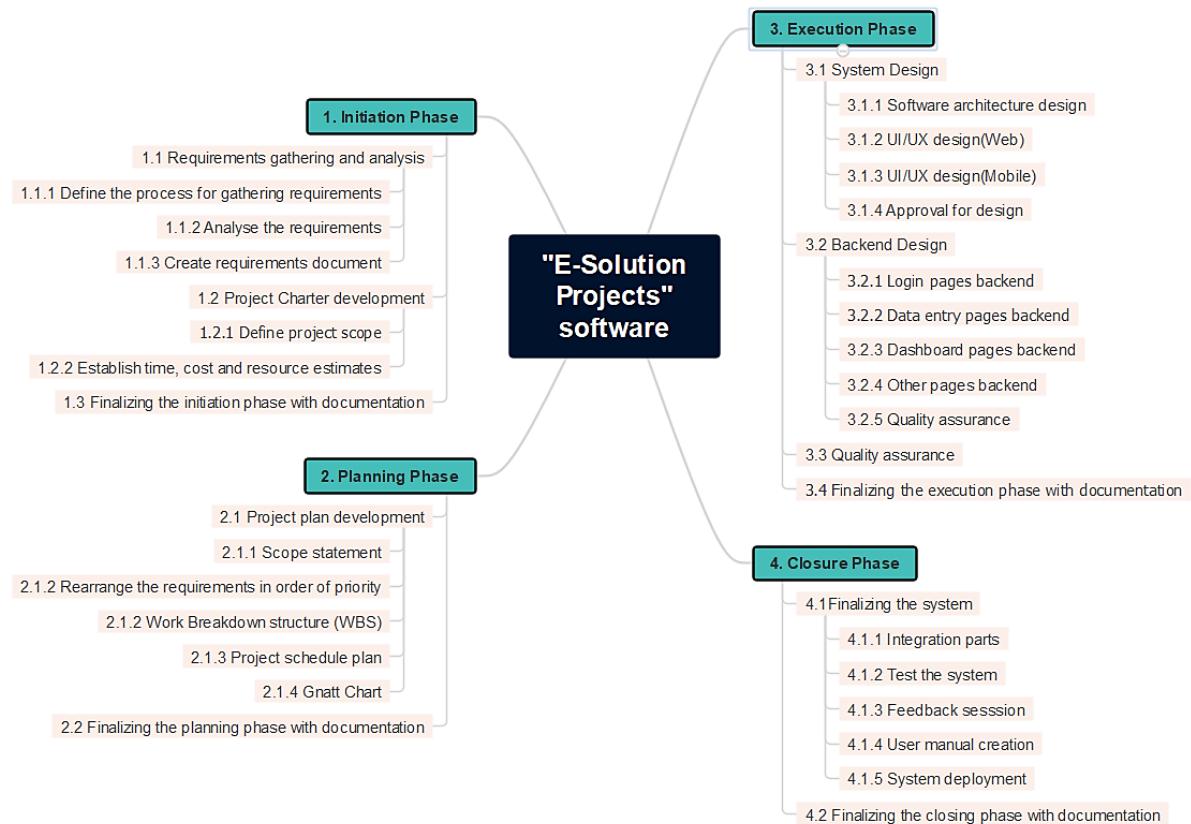


Figure 2. 9 Work Breakdown Structure for the project

One of the most common and effective methods of displaying activities (tasks or events) displayed against time is a Gantt chart, which is frequently used in project management. A list of the activities is located on the chart's left side, and a suitable time scale is located along the top. A bar is used to symbolize each activity, and the position and length of the bar correspond to the activity's beginning, middle, and finish dates.

This allows us to quickly determine the following things:

- What each of the actions requires.
- When the start and end of each action.
- How much time is allocated to each activity?
- Where, and to what extent, certain activities overlap with others.
- The time when the entire project will begin and end.

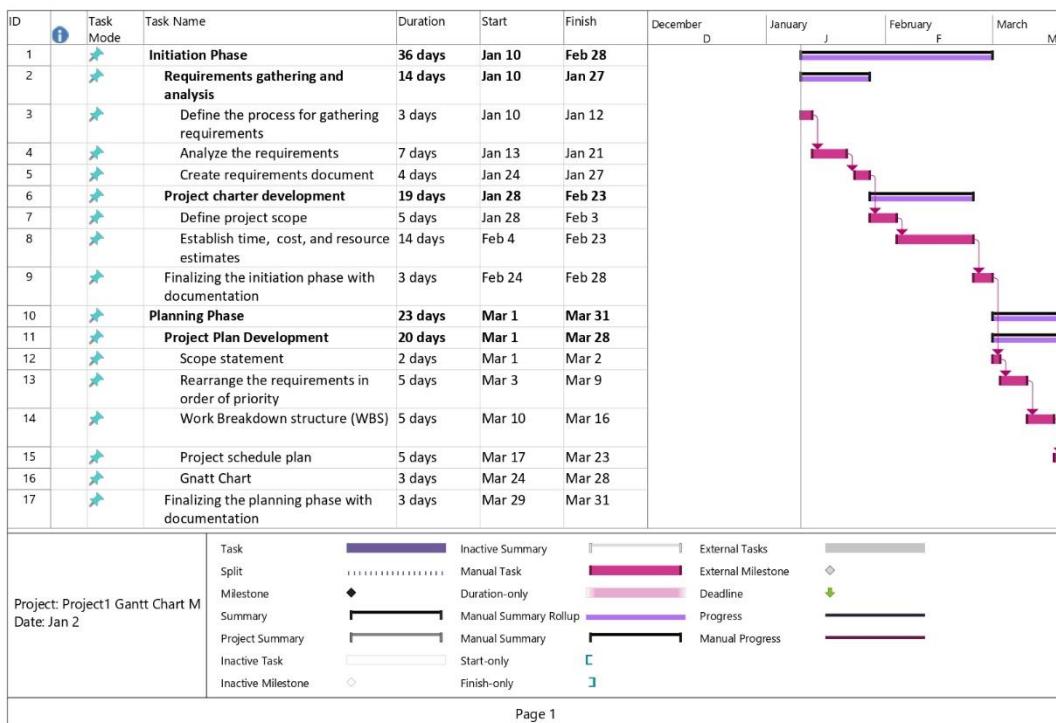


Figure 2. 10 Gantt Chart for the project I

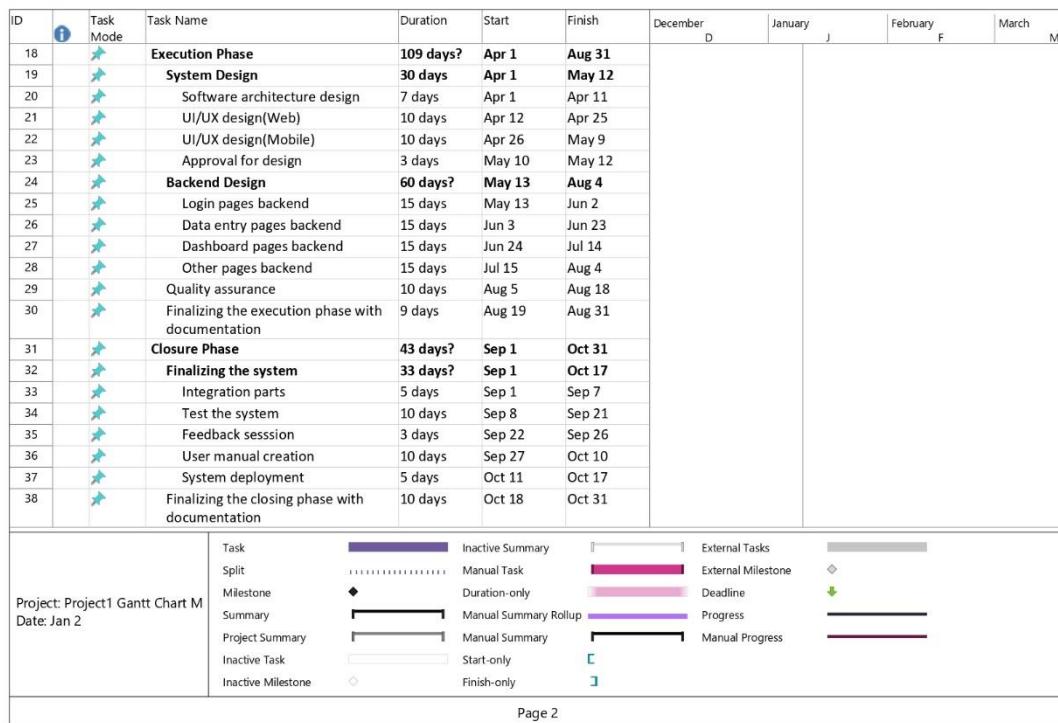


Figure 2. 11 Gnatt Chart for the project 2

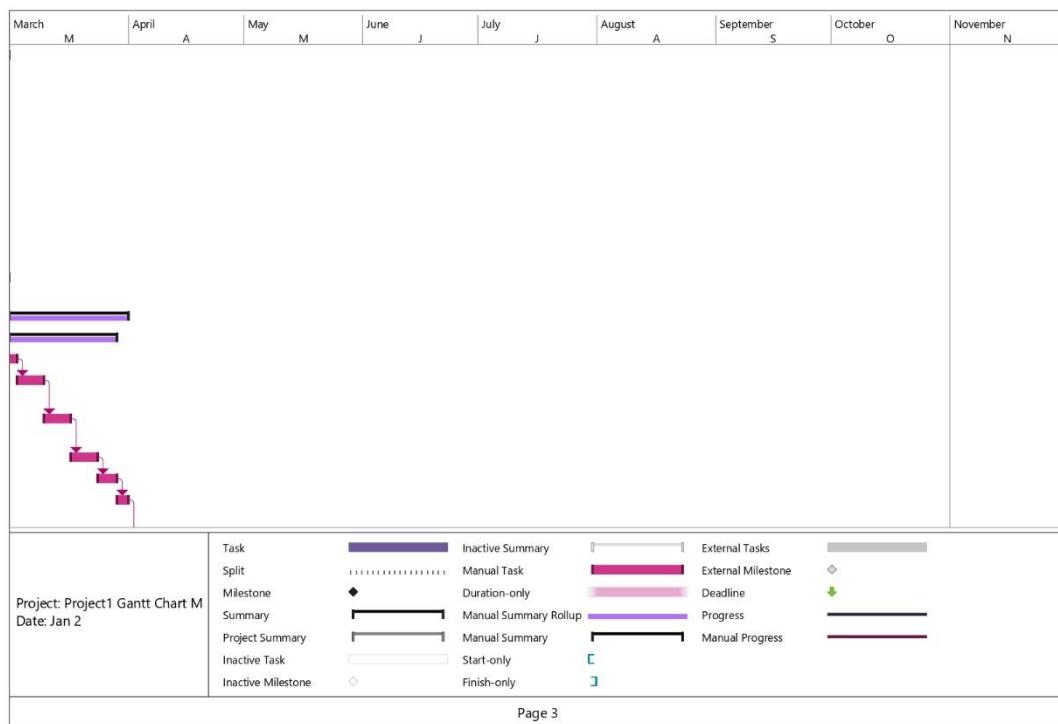


Figure 2. 12 Gnatt Chart for the project 3

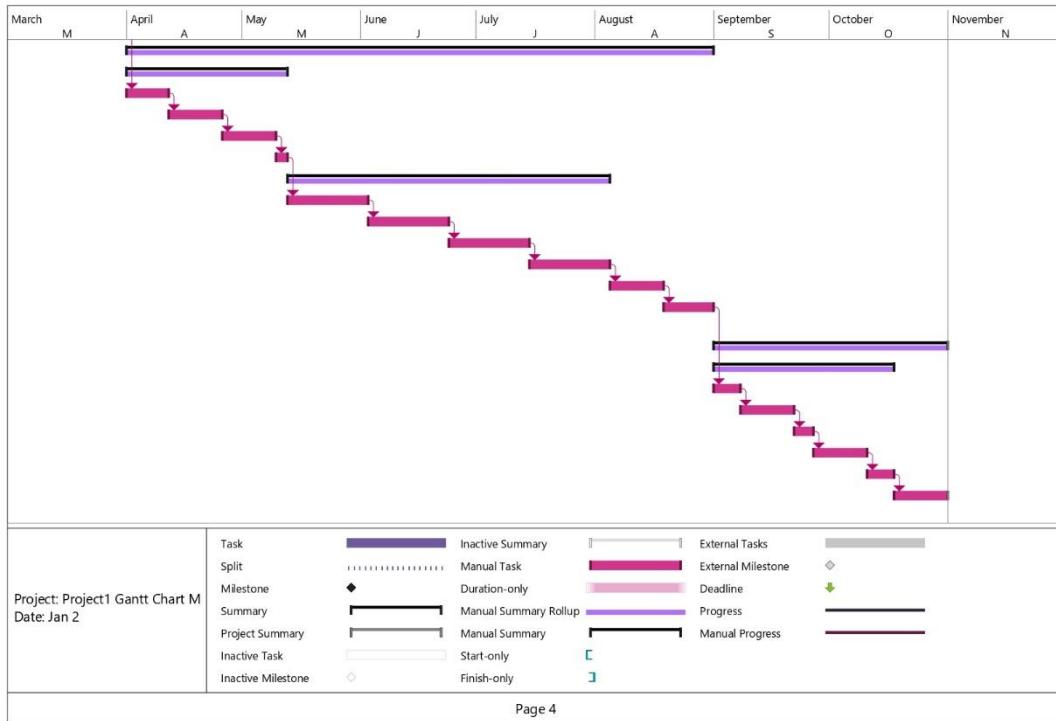


Figure 2. 13 Gantt Chart for the project I

Conclusion:

The WBS and Gantt Chart above indicate that the project can be completed by the deadline. The project can be completed before the month of November if we follow the timetables provided, even though the deadline is December 31, 2023. Therefore, we have more than enough time and two more months to deal with unforeseen circumstances.

Therefore, it is evident that this project is Time / Schedule feasible.

2.3 Economic Feasibility

A	B	C	D	E
Project Budget				
1	Project Name:	"E-Solution Projects" software		
2	Department:	IT		
3	Supervisor Name:	Ryan Wickramaratne		
4				
5				
6				
7				
8				
9	WBS	Project Tasks	BUDGETED	ACTUAL
10	1.1	Requirements gathering and analysis	200.00	150.00
11				50.00
12	1.2	Project charter development	300.00	200.00
13				100.00
14	1.3	Finalizing the initiation phase with documentation	200.00	350.00
15				(150.00)
16	2.1	Project plan development	300.00	400.00
17				(100.00)
18	2.2	Finalizing the planning phase with documentation	200.00	350.00
19				(150.00)
20	3.1	System Design	200,000.00	150,000.00
21				50,000.00
22	3.2	Backend Design	200,000.00	150,000.00
23				50,000.00
24	3.3	Quality assurance	200,000.00	100,000.00
25				100,000.00
26	3.4	Finalizing the execution phase with documentation	1,000.00	850.00
27				150.00
28	4.1	Finalizing the system	100,000.00	85,000.00
29				15,000.00
30	4.2	Finalizing the closing phase with documentation	200.00	350.00
31				(150.00)
32	SUBTOTAL		702,400.00	\$487,650.00
				214,750.00

Figure 2. 14 Project Budget Plan

The chart above is an economic evaluation of the project that deals with monetary elements that can be evaluated, measured, and compared. This project budget plan displays the cost and value of the project. The actual value is less than the budgeted value, and it is also less than the client's budget (client budget - \$50,000).

Conclusion:

There are no economical obstacles to the project's implementation, according to the above economic feasibility study.

Based on that, the development of the "E-Solution Projects" system for E-Solutions Software Technologies is Economically Feasible.

2.4 Legal Feasibility

Although a well-written software licensing agreement should hold the interest of both the developer and the client, the protection of intellectual property rights is of utmost importance to the software developer. So that there won't be a legal issue when creating the system for E-Solutions Software Technologies since this project can operate under the following legal conditions.

- 1) Software development agreements → The development of a software application for E-Solutions Software Technologies is permitted in accordance with this agreement. Therefore, we may give them access to the system while transferring their intellectual property rights.
- 2) Framework software development agreements → With the use of this Framework Agreement, we are eligible to define our respective rights, responsibilities, and liabilities in relation to developing the system for the E-Solutions Software Technologies.
- 3) Source code license agreement → We can protect our source code doubly under this source code license agreement. The source code may be made accessible to E-Solutions Software Technologies with this agreement under tight licensing conditions. Additionally, the source code may be classified as confidential information and be subject to disclosure limitations as well as security requirements.
- 4) Download terms and conditions → With the system we can provide the terms and conditions for downloading including typical clauses addressing the ordering procedure, pricing and payments, distant selling policies, warranties, liability restrictions, and interpretations.
- 5) Software maintenance agreement → This legal contract outlines our company's obligation to the E-Solutions Software Technologies' need for technical support and upgrades for an existing software product.

Conclusion:

There are no legal obstacles to the project's implementation, according to the client's requirements and conditions.

Based on that, the development of the "E-Solution Projects" system for E-Solutions Software Technologies is Legally Feasible.

2.5 Operational Feasibility

The client company needed a system to enhance productivity and reduce the expenses of the company. So, with our new automated proposing system they can replace their current, manual, error-prone system. With our proposed system the client can resume operations effectively. The proposed system offers every functionality to meet the client's needs while staying within budget.

Our proposed system includes all of the functionality that the client E-Solutions Software Technologies company requires to efficiently organize these projects and operations. With the help of our system, they can control their available resources, address issues, and keep in touch with all project stakeholders.

Our system's ability to make collaboration between team members, team leads, the project manager, and the project director of the client company exceedingly simple is one of its major benefits. All communication can be maintained using this system in one location. Furthermore, our system enables us to send significant notifications to the appropriate parties immediately, as well as access the timetable and status updates for daily operations with just a single click.

By using remote access, our system is simple to maintain. This makes it easier to resolve technical problems and lowers maintenance costs. System problems can easily be fixed at any anytime by our company's IT professionals. This remote access operating feature can help ensure the quality of our product, particularly in unforeseen circumstances like the COVID-19 pandemic. This functionality eliminates the need for our IT specialists to travel

to the location to patch issues and perform maintenance. Additionally, the client does not need to bring their devices in to be fixed.

Our device has on-screen tutorials, which can save users time and motivate them to learn on their own by giving them the appropriate help when they need it. We can greatly improve our product's usability and uptake with on-screen assistance while also lowering user frustration. This feature makes users feel good and helps them understand how our product has helped them with an issue, an inquiry, or the learning of a new skill. As a result, the time it takes for our client company employees to fully utilize our system is reduced thanks to in-app guidance.

The system we provide does not put the client company in a novel situation. The project manager, project director, team members, and team leads of the E-Solutions Software Technologies company are all experienced professionals. They all have quite a bit of experience with the traditional project management system. The employees of the client company can use all of our system's capabilities despite the fact that it is a little bit more advanced than the old system they used. But with our helpful hints and in-app instructions, they can learn all of the functions of our system. And we give a comprehensive walkthrough of the system in the user manual, from A to Z.

Conclusion:

Therefore, there are no operational difficulties in the project's implementation. The information above describes how we address every potential operational problem.

Based on that, the development of the "E-Solution Projects" system for E-Solutions Software Technologies is Operationally Feasible.

3. Comparison of Current and Proposed System

Table 2. 3 Comparison of Current and Proposed System

Current System	Proposed System
Staff Members, Project Managers, and Project Directors do not have separate logins.	Staff Members, Project Managers, and Project Directors have separate logins.
There are no different dashboards for the three sorts of users listed above.	There are 3 different types of dashboards for the three sorts of users listed above.
Project Directors cannot access real-time information about the number of employees, who are now active/offline, and who are unproductive.	Project Directors can access real-time information about the number of employees, who are now active/offline, and who are unproductive.
When entering, deleting, or updating data, the system generates problems on various pages.	When entering, deleting, or updating data, the system does not generate problems on various pages.
The system lacks essential pages such as the Invoicing page, the Time and Attendance page, the Running Projects page, the Project Profile page, and the Teams page.	The system consisting essential pages such as the Invoicing page, the Time and Attendance page, the Running Projects page, the Project Profile page, and the Teams page.
The system does not allow to designate of Project Managers for existing projects.	The system allows designating Project Managers for existing projects.

There is no stopwatch to keep track of time of every project.	There is a stopwatch to keep track of time of every project
There is no Kanban Board View system in place to monitor project progress.	There is a Kanban Board View system in place to monitor project progress.
There is no effective way to create tasks and display them in the system's ongoing tasks.	The system comes with task cards which is an effective way to create tasks and display them in the system's ongoing tasks.
There is no efficient way to update the status of a task.	Task cards in a workflow are able to shift from one task status to another by dragging and dropping.
There is no effective way in the system to reveal further details about the task.	Each card in the system displays a popup with more information about the task when clicked.
There is no efficient way to allocate tasks to team members in the system.	Project managers can delegate work to specific team members by assigning cards to them.
With the current system, the Project Director cannot create teams for a given project, assign personnel to the teams, or assign a team leader.	The Project Director can form teams for a specific project, assign personnel to the teams, and choose a team leader.
Like a project manager, the team leader is unable to delegate work to the team members.	The team leader can give the team members tasks to perform.

Users cannot create checklists to display mini-tasks and progress toward completion.	Users can create checklists to show mini-tasks and progress toward completion.
There are no deadline alerts or notifications in the system.	The system includes deadline notifications and alarms.
There are no automated Email notifications in the system.	Automated email notifications are included in the system.
The Dashboard view displays less team progress and provides fewer insights into team performance.	Team progress and team performance should be shown in the Dashboard view.
There is no Calendar view to provide a monthly view of project management.	There is a Calendar view to provide a monthly overview of project management and an easy way to adjust start and due dates by dragging and dropping.
Users are unable to attach files, notes, or comments to running tasks.	Users can add files, notes, and comments to cards.
The system lacks a discussion function that allows users to communicate with one another, tag members in comments, and organize their work.	A discussion feature is included in the system so that users can communicate with one another, tag other users in comments, and organize their work.
Project managers are unable to track team members' time spent on projects and tasks.	Project managers can keep track of how much time each team member spends on projects and tasks.

The system does not include a function that alerts employees when they are idle for an extended period of time.	If an employee is inactive for a long period of time, the system can notify them.
The system does not have an automatic payroll calculation feature.	The system has an automatic payroll calculation feature to calculate the wages that the employees are due in a fair and precise manner.
Payroll and invoicing activities cannot be automated by the system.	Invoicing and payroll operations can be automated in the system.
When users push the submit button, the confirmation screen takes a long time to load.	The confirmation screen loads in no more than two seconds after pressing the submit button.
The home page takes a long time to load.	The home page loads in 3 seconds or less.
Sensitive data is not securely stored.	Sensitive data is securely stored.
The system does not have enough storage capacity.	Up to 10TB of data can be stored on the system.
The existing system makes extensive use of memory.	The system only uses up to 8GB of RAM.
Customer information is not encrypted in accordance with encryption standards.	Customer information is protected using level 4 encryption.
The system does not support the most recent versions of Windows and iOS.	The most recent versions of Windows and iOS are supported by the system.

The system is not portable. This means that switching from one operating system to another causes issues.	When switching from one OS to another, the system doesn't cause any issues, hence the system is portable.
Most of the time, the system is unavailable to end users due to maintenance.	99.999% of the time, the system is accessible to users.
In web-based applications, the system response to user activity takes a long time.	In web-based applications, the system reacts to user activity in under 1 second.
Every failed attempt by a user is not recorded.	Every unsuccessful attempt by a user to access data is documented on an audit trail.
The system cannot accommodate 100 users without degrading its performance.	The system is able to accommodate 1000 users without experiencing performance issues.
In fewer than 24 hours, the system cannot recover from any operational disruption.	The system can recover from any operational disturbance in less than 4 hours.
If the automated email services go down, they will be inaccessible for about 3 days.	The automated email systems can be out of operation for about 3 hours if they do.
If a big problem occurs in the system, the company cannot be completely functioning within two days.	If a major event occurs on the system, we can restore full system functionality within 2 days.
Every user has access to the employee's personal information.	The employee's personal information only can be seen by users with the role of "Project Director."

When editing essential pages that require assistance from other windows, the remainder of the windows do not stay open.	When editing a project profile, the other windows remain open.
The system does not ensure data integrity by retaining backups of all database updates for each record transaction.	The system preserves data integrity by making regular backups of all database updates for each record transaction.
The system interface is extremely complicated.	The system interface is simple to use and friendly to users.
All security configuration changes are not logged by the system.	All modifications to the security settings is logged in by the system.
All audit log entries that are not associated with the user who performed the action.	All audit log entries connected to the user who carried out the action.
The system is unable to track and erase test transactions.	The system offers a method to track and delete test transactions.
The system is incapable of providing a responsive user interface. (From smartphones to monitors).	A responsive user interface is offered by the system. (from smartphones to large displays)
The system does not have an easy installation or upgrade process.	The system allows for automatic installation and updating procedures.
There is no way to export all data to machine-readable format from the system.	There is a method for exporting all data to a machine-readable format provided by the system (XML or JSON)

In the event of a system failure, the system is unable to immediately switch to a redundant environment.	In the event of a system breakdown, the system is able to automatically switch to a redundant environment.
The system is not compatible with normal backup/recovery software.	Standard backup/recovery tools such as Veeam, and Zerto are compatible with the system.
The system does not support on-screen contextual prompts that instruct users on how to operate the software.	The system gives on-screen contextual prompts that instruct users on how to use the software.

4. Conclusion for the Feasibility Study

E-Solutions Software Technologies information and technology company that provides technical and programming knowledge to both local and international businesses. They require a new automated system to replace the current, manual, error-prone system. So our project goal is to enhance productivity and reduce expenses of the E-Solutions Software Technologies company with a new automated system.

When developing the system, we must take into account four key actors (Project Director, Project Manager, Team Lead and Team Member). The feasibility study in the report is solid. The technical feasibility, time/schedule feasibility, economic feasibility, legal feasibility, and operational feasibility were all taken into account in this study. According to the comparison between the present and proposed system, our system addresses all of the client's specific requirements and problems.

Activity 3

3.1 Requirements

3.1.1 Introduction to Requirements in Systems Analysis & Design

Every software project needs a complete set of requirements. High-level requirements specify the demands and objectives of the product's business. Additionally, they define the attributes, capabilities, behaviors, and performances that stakeholders demand. The why, what, and how of an application used by a business can be identified and clarified using software requirements.

Software requirements can serve as a roadmap to help a development team construct the right product fast and with the least amount of cost revision when they are well documented. Depending on the target audience and the project's maturity, an IT organization may develop different types of software requirements and documents for a specific project. The needs of business executives, project managers, and software developers are frequently taken into account when a company drafts multiple requirements documents.



Figure 3. 1 Requirements in Systems Analysis & Design

3.1.2 Types of Requirements in Systems Analysis & Design

There are several types of requirements in Systems Analysis & Design as follows.

- 1) Business Requirements.
- 2) User Requirements.
- 3) Domain Requirements.
- 4) System Requirements.

1) Business Requirements

Many software projects are motivated by business needs. Project goals that are measurable for the company, users, and other stakeholders are described in a Business Requirements Document (BRD). The Business Requirements Document is created at the outset of the project by business analysts, executives, and other sponsors. The purpose of the project is explained by the Business Requirements Document. This outlines the advantages that the producing company or its clients anticipate from the product.

The Business Requirements Document is used as a starting point for more thorough document preparation with clients by software development contractors. The statements that make up a Business Requirements Document can be one or more.

Following is the basic format of a BRD statement.

"The [project or product name] software will [meet business goal] in order to [fulfill business benefit]."

The following is an example BRD statement for E-Solutions private Limited.

The "E-Solution Projects" software will allow the Project Managers to assign projects to the employees quickly in order to save time and energy.

2) User Requirements.

The special demands or expectations of the software's users are reflected in the user requirements. Such as what functions users must be able to carry out. In a User Requirements Document (URD), narrative material is typically used to describe user requirements. The user typically approves user requirements, which are the main source of information for formulating system requirements.

Finding out what the user truly wants a piece of software to do is a crucial and challenging phase in its development. This is due to the fact that the user frequently struggles to fully express their requirements and desires, and that the information they supply may also be unreliable, imprecise, or inconsistent.

The business analyst is in charge of fully comprehending what the consumer wants. Because of this, user requirements and system requirements are typically treated differently. In order to ensure that the requirements meet certain quality standards, the business analyst thoroughly examines user requirements before carefully creating and documenting a set of high-quality system requirements.

User requirements are much like the user stories in the business case that illustrate how consumers interact with the system. Although there isn't a single, widely acknowledged format for user requirements statements, the following is common.

"The [Type of the user or position] shall [user's interaction with the software] in order to [business goal or a result]."

The following is an example BRD statement for E-Solutions Private Limited.

"The Project Director shall be able to create a project and a “project profile” for each project to manage purchase orders coming to the E-Solutions Private Limited."

3) Domain Requirements.

The requirements relating to a specific category, such as software, industry, or other project domain, are known as domain requirements. Both functional and non-functional requirements may apply to a domain. These are fundamental capabilities that every system made up of particular domains must possess. Domain requirements have one thing in common: they adhere to generally recognized feature sets or established standards for that particular category of software project.

Typically, the military, healthcare, and financial industries will have domain requirements. They are not user-specific and can only be identified from that particular domain. Examples of domain needs include software for schooling or medical equipment.

Example no 1:- Medical equipment software

- Software for medical equipment must be created in accordance with the requirements of IEC 60601 for the performance and basic safety of medical electrical equipment.
- Even though the program is usable and functional, it cannot be used in production since it does not adhere to the standards set out by the domain.

Example no 2:- Educational software

- It is necessary to create such software to efficiently maintain an institute's records.
- The capability of accessing the list of professors and list of students by grade is a domain requirement for such software.

3.2 System Requirements

3.2.1 Introduction to System Requirements in Systems Analysis & Design

The building blocks that developers utilize to construct the system are called system requirements. These statements outline what the system "must" perform. These requirements, which the customer provides in order to be satisfied, are both wide and precise. The customer should express exactly what they want and how they want it in the statement. Functional requirements and Non-Functional requirements are the two categories of system requirements.

A Functional Requirement outlines what a user requires in order to complete their task. An example of a functional requirement would be for a system to be able to enter and print cost estimates. All other requirements not covered by the functional requirements are described as non-functional requirements. The system architecture includes specifics about how to implement non-functional requirements, such as accessibility, speed, and performance.

The best method for satisfying user needs while lowering implementation costs is through system requirements. Customers, businesses, or even other large entities that establish project requirements may provide system requirements. System requirements can save a business a lot of money and time, but they can also cost the business money and effort. They are the most crucial component of any project since if they are not met, the project cannot be considered finished.

Following is the basic format of a Software Requirements Specification (SRS) statement.

"The [feature/function of the software] shall [perform an action based on user inputs and deliver matching outputs]."

The following is an example SRS statement for E-Solutions Private Limited.

"The Print button shall direct to the printing window, which shall provide an overview of all

3.2.2 Introduction to Functional Requirements

The most straightforward definition of a functional requirement is “Something the system have to do.” The kind of software requirements known as functional requirements are those that end users specifically want as fundamental features of the system. The system will not work properly if a functional requirement is not met. It will be unable to do an operation that is necessary for it to function effectively, which is why. Therefore, these functional requirements should unavoidably be included in the system as part of the agreement.

We can better understand the desired behavior of the system with the aid of functional software requirements. Functional requirements might outline a system's expected performance of a computation, manipulation of data, business process, user interface, or any other specialized feature. Examining the system's inputs and outputs might help us to understand the concept of a functional requirement. Functional requirements outline what the system must output and how it must react to various inputs.

In contrast to non-functional needs, they are essentially user-stated criteria that are visible in the finished product. For instance, a doctor should be able to access his patients' information in a hospital management system. There may be multiple exchanges or conversations between the system and the outside world for each high-level functional requirement. All scenarios must be listed in order to completely define the functional requirements.

3.2.3 Examples of Functional Requirements

For clarification, it can be useful to look at some examples of functional requirements. Functional requirements typically include both user requirements and product features as follows.

- System ability to report on how many transactions were correctly handled.
- What the system does after a user clicks a particular button, and where they are directed.
- What takes place if the system is attacked

- When a user logs in to the system, the way they are authenticated.
- Every time an order is placed, the system needs to send a confirmation email.
- Visitors to blogs must be able to subscribe to the newsletter by entering their email address into the system.
- Users must be able to use their phone numbers to verify their accounts on the system.
- After the user submits personal information, the system should send an approval request.
- If a user wants to credit an issued invoice, they should be able to utilize a search option to look through numerous invoices.
- A confirmation email should be sent by the system after the creation of a new user account.

3.2.4 Functional Requirements for the E-Solutions Private Limited New System

- There must be 3 different logins for Staff Members, Project Manager and Project Director.
- Different dashboards must be created for each of the three sorts of users mentioned above.
- The dashboard for project directors should display real-time information on the number of employees, who are online/active at the moment, and who is being unproductive.
- Every user should be able to add, remove, and update data in the system.
- The system should include the following windows: the dashboard page, the time and attendance page, the running projects page, the project profile page, the teams page, and the invoice page.
- The project profile window should show project employee costs, tasks assigned to the project, and the designated project manager.
- The system should only enable one Project Manager to be assigned to each project.

- Every project should have a stopwatch to keep track of time.
- To monitor project progress, every project should have a Kanban Board View system.
- Creating task cards for each project must be possible.
- Task cards in a workflow should be able to shift from one task status to another by dragging and dropping.
- Each card in the system should display a popup with more information about the task when clicked.
- Project managers should be able to delegate work to specific team members by assigning cards to them.
- The Project Director should be able to form teams for a specific project, assign personnel to the teams, and choose a team leader.
- The team leader should be able to give the team members tasks to perform.
- Users should be able to create checklists to show mini-tasks and progress toward completion.
- The system should include deadline notifications and alarms.
- Automated email notifications should be included in the system.
- Team progress and team performance should be shown in the Dashboard view.
- The Calendar view should provide a monthly overview of project management and an easy way to adjust start and due dates by dragging and dropping.
- Users should be able to add files, notes, and comments to cards.
- A discussion feature should be included in the system so that users can communicate with one another, tag other users in comments, and organize their work.
- Project managers should be able to keep track of how much time each team member spends on projects and tasks.
- If an employee is inactive for a long period of time, the system should be able to notify them.
- The system needs to have an automatic payroll calculation feature to calculate the wages that the employees are due in a fair and precise manner.
- Invoicing and payroll operations should be automated in the system.

3.2.5 Introduction to Non-Functional Requirements

In software engineering, non-functional requirements are defined as those that specify how the system operates. This non-functional need is concerned with the process the system uses to carry out a particular function. These are essentially the quality requirements outlined in the project contract that the system must meet. Nonfunctional requirements outline how the system should function rather than being tied to system functionality. Depending on the project, different aspects may be given different levels of priority or implementation.

Non-functional requirements can appear to be less significant than functional requirements at first look, but both are necessary for a sound system. While non-functional requirements do not affect the system's functionality, they do affect how well it will function. So, the main focus of non-functional requirements is system usability. Users may lose interest in the system and look elsewhere if non-functional requirements are not addressed. In order for a system to function well, at least certain non-functional needs must be met. It is necessary to have both a basic understanding of the system's functionality and a good understanding of the environment in which the system will operate in order to identify non-functional requirements.

The following are some non-functional requirements that must be met in a system.

- Speed:- The system's ability to do tasks quickly.
- Availability:- How long is the system operational? For instance, does it run all year round, or just during the day?
- Capacity:- How much the system can manage before reaching its limits.
- Reliability:- The system's dependability
- Usability:- How simple it is for the consumer or end user to use the system.

3.2.6 Examples of Non-Functional Requirements

Examples of non-functional requirements make it easier to comprehend what they are.

- The duration that it took for a given page to load. For instance, with 5,000 users logged in simultaneously, the website pages should load in 3 seconds.
- The required processing time for some requests. For instance, 20 million users shouldn't cause the system's performance to suffer.
- The required level of system availability.
- When maintenance will be conducted, and which tasks can be done at certain times.
- The maximum number of concurrent users the system can support.
- The system should be accessible for people with disabilities.
- The system should be accessible for people who are color blind.
- Passwords should not be viewable at the point of entry.
- The access permissions for the system data may only be changed by the system data administrator.
- Online payments should be available at any time.
- The system should achieve 100% uptime.
- Within 24 hours after the commencement of the installation, the new installation of the system must be ready for use for the first time.

3.2.7 Non-Functional Requirements for the E-Solutions Private Limited

- The confirmation screen should load in no more than two seconds after pressing the submit button.
- The home page should load in 3 seconds or less.
- Sensitive data should be securely stored.
- Up to 10TB of data should be able to be stored on the system.
- The system should only use up to 8GB of RAM.
- Customer information should be protected using level 4 encryption.
- The most recent versions of Windows and iOS should support the system.
- When switching from one OS to another, the system shouldn't cause any issues, hence the system should be portable.
- For 99.999% of the time, the system should be accessible to users.
- In web-based applications, the system should react to user activity in under 1 second.
- Every unsuccessful attempt by a user to access data should be documented on an audit trail.
- The system should be able to accommodate 1000 users without experiencing performance issues.
- The system should recover from any operational disturbance in less than 4 hours.
- The automated email systems should be out of operation for about 3 hours if they do.
- If a major event occurs on the system, we must take steps to restore full system functionality within 2 days.
- The employee's personal information should only be seen by users with the role of "Project Director."
- When editing a project profile, the other windows should remain open.
- The system should preserve data integrity by making regular backups of all database updates for each record transaction.
- The system interface should be simple to use and friendly to users.
- All modifications to the security settings should be logged in by the system.

- All audit log entries should be connected to the user who carried out the action.
- The system should be able to offer a method to track and delete test transactions.
- A responsive user interface must be offered by the system. (from smartphones to large displays)
- The system must allow for automatic installation and updating procedures.
- A method for exporting all data to a machine-readable format must be provided by the system (XML or JSON)
- In the event of a system breakdown the system should be able to automatically switch to a redundant environment.
- The system should be able to give on-screen contextual prompts that instruct users on how to use the software.
- Standard backup/recovery tools such as Veeam, and Zerto should be compatible with the system.

3.3 Requirements gathering techniques

3.3.1 Review of Procedural Forms

A good way to learn about present and potential systems is to speak with people who regularly use or are interested in a system. A more direct technique to see how an existing system functions is to observe current system users. There are limitations to both interviewing and observing. In order to learn more about current systems and the organizations they support, methods for determining system requirements can be improved by reviewing through the system and organizational documentation.

The study of procedural forms will provide us with a solid starting point for obtaining project requirements. We will ensure that the project gets off to a good start by learning about the company's current state and what has previously been valued. It could be an expensive and time-consuming operation. It will take a lot of man-hours to go through the mountains of paperwork. When it comes time to prepare surveys or conduct interviews in the future, the significant investment we made early in the process should pay off well.

We can find the following information by reviewing Procedural Forms.

- We can uncover flaws in existing systems. For example, missing information or repetitive steps.
- If particular information or information processing were available, we could identify ways to meet new demands. For example, sales analysis depends on customer type.
- We can uncover organizational orientation that will influence the requirements of information systems. For example, attempting to connect consumers and suppliers more closely to the firm.
- We can locate the titles and names of key individuals who are interested in existing systems. For example, the name of a sales manager who directed a study of significant customers' purchasing habits.

- We can identify organizational or individual values that can assist in determining priorities for various capabilities sought by different users. For example, retaining market share even if it entails decreasing short-term profitability.
- We can identify unusual information-processing scenarios that cannot be discovered by any other needs determination technique. As an example, special handling may be required for a few high-volume customers that require the adoption of specialized customer ordering procedures.
- We can determine why existing systems are built the way they are, which can suggest improvements that were previously absent from software but may now be viable and desired. For instance, information on a customer's purchases of rival items, which was not available at the time the existing system was built but is now available from a number of sources.

A written software work procedure is one kind of helpful document. This procedure explains how a specific job or task is carried out, along with the data and information that are used and produced while the job is being carried out. Business forms are a second sort of document that systems analysts find helpful. Forms are used for all kinds of commercial activities, including documenting orders, confirming the payment of bills, and listing the items that have been dispatched. Since they clearly state what data flow in or out of a system, forms are crucial for understanding it. A report produced by current systems is the third category of useful document. We may move backward from the information on the report to the data that must have been required to make it because reports are the main output for some types of systems.

The data gathered through interviews, surveys, and observations can be supplemented by information from document analysis. Organizational documentation, for instance, may assist in making sense of some of the interviewee's replies if any of them are confusing. Understanding why a person performs particular things while being observed may also be helped by reviewing previous documents. Documents may be out-of-date while they are being analyzed, thus the analyst must make sure they are valid or not. Depending on the company and the system, reviewing documents might also take a long time.

3.3.2 Interviews

One of the simplest yet most effective methods for acquiring requirements is conducting interviews. Since an interview is essentially just two people having a somewhat more structured conversation, anyone can learn how to conduct them. Finding smart questions to ask is frequently the most difficult part. Understanding the demands of our stakeholders is obviously necessary for requirements management success. Clients, users, decision-makers, and anybody else who has an interest in the future system we are creating are all potential stakeholders. In order to develop a system that is well adapted to the needs of stakeholders, requirements analysts should employ interviews to gain a deeper understanding of their problems.

The first goal of the interviewer should be to learn the fundamentals about the stakeholder and his or her company. After that, ask one or two of the colleagues to evaluate the interview questions the interviewer intends to ask. The interviewer must keep in mind that the objective is to obtain comprehensive and consistent input for the design of the system in order to prevent the interview script from becoming an obstacle rather than a help. After that, if any follow-up questions arise throughout the interview, they must be asked directly. The interview form should be created in a way that makes it simple to record the answers or write them down or enter them into a computer. The interviewer must keep in mind that these strategies aid in keeping the conversation's attention on its subject matter rather than the process of recording it.

Advantages →

- Interviews facilitate simple speech correction. In an interview, any misunderstanding or error can be simply corrected mostly due to the interviewer and interviewee's personal presence before the interview board.
- Interviews contribute to the development of a relationship. An interview can be used to establish a relationship between the interviewer and the interviewee (client). It improves the parties' understanding and cooperation with one another.

- The selection of the right requirements is assisted by interviews. Interviews are a good way to find all the requirements and expectations of the client because they reveal a lot about the client's needs.
- Interviews assist in gathering primary data. When necessary, interviews can assist in gathering new, innovative, and fresh information.
- Interviews assist in the collection of sufficient data. The interviewing technique can be used to get enough data because the interviewer is free to ask any question.
- Interviews can help us find the requirements more quickly. With the interview, communication can be established in a remarkably short amount of time.
- The knowledge of both the interviewer and the interviewee is increased during any interview. Their opinions and ideas are open to exchange.
- Interviews assist to identify the problem's root causes. Executives in the corporate world must deal with a variety of issues. Interviewing might be done to investigate or discover the true causes of the issue.

Disadvantages →

- The interview procedure is not complete. Only conducting interviews will not provide the necessary requirements. Compared to the interview, the written test is more significant.
- Since there is no actual evidence of what was mentioned during the interview, there may be some uncertainty in the future.
- For a successful interview, careful attention is necessary. The interviewer and the interviewee, however, can occasionally be seen to be paying less attention to one another. Because of this, accurate data cannot be gathered.
- When asked questions that have nothing to do with the field, the interviewee could feel let down. Because of this, crucial information could be missed.
- One of the main restrictions on the interview process is time. The interview method is time-consuming since it takes a lot of preparation, time to conduct interviews, and time to interpret the results.
- The interviewer's biases could potentially affect the interview process at any time.
- Interview techniques are typically expensive.

Structured Interviews →

A structured interview is one in which the interviewer arrives for the interview prepared with a prearranged list of questions. A client's expectations and needs are the main focus of this quantitative research technique. Closed-ended questions are allowed in structured interviews, and the clients must adhere to certain rules. The uniform character of these interviews makes them ideal for gathering information with numerical values. Further study and analysis of the client's requirements are done using the information from these interviews. When gathering requirements from a large client base, a structured interview works well.

This type of interview has a set format, is rigid, more successful, and requires less interaction between the interviewer and client. These interviews are simple to repeat and quantify because the questions are fixed, which makes determining the validity easier. However, the lack of specific information about the customer in this type of interview makes it challenging to determine customer behavioural inputs.

Unstructured Interviews →

Unstructured interviews don't follow a predetermined format or set of questions. Based on the client's responses, the interviewer might adjust their questions and uncover more information. This type of interview uses an informal style and open-ended questions. It resembles a discussion between the interviewer and the customer more than anything else. Unstructured interviews are a type of qualitative study that provide the interviewer total flexibility to ask any questions they want.

There is no set order or format that the Interviewer must follow. However, the individual in charge must be well-versed in the topic at hand and possess the necessary expertise. These unplanned, inconsistent interviews provide opportunities for collecting data for exploratory research. Unstructured interviews depend on unpredictability and are characterized by a high degree of descriptiveness due to the lack of formalized sequencing. When it comes to providing a high-quality product, this type of interview is ideal. The discussion can go on for longer than normal in this situation because it is not time limited.

During an unstructured interview, characteristics like friendliness, maturity, and social skills are quite important.

Differences between Structured and Unstructured Interviews →

Table 3. 1 Differences between Structured and Unstructured Interviews

Structured Interviews	Unstructured Interviews
An interview that follows a specific format is called a structured interview.	In an unstructured interview, the interviewer tries to start a discussion with the client freely.
The interviewer has a specific questionnaire that they have prepared in advance.	There is no predetermined structure or pattern that permits the interviewer to mix informal conversation with interview questions.
These interviews have a directive aspect, which means that a step-by-step process is followed with the appropriate guidelines.	The interviewer is not required to follow any predetermined guidelines or leads because this type of interview is non-directive in nature.
When conducting a structured interview, the interviewer must adhere to a predetermined order of questions and a defined interview format.	An open-ended interview allows the interviewer to ask whatever they want.
Data that is quantitative is produced by structured interviews.	Data for qualitative research are gathered through unstructured interviews.

Although this data can be measured, it doesn't provide details about how the client behaved in a given circumstance or incident.	The interviewer can learn more about a person's situational awareness and acquire in-depth information.
Closed-ended interview questions are used in this format.	There are open-ended questions throughout this interview.
The questions in a structured interview are the same for all clients.	The questions in an unstructured interview vary from client to client.
For the purpose of acquiring pertinent information, structured interviews are combined with surveys and questionnaires.	Unstructured interviews are conducted using tools like audio recorders, cameras, and telephones.
A comparison can be made with the data gathered during a structured interview.	The data samples do not permit comparison because an unstructured interview is more likely to differ for each individual.

3.3.3 Questionnaires

Understanding the requirements of the project's diverse stakeholders might be difficult. Having a lot of input is good, but businesses must monitor costs while the project is still in its early phases. It is inexpensive to obtain data for a project's requirements via a survey or questionnaire. Additionally, questionnaires do away with limitations like a geographically scattered stakeholder base or time limits. Additionally, the accuracy of the responses' recording is assured to us. A survey or questionnaire, whether conducted online or on paper, eliminates any potential for tone interpretation or information that might be misunderstood.

When thinking about using a questionnaire, the main choice to make is whether to collect data using a hard or soft copy. Since there is no need for printing or additional data entry, an online, soft copy provides obvious financial advantages. If some stakeholders do not have stable computers and internet connections, a paper copy might improve accessibility.

Generally speaking, surveys provide us with discrete chunks of data. As a result, it is simple to analyze the data and present it in the format of graphs, charts, and visualizations. There will be few opportunities for interpretation in the datasets we have. During a conversation, for instance, there isn't any body language to understand. Utilizing a survey also allows us to include all project participants without having to create representative samples, which is another benefit. As a result of feeling more appreciated, stakeholders may become more involved.

Advantages →

- Numerous stakeholders or other information sources may be reached with a questionnaire.
- In a short length of time, a questionnaire is a great instrument for gathering a lot of targeted data.
- After using the interviewing technique, the questionnaire method can be effective in validating presumptions.
- An excellent way for gathering statistical preference data is the use of questionnaires.

- A questionnaire only requires minimal planning.
- A questionnaire involves little time and effort from the stakeholders.

Disadvantages →

- Particularly for big questionnaires, the response rate is frequently low.
- Typically, responses are restricted to the scope of the questions being asked.
- To create well-made surveys, qualified and experienced professionals are needed.
- Development times may be lengthy.
- Additional analysis is necessary to resolve informational conflicts and inconsistencies from stakeholders.

Open-Ended Questionnaires →

A survey question that allows respondents to provide their own answers is known as an open-ended question. In order to completely express themselves when responding to questions, respondents must use open-ended questions. In contrast to closed-ended questions, open-ended questions have a wider emphasis and enable responders to give in-depth responses. They also let the respondent speak freely, giving them a deeper understanding of their views, hopes, and experiences.

Close-Ended Questionnaires →

An open-ended question can have more than one possible answer, whereas a closed-ended question only allows for a limited number of predefined answers. In other words, the reply must select an answer from the few possible choices given in the question. The response is not permitted to offer alternatives to these choices. In quantitative research, closed-ended questions are frequently used to collect statistical data from participants. Additionally, the number of possible answers in closed-ended questions is not constant. Depending on the survey's goals, there may be 2 or more choices listed.

Table 3. 2 Open-Ended Questionnaires vs Close-Ended Questionnaires

Open-ended Questions	Close-ended Questions
Questions with lengthy responses that should be answered.	Short answers should be provided to questions.
A lot of the time, answers are explicative and detailed.	The responses are brief and accurate.
Collecting qualitative data is the aim of open-ended questions.	Collecting quantitative data is the aim of closed-ended inquiries.
Essay questions are a typical example of an open-ended question. Usually, the respondent must give more than a single word in response to these queries.	Multiple choice questions are a common form of closed-ended question. Respondents are given a variety of answer choices in the multiple-choice questions, which can be formatted as checkboxes.
Questions usually start with the words "how," "why," "explain," or "describe."	Words like "is" and "would you do" are used to start questions.
Common techniques for gathering data include observation, interviews, and focus groups.	Quantitative data gathering techniques, such as surveys and questionnaires, are used to pose questions.
Open-ended inquiries necessitate a longer time for responses from responders.	Closed-ended questions, on the other hand, can be answered quickly because they are simple and to the point.

3.3.4 Observations

Observation should be used in the early stages of gathering project requirements. We can learn insights from seeing stakeholders and end users in action that are not possible from talking alone. When modifying, adapting, or improving an existing product or procedure as part of our project, observation will be useful. Despite how beneficial it might be, not every project is a suitable fit for direct observation. The following situations are ones when making observations will be helpful.

- When the goal is to enhance a current product or procedure, observations can be useful.
- When stakeholders find it difficult to properly express their requirements in words, observations might be useful.
- In repeating processes, observations might be useful.
- When the accuracy of other data may be in question, observations can be useful.

Participant Observation →

Participant observation entails being a part of the group being observed while also observing the scenario, events, or activities from within. Participant observation entails being a member of the group being watched and viewing the circumstances, events, or activities from within. The observer participates in all of the group's activities and freely engages with the other participants.

So, instead of studying their behaviour or other activities as an outsider but as a part of the observed group, the observer adopts the observed group's way of life or his own. The observer must therefore continue to be a part of the group he is trying to examine in order to conduct this type of observation. It is crucial for participant observation to be successful that the subjects being observed have no doubts regarding the researcher's intentions.

Advantages:

- Participant observation allows for the study of the respondent's natural behavior.
- The observer establishes a strong relationship with the respondents during participant observation. The relationship between an observer and the group members might be very close.
- Participant observation allows the observer to conduct a thorough and comprehensive analysis of the group and get knowledge into its true nature.
- During participant observation, the observer can more fully comprehend the respondents' emotions than an outsider.
- The main benefit of participant observation is that it gives the observer a chance to talk with the group about their varied activities.

Disadvantages:

- The observer may lose his objectivity if he joins the group and actively participates in it.
- The unbiased analysis of the observer is destroyed by his sentimental and emotional ties to the group.
- The researcher establishes a unique position for himself in the group as a result of his close association and emotional involvement with the group members.
- Many important occurrences seem to be of little or no consequence to the participant observer because of their extensive familiarity. As a result, he overlooks a lot of crucial details. However, because it is unfamiliar to him, a stranger is very attentive to even the smallest detail.
- The observer limits himself to a specific group when doing participant observation. As a result, his experience becomes incredibly broad yet very narrow in scope.

Non-Participant Observation →

Non-participant observation is when the observer observes the group inactively from a distance while not taking part in the activity. The observer in this case does not attempt to persuade individuals or engage in group activities. Pure non-participant observation is really challenging, though. Really, it is impossible to image a relationship in which the researcher is constantly there but never takes part.

Some of the routine activities are actively participated in by the observer, while others are observed passively from a distance. Because of this, many sociologists only consider non-participant observations to be quasi-participant observations. It is simpler for the observer to play both roles than to fully disguise himself.

Advantages:

- The neutrality or objectivity can be upheld during non-participant observation. In this kind of observation, the observer provides an unbiased and detached evaluation of the group.
- When doing non-participant observation, the researcher acts impartially. As a result, each group member accords with him a unique status and helps him with his research.
- When doing non-participant observation, the researcher acts impartially. As a result, each group member accords him with a unique status and helps him with his research.
- The majority of the time, people are not embarrassed to reveal their faults, secrets, or casual habits to an unknown person. However, they always find it difficult to tell a known individual these details.
- The researcher does not even overlook a small detail during non-participant observation. The researcher carefully weighs the benefits and drawbacks of each phenomenon under observation.
- The researcher always keeps an unbiased stance when doing non-participant observation.

Disadvantages:

- The observer in non-participant observation lacks clarity regarding specific events or activities. By posing several questions to the group members, the observer cannot eliminate his doubts.
- Only the events that are taking place in front of the observer can be seen by him.
- Members of a group develop mistrust for anyone who observes them unbiasedly. The group members feel nervous while speaking to a stranger or outsider, and they only give formal information in an unnatural manner.
- When members of a group are aware that an outsider is judging their behavior, they always feel very uncomfortable.

3.3.5 Focus Groups

The focus group resembles both a brainstorming session and an interview. In order to get input on a product, service, or concept, focus groups are interactive, facilitated discussions with a carefully chosen group of people. Facilitators assemble customers, clients, or consumers and start conversations in focus groups to get feedback and insights. Focus groups can be an effective method of gathering information about the particular insights facilitator wants including learning what a consumer or end-user wants from a product.

For continued feedback, a focus group might be held before a launch or afterwards. Focus groups include external stakeholders, but brainstorming sessions often involve internal ones. This is the main distinction between the two styles of meetings. The Scribe will record the discussion as it is being facilitated by the facilitator, and a video recording or audio recording may also be made for subsequent analysis and review. The focus group conversations are often facilitated by a moderator or facilitator by way of questions. They take notes on the responses, follow up with additional questions, and solicit comments or feedback from other participants. Normally, focus groups last between one and two hours.

Advantages:

- In comparison to very specific and structured quantitative research studies, focus groups offer deeper and broader insights into the brand and associated subjects.
- Focus groups offer another significant benefit in that they enable small-group interaction research as compared to individual responses on surveys. Moderators gain information from responses through this interaction, and they also get to monitor the group and notice emotional reactions, contradictions, tension, rage, annoyance, and other feelings that are not revealed through a formal questionnaire.
- Focus groups typically don't require any technology and are completed in a few hours on a single day. Hence, the expense of using this method is minimal.
- The moderator can also see the dynamics of the focus group participants as they exchange opinions with one another, which is beneficial.
- In businesses that are changing quickly, saving time during the research phase of product development is crucial, especially if a thorough focus group enables to acceleration of the launch of a product.

Disadvantages:

- Focus group participants don't always express their genuine ideas, which is one disadvantage. Some people are hesitant to speak to huge crowds or strangers because they are naturally shy.
- The price may vary depending on the cost of supplies, renting a space, working with a market research company, and paying respondents for their time. And analyzing the focus group results to produce a report or presentation might take a lot of time.
- One or two dominating participants in a focus group can distort the results and make other participants hesitant to share their thoughts.
- If the focus group is held at a remote location, transportation challenges can occur. Companies should either choose an alternative location or arrange transportation for attendees if the location cannot be reached by public transportation.
- Focus groups are significantly more expensive to conduct than surveys and questionnaires.

3.3.6 Discovery Prototyping

Another modern requirement collecting technique is prototyping. Prototyping is an iterative process that mainly relies on user input. A working model of the anticipated product is provided as a prototype, allowing for early feedback on requirements before the actual product is built. Therefore, it can be thought of as a test version of the final product.

It is impossible to use prototyping as a stand-alone method of requirement collecting because it depends on client engagement. Prior to starting the prototyping process, the analyst must conduct interviews or carry out another type of requirement collecting. Since the client is so actively involved in the process, prototyping is particularly effective in defining requirements. While the analyst enters requirements into a functioning system, the client will continue to be seated next to them, giving them instructions. This will enable the user to view the results of their demand immediately.

The client may alter some of his or her requirements at that time. It's possible that they will realize that what they offered was not what they had in mind. At this point, the client can go back and change any incorrect information that appears on the form. In order to reach a comprehensive set of specs, the user and analyst will go through additional iterations. The final prototype will serve as the basis for creating the real system.

Advantages:

- Prototyping helps to increase the quality of specifications and requirements offered to clients.
- For years to come, good prototyping can guarantee product quality and cost savings.
- The majority of clients want to be involved in all the minor elements of their project. Customers can view and engage with a working model of their project through prototyping, which is a participation-required process.
- Customers can adjust model specifications, request project revisions, and provide fast feedback when using prototypes.

- Most significantly, prototyping helps prevent misinterpretations and misunderstandings during the development process.
- Nothing pleases clients more than projects that are completed within budget. The quality of the requirements and specifications delivered to clients is improved via prototyping.

Disadvantages:

- Developers may become sidetracked from conducting a thorough project analysis if they concentrate too much on a limited prototype.
- Customers mistaking a prototype for the finished product is the worst-case scenario for any prototype. Customers might not understand that a rough prototype only requires finishing or polishing after seeing it.
- Customers may mistakenly believe that the prototype truly represents the performance of the finished product. Customers could develop fond of prototype features that won't be included in the finished product.
- Developers and clients must agree on the same project goals in order for a project to be successful. Conflicts between teams and missions may arise if clients demand that every feature from a prototype be incorporated in the finished product.
- By their very nature, prototypes are intended to be created rapidly. A sophisticated prototype that takes too long to construct might cause problems and cause time and money budgets to be exceeded.

3.3.7 Joint requirements planning (JRP)

A widely popular technique for gathering requirements is Joint Development Application (JAD). Through a systematic approach that include discussion groups with the aim of obtaining requirements and specifications, it brings system developers and users together in a productive and creative space. To put it briefly, the ultimate goal of JAD is to involve the client in the development process and create an end result that will more effectively suit the requirements of the customers.

Through a series of workshops, JAD gives clients complete control over the development of their projects and gives them the opportunity to contribute to the creation of their applications. JAD sessions are necessary when an organization requires some technical advice from technical expertise. Nothing is more efficient than setting up a JAD session when a technical and business team are attempting to work together to reach decisions about important issues.

We must keep in mind that the JAD team is the project's core when implementing the JAD process. Additionally, since they will be accountable for the overall success of a JAD session, we must carefully choose and include the stakeholders. To keep our JAD session on course, the following critical duties must be fulfilled.

- Executive sponsor:- This person, who typically comes from the customer's organization as the manager of the business division, is free to make all important project-related decisions.
- Facilitator:- This individual is crucial to the process because they are in charge of organizing, carrying out, and managing the session. The facilitator should be well-versed in the subject and have a lot of experience.
- Stakeholder:- The method's primary focus is on a stakeholder. JAD sessions are useless without their participation. JAD sessions enable stakeholders to participate actively in the project and receive the product they require.
- Scribe (Recorder):- The task of the scribe is to record each step of the JAD procedure.
- IT Representative:- Technical guidance is provided by the IT representative, who also assists the JAD team in creating logical models for a prototype.

Advantages:

- JAD makes it possible to create better, error-free software and handle issues more quickly.
- The organization and its clients work together to reduce all risks.
- JAD decreases project development costs and time requirements.
- The quality of the system is improved by clearly specified requirements.
- Close communication makes progress happen more quickly.
- JAD challenges the group to push one another to complete tasks more quickly and on schedule.

Disadvantages:

- It's challenging to retain focus and align goals when there are disagreements among team members.
- The amount of time needed to complete a JAD project will depend on its complexity.

3.4 Methodology and Model to develop a system for E-Solutions private Limited

3.4.1 Choosing Methodology for the proposed system

The truth is that there isn't a single methodology that works for all projects and organizations. The decision to use a method is mostly influenced by variables including the project's nature, size, and resources needed, among other things. So, before choosing which methodology is appropriate for the business, it is important to first examine the main distinctions between the two.

Table 3. 3 Traditional vs Agile Methodology

Traditional Methodology	Agile Methodology
Simple software development is done with it.	Complex software development uses it.
After the development phase is complete, testing is carried out according to this process.	The testing and development stages are carried out simultaneously in this methodology.
It is organized in a linear manner.	It is organized in an iterative manner.
Less security is offered.	It offers a great level of security.
In comparison to Agile development, there is less client participation.	In contrast to traditional software development, there is a lot of client interaction.
The development model is fixed.	It encourages a flexible model of development.

The software has less features available.	All the features that users require are offered by it.
Using this process leads to lower development costs.	The expense of employing this process for development is significant.
There are 5 main phases to it.	There are mainly 3 stages to it.
When it comes to customer communication, traditional software development methodologies are formal.	<p>The approaches used in agile software development are informal.</p> <p>This means, Customers are more willing to interact with companies that use agile software development methodologies.</p>
<p>To begin with, traditional methods for software development use a predictive strategy.</p> <p>Because the product is created through rigorous and precise planning, the software development processes are fully specified and predicted.</p> <p>This method for project development has defined times and costs, thus changes are not allowed.</p>	A flexible approach is adopted because software development methodologies are based on the concept of continuous design refinement and testing is based on team and customer response.
Software development companies utilize it less frequently.	It is commonly used by software development companies.
Most of the time beginners utilize it.	Most of the time experts use it.

According to the list of functional and non-functional requirements, the given project contains a lot of demands. So developing this program is not straightforward. As a result, there must be many testing and development phases as well as numerous client interactions. Additionally, since this project is complicated, the methodology we utilize should be flexible. The project's outcome must be perfect despite the expense. The **Agile Methodology** is the optimal methodology to employ for developing the system, based on these factors.

Software like "E-Solution Projects" is difficult to create since there are several clients' needs that are frequently challenging to meet. The ideal methodology to use when working on large projects is the agile methodology. Unlike simple projects, which only have one stage, complicated projects might have multiple interrelated phases and multiple stages that each depend on one another. For such massive and complicated projects, agile methodologies are preferred.

For projects like the "E-Solution Projects" software, numerous prototypes must be created, and numerous meetings must be held to acquire further customer requirements. The highest priority in Agile is customer engagement. As feedback is generated and considered after each iteration, the client is given high attention in its frameworks. Among project management methodologies that happen to require more customer interaction, Agile is unquestionably the best.

As stated earlier, there must be multiple customer interactions in order to obtain additional requirements and carry out modifications. It is better to utilize Agile in this kind of situation since it breaks a project up into sections (called iterations), with each release being provided to the customer after every single iteration. The success of these iterations also makes it simple to predict if the project will be a success. So, this methodology is preferred for this project since it has a high acceptance rate for feedback and modification. **With this Agile methodology, better results can be produced within the entire project delivery time because of the process's high level of flexibility and continual opportunity for feedback.**

3.4.2 Choosing Model from selected methodology for the proposed system

Scrum and Extreme Programming (XP) are the 2 main models of the Agile methodology. It is difficult to choose between these two models a model that is appropriate for the project. So, before choosing which model is appropriate for the business, it is important to first examine the main distinctions between the two.

Table 3. 4 Scrum Vs XP

Scrum	XP
The Scrum framework refers to teamwork in iterations as "Sprint," which lasts between two and one months.	Only 1-2 weeks of teamwork are allowed in Extreme Programming (XP).
The timetable and rules of Scrum models cannot be changed.	The set timelines for Extreme Programming are flexible.
Scrum places a strong emphasis on self-organization.	Strong engineering techniques are emphasized by Extreme Programming.
The team decides how the product will be created in the Scrum framework.	The team must adhere to a rigid priority order or predetermined priority order when using extreme programming.
Software engineering techniques that developers should employ are not emphasized in Scrum.	Developers should apply programming methodologies, according to Extreme Programming (XP), to guarantee a superior result.

The product owner assigns tasks a priority, but the development team has the ability to adjust those priorities as needed in the future.	The customer prioritizes the tasks, and the development team is unable to alter these priorities.
The project has less customer participation.	The project involves the customer more.

Scrum and XP are both agile methodologies, hence they have many similarities. They both insist that work should be done gradually. Both strategies place a strong emphasis on the benefits of team autonomy, open communication, and giving top priority to tasks that add the most value. Both methods break down requirements into manageable chunks. When it comes to roles, there are also overlaps. For instance, the Product Owner in the scrum and the client in XP both provide similar functions.

There are significant distinctions between the 2, though. As I've already explained, scrum is a method that organizations and teams may use to manage projects of all types. Scrum, for example, doesn't specify which software engineering techniques developers must employ. On the other hand, extreme programming, as its name suggests, really cares about programming. This approach emphasizes the importance of the programming methods that programmers should employ to produce a high-quality product that is definitely appropriate for the E-Solution project.

Therefore, the **XP model** is the best to adopt since "E-Solution Projects" software is a complex program that requires a programming-skills-oriented approach to produce a high-end final result. In comparison to Scrum, the XP methodology is preferable because of its frequent iterations and high customer involvement levels.

In comparison to XP, Scrum is applied with a bit more rigidity. Changes cannot be made in Scrum after the sprint planning session. Once the features and project scope are determined upon during the meeting, nothing can be changed during the sprint. The product owner or team must wait until after the sprint to propose any internal modifications. Our

E-Solution project, however, requires a more flexible model to handle its complexity. Comparatively, XP offers more flexibility because it permits new additions and modifications even mid-sprint. Additionally, if an activity hasn't started when the update is made, we can replace it during a sprint.

XP is not for beginners and works best in smaller teams of fewer than ten people. We must be able to work rapidly and make modifications. It is better to use XP for this project as we have a small team with experience in software development. Additionally, as was already indicated, this project has a strong requirement for client orientation. The ideal option is to adopt XP because it incorporates customer requirements into the development process and even depends on them for testing and approval. For this project, the main benefit of XP is that the entire process is transparent and accountable. The developers make concrete promises about what they will do and show specific results.

Since this is a significant project, the client won't be pleased if the working software isn't deployed right away. But this XP model produces functional software more quickly. All bugs are found through regular testing during the development phase. Furthermore, using XP makes customer-approved validation tests to evaluate the successful completion of a coding block guarantees that only what the customer wants is implemented. **So, with this XP model, better results can be produced within the entire project delivery time because of the process's high level of flexibility and continual opportunity for feedback.**

3.5 Conclusion to Activity 3

In system development, numerous types of requirements can be used, including Business Requirements, User Requirements, Domain Requirements, and System Requirements. There are two sorts of system requirements: functional requirements and nonfunctional requirements. Non-functional requirements are a set of specifications that describe the system's operation while functional requirements are the desired operations of a program. There are various approaches for gathering requirements. Interviews, Questionnaires, Observations, Focus Groups, Discovery Prototyping, and Joint Requirements Planning are among them (JRP). And, in my opinion, the ideal methodology for the particular project is Agile Methodology, and the best model to apply is XP (Extreme Programming).

Activity 4

4.1 Entity Relationship Diagram (ERD)

4.1.1 Introduction to Entity Relationship Diagram

Entity Relationship Diagram (ERD) is a diagram that shows the relationship between entity sets contained in a database. In other words, ER diagrams help in the explanation of database logical structure. Entities, attributes, and relationships are the three main components that ER diagrams are built on.

Rectangles are used to represent entities, ovals are used to define attributes, and diamond shapes are used to show relationships in ER Diagrams. An ER diagram appears to be quite similar to a flowchart at first glance. The ER Diagram, on the other hand, contains many specific symbols, and the meanings of these symbols distinguish this model. The entity framework infrastructure is represented by the ER Diagram.

Entity Relationship Model (ER Model) is a high-level conceptual data model diagram. The ER model aids in the methodical analysis of data requirements in order to create a well-designed system. The ER Model is a representation of real-world things and their relationships. Before we implement our system, we should create an ER Model in a system.

ER Modelling assists us in doing a systematic analysis of data requirements in order to create a well-designed system. As a result, completing ER modelling before implementing our system is considered best practice.

4.1.2 Why use Entity Relationship Diagram (Benefits)

- The most significant advantage of an ERD is that it gives a visual representation of the design.
- An ERD explains the important elements of a specific database and how they relate to one another.
- ERD is simple to design and straightforward to understand. In essence, the representatives can be shown this design to get their approval and confirmation.
- The ERD model can be applied relatively flexibly because new relationships can be simply derived from the ones that currently exist.
- ERD supports in defining the concepts used in entity relationship modelling.
- These diagrams show how all of our tables should relate, as well as what fields will be on each table.
- This helps in the definition of entities, attributes, and relationships.
- ER diagrams may be converted into relational tables, allowing us to create systems quickly.
- System designers can utilize ER diagrams as a roadmap for implementing data in specific software applications.
- With the help of an ERD diagram, the system designer gains a better knowledge of the information that will be stored in the database.
- The ERD Diagram helps us to explain the system's logical structure to users.

4.1.3 ER Diagrams Symbols & Notations

The rectangle, oval, and diamond are the three primary symbols used to represent relationships between elements, entities, and attributes in the Entity Relationship Diagram. Some sub-components in the ER Diagram are based on the main elements. The ER Diagram is a visual representation of data that shows how different ERD Symbols and Notations are used to explain how data is related to one another.

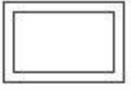
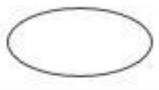
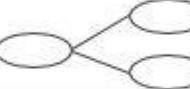
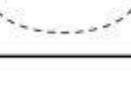
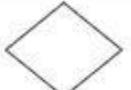
Entity Set	Strong Entity Set	
	Weak Entity Set	
Attributes	Simple Attribute	
	Composite Attribute	
	Single-valued Attribute	
	Multivalued Attribute	
	Derived Attribute	
	Null Attribute	
Relationship	Strong Relationship	
	Weak Relationship	

Figure 4. 1 ER Diagrams Symbols & Notations

4.1.4 Basic 3 components of the ER Diagram

There are 3 basic components in ER Diagram:

1. Entity
2. Attribute
3. Relationship

1) Entity

An ERD entity is a simply definable concept within a system. A system entity can be a place (e.g. School), a person/role (e.g. Student), an object (e.g. Invoice), an event (e.g. Transaction), or a concept (e.g. Profile). In ERD, the term "entity" is frequently used instead of "table," however the two terms are interchangeable.

When determining entities, we must consider them to be nouns. An entity in an ER model is represented by a rectangle. Inside the rectangle is written the Entity's name. Entities must have an attribute and a unique key as part of their characteristics. Every entity is built up of a set of 'attributes' that define it.

Ex: - * Peron – Doctor, Teacher, Student

* Place – School, Office, Building

* Object – Pen, Bus, Machine

* Event – Promotion, Sale, Renewal

* Concept – Account, Course, Profile

An entity set is a collection of entities that are similar in nature. It may contain entities with attribute values that are similar. It's an object for which we'd like to model and store data. Properties, often known as attributes, are used to represent entities.

Each attribute has its own set of values. For example, student entity may have properties such as name, age, and class. If we want to store data about students in School database,

the entity in the diagram is represented by rectangles. Inside the rectangle is written the Entity's name.



Figure 4. 2 Student Entity

2) Attribute

The term "attributes" is also used to refer to a column. A property or characteristic of the entity that holds it is called an attribute. An attribute has a name that identifies the property and a type that describes the type of attribute, such as varchar for a string and int for an integer. When creating an ERD for physical database development, it's critical to choose data types that are supported.

The attributes are represented by an oval shape. The attribute's name is written inside the oval shape, and a line connects it to its entity. An entity with some attributes is shown in the ER diagram below.

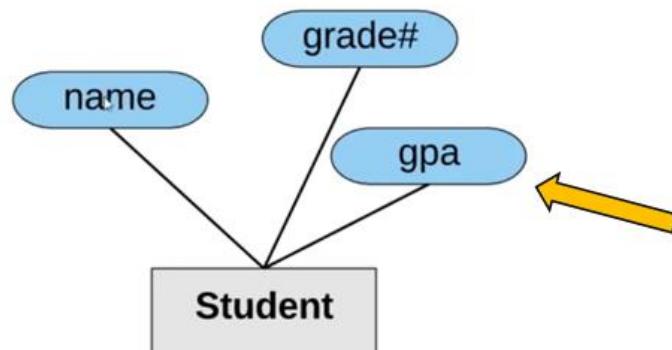


Figure 4. 3 Relevant attributes for student entity

3) Relationship

The term "relationship" refers to the relationship that exists between two entities. A relationship is nothing more than a link between two or more entities. A relationship between two entities shows that the two entities are linked in some way. For example, a student might take in a class. As a result, the entities Student and Course are linked, and a relationship is shown as a connector joining them.

The relationship type is represented as a diamond in the ER diagram, with lines linking the items. For example, when we combine the Student Entity and the Class Entity, Student can take a class. Class can be taken by student.

4.1.5 Entity Relationship Diagram for E-Solutions Pvt Ltd

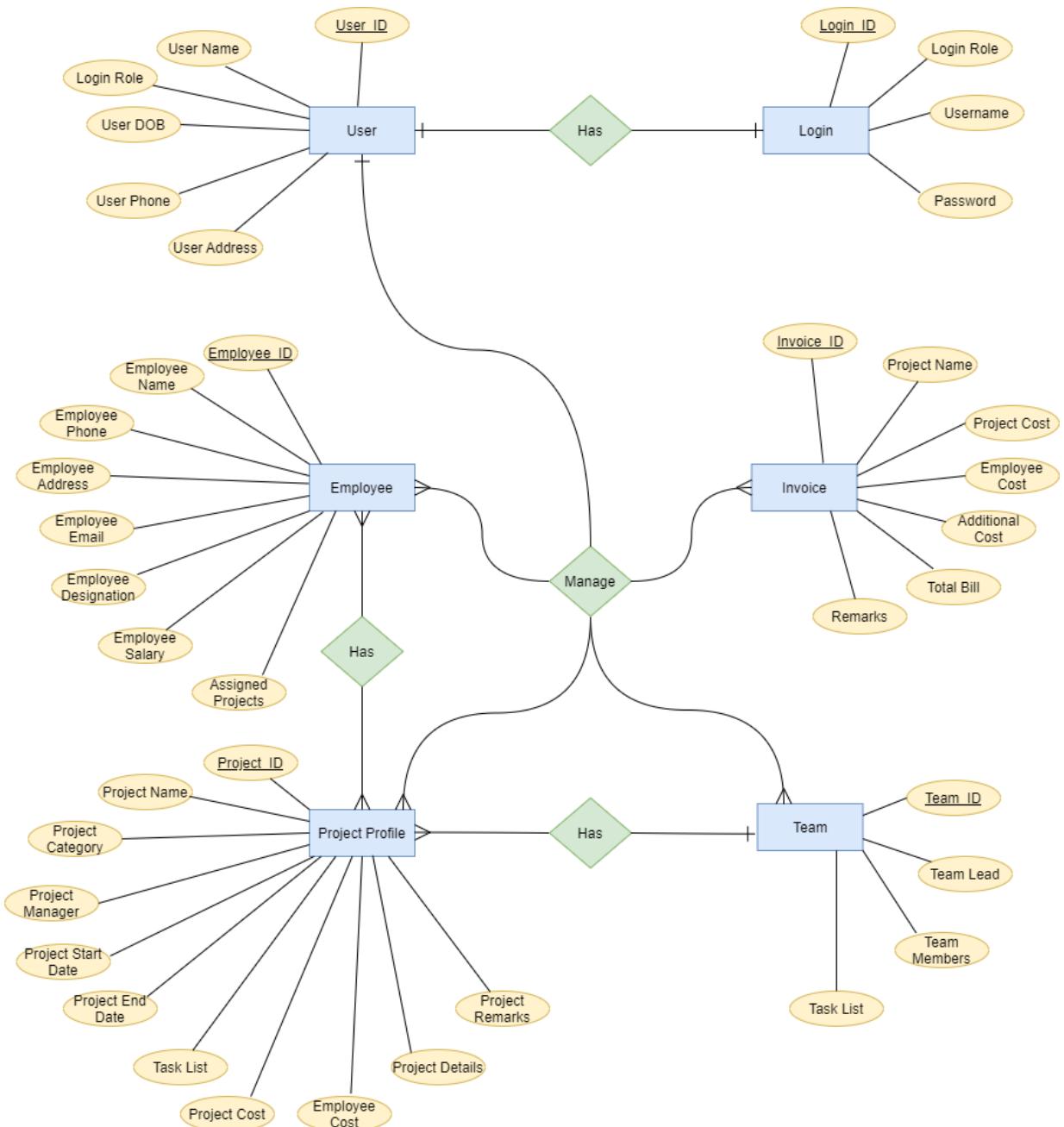


Figure 4. 4 Entity Relationship Diagram for E-Solutions Pvt Ltd

4.2 Logical Database Design

4.2.1 Introduction to Logical Database Design

The process of translating or mapping an application domain's conceptual schema into a schema for the data model supporting a specific DBMS, such as the relational or object-oriented data model, is known as logical database design. The mapping that resulted from attempting to accomplish two different sets of objectives can be understood.

- 1) Aim for representation.

maintaining the capacity to identify and capture all states that are valid for the conceptual schema

- 2) Goals for data management

addressing problems with querying the logical schema, cost of storage, and cost of constraint management. The majority of this entry's attention is given to relational databases' mapping of entity-relationship (ER) diagrams.

The entities and relationships represented in an ERD are focused on the requirements of the business. The necessity to satisfy database design has not yet been considered. The real design of a database is represented by a logical database. It is concerned with the transformation of a logical design into a schema-level design that will be turned into a relational database. When modelling a logical database, ERD is used as the foundation, and then primary keys, foreign keys, and constraints are defined. Relationships are sometimes resolved by adding new tables, such as a Linked table for a many-to-many relationship.

Because the ERD and logical database represent the business requirement and database schema, comparing them both helps to identify the differences, ensuring that the database follows the initial business's user requirements regardless of modifications.

4.2.2 Why use Logical Database Design (Benefits)

- Programming is not required for data selection or retrieval.
- Standard user interfaces are simple to use, and you can verify that all user input has been provided.
- A simple selection screen is provided by this.
- The pre-generated selection screen can be changed to suit our requirements. It provides check routines to see if user input is accurate, full, and credible.
- This provides a good selection of data.
- For database accesses, this has central authorization checks.
- Improvements such as enhanced performance are instantly applicable to any report applications that use the logical database.
- Data retrieval requires less coding than it would with traditional internal tables.
- Check functions verify that user input is accurate, complete, and credible.
- Maintain the application logic-determined hierarchical data view while providing good read access performance.

4.2.3 Logical Database Design for E-Solutions Pvt Ltd

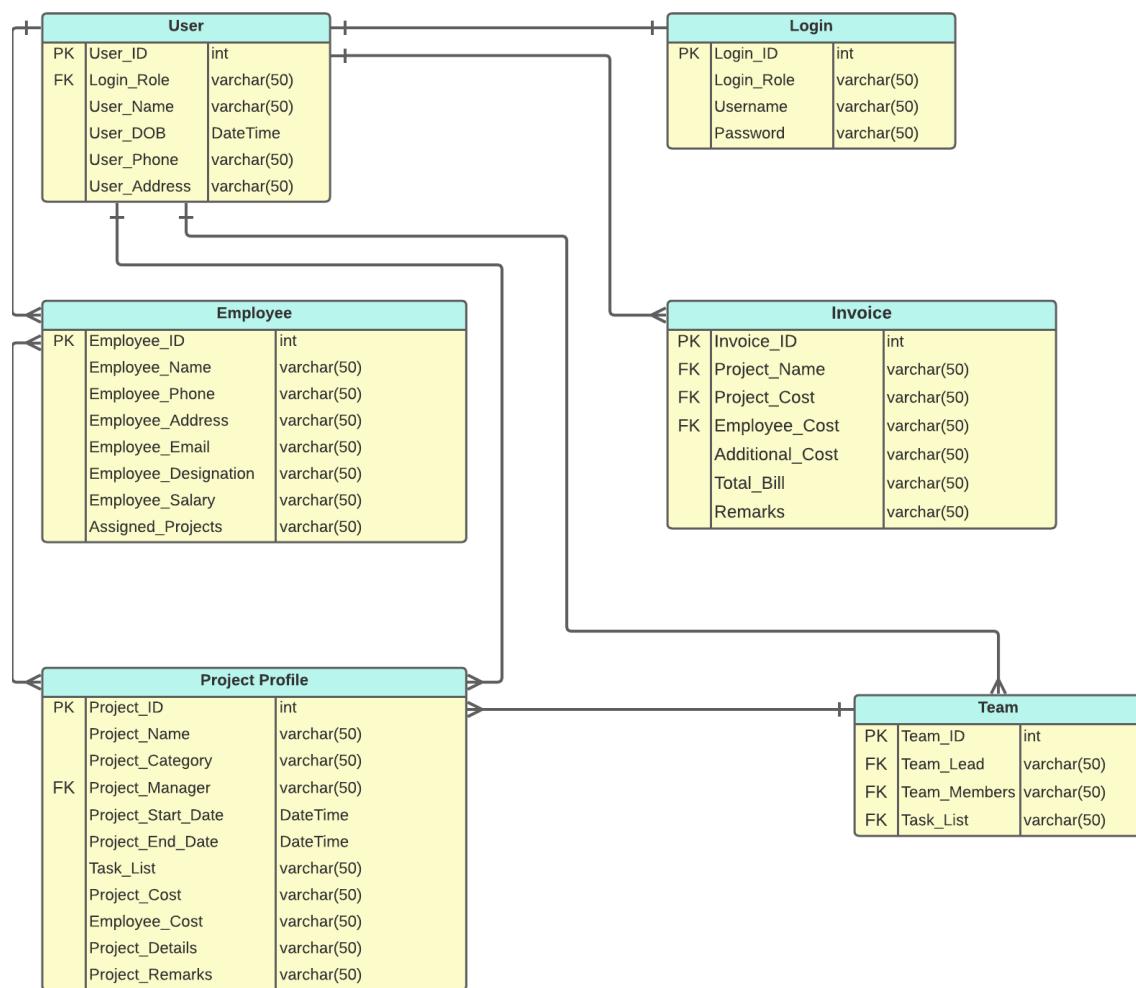


Figure 4. 5 Logical Database Design for E-Solutions Pvt Ltd

4.3 Class Diagram

4.3.1 Introduction to Class Diagram

The Unified Modeling Language (UML) can assist us in modeling systems in a variety of ways. The class diagram is one of the more common UML types. Class diagrams are a form of structural diagram that are used by system engineers to document software architecture. They specify what must be included in the system being modeled.

UML was created as a standard model for describing an object-oriented programming method. Class diagrams are the building blocks of UML because classes are the building blocks of objects. A class diagram's many components can represent the classes that will be programmed, the principal objects, or the interactions between classes and objects.

The shape of the class is a rectangle with three rows. The top row carries the class's name, the middle row contains the class's characteristics, and the bottom part expresses the methods or operations that the class may employ. Classes and subclasses are placed together to demonstrate the object's static relationship.

4.3.2 Why use Class Diagram (Benefits)

Class diagrams provide numerous advantages to any organization. UML class diagrams can be used to:

- Draw data models for information systems, no matter how simple or complex they are.
- Improve the understanding of an application's overall schematics.
- Visually articulate any system-specific requirements and convey that knowledge throughout the organization.
- Create thorough charts that indicate any specific code that must be designed and implemented in accordance with the defined structure.
- Give an implementation-independent description of types that are employed in a system and are later passed between its components.

4.3.3 Class Diagram for E-Solutions Pvt Ltd

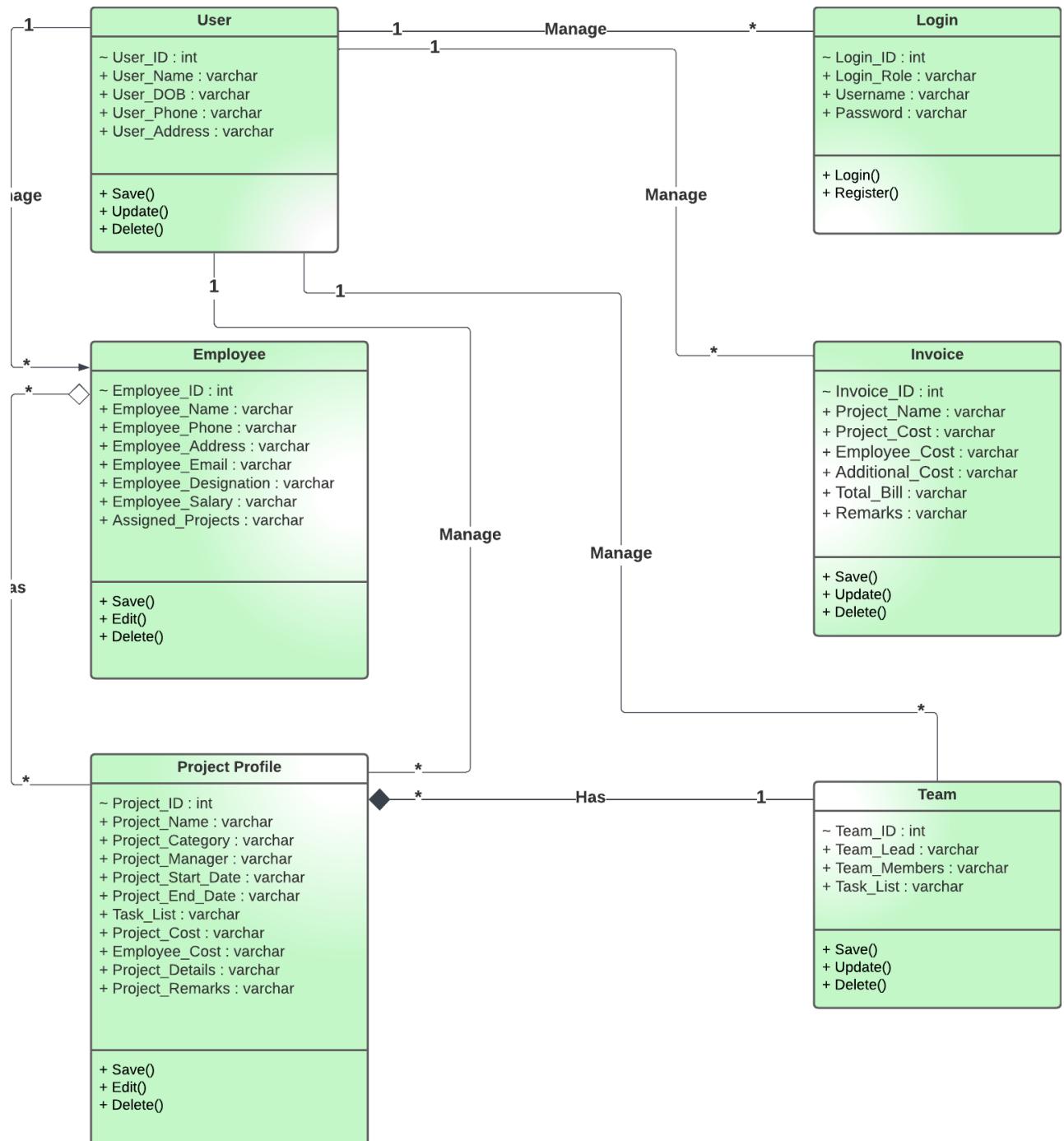


Figure 4. 6 Class Diagram for E-Solutions Pvt Ltd

4.4 Use Case Diagram

4.4.1 Introduction to Use Case Diagram

Use case diagrams are used to illustrate the dynamic nature of a system. As other 4 diagrams (activity, sequence, collaboration, and State chart) also serve the same objective, this definition is too general to adequately describe the objective. A use case diagram in the Unified Modeling Language (UML) can summarize the details of our system's users (also known as actors) and their interactions with the system. We employ a set of specific symbols and connectors to create one.

Use case diagrams are used to collect a system's requirements, taking into account both internal and external influences. Most of these needs are for the design. In order to prepare use cases and identify actors, a system must first be analyzed to determine all of its functionalities. Use case diagrams are created to show the outside perspective after the initial task has been completed.

A good use case diagram can assist our team in discussing and representing following things:

- Interactions between our system or application and individuals, organizations, or external systems.
- Goals that our system or application assists those entities (referred to as actors) in achieving.
- Our system's capabilities.

4.4.2 Use Case Diagram main components

UML Use Case diagrams work best when kept simple, just like flowcharts, wireframes, and many other visual aids used in the field. The most effective UML use case diagrams are simple ones. We'll use our fancy graphic design skills in other aspects of our project instead of displaying them here. The following three elements serve as the foundation for all of our diagrams.

- 1) Actors: Whatever is engaging with our system is represented by actors. An actor could be an individual, a company, a team, or something else entirely. Anything that exists outside of the system and engages with it in some way serves as an actor.
- 2) Systems: Our system is made up of a series of acts and exchanges done by actors, also referred to as a "scenario."
- 3) Goals: The results of an actor's interactions with the system are our goals. In certain circumstances, our system will result in a number of different outputs, yet in other circumstances, only one outcome will be achieved. Before moving on, we should think about adjusting our approach if we see any difficult or incomplete stages toward achieving a goal.

4.4.3 Why using the Use Case Diagram (Benefits)

- Use cases assist in capturing a system's functional requirements.
- The use cases can be tracked.
- The budgeting, planning, and validating work can be built upon use cases.
- Each iteration of the use case can go from a way of recording requirements to development instructions for programmers, to a test case, and lastly into user documentation.
- Alternative use case paths capture extra behaviour that could increase system robustness.
- Use cases have been shown to be simple for business users to comprehend, making them a great link between system developers and end users.

4.4.4 Use Case Diagram for E-Solutions Pvt Ltd

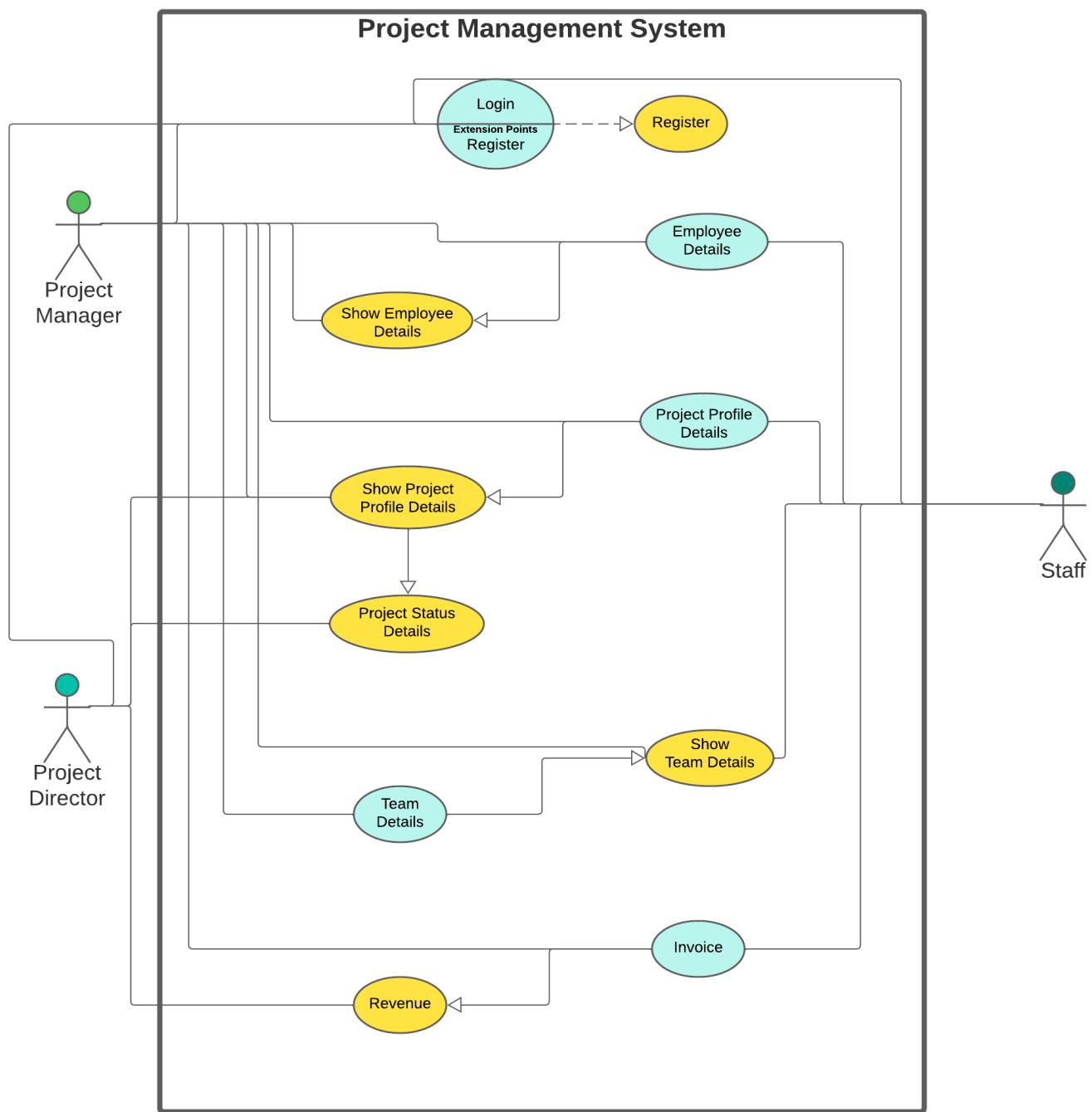


Figure 4. 7 Use Case Diagram for E-Solutions Pvt Ltd

4.5 Data Flow Diagram (DFD)

4.5.1 Introduction to Data Flow Diagram (DFD)

The information flow for any system or process is represented in a data flow diagram (DFD). It displays data inputs, outputs, storage locations, and routes between each destination using predefined symbols such rectangles, circles, and arrows as well as brief text labels. Data flow diagrams can be as basic as hand-drawn process overviews or more complex, multi-level DFDs that explore deeper and deeper into the handling of the data. Both a new system and an existing one can be analyzed using them.

A DFD works for both technical and nontechnical audiences, from developers to CEOs, and like all the finest diagrams and charts, it can frequently graphically "express" things that are difficult to explain in words. That is why DFDs have remained so popular over the years. The ability to visualize interactive, real-time, or database-oriented software or systems is less common today, despite the fact that they still function well for data flow software and systems.

4.5.2 Why use Logical Data Flow Diagram (Benefits)

- The system's boundaries can be described with the help of this.
- This helps to convey to consumers knowledge of the current system.
- This uses a simple graphical design that is simple to recognize.
- In-depth illustrations of the system's components may be provided in this method.
- This serves as a component of the system documentation file.
- DFDs are simpler for both technical and nontechnical audiences to understand.
- This validates the logic underlying the data flow within the system.

4.5.3 Data Flow Diagram Symbols & Notations

Three popular symbol systems have their designers' names as part of their names

- Yourdon and Coad
- Yourdon and DeMarco
- Gane and Sarson

Yourdon-Coad and Yourdon-DeMarco used circles for processes, whereas Gane and Sarson use rectangles with rounded corners, also known as lozenges. This is one of the fundamental differences between their symbols. Other symbol variations are also in use, so the main thing to remember is to be consistent and clear in the shapes and notations we use to collaborate and communicate with others.

The symbols represent the 4 parts of data flow diagrams according to DFD rules or guidelines for any format.

1. External entity: An external entity is a system that interacts with the diagrammed system by sending or receiving data from it. They serve as the locations where information enters or exits the system. They could be a company, an individual, a computer system, or a business system from the outside. They are also referred to as actors, sources, sinks, and terminators. Usually, they are shown along the diagram's edges.
2. Process: Any procedure that modifies data and generates an output is referred to as a process. It could carry out calculations, sort data logically, or control data flow according to business standards. The procedure is identified with a simple label, such as "Submit payment."
3. Data store: A data store is a file or archive that stores information for future purpose, such as a database table or a registration form. Simple labels, such "Orders," are given to each data storage.
4. Data flow: The path that data takes between external entities, processes, and data repositories is known as the data flow. Arrows are used to illustrate the interface with the other components, and they are often labeled with a brief description of the data, such as "Billing details."

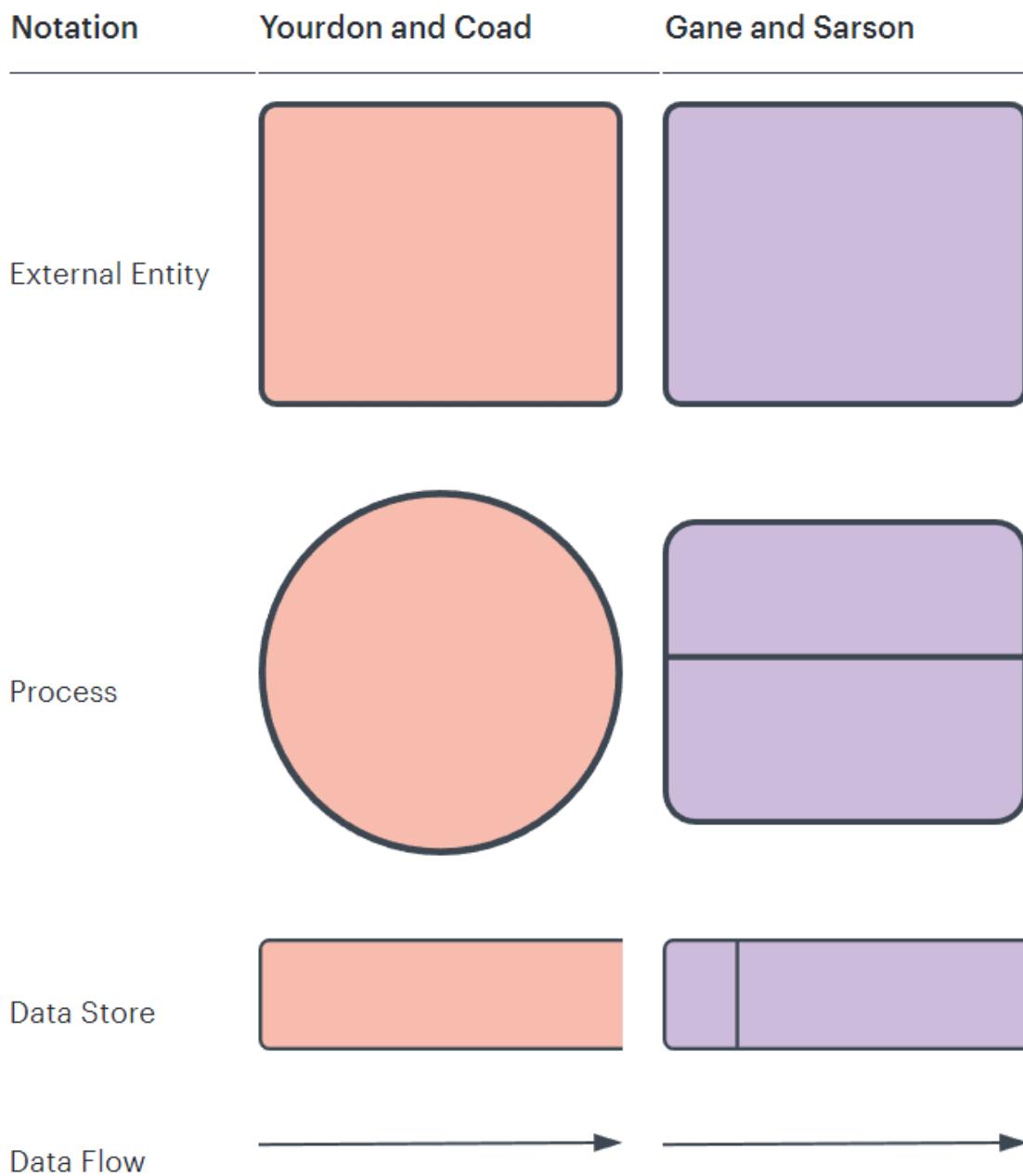


Figure 4. 8 Data flow diagram notations

4.5.4 Data Flow Diagram Levels

Context-Level Diagram (Level 0)

Context Diagram is another name for DFD Level 0. It provides a general summary of the entire system or process that is being studied or modeled. The system is displayed as a single, high-level process, together with its relationship to external entities, in what is intended to be an overview view. A large audience, involving stakeholders, business analysts, data analysts, and developers, should be able to understand it with ease.

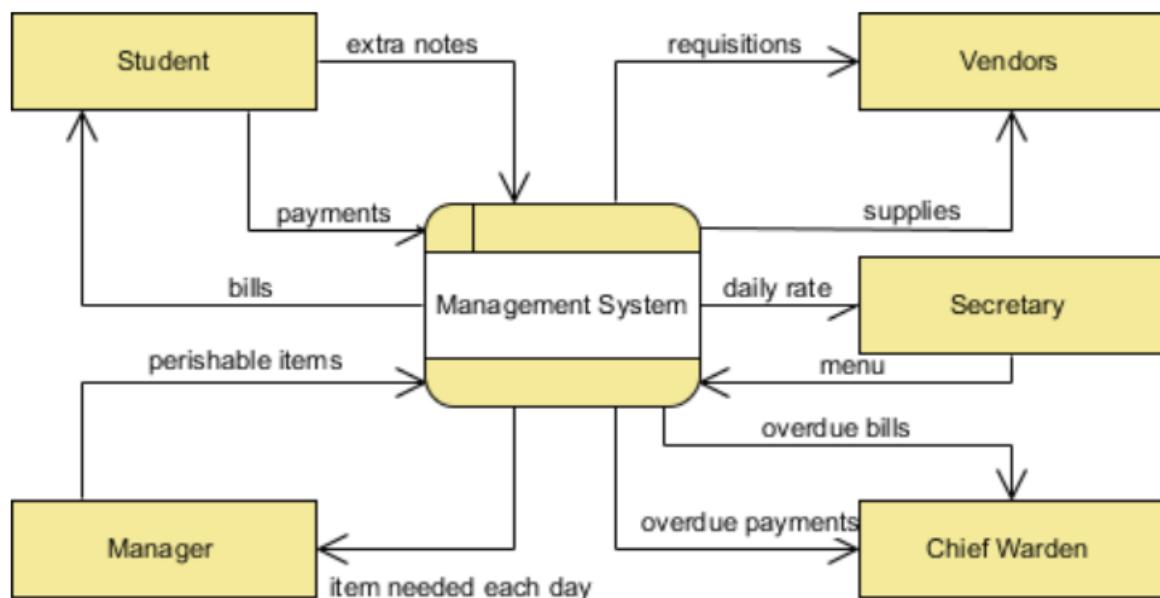


Figure 4. 9 Example Context-Level Diagram (Level 0)

Level 1 DFD

The Context Level Diagram is broken down more specifically in DFD Level 1. As we break down the high-level process of the Context Diagram into its sub - processes, this will emphasize the key objectives carried out by the system. There are several inputs and outputs from the processes to the external entities even though the level 1 DFD that follows just has three processes. In the diagram, there may end up being a few cross lines between them. We could use multiple views of the same external entity in the DFD to get around this issue.

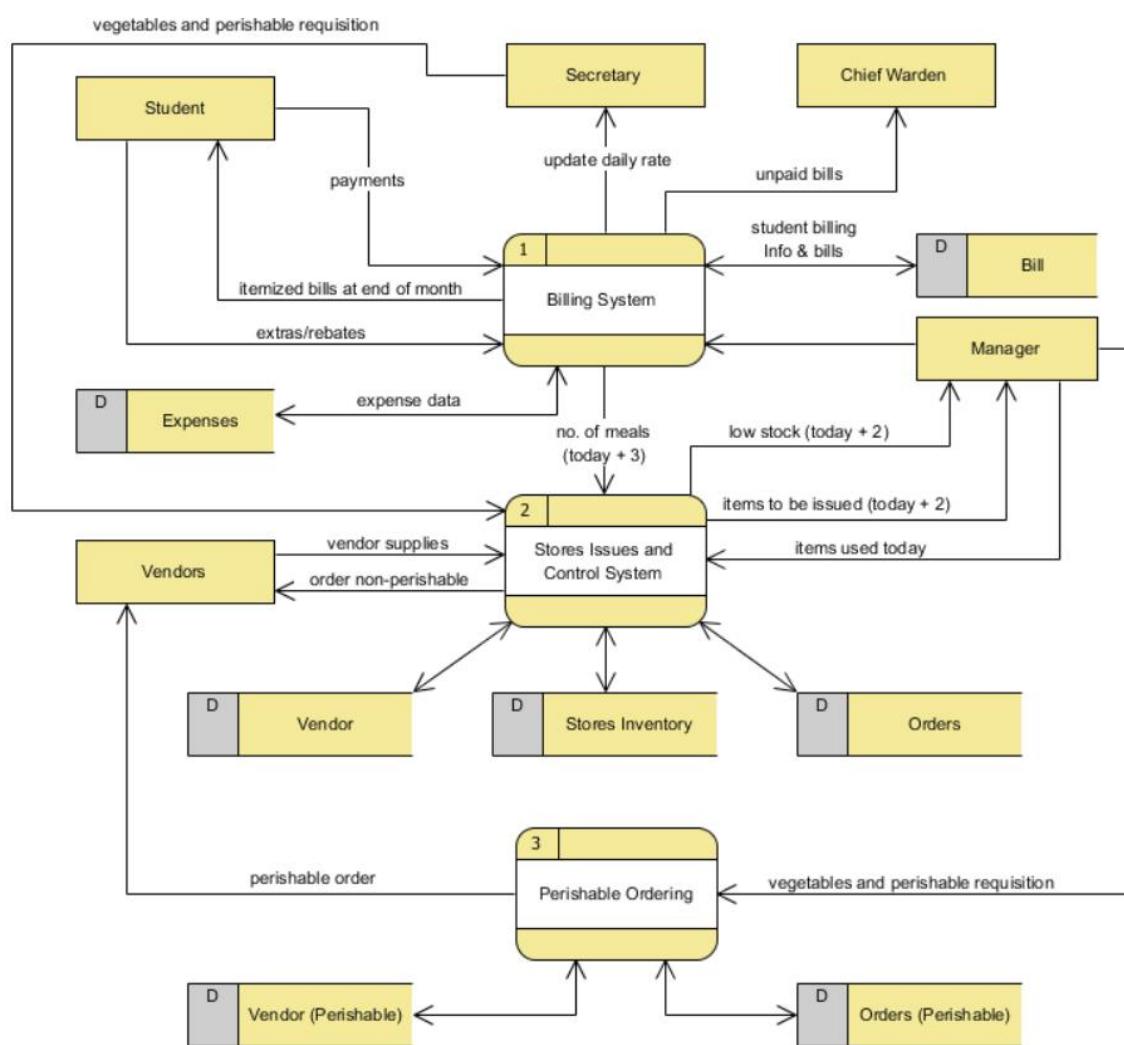


Figure 4. 10 Example Level 1 DFD

Level 2 DFD

If a process links several external entities with a lot of data flow, we might first divide that specific process and the linked external entities into a different diagram. We can simplify the procedure into a different level of DFD that is comparable to a context diagram. And in doing so, we can much more easily assure that they are consistent. DFD Level 2 digs more deeply into some elements of Level 1. To provide the required amount of detail regarding how the system works, additional text could be needed.

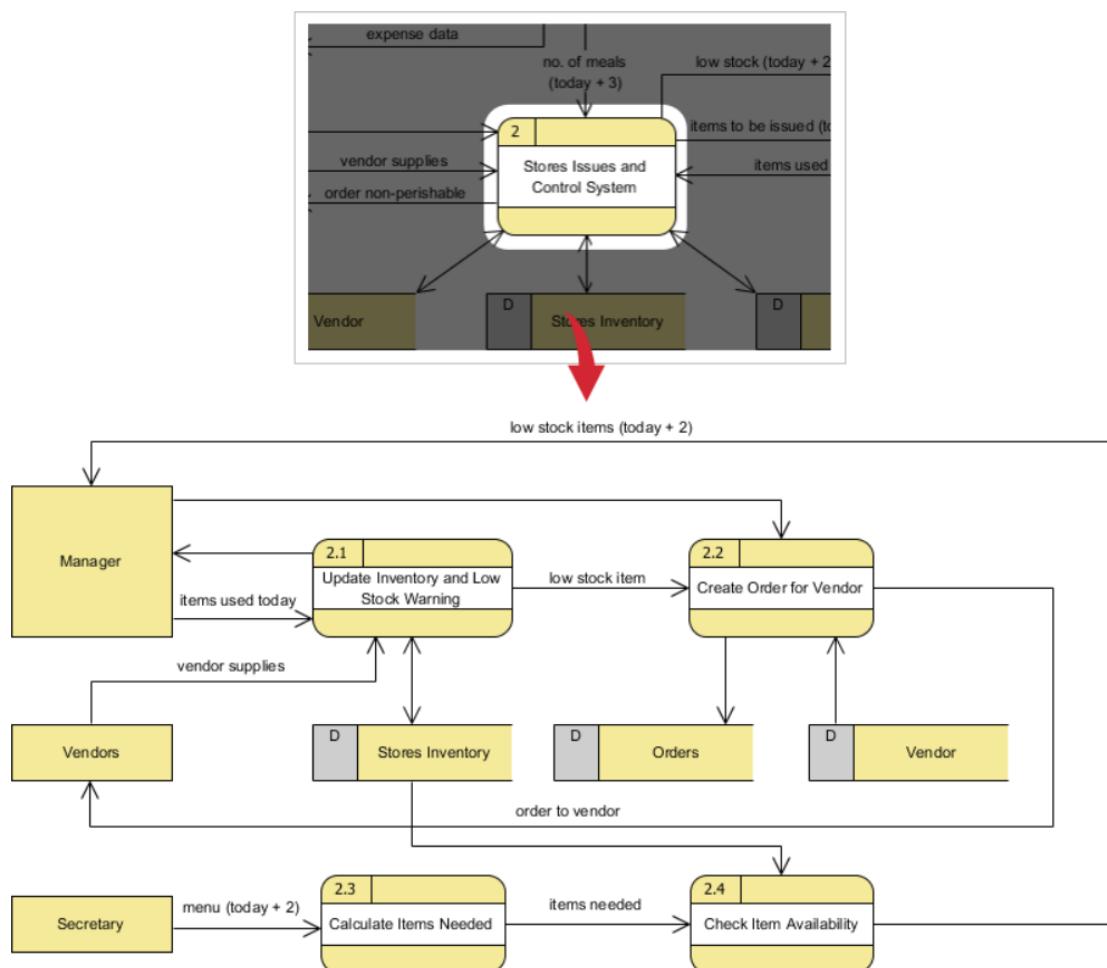


Figure 4. 11 Example Level 2 DFD

4.5.5 Difference between a logical DFD and physical DFD

There are two types of data flow diagrams: logical and physical. A diagram of the logical data flow focuses on the company and how it runs. It is not worried about how the system will be put together. By concentrating on the tasks carried out by the system, such as data gathering, data to information translation, and information reporting, we can overlook implementation details like computer configuration, data storage technology, communication, or message passing techniques.

The hardware, software, files, and users of the system are all shown in the physical data flow diagram, along with how the system will be implemented. It is designed in such a way that the logical data flow diagrams' descriptions of the processes are accurately carried out to accomplish the business's objective.

4.5.6 Logical Data Flow Diagram benefits with an example

- When used to communicate with project users, a logical diagram that is produced with current business information and business activities in the center provides an excellent communication tool.
- The system that results from logical DFD is more stable because it is based on business events rather than a specific technology or physical setup.
- Analysts can comprehend the business under study and determine the motivation behind implementation plans by using logical DFD.
- Due to the fact that business functions are not frequently changing, systems built using logical DFD will be simpler to manage.
- Logical DFD is frequently simpler to create than physical DFD because it only uses files or databases as data stores, making it less complex.
- A logical DFD can be simply modified to create a physical DFD.

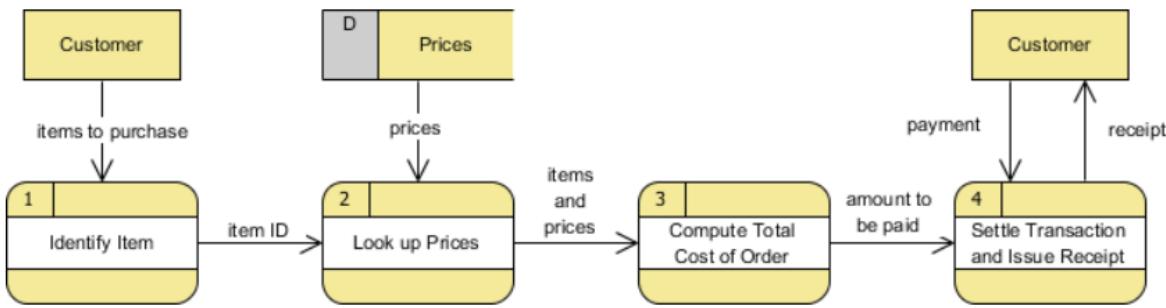


Figure 4. 12 Logical Data Flow Diagram (Grocery Store)

4.5.7 Physical Data Flow Diagram benefits with an example

- Defining the difference between manual and automated processes: Both manual and automated operations require extensive paperwork and computer programs, respectively.
- Providing a more thorough description of processes than logical DFDs: This outlines each step in the data processing process.
- Putting processes in a specific order that they must be completed: It is described the series of actions that produce a significant outcome. For instance, an update needs to be made before a summary report is produced.
- Determining where to store temporary data. The description of temporary storage includes examples like a sales transaction record for a customer receipt in a grocery store.
- Naming files and printouts by their real names: Actual filenames and reports are described in logical data flow diagrams so that programmers can connect them to the data dictionary during the system's development phase.
- Adding measures to guarantee that the processes are carried out correctly: When data is entered, updated, deleted, or processed in any other way, certain conditions or validations of the data must be satisfied.

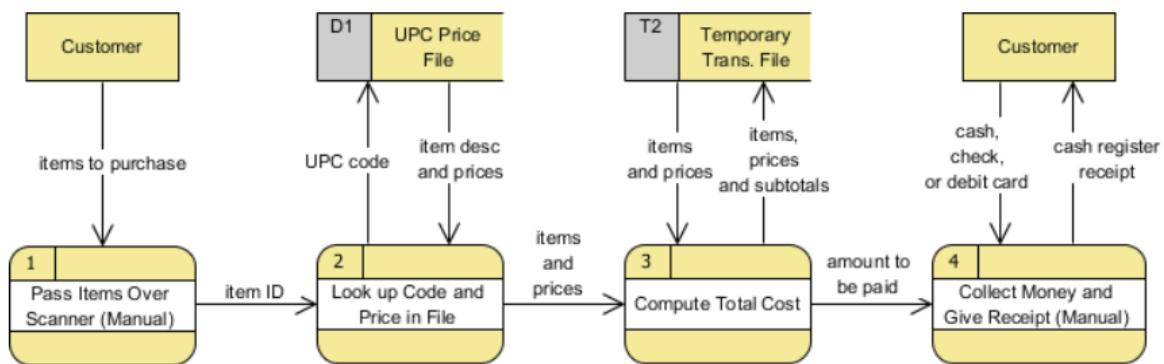


Figure 4. 13 Physical Data Flow Diagram (Grocery Store)

4.5.8 Data Flow Diagram for E-Solutions Pvt Ltd (Level 0)

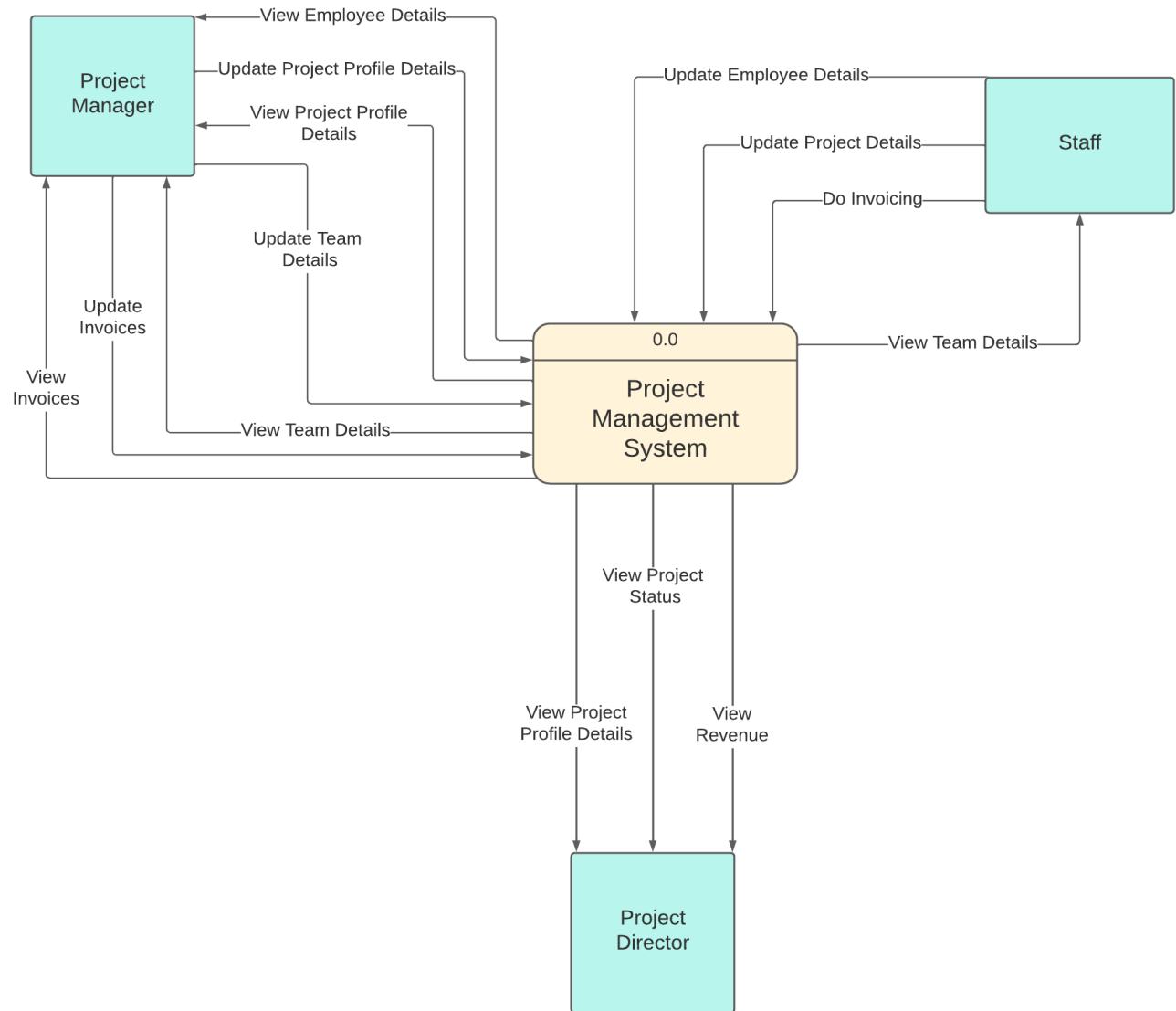


Figure 4. 14 Data Flow Diagram for E-Solutions Pvt Ltd (Level 0)

4.5.9 Data Flow Diagram for E-Solutions Pvt Ltd (Level 1)

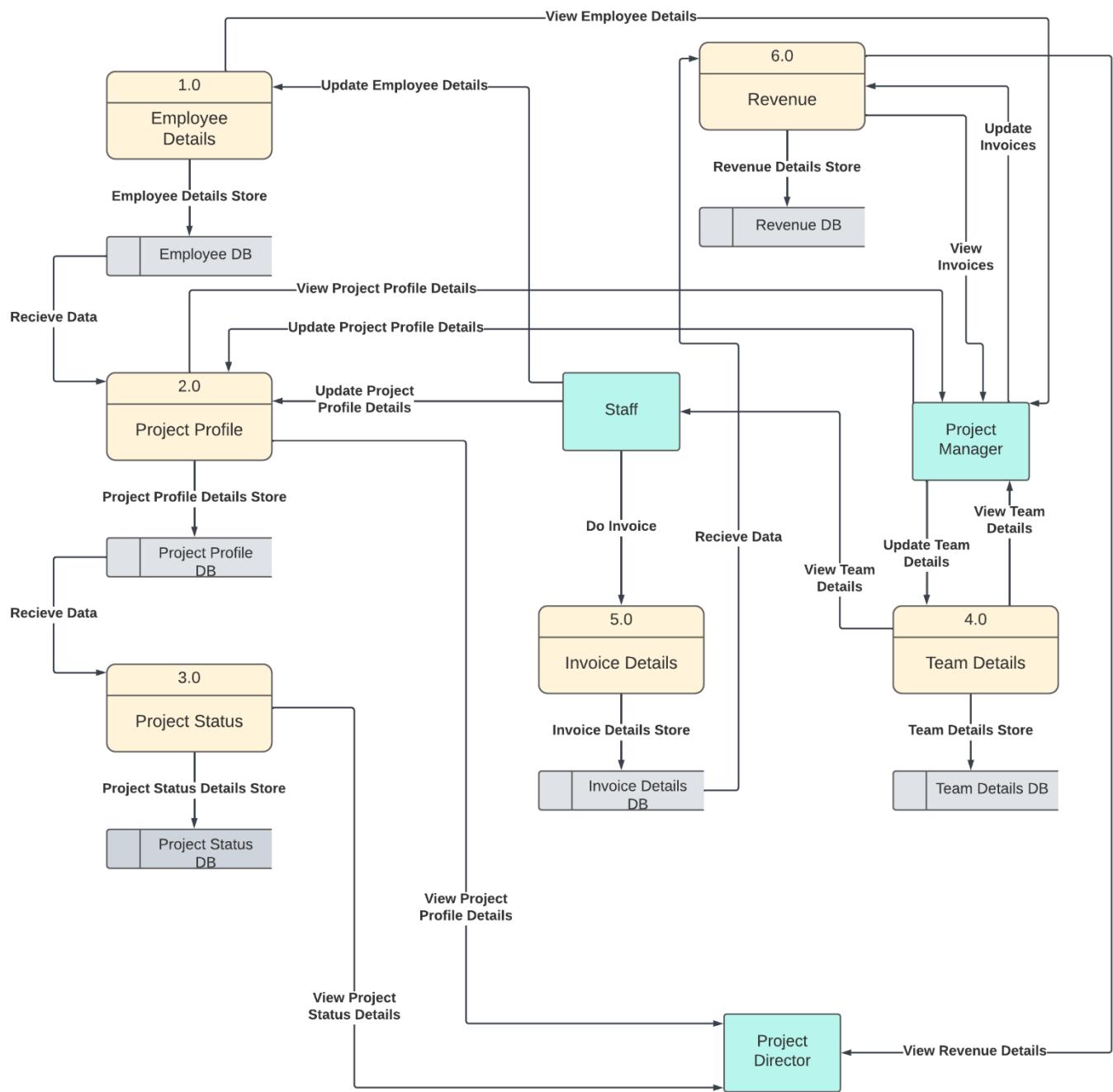


Figure 4. 15 Data Flow Diagram for E-Solutions Pvt Ltd (Level 1)

4.6 Flow Chart Diagram

4.6.1 Introduction to Flow Chart Diagram

A flowchart is a diagram that shows how a system, computer algorithm, or process works. They are frequently used in many different fields to analyze, organize, develop, and describe frequently complex processes in simple, understandable diagrams. Rectangles, ovals, diamonds, and possibly many other shapes are used in flowcharts, also known as flow charts, to indicate the type of step and connecting arrows to indicate flow and sequence. Simple hand-drawn charts and detailed computer-drawn diagrams that show numerous steps and paths are both acceptable forms of these diagrams. Considering all the different variations, flowcharts rank among the most widely used diagrams on the world, being utilized by both technical and non-technical professionals in a wide range of industries.

In the 1920s and 1930s, flowcharts were first used to describe business operations. Frank and Lillian Gilbreth, industrial engineers, presented the "Flow Process Chart" to the American Society of Mechanical Engineers (ASME) in 1921. Gilbreth's tools were used by industrial engineer Allan H. Morgensen to provide conferences on increasing productivity to businesspeople at his company in the early 1930s. Two of Morgensen's students from the 1940s, Art Spinanger and Ben S. Graham, helped the techniques become more popular. Using flow process diagrams, Graham, a director at Standard Register Industrial, modified them for information processing. Based on the Gilbreths' original research, ASME approved a symbol system for flow process charts in 1947.

Herman Goldstine and John Van Neumann both utilized flowcharts to create computer systems in the late 1940s, and diagramming quickly gained popularity for many kinds of algorithms and computer programs. Despite the usage of pseudocode, a mixture of words and coding language intended for human understanding, flowcharts are still employed in programming today. It is frequently used to show finer details and get closer to the final product.

4.6.2 Why use Flow Chart Diagram (Benefits)

- The rationale of the system is more effectively explained through flowcharts.
- During the design of a system, flowcharts serve as a blueprint guide.
- The debugging process is supported by flowcharts.
- It is simple to analyze systems with the support of flowcharts.
- Better documentation is provided.
- Flowcharts are a useful form of official documentation.
- Software flaws are simple to track down.
- Simple to understand.
- It is possible to reuse the flowchart to avoid future inconveniences.

4.6.3 Flow Chart Symbols & Notations

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

Figure 4. 16 Flow Chart Symbols & Notations

4.6.4 Flow Chart Diagram for E-Solutions Pvt Ltd

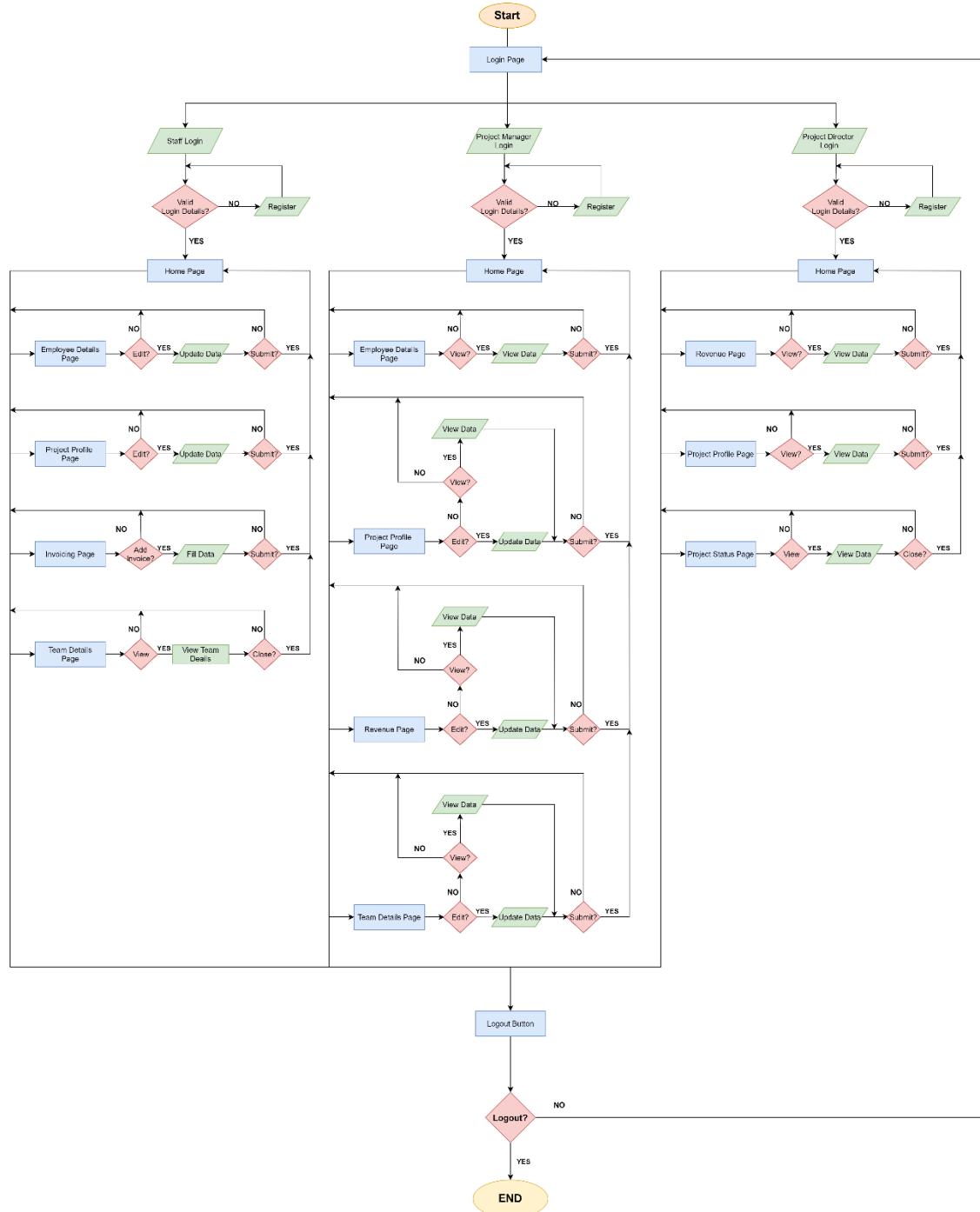


Figure 4. 17 Flow Chart Diagram for E-Solutions Pvt Ltd

4.7 Activity Diagram

4.7.1 Introduction to Activity Diagram

Activity Diagrams are used to represent the flow of control in a system and to relate to the processes associated with the execution of a use case. Activity diagrams are used to model both sequential and concurrent activities. In essence, we use an activity diagram to visually represent workflows. An activity diagram emphasizes the state of flow and the order in which it occurs. Using an activity diagram, we can explain or show what leads to a specific event. Structure diagrams, interaction diagrams, and behavior diagrams are the three main types of diagrams that UML describes. An activity diagram is a behavioral diagram, means it shows how a system behaves.

In an activity diagram, the control flow is shown from a start point to an end point, along with the numerous decision pathways that can be taken while the activity is being carried out. Using an activity diagram, we can show both concurrent and sequential processing of activities. Their main purpose is to represent the dynamic elements of a system, and they are utilized in business and process modeling. And also the flowchart and an activity diagram are extremely similar.

4.7.2 Activity Diagram Basic Components

It is important to first comprehend the structure of an activity diagram before we start creating one. The following are some of the most typical elements of an activity diagram.

- Action: An action where users or software carry out a certain task.
- Decision node: The control flows leaving a decision node will have guard conditions that, if satisfied, will allow control to flow.
- Control flows: A control flow illustrates how control moves from one action to the next.
- Start node: The start of the action is symbolized by this.
- End node: Shows the activity's last step.

4.7.3 Why use Activity Diagram (Benefits)

- Show the algorithm's logic in action.
- Describe the actions that are taken in a UML use case.
- A workflow or business process between users and the system will be illustrated.
- By making complex use cases more understandable, we can improve any process.
- They may be enhanced with components from different notations.
- Automatic verification is possible for them.

4.7.4 Activity Diagram Symbols & Notations

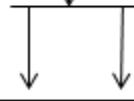
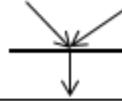
Sr. No	Name	Symbol
1.	Start Node	
2.	Action State	
3.	Control Flow	
4.	Decision Node	
5.	Fork	
6.	Join	
7.	End State	

Figure 4. 18 Activity Diagram Symbols & Notations

4.7.5 Activity Diagram for E-Solutions Pvt Ltd (For the Staff)

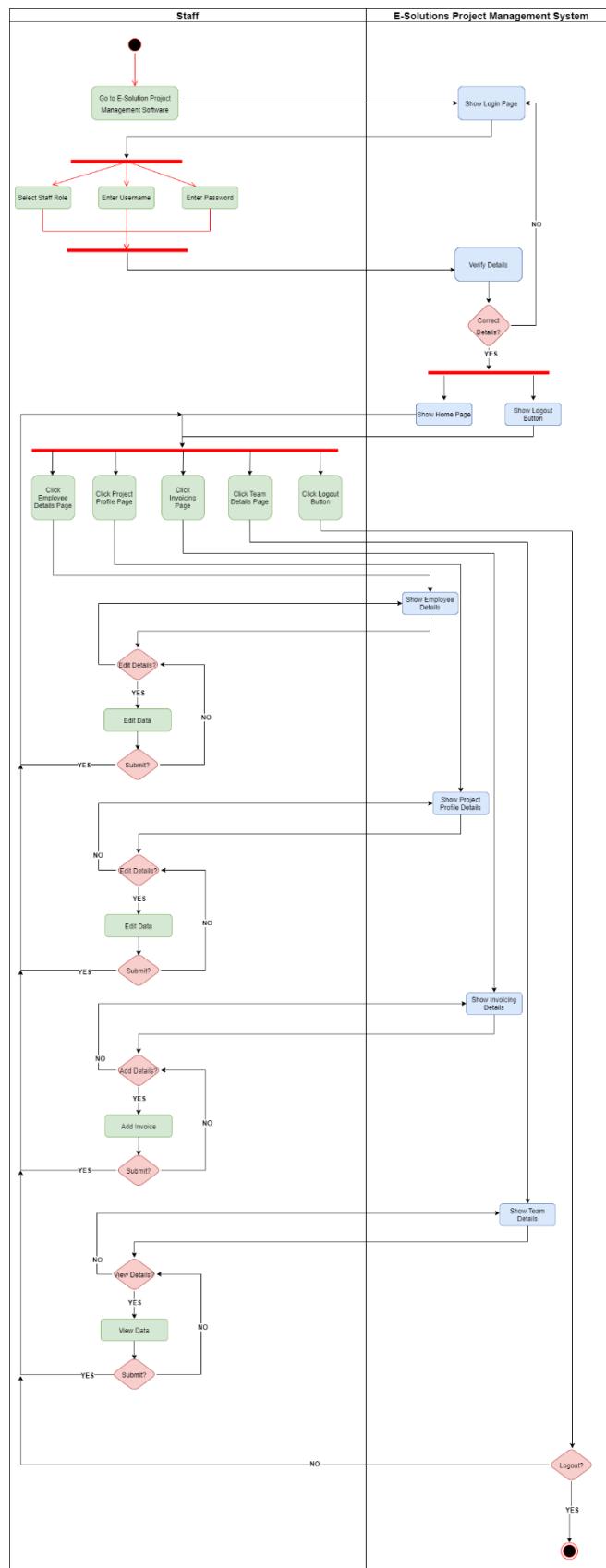


Figure 4. 19 Activity Diagram for E-Solutions Pvt Ltd (For the Staff)

4.7.6 Activity Diagram for E-Solutions Pvt Ltd (For the Project Manager)

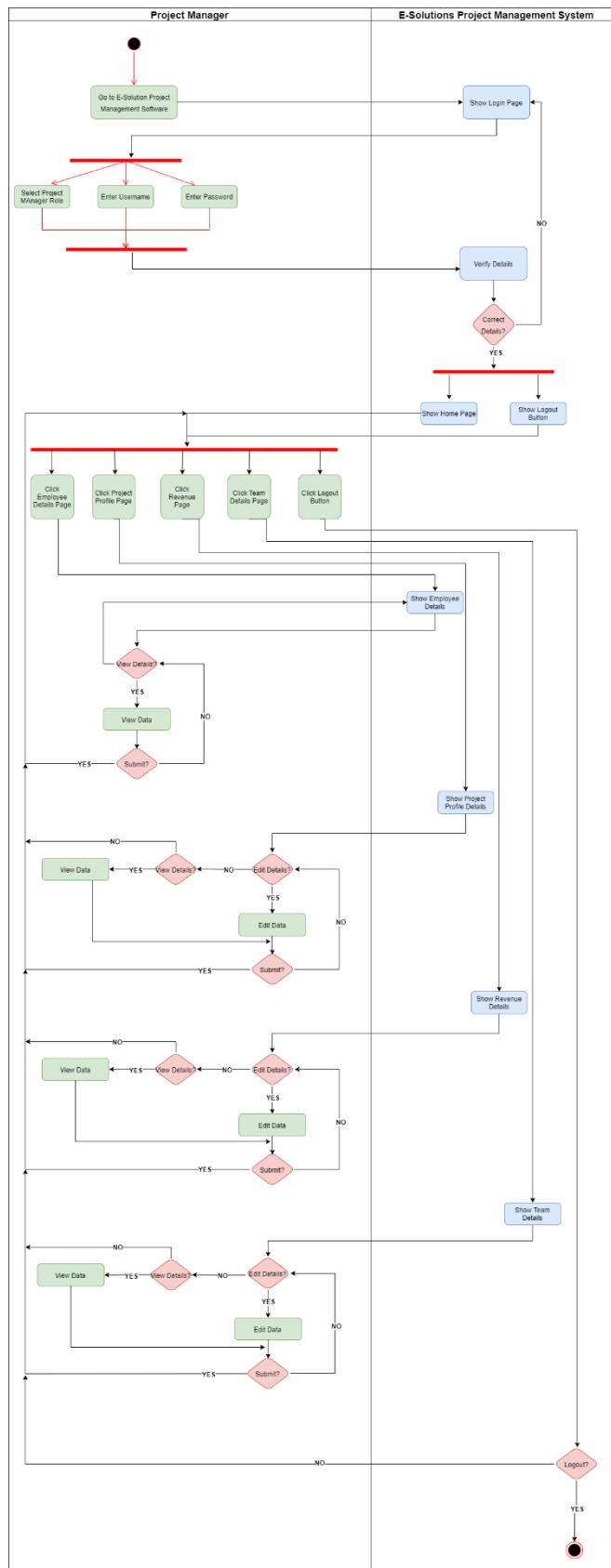


Figure 4. 20 Activity Diagram for E-Solutions Pvt Ltd (For the Project Manager)

4.7.7 Activity Diagram for E-Solutions Pvt Ltd (For the Project Director)

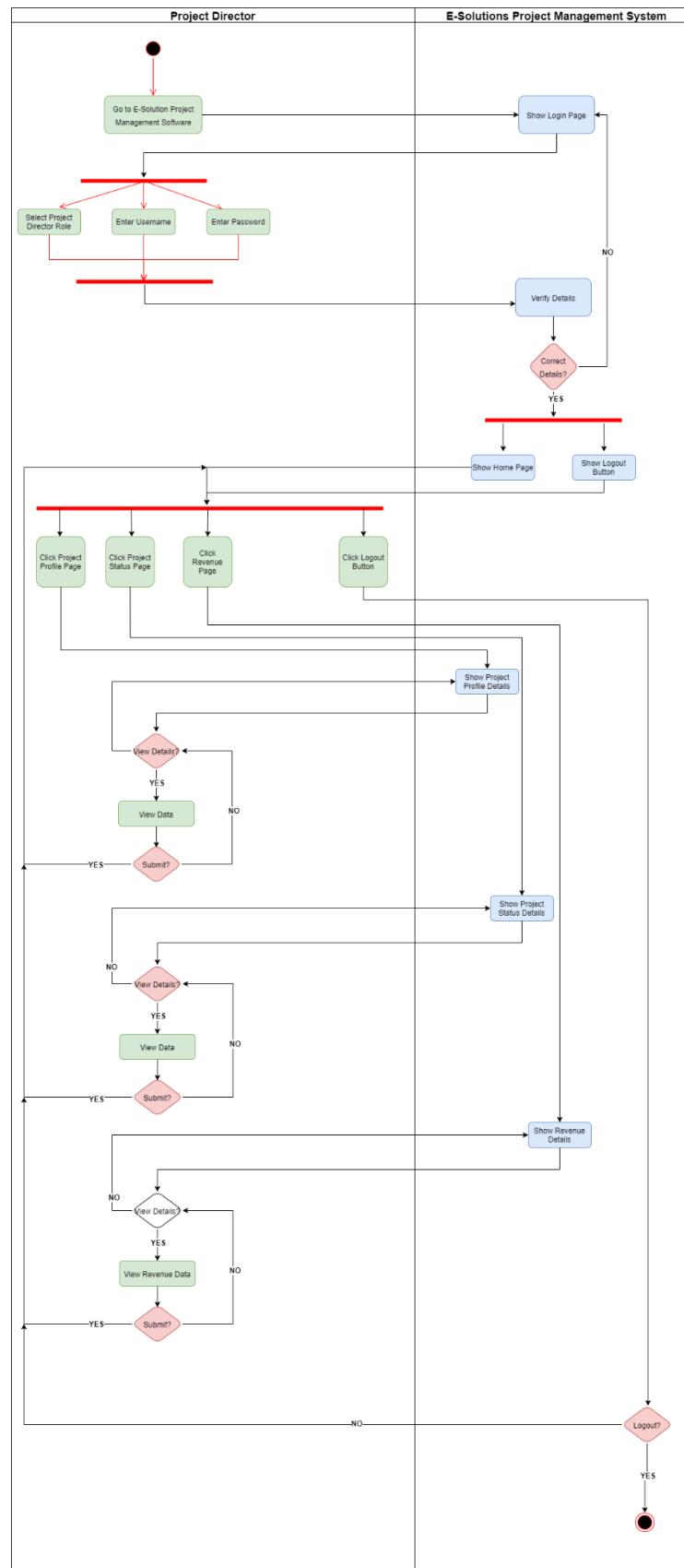


Figure 4. 21 Activity Diagram for E-Solutions Pvt Ltd (For the Project Director)

4.8 Sequence Diagram

4.8.1 Introduction to Sequence Diagram

A sequence diagram is a diagram created using the Unified Modeling Language (UML) that shows the flow of messages sent and received by objects during an interaction. A group of objects that are symbolized by lifelines and the messages they exchange over the course of an interaction make up a sequence diagram. The order in which messages are transferred between objects is shown in a sequence diagram. Sequence diagrams can also display the command chains that link items. A customer, teller, or bank manager, for instance, could be represented by a lifeline in a sequence diagram for a banking scenario. Messages transmitted between them serve as a representation of the interaction between the customer, teller, and manager. The objects and the communications exchanged between them are represented in the sequence diagram.

4.8.2 Sequence Diagram Basic Components

- **Class Roles or Participants:** The behaviors of objects are described by their class roles. Instead of listing object properties, use the UML object symbol to represent class roles.
- **Activation or Execution Occurrence:** The amount of time an item requires to finish a job is represented by activation boxes. Use a thin gray rectangle positioned vertically on the lifeline of the object to indicate that it is busy processing information or waiting for a response.
- **Messages:** Arrows that signify communication between objects are messages.
- **Lifelines:** Vertical dashed lines called lifelines show the object's presence across time.

4.8.3 Why use Sequence Diagram (Benefits)

- Any real application or system can be explored with the help of a sequence diagram.
- Sequence diagrams are used to show how messages move between different objects.
- Diagrams of the sequence are simpler to maintain.
- The generation of sequence diagrams is simpler.
- Sequence diagrams are simple to update in response to system changes.
- Reverse and forward engineering are both possible using a sequence diagram.

4.8.4 Sequence Diagram Symbols & Notations

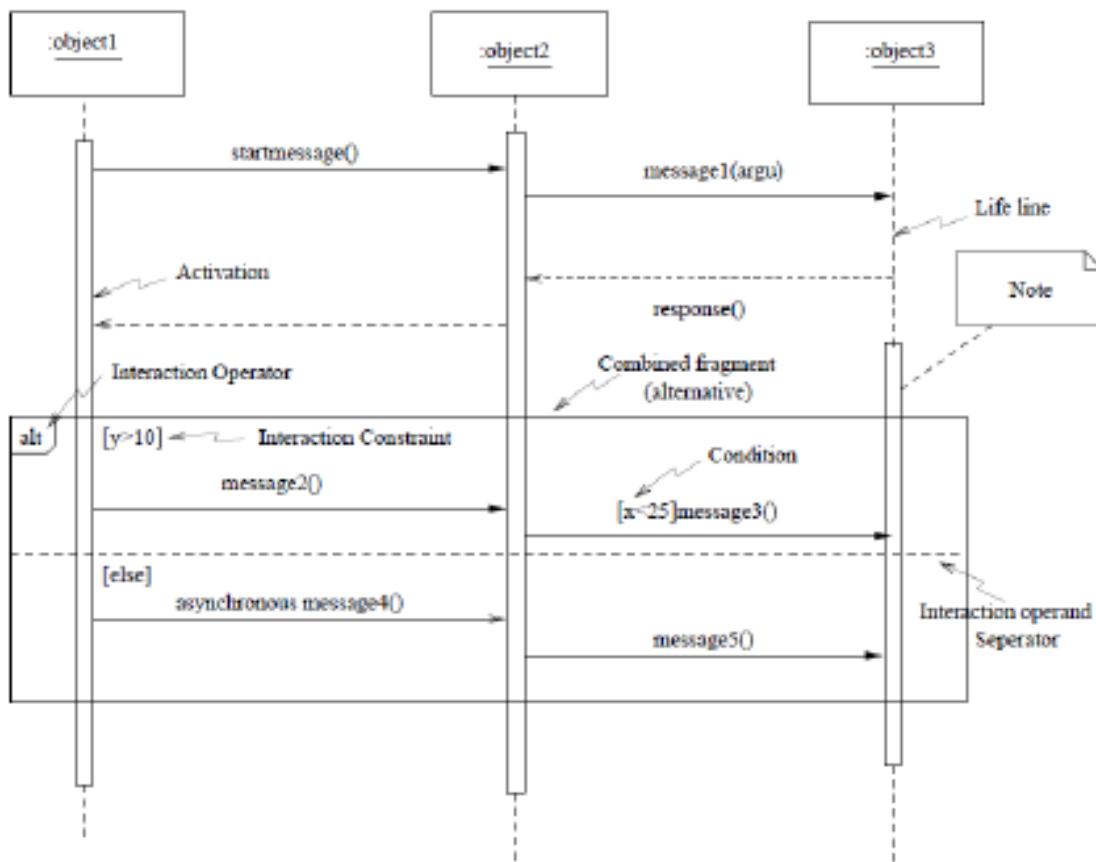


Figure 4. 22 Sequence Diagram Symbols & Notations with an example

4.8.5 Sequence Diagram for E-Solutions Pvt Ltd (For the Staff)

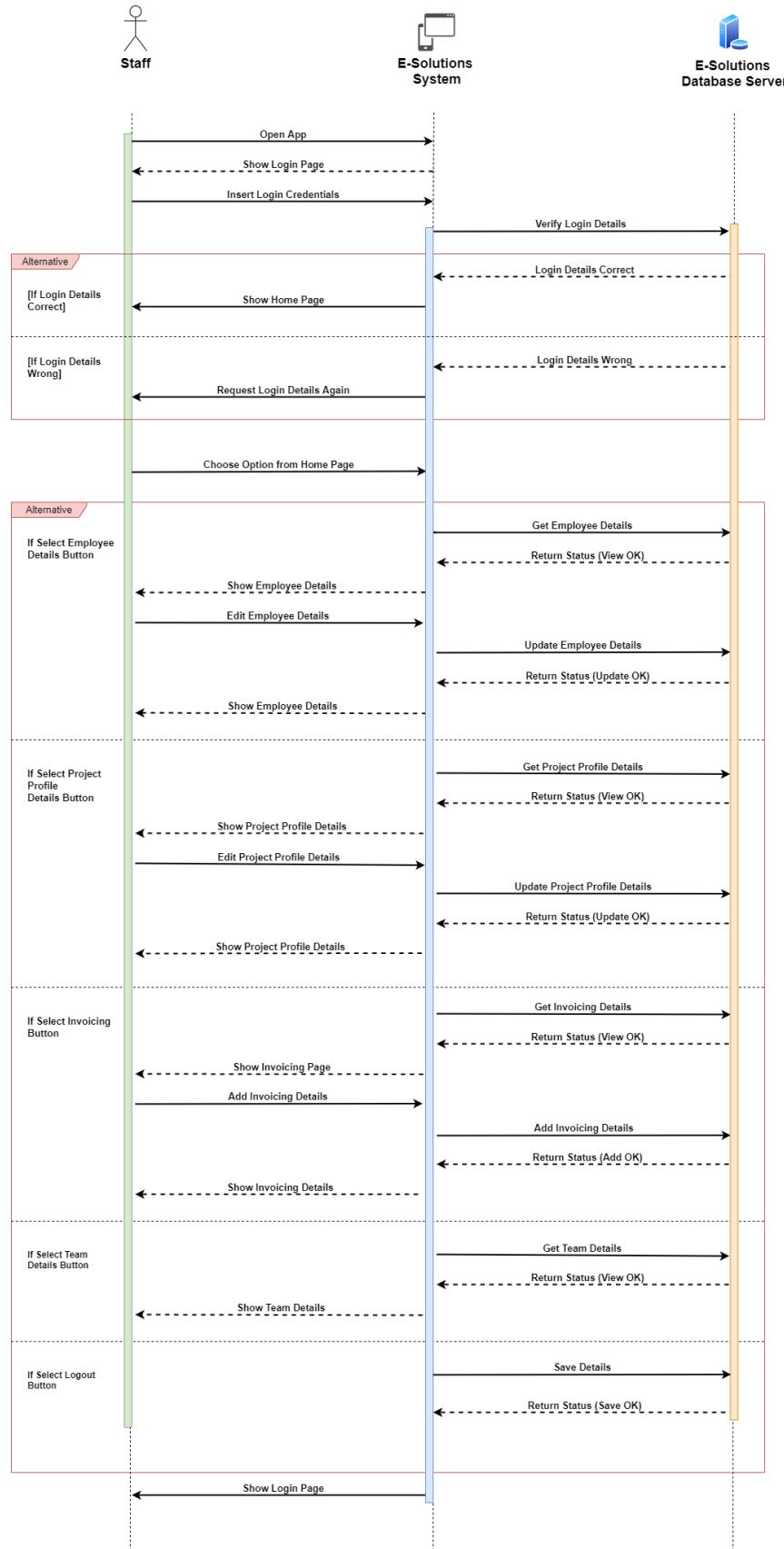


Figure 4. 23 Sequence Diagram for E-Solutions Pvt Ltd (For the Staff)

4.8.6 Sequence Diagram for E-Solutions Pvt Ltd (For the Project Manager)

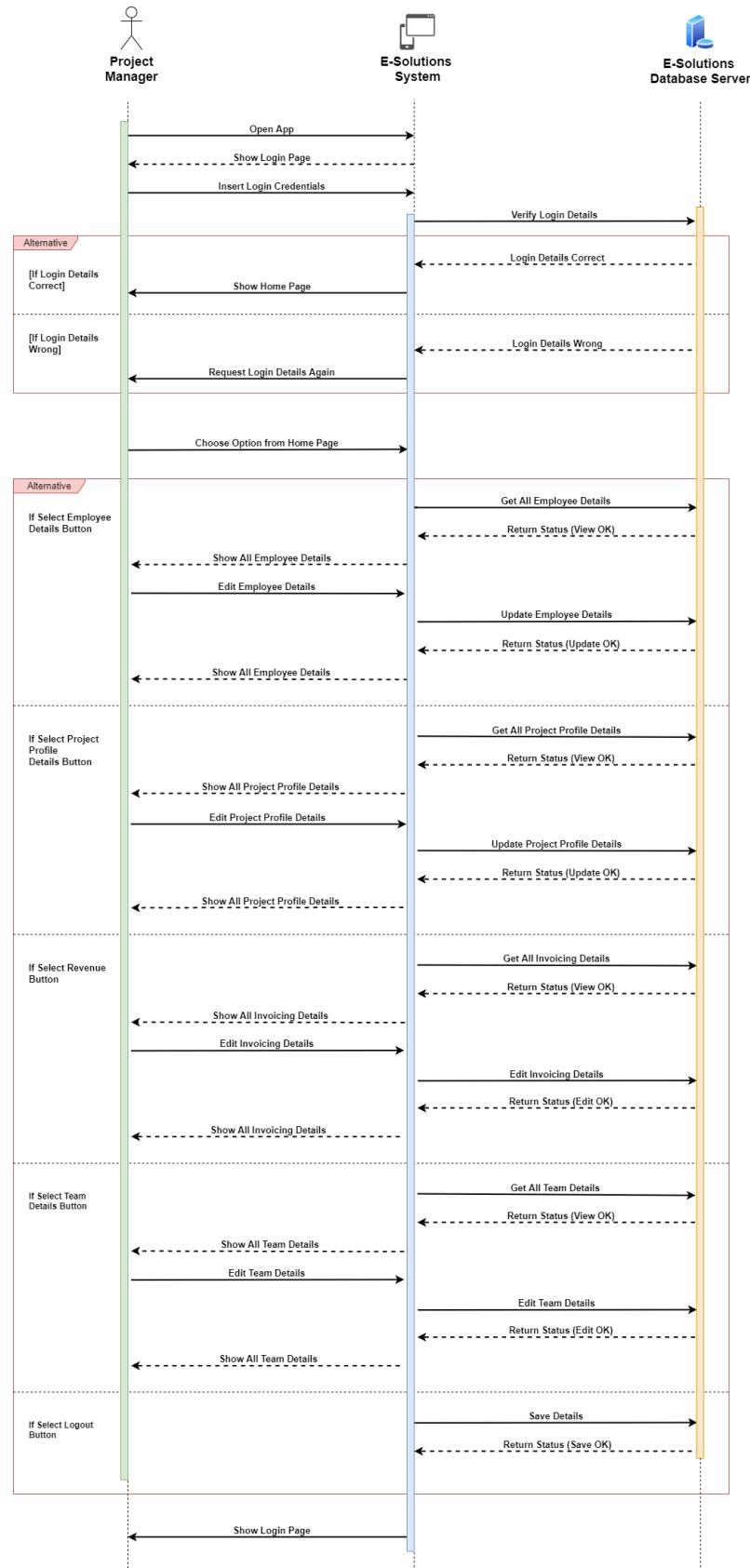


Figure 4. 24 Sequence Diagram for E-Solutions Pvt Ltd (For the Project Manager)

4.8.7 Sequence Diagram for E-Solutions Pvt Ltd (For the Project Director)

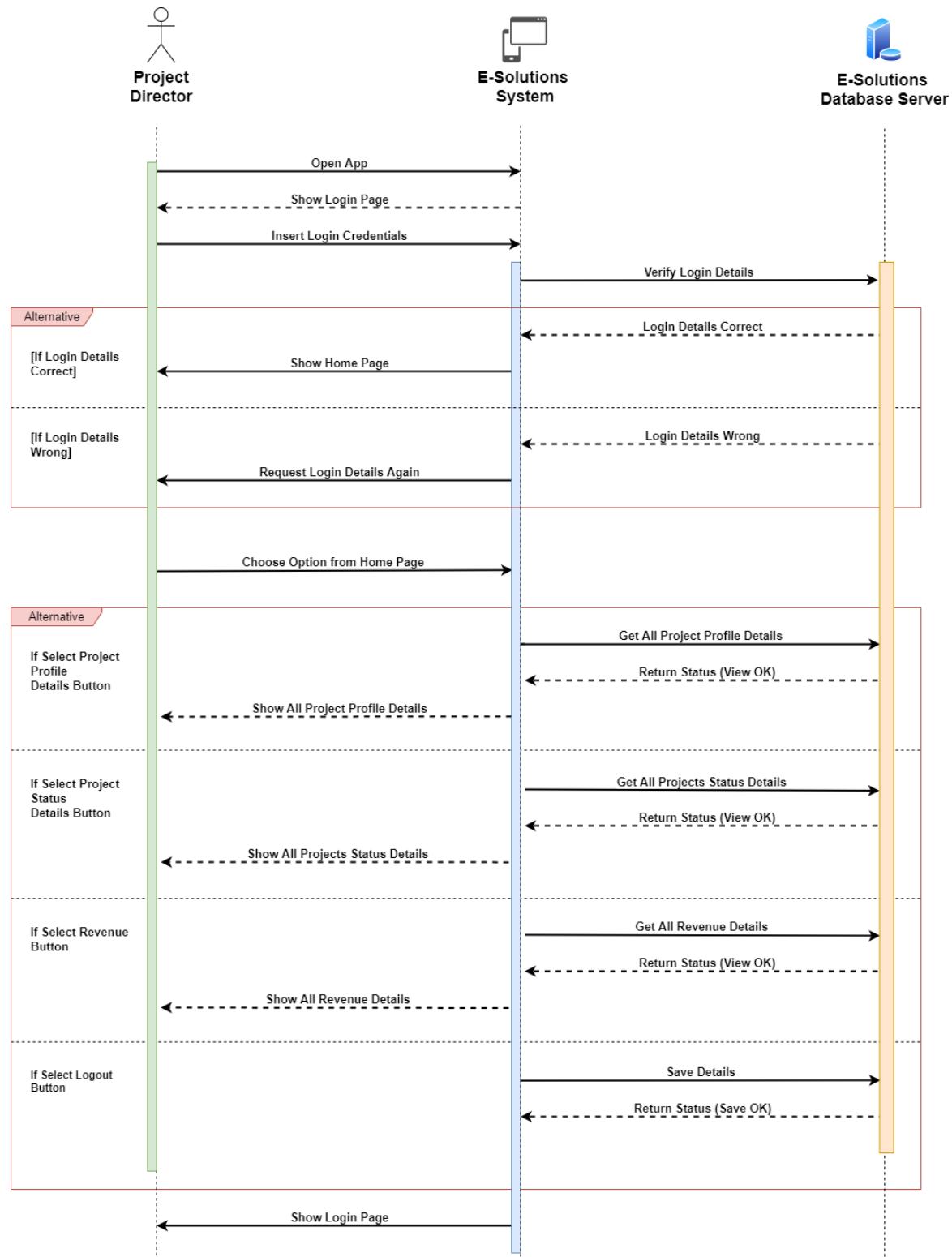


Figure 4. 25 Sequence Diagram for E-Solutions Pvt Ltd (For the Project Director)

4.9 Wireframe Diagram

4.9.1 Introduction to Wireframe Diagram

The term "wireframe" originally referred to a graphic representation of three-dimensional objects, such as those used in product design and development. These days, it's also used to define 3D modeling utilized in computer animation, as well as in the creation of 2D web pages and mobile apps. A wireframe, also known as a wireframe diagram, in web design is a greyscale visual depiction of the functionality and structure of a single web page or mobile app screen. Before visual design and content are added, a page's basic structure is established using wireframes, which can be generated on paper, directly in HTML/CSS, or with software programs.

One of the initial steps in developing interfaces for mobile applications and websites as well as one of the most crucial phases that have an impact on the future of any digital product, is the creation of wireframes. It's important to recognize the value of wireframing while creating a website or mobile application. In the meantime, a poor user experience may be the reason that individuals never even start using it. While developing web apps, this should also be taken into account as it is particularly crucial for mobile applications. Buttons, checkboxes, and other items must be arranged in a practical way to allow for simple and convenient program navigation.

4.9.2 Wireframe Diagram Basic Components

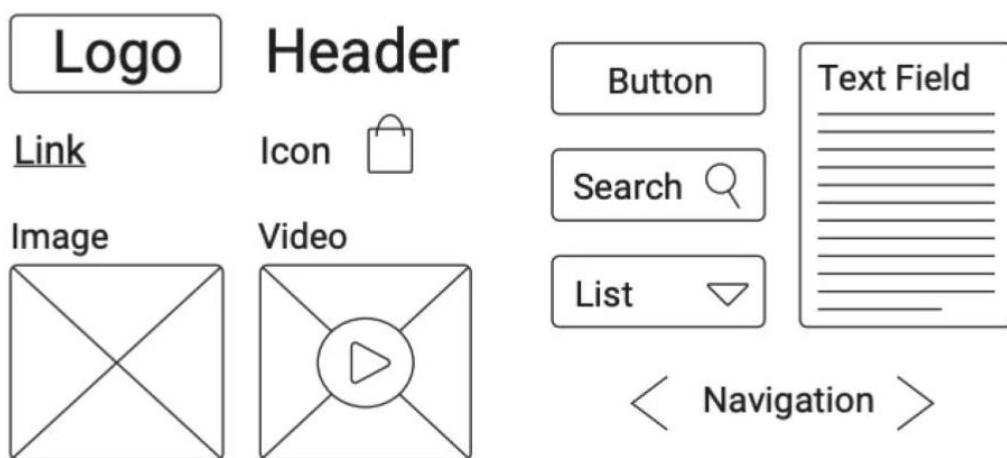


Figure 4. 26 Wireframe Diagram Basic Components

4.9.3 Why use Wireframe Diagram (Benefits)

- Before design or development, wireframes guarantee that all layers of content are taken into account, assist in visualizing user journeys, and support in brainstorming about website navigation.
- With wireframing, we can "test" the viability of the website architecture as soon as possible.
- Project scope creep and costs are decreased by wireframing.
- A client can better comprehend more practical functionality like filtering, widgets, or jQuery by wireframing the website's components and outputs.
- Engaging in the wireframing process is essential for improving user experience (UX).
- We can use wireframing to test layouts, meet goals, and test calls to action as well as to sketch out important pages.
- The web designer can better comprehend what has to be accomplished from the perspective of the user by using wireframes.
- Wireframes assist in breaking the design process down into smaller, simpler, and faster steps rather than mocking up a whole design with finished aspects like site architecture, styles, and brand and logo.
- Wireframes will surely save time and money in the long term, despite the fact that they may first look like an extra delivery.

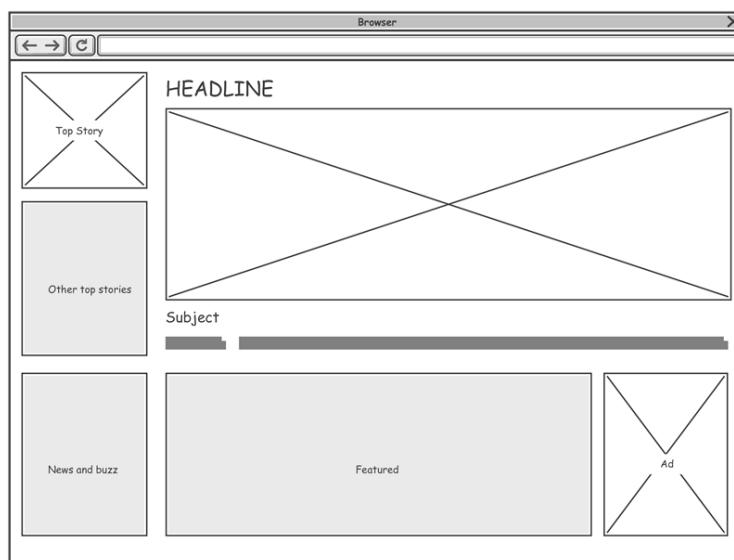


Figure 4. 27 Wireframe Diagram example

4.9.4 Wireframe Diagrams for E-Solutions Pvt Ltd

The wireframe shows a login form titled "USER LOGIN". It includes fields for "Select Role" (a dropdown menu), "Username" (text input), and "Password" (text input). Below the form are two buttons: "Sign In" and "Register". The background features a large, stylized 'X' shape.

Figure 4. 28 Wireframe Diagrams for E-Solutions Pvt Ltd - Login Page

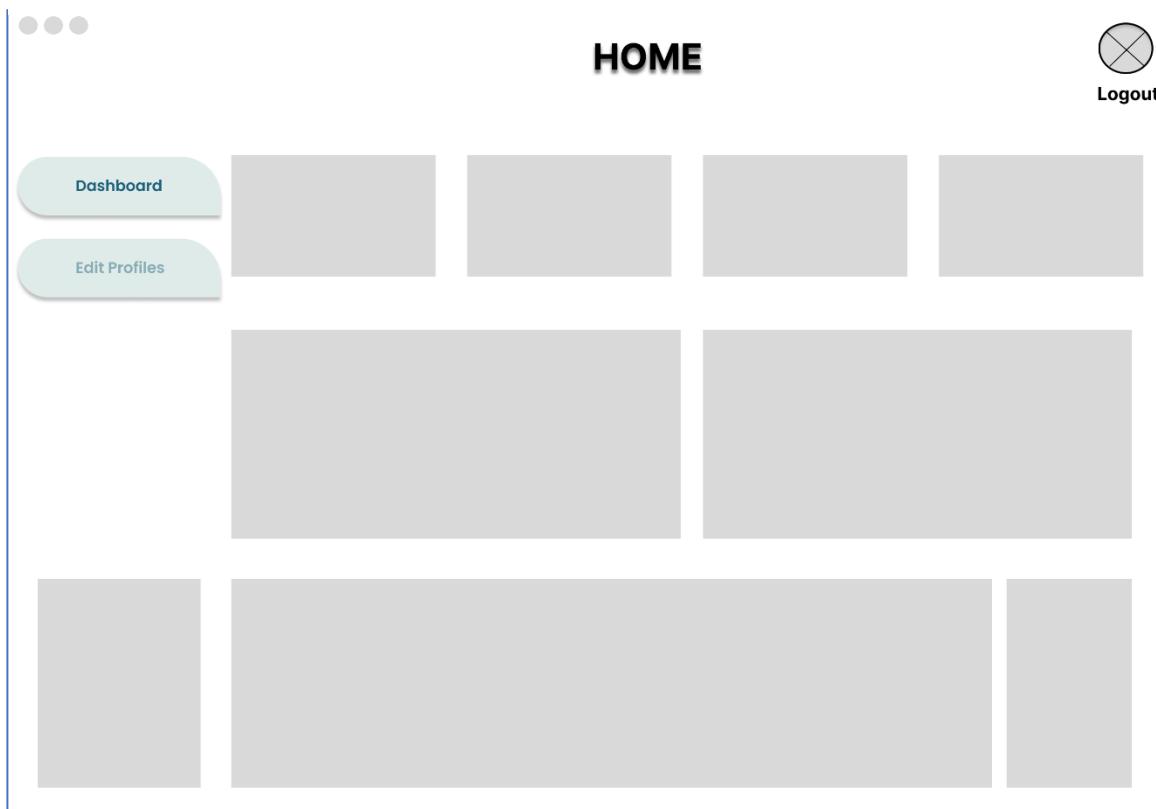


Figure 4. 29 Wireframe Diagrams for E-Solutions Pvt Ltd - Home Page

Edit Profiles


Employee Details

Employee Details

Project Profile


Team Details

Invoicing


LOGOUT

Employee Details

Employee ID

Name

Phone

Address

Email

Designation

Salary

Assigned Projects

Submit **Edit** **Delete**

HOME

Figure 4. 30 Wireframe Diagrams for E-Solutions Pvt Ltd - Employee Deatils Page

Edit Profiles


Employee Details

Project Profile


Team Details

Invoicing


LOGOUT

Project Profile

Project ID **Project Name**

Category **Project Manager**

Start Date **End Date**

Task List

Project Cost **Employee Cost**

Project Details

Remarks

Submit **Edit** **Delete**

HOME

Figure 4. 31 Wireframe Diagrams for E-Solutions Pvt Ltd - Project Profile Page

Team Details

Employee Details	Team ID	<input type="text"/>
Project Profile	Team Lead	<input type="text"/>
Team Details	Team Members	<input type="button" value="▼"/>
Invoicing	Task List	<input type="button" value="▼"/>

Logout

Submit **Edit** **Delete**

Figure 4. 32 Wireframe Diagrams for E-Solutions Pvt Ltd - Team Details Page

Invoicing

Employee Details	Invoice ID	<input type="text"/>
Project Profile	Project Name	<input type="text"/>
Team Details	Project Cost	<input type="text"/> <input checked="" type="checkbox"/>
Invoicing	Employee Cost	<input type="text"/> <input type="checkbox"/>
	Additional Cost	<input type="text"/> <input type="checkbox"/>
	Calculate	
	Total Bill	<input type="text"/>
	Remarks	<input type="text"/>

Logout

Submit **Edit** **Delete**

Figure 4. 33 Wireframe Diagrams for E-Solutions Pvt Ltd - Invoicing Page

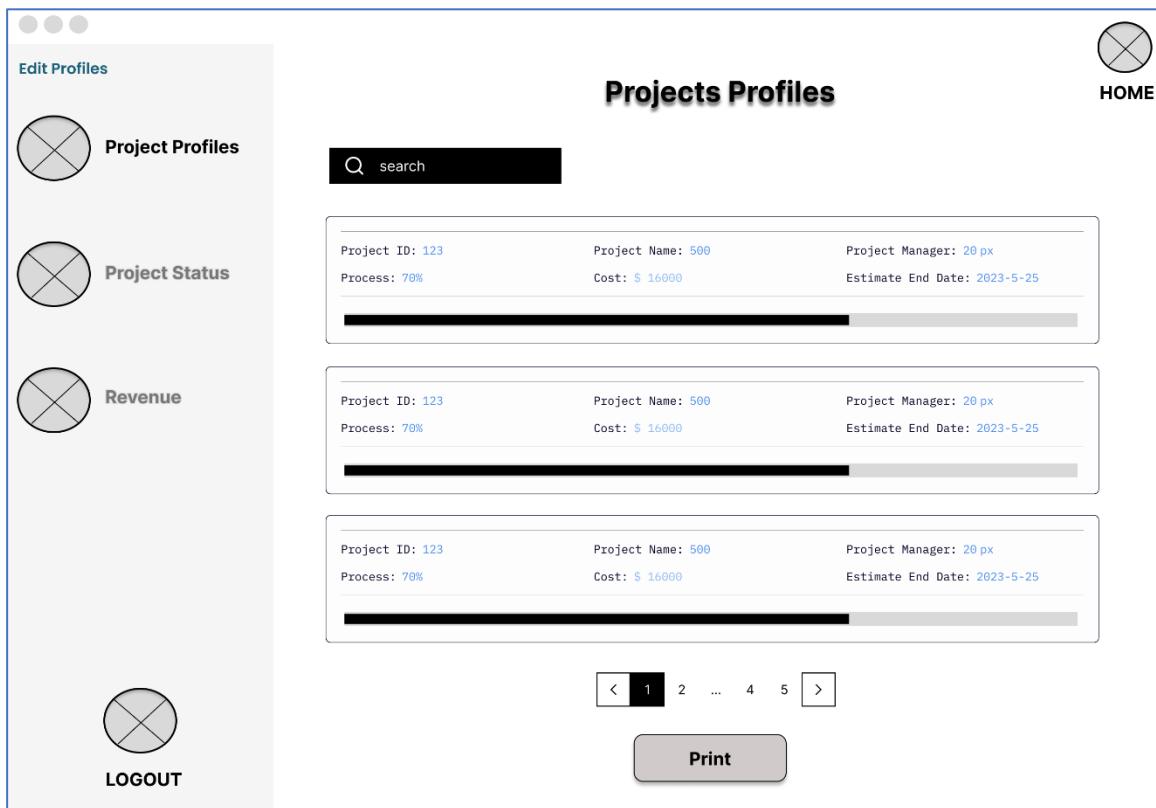


Figure 4. 34 Wireframe Diagrams for E-Solutions Pvt Ltd - Project Profiles Page



Figure 4. 35 Wireframe Diagrams for E-Solutions Pvt Ltd - Projects Status Page

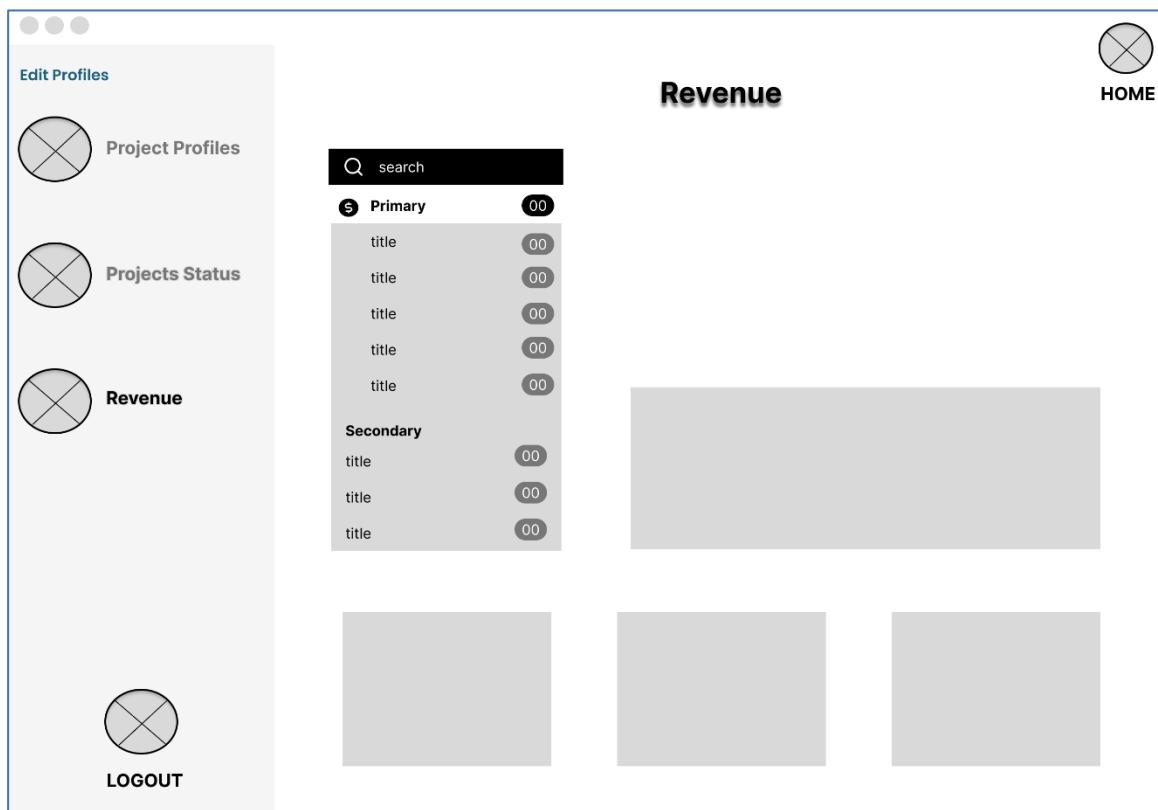


Figure 4. 36 Wireframe Diagrams for E-Solutions Pvt Ltd - Revenue Page

4.10 Prototype Diagram

4.10.1 Introduction to Prototype Diagram

The model form of a product is called a prototype. In order to evaluate a product's features or find flaws so that the final version can be improved, an early, affordable sample of the product is utilized. Using prototypes offers the chance to get insightful product feedback from partners, customers, or stakeholders. Using this knowledge, a product that satisfies their needs can be created.

By using a prototype, we may hide from the client how difficult it is to create new instances. Instead, than starting from scratch and potentially involving expensive processes, the idea is to duplicate an existing object. The current object serves as a prototype and holds the object's current state. Only when necessary may the newly duplicated object make the same property changes. This method saves money and time, especially when creating objects is a laborious process.

4.10.2 Why use Prototype Diagram (Benefits)

- Before the product is released to the general public, gather opinions on its functionality from users and other stakeholders.
- Before the general release, point out potential improvements and assist in finding bugs and usability problems.
- Contribute to cutting wasteful spending.
- Boost cooperation and efficiency within the team.
- Give the user the chance to interact with a functioning model of their product.
- Help create a physical product from an abstract idea while saving money.
- Before moving forward with the product idea, determine whether it is a weak one or will cost a lot of money.

4.10.3 Prototype Diagrams for E-Solutions Pvt Ltd

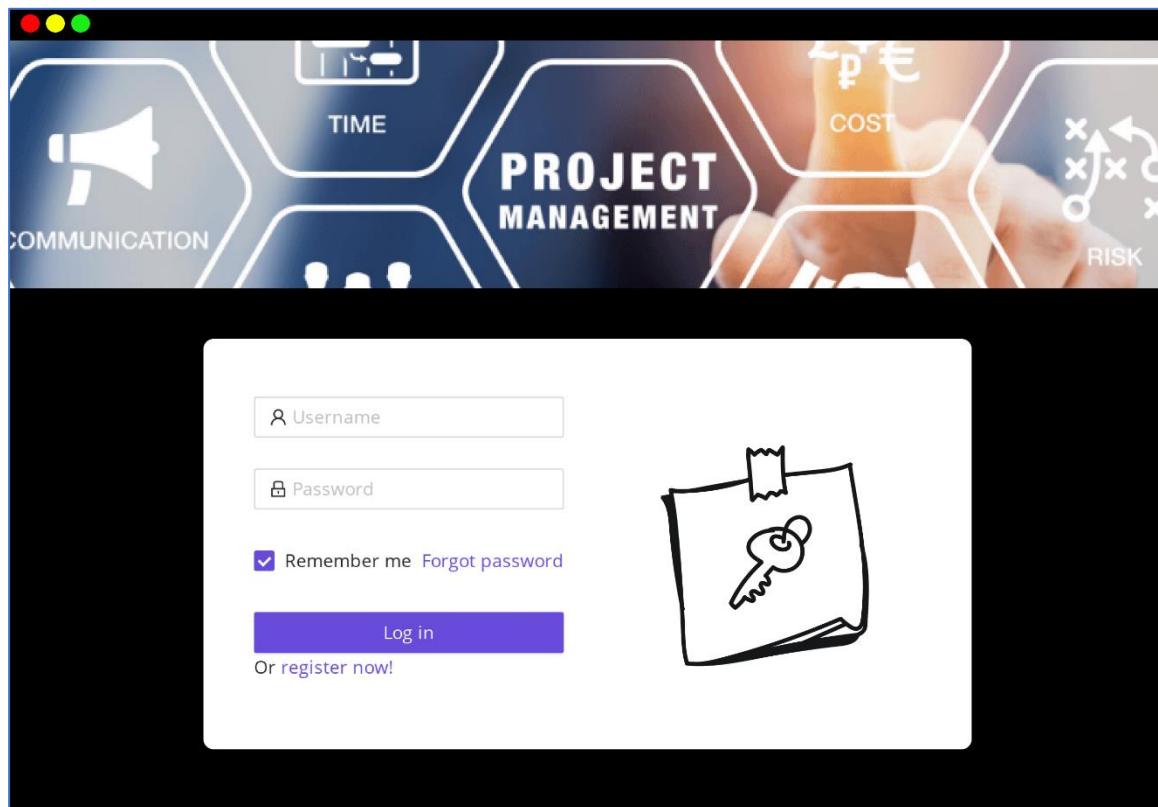


Figure 4. 37 Prototype Diagrams for E-Solutions Pvt Ltd - Login Page

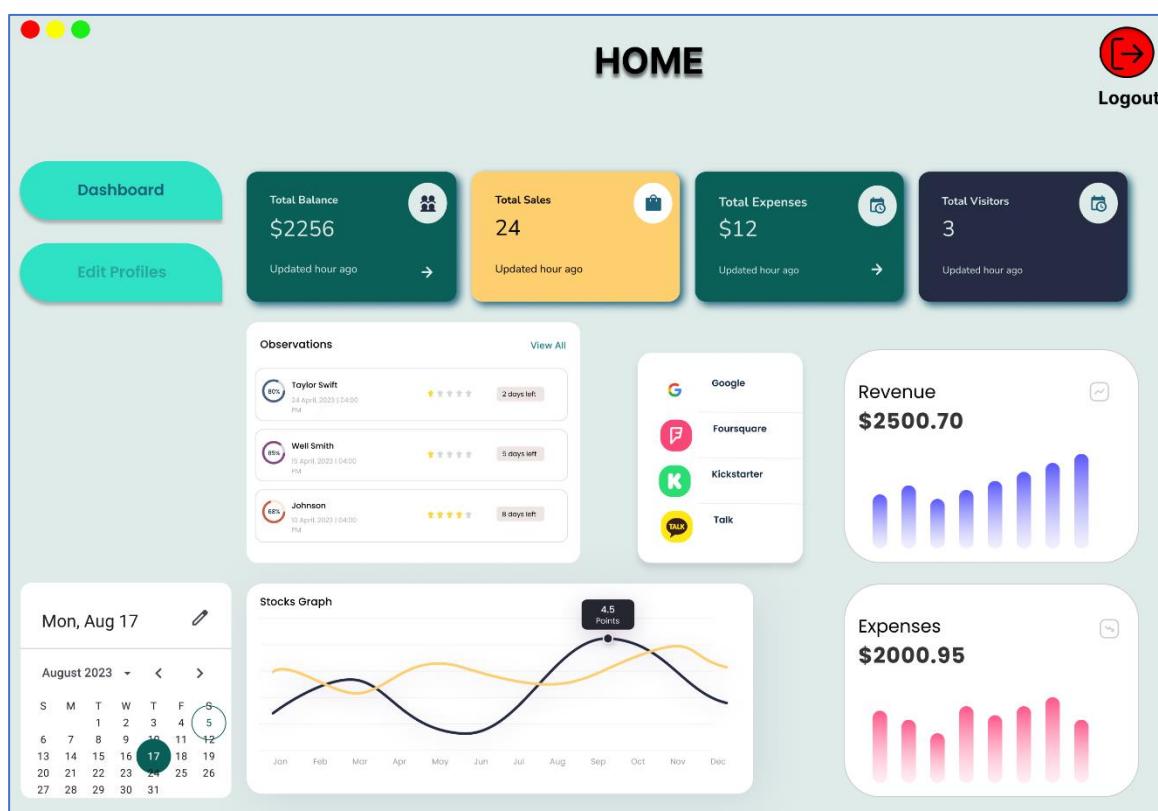


Figure 4. 38 Prototype Diagrams for E-Solutions Pvt Ltd - Home Page

Employee Details

E-Solutions

Edit Profiles

- [Employee Details](#)
- [Project Profile](#)
- [Invoicing](#)
- [Team Details](#)
- [Settings](#)
- [Sign out](#)

Employee ID:

Name:

Phone:

Address:

Email:

Designation:

Salary:

Assigned Projects:

Submit **Edit** **Delete**

HOME

Figure 4. 39 Prototype Diagrams for E-Solutions Pvt Ltd - Employee Details Page

Project Profile

E-Solutions

Edit Profiles

- [Employee Details](#)
- [Project Profile](#)
- [Invoicing](#)
- [Team Details](#)
- [Settings](#)
- [Sign out](#)

Project ID: <input type="text"/>	Project Name: <input type="text"/>
Category: <input type="text"/>	Project Manager: <input type="text"/>
Start Date: <input type="text"/>	End Date: <input type="text"/>
Task List: <input type="text"/>	
Project Cost: <input type="text"/>	Employee Cost: <input type="text"/>
Project Details: <input type="text"/>	
Remarks: <input type="text"/>	

Submit **Edit** **Delete**

HOME

Figure 4. 40 Prototype Diagrams for E-Solutions Pvt Ltd - Project Profile Page

The screenshot shows a software prototype for the 'Invoicing' module. On the left is a vertical sidebar with a blue header containing the 'E-Solutions' logo and navigation links: 'Edit Profiles', 'Employee Details', 'Project Profile', 'Invoicing' (which is highlighted in blue), 'Team Details', 'Settings', and 'Sign out'. The main area has a light gray background with a title 'Invoicing' at the top right. It contains several input fields and buttons. The fields include 'Invoice ID' (text box), 'Project Name' (text box), 'Project Cost' (text box with a checked checkbox to its right), 'Employee Cost' (text box with an unchecked checkbox to its right), 'Additional Cost' (text box with an unchecked checkbox to its right), 'Total Bill' (text box), and 'Remarks' (text box). Below these fields are three buttons: 'Calculate' (green), 'Submit' (yellow), 'Edit' (yellow), and 'Delete' (yellow). A green circular 'HOME' button is located in the top right corner of the main area.

Figure 4.41 Prototype Diagrams for E-Solutions Pvt Ltd - Invoicing Page

The screenshot shows a software prototype for the 'Team Details' module. The layout is similar to the Invoicing page, with a blue sidebar on the left and a light gray main area. The sidebar includes 'Edit Profiles', 'Employee Details', 'Project Profile', 'Invoicing', 'Team Details' (highlighted in blue), 'Settings', and 'Sign out'. The main area features a title 'Team Details' at the top right. It contains four input fields with dropdown arrows: 'Team ID' (text box), 'Team Lead' (text box), 'Team Members' (dropdown arrow), and 'Task List' (dropdown arrow). At the bottom are 'Submit' (yellow), 'Edit' (yellow), and 'Delete' (yellow) buttons. A green circular 'HOME' button is in the top right.

Figure 4.42 Prototype Diagrams for E-Solutions Pvt Ltd - Team Details Page

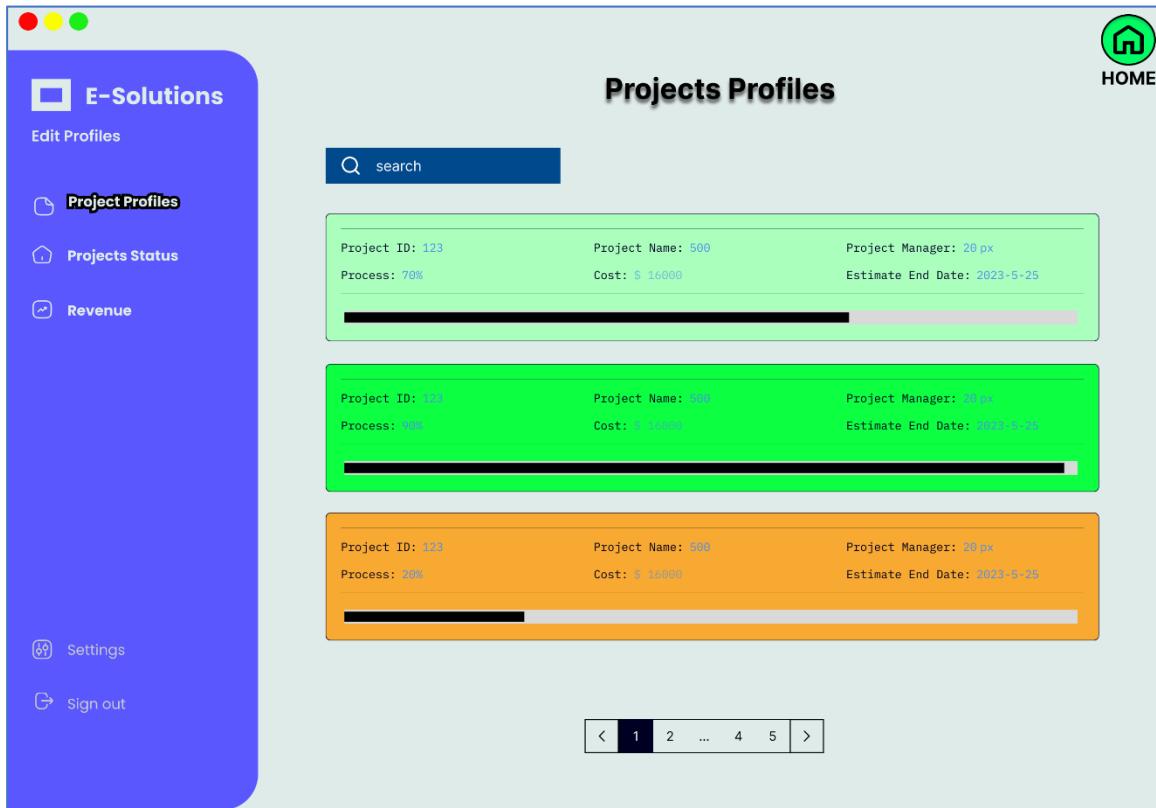


Figure 4. 43 Prototype Diagrams for E-Solutions Pvt Ltd - Projects Profiles Page

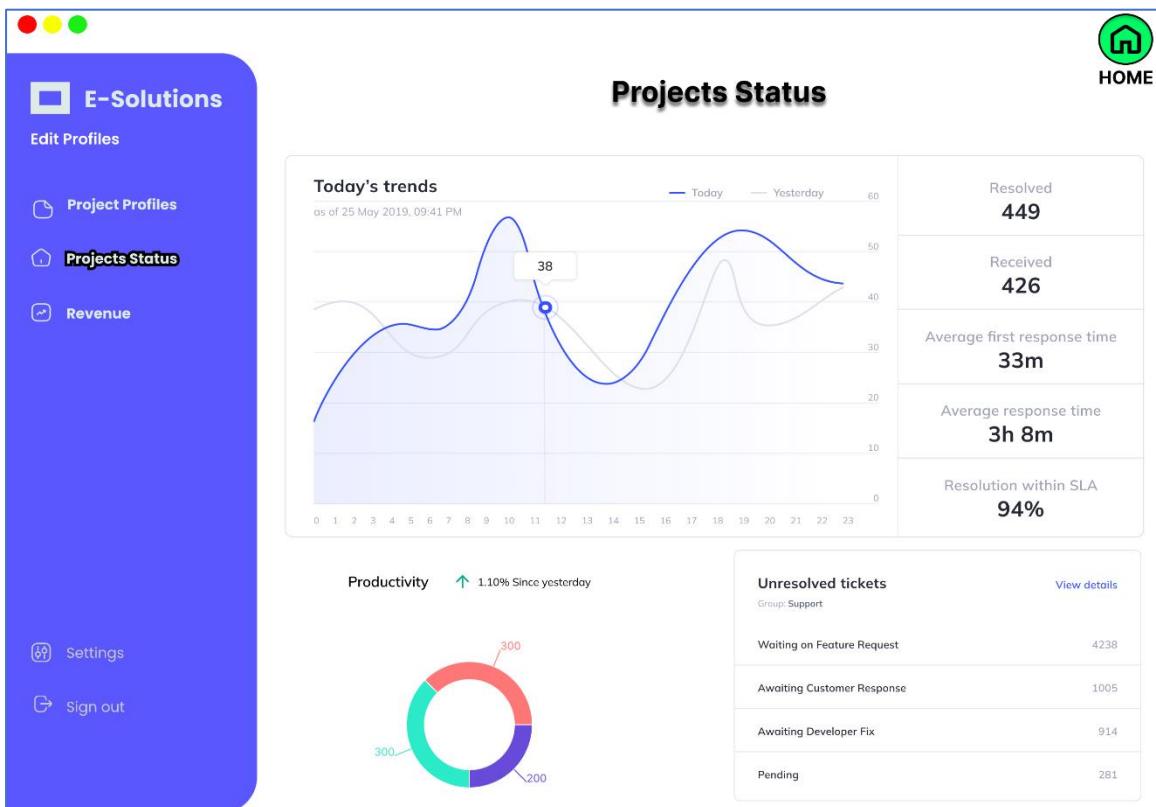


Figure 4. 44 Prototype Diagrams for E-Solutions Pvt Ltd - Projects Status Page

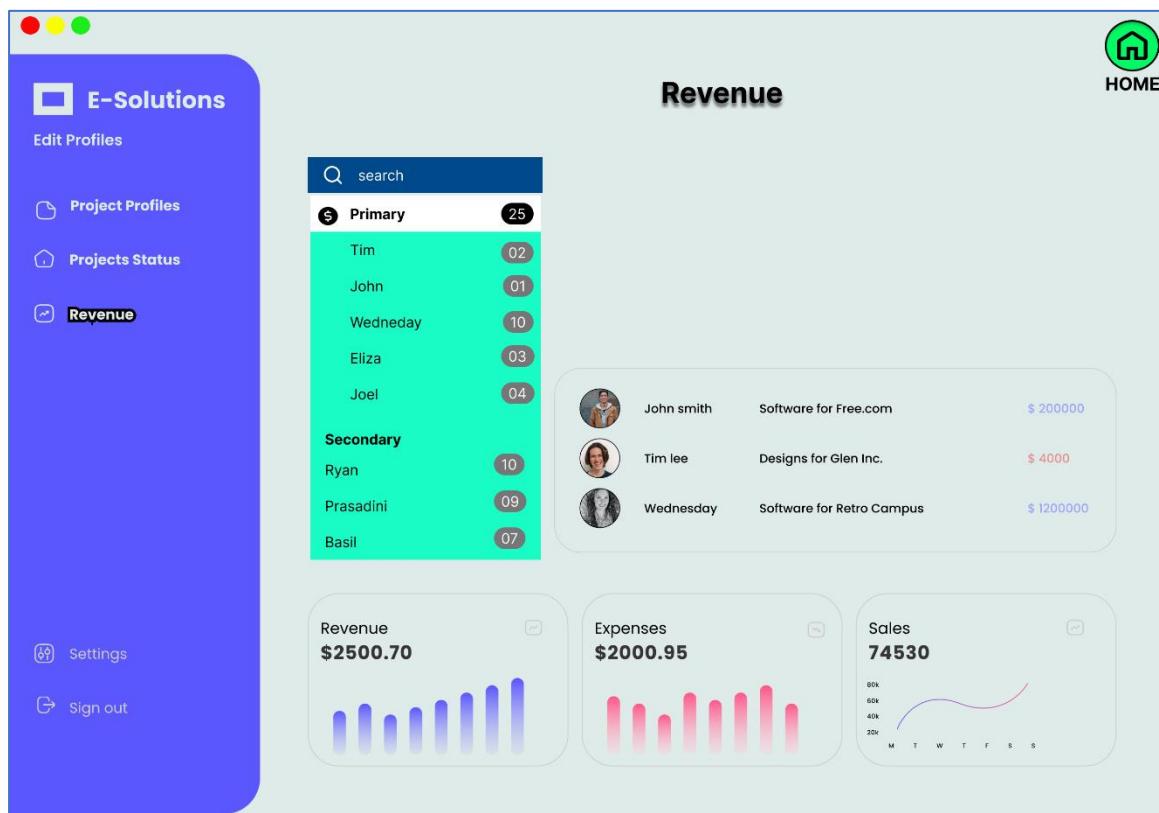


Figure 4. 45 Prototype Diagrams for E-Solutions Pvt Ltd - Revenue Page

4.11 Tools and Techniques

4.11.1 Tools and Techniques that can use to create ERD Diagram

- Vertabelo
- Lucidchart
- Visual Paradigm Online
- Draw.io
- Microsoft Visio
- Gliffy
- SqldbM ER

4.11.2 Tools and Techniques I used to create ERD Diagram

Because Lucidchart's DB diagram tool facilitates a visual workspace that blends diagramming, data visualization, and collaboration to speed up comprehension and spur innovation, I was able to design a logical database diagram with it. Whether my diagrams are conceptual or physical, their ER diagram tool makes database modeling simpler. Their ER diagram maker allowed me to quickly drag and drop common ERD shapes and symbols onto the canvas.

4.11.3 Tools and Techniques that can use to create Logical Database Diagram

- Visual Paradigm ERD Tools
- Vertabelo
- Lucidchart
- DeZign for Databases
- Erwin Data Modeler
- Aqua Data Studio ER Modeler

4.11.4 Tools and Techniques I used to create Logical Database Diagram

Using the simple-to-use Logical Database Diagram tool from Lucidchart, I was able to construct a logical database diagram that will help to visualize, comprehend, and enhance the database architecture. I was able to design database diagrams with personalized text, layout, and field relationships. Diagramming was simple thanks to their shape collection. Numerous available shapes, plus I could use imported and customized shapes to make my own shape libraries.

4.11.5 Tools and Techniques that can use to create Class Diagram

- Lucidchart
- Gleek.io
- Diagrams.net
- Cacoo
- Gliffy
- EdrawMax
- Microsoft Visio Pro

4.11.6 Tools and Techniques I used to create Class Diagram

I was able to make a Class Diagram using the Lucidchart Class Diagram tool because it made it easier to see the links between classes, map out a system's structure, and utilize UML shape libraries. I was able to quickly begin designing class diagrams thanks to Lucidchart's UML diagram templates, which feature connectors and standardized notation forms.

4.11.7 Tools and Techniques that can use to create Use Case Diagram

- Creately
- Visual Paradigm
- SINVAS UML Community Edition
- ClickCharts
- AgroUML
- WhiteStarUML
- Astah Community
- Violet

4.11.8 Tools and Techniques I used to create Use Case Diagram

I was able to rapidly and efficiently generate use case diagrams using Lucidchart software, in addition to other kinds of UML diagrams. All of the UML elements had labels and were located on the interface's right side. As a result, I had no trouble understanding how to use each element. To create a use case diagram, I had to first choose the Use Case Diagram section to open a special window with all the necessary components. After creating the structure on the canvas, I simply added toolbar items and double-clicked each one to give it a label and add the necessary titles.

4.11.9 Tools and Techniques that can use to create Data Flow Diagram

- EdrawMax
- Miro
- Visual Paradigm Online
- Creately
- Diagrams.net
- FigJam

4.11.10 Tools and Techniques I used to create Data Flow Diagram

I was able to make a data flow diagram utilizing the data flow diagram tool from Lucidchart since it helps to take any information flow for a process or system and compress it into a logical, understandable graphic. I was able to visually illustrate complex processes and systems so that even the least tech minded person could grasp them by using standardized circles, arrows, and text labels. Additionally, whether I was creating my data flow diagram from scratch or adapting one of their many DFD templates, Lucidchart made it simpler than ever to get started.

4.11.11 Tools and Techniques that can use to create Flow Chart Diagram

- Microsoft Visio
- Textografo
- Lucidchart
- diagrams.net
- SmartDraw
- Whimsical
- ConceptDraw Diagram

4.11.12 Tools and Techniques I used to create Flow Chart Diagram

Since Draw.io's diagram tool has a variety of templates I can use for different types of flow diagrams, I was able to make a flow chart diagram using it. I could alter one of these to fit my workflow or use them as examples to show what is possible. This software comes with its flowcharts shape library turned on by default. I was able to add each shape from the shape library as needed to the drawing area and draw the relationships between them.

4.11.13 Tools and Techniques that can use to create Activity Diagram

- Diagrams.net
- Lucidchart
- GenMyModel
- yUML
- Miro
- SmartDraw
- Moqups
- Creately

4.11.14 Tools and Techniques I used to create Activity Diagram

Using Creately, which offers purpose-built form libraries to precisely portray how systems act, I was able to create an Activity Diagram. With the aid of this software, intuitive modeling was possible with shapes that are adaptive and change depending on the environment in which they are used. This tool's basic drag and drop tools made it simple to construct activity diagrams, which was helpful.

4.11.15 Tools and Techniques that can use to create Sequence Diagram

- GitMind
- Lucidchart
- Visual Paradigm
- StarUML
- Gliffy
- Creately
- UMLetino

4.11.16 Tools and Techniques I used to create Sequence Diagram

Due to Drawio's diagram tool's assistance in editing the markup to update a diagram that was already on the page, I was able to build a sequence diagram. Since I was creating a simple design, I could drag and drop shapes from their entire UML library. This software made it easy for me to edit the sequence diagram once it was created.

4.11.17 Tools and Techniques that can use to create Wireframe Diagram

- Figma
- Adobe XD
- Sketch
- Balsamiq Wireframes
- Justinmind
- Lucidchart
- MockFlow
- Moqups

4.11.18 Tools and Techniques I used to create Wireframe Diagram

Using the Figma diagram tool, I was able to construct a wireframe diagram. I was putting the system's blueprints together when I created wireframes. Figma made it simple to investigate how my website should operate to create the prototype from a blueprint. With pre-built wireframe website elements like navigation bars, features, headers, buttons, and more, I was able to save time by using Figma.

4.11.19 Tools and Techniques that can use to create Prototype Diagram

- Figma
- Adobe XD
- Framer
- Mockplus
- Vireflow
- Photoshop

4.11.20 Tools and Techniques I used to create Prototype Diagram

I was able to produce prototype diagrams thanks to Figma's diagram tool because it allowed me to develop my wireframes into prototypes without writing any code. I was able to convert my wireframes into clickable, interactive prototypes directly from my browser using Figma. Using Figma's plug-ins and community-shared documents, I was able to add interactivity to my wireframes without having to write any code to convert them into prototypes.

4.12 System Feedback

The screenshot shows a Google Forms survey titled "USER FEEDBACK FORM". The form is designed to evaluate and enhance the system of Polly Pipe in Braintree. It includes fields for name, department, and login ease, and rating scales for system speed and security. The form is currently being viewed by a user named "ryandilthusha@gmail.com".

USER FEEDBACK FORM

Fill this form evaluate and to enhance the the system of Polly Pipe in Braintree. Your honest answer is highly expected.
If you have any issues contact us by ryandilthusha@gmail.com

ryandilthusha@gmail.com (not shared) [Switch account](#)

Write your name ?
Your answer _____

What is your department ?
Your answer _____

It is easy to log into the system ?

YES
 NO

How do you rate the speed of the system ?

1 2 3 4 5
Very Slow Vert Fast

How do you rate the security of the system ?

1 2 3 4 5
Very Unsecure Very Secure

Figure 4. 46 Feedback form part 1

USER FEEDBACK FORM

How do you rate the security of the system ?

1 2 3 4 5

Very Unsecure Very Secure

Is it easy to access authorized forms ?

Yes
 No
 Sometimes having errors

Does this system design help you to increase work efficiency ?

Yes
 No

If above answer is "NO", what is the reason ?

Your answer

How satisfied are you with our product ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied were you with the instructions before use ?

1 2 3 4 5

Figure 4. 47 Feedback form part 2

USER FEEDBACK FORM

How satisfied were you with the instructions before use ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied is the quality of the product ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How easy is it to find data from the database

1 2 3 4 5

Very Hard Very Easy

How would you rate the design of the system ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied are you with our service ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied are you with our team?

!  

Figure 4. 48 Feedback form part 3

USER FEEDBACK FORM

How would you rate the design of the system?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied are you with our service ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied are you with our team?

1 2 3 4 5

Very Unsatisfied Very Satisfied

Would you like to a representative to contact you ?

Yes
 No

Tell us how we can improve ?

Your answer

Submit Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Figure 4. 49 Feedback form part 4

4.13 User Documentation for E-Solutions Pvt Ltd Project Management System

Contents

- 1) System Description
- 2) System References
- 3) System Features
- 4) System Requirements
- 5) User Interfaces

1) System Description

This system is made for Staff Members (Including Team Leaders), Project Manager and Project Director in the E-Solutions Pvt Ltd to use. I've provided login credentials to access the system hence it enhances the security of the system. There are 3 different types of dashboards for each of the 3 user types. Additionally, the system's functionality and user interfaces vary slightly depending on the type of user. For example, a Staff Member can only add an invoice by invoice page, however the Project Manager can see all invoices and change them. However, the Project Director can only view the company's detailed revenue and costs.

2) System References

I used latest platforms, plugins and technologies when designing this system

- Figma
- Draw io
- Lucidchart
- Creatly

3) System Features

This system is coming with small file containing the database base, source code along with the setup. User can easily install the setup by double clicking the “SETUP.exe” file to any location user want.

4) System Requirements

This system can be installed to any basic computers which having Windows Operating System.

5) User Interfaces

Login Page →

After loading the system 1st thing all users can see is the login form. User should give login credentials in the Username and Password fields which system developers have given separately. After entering login credentials user can simply enter to the system by clicking Login button. Users can continue using the Forget Password link if they are unable to recall their username and password. Users who are new to the system can register by clicking the Register button.

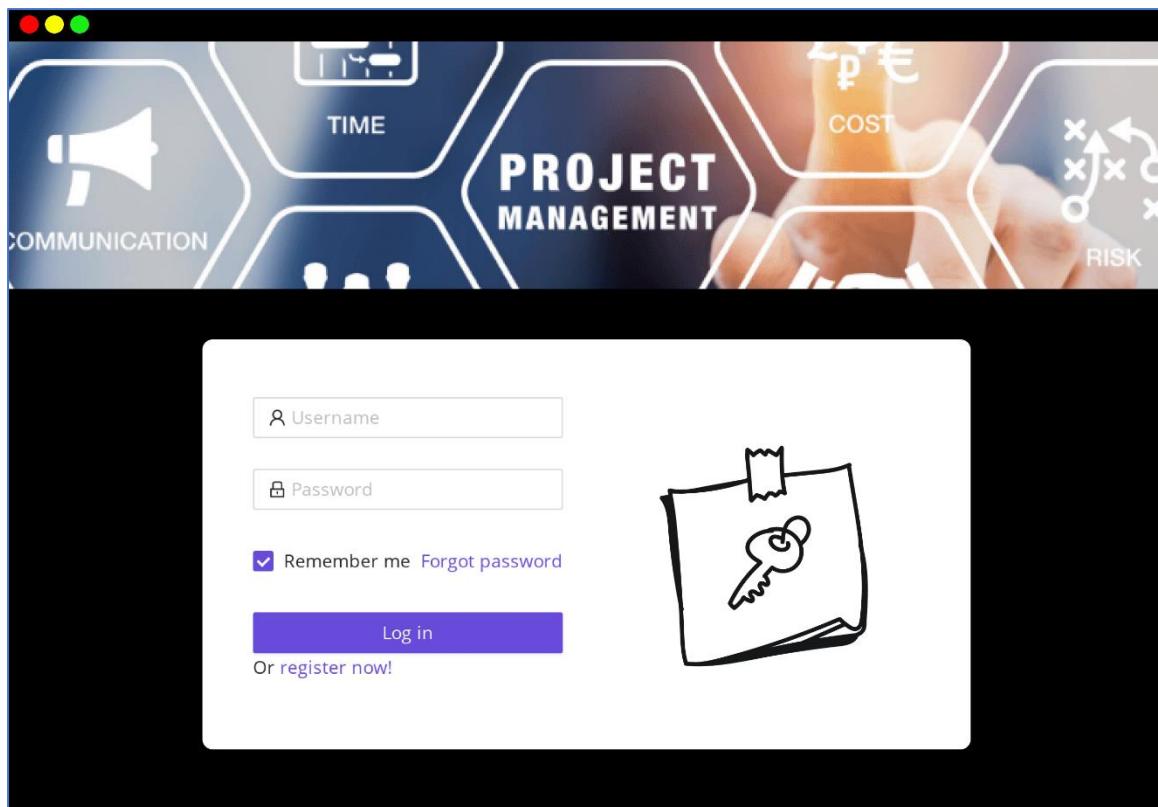


Figure 4. 50 Login Page Interface

Home Page →

- After giving correct login credential user directed to home page.
- User can surf through this interface and can see the dashboard that giving vital information about the running projects (For each type of user this can be different)
- User can navigate details by scrolling down.
- User can go to edit profiles by clicking “Edit Profiles” button.
- User can go back to the Login Page by clicking the Logout button which placed in top right corner.

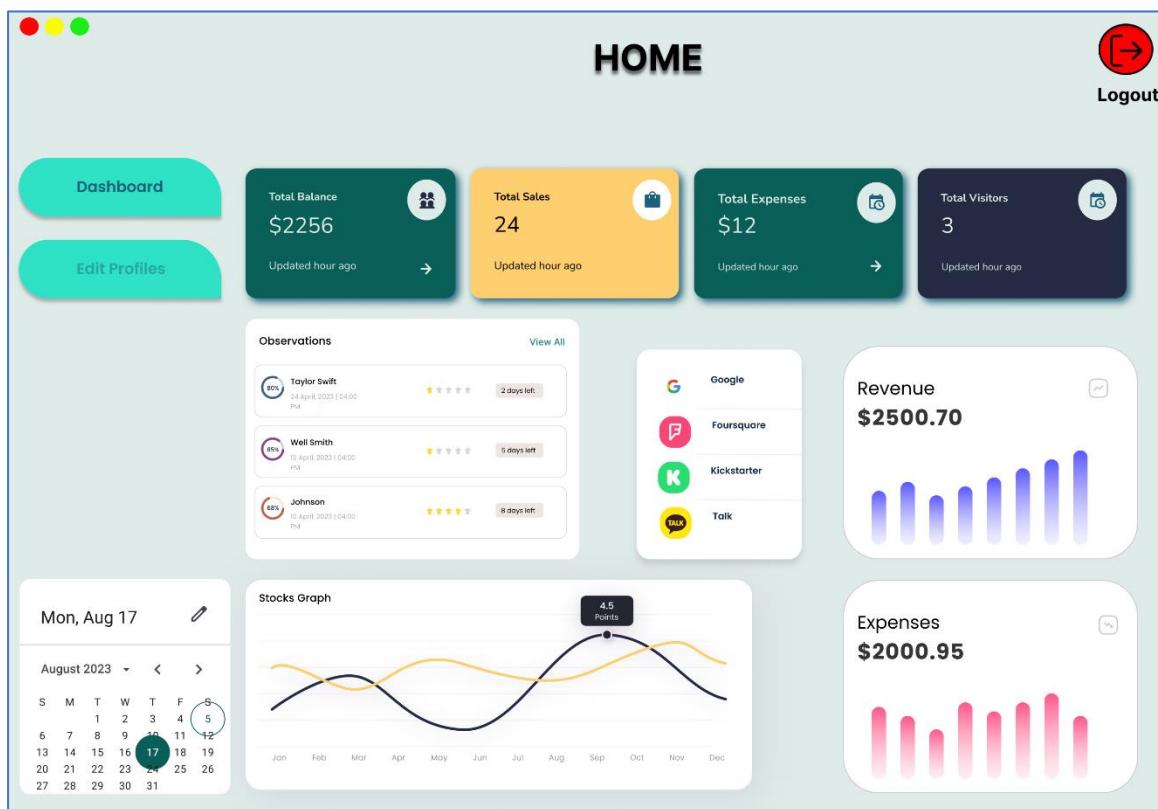


Figure 4. 51 Home Page Interface

Employee Details Page →

- User can go to this form by clicking on the “Edit Profiles” button at the Home Page where we can see Dashboard or by navigation bar.
- To save new records user must click on “Submit” button. User can simply save new records by filling the text box fields and hitting the Submit button.
- When updating a specific record, user should click Edit button on the related row. User can make changes on the fields then by hitting the Edit button the record will be updated.
- When deleting a specific record, user should click on the Delete button on the related row.
- User can go back to home page by clicking the Home button at the top right corner or through navigation bar at the left.
- User can exit from the system by click on the Signout button.
- User can navigate through each page by clicking this links on the navigation bar at the left.

The screenshot shows the 'Employee Details' page interface. The left sidebar is blue and contains the following navigation items:

- E-Solutions**
- Edit Profiles**
- Employee Details** (highlighted)
- Project Profile**
- Invoicing**
- Team Details**
- Settings**
- Sign out**

The main content area is titled "Employee Details". It contains the following form fields:

Employee ID	<input type="text"/>
Name	<input type="text"/>
Phone	<input type="text"/>
Address	<input type="text"/>
Email	<input type="text"/>
Designation	<input type="text"/>
Salary	<input type="text"/>
Assigned Projects	<input type="text"/>

At the bottom of the page are three buttons: "Submit", "Edit", and "Delete".

Figure 4. 52 Employee Details Page Interface

Project Profiles Page →

- User can go to this form by clicking on the “Edit Profiles” button at the Home Page where we can see Dashboard or by navigation bar.
- To save new records user must click on “Submit” button. User can simply save new records by filling the text box fields and hitting the Submit button.
- When updating a specific record, user should click Edit button on the related row. User can make changes on the fields then by hitting the Edit button the record will be updated.
- When deleting a specific record, user should click on the Delete button on the related row.
- User can go back to home page by clicking the Home button at the top right corner or through navigation bar at the left.
- User can exit from the system by click on the Signout button.
- User can navigate through each page by clicking this links on the navigation bar at the left.

The screenshot shows the 'Project Profile' page interface. On the left, there is a vertical sidebar with a blue background containing the following navigation items:

- E-Solutions**
- Edit Profiles**
- Employee Details**
- Project Profile** (highlighted with a green icon)
- Invoicing**
- Team Details**
- Settings**
- Sign out**

The main content area is titled "Project Profile". It contains the following form fields:

Project ID	<input type="text"/>	Project Name	<input type="text"/>
Category	<input type="text"/>	Project Manager	<input type="text"/>
Start Date	<input type="text"/>	End Date	<input type="text"/>
Task List	<input type="text"/>		
Project Cost	<input type="text"/>	Employee Cost	<input type="text"/>
Project Details	<input type="text"/>		
Remarks	<input type="text"/>		

At the bottom of the page are three buttons: "Submit", "Edit", and "Delete".

Figure 4. 53 Project Profile Page Interface

Invoicing Page →

- User can go to this form by clicking on the “Edit Profiles” button at the Home Page where we can see Dashboard or by navigation bar.
- To save new records user must click on “Submit” button. User can simply save new records by filling the text box fields and hitting the Submit button.
- When updating a specific record, user should click Edit button on the related row. User can make changes on the fields then by hitting the Edit button the record will be updated.
- When deleting a specific record, user should click on the Delete button on the related row.
- User can go back to home page by clicking the Home button at the top right corner or through navigation bar at the left.
- User can exit from the system by click on the Signout button.
- User can navigate through each page by clicking this links on the navigation bar at the left.
- User can calculate the Total Cost by clicking the “Calculate” button.

The screenshot shows the 'Invoicing' page interface. On the left, there's a sidebar with a blue background containing the 'E-Solutions' logo and several navigation links: 'Employee Details', 'Project Profile', 'Invoicing' (which is currently selected and highlighted in blue), 'Team Details', 'Settings', and 'Sign out'. The main content area has a light gray background and is titled 'Invoicing' in bold. It contains the following fields:

- 'Invoice ID' with an input field.
- 'Project Name' with an input field.
- 'Project Cost' with an input field and a checked checkbox.
- 'Employee Cost' with an input field and an unchecked checkbox.
- 'Additional Cost' with an input field and an unchecked checkbox.
- A green 'Calculate' button.
- 'Total Bill' with an input field.
- 'Remarks' with an input field.
- At the bottom, there are three buttons: 'Submit' (yellow), 'Edit' (yellow), and 'Delete' (yellow).

Figure 4. 54 Invoicing Page Interface

Team Details Page →

- User can go to this form by clicking on the “Edit Profiles” button at the Home Page where we can see Dashboard or by navigation bar.
- To save new records user must click on “Submit” button. User can simply save new records by filling the text box fields and hitting the Submit button.
- When updating a specific record, user should click Edit button on the related row. User can make changes on the fields then by hitting the Edit button the record will be updated.
- When deleting a specific record, user should click on the Delete button on the related row.
- User can go back to home page by clicking the Home button at the top right corner or through navigation bar at the left.
- User can exit from the system by click on the Signout button.
- User can navigate through each page by clicking this links on the navigation bar at the left.

Figure 4. 55 Team Details Page Interface

Project Profiles Page →

- User can go to this form by clicking on the “Edit Profiles” button at the Home Page where we can see Dashboard or by navigation bar.
- When updating a specific record, user should click record on the related row. User can make changes on the fields then by hitting the Enter button the record will be updated easily.
- When deleting a specific record, user should click on the Delete button on the selected row.
- User can go back to home page by clicking the Home button at the top right corner or through navigation bar at the left.
- User can exit from the system by click on the Signout button.
- User can navigate through each page by clicking this links on the navigation bar at the left.

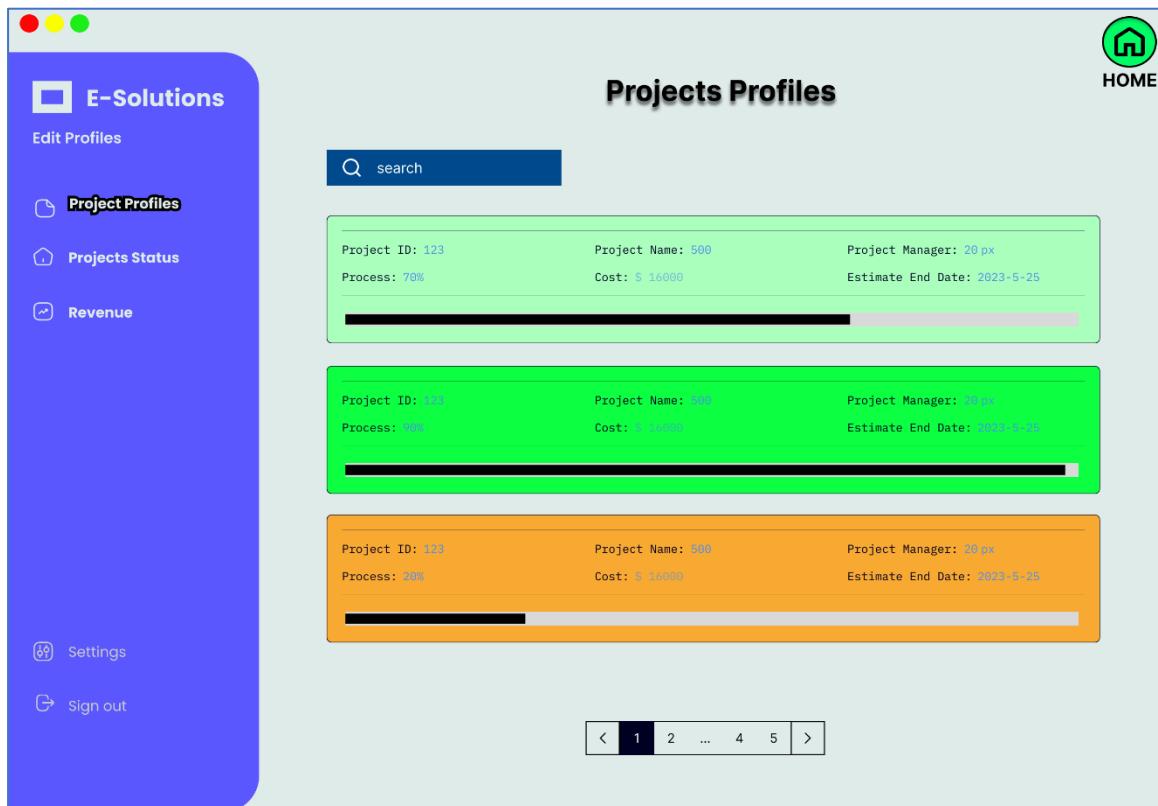


Figure 4. 56 Project Profiles Page Interface

Projects Status Page →

- User can go to this form by clicking on the “Edit Profiles” button at the Home Page where we can see Dashboard or by navigation bar.
- At here only user can view the data and analyze them.
- User can go back to home page by clicking the Home button at the top right corner or through navigation bar at the left.
- User can exit from the system by click on the Signout button.
- User can navigate through each page by clicking this links on the navigation bar at the left.

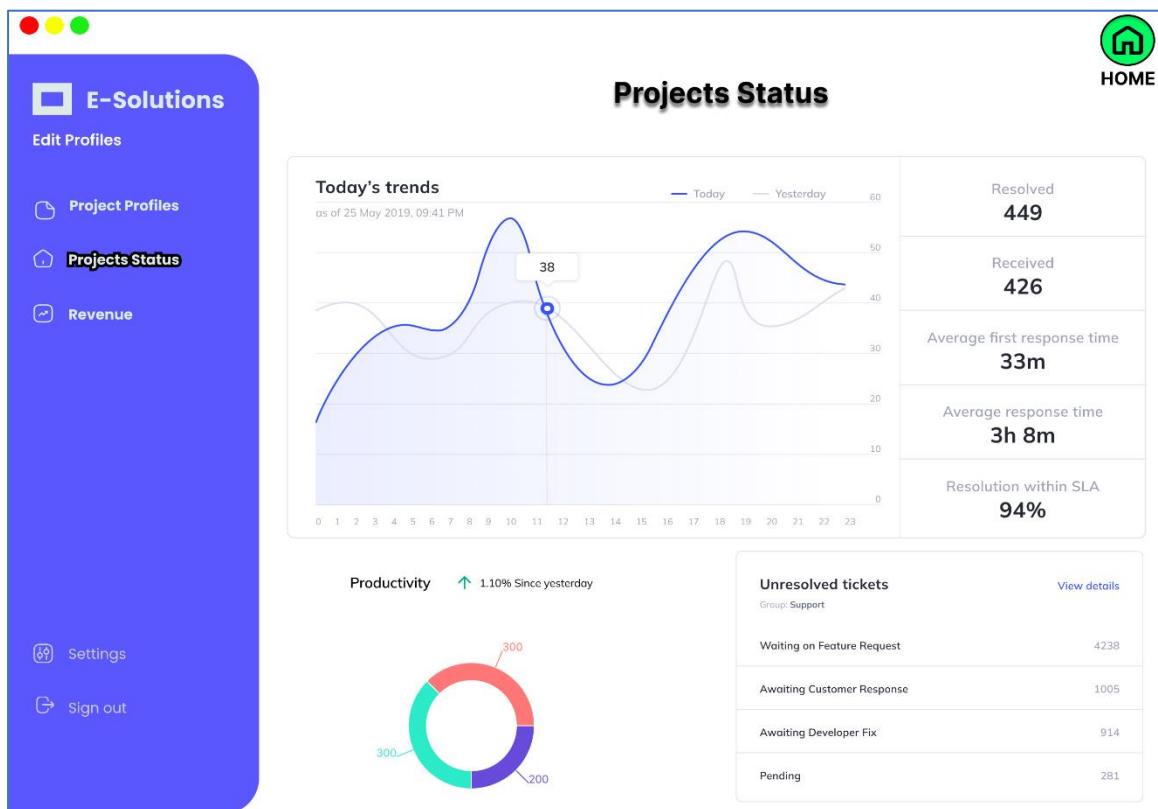


Figure 4. 57 Projects Status Page Interface

Revenue Page →

- User can go to this form by clicking on the “Edit Profiles” button at the Home Page where we can see Dashboard or by navigation bar.
- When updating a specific record, user should click record on the related row. User can make changes on the fields then by hitting the Enter button the record will be updated easily.
- When deleting a specific record, user should click on the Delete button on the selected row.
- User can go back to home page by clicking the Home button at the top right corner or through navigation bar at the left.
- User can exit from the system by click on the Signout button.
- User can navigate through each page by clicking this links on the navigation bar at the left.

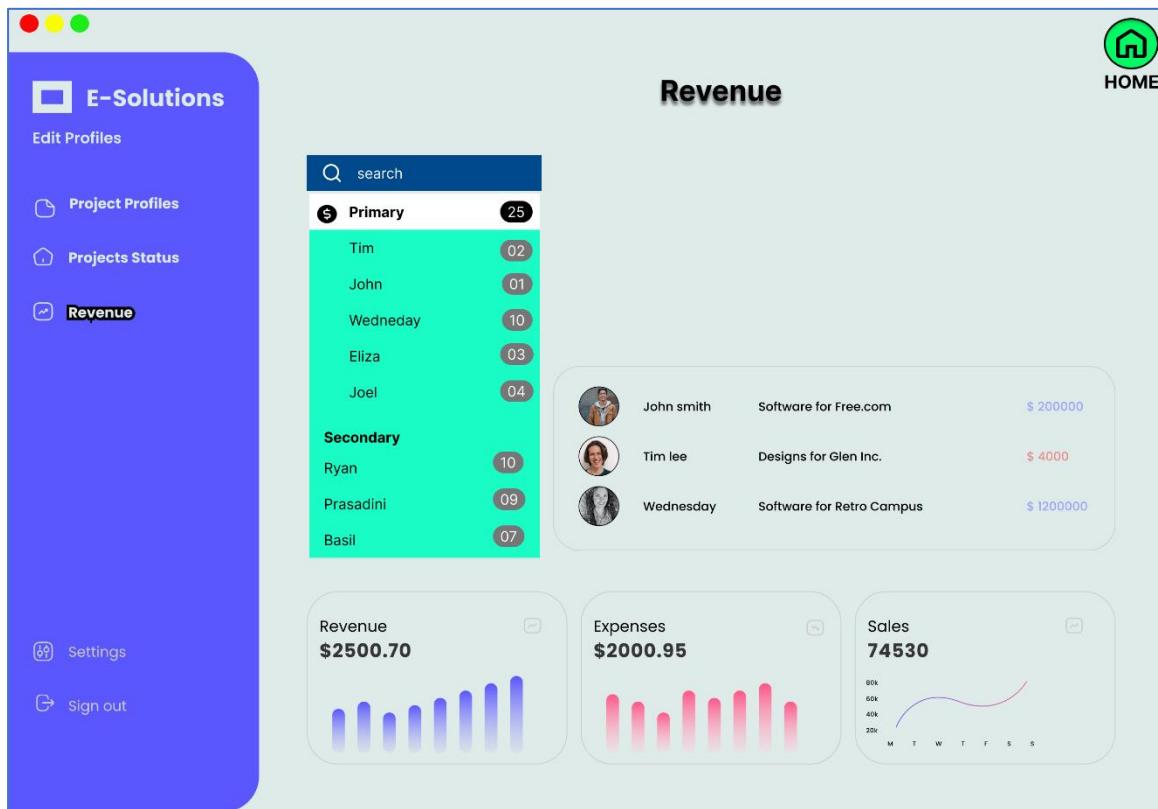


Figure 4. 58 Revenue Page Interface

4.13 Conclusion to Activity 4

It's essential to know how to design systems by using the theories we are familiar with. There are ten different sorts of system designs provided in this chapter. Entity Relationship Design, Logical Database Design, and Class Diagram have similar flows in them. The Use Case Diagram, Data Flow Diagram, Flow Chart Diagram, Activity Diagram and Sequence Diagram have similar characteristics. And Wireframe designs and Prototypes follow the same flow. So, it was easier to design these diagrams by understanding their connections. By drawing the ERD diagram, I was able to develop other all diagrams by following its data flows and entities.

Conclusion

This entire assignment is based on the implementation of a Project Management system design for the selected company (E-Solutions Pvt Ltd). The purpose of this assignment is to improve Systems Analysis skills & Designing skills.

This report includes basic system analysis and designing theories with the System Development Life Cycle details relating to the given scenario. System designing methodologies, business case and feasibility study, investigation techniques and tools to investigate, user and system requirements, documentation for different phases, design elements and features, and designing techniques and tools have been discussed in this assignment.

References

Systems Analysis and Design (SAD) Tutorial. [no date].

Available at: <https://www.w3computing.com/systemsanalysis/>

[Accessed: 1 January 2023].

7 Phases of the System Development Life Cycle Guide. 2001.

Available at: <https://www.clouddefense.ai/blog/system-development-life-cycle#:~:text=The%20new%20seven%20phases%20of,testing%2C%20implementation%2C%20and%20maintenance.>

[Accessed: 1 January 2023].

Your Trusted IT Consulting & Software Development Partner | Improving. [no date].

Available at: <https://improving.com>

[Accessed: 1 January 2023].

Types of Software Development Methodologies and Their Pros and Cons. 2021.

Available at: <https://www.innuy.com/blog/types-of-software-development-methodologies-and-their-pros-and-cons/>

[Accessed: 1 January 2023].

News Team, K.P. [no date]. Traditional vs. Agile Software Development Methodologies.

Available at: <https://www.kkipartners.com/blog/traditional-vs-agile-software-development-methodologies>

[Accessed: 1 January 2023].

Software Engineering | Classical Waterfall Model - GeeksforGeeks. 2018.

Available at: <https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/>

[Accessed: 1 January 2023].

When should you use Waterfall Model - GeeksforGeeks. 2020.

Available at: <https://www.geeksforgeeks.org/when-should-you-use-waterfall-model/>

[Accessed: 1 January 2023].

Software Engineering | Prototyping Model - GeeksforGeeks. 2018.

Available at: <https://www.geeksforgeeks.org/software-engineering-prototyping-model/>

[Accessed: 1 January 2023].

Software Engineering | Phases of Prototyping Model | Set - 2 - GeeksforGeeks. 2018.

Available at: <https://www.geeksforgeeks.org/software-engineering-phases-prototyping-model-set-2/>

[Accessed: 1 January 2023].

Prasanna 2022. Advantages and Disadvantages of Prototype Model | Prototyping Model in Software Engineering for Testing.

Available at: <https://www.aplustopper.com/advantages-and-disadvantages-of-prototype-model/>

[Accessed: 1 January 2023].

What is Prototype model- advantages, disadvantages and when to use it? 2020.

Available at: <https://tryqa.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/>

[Accessed: 3 January 2023].

Software Engineering | Spiral Model - GeeksforGeeks. 2018.

Available at: <https://www.geeksforgeeks.org/software-engineering-spiral-model/>

[Accessed: 3 January 2023].

What is Spiral Model and How is it Used? 2019.

Available at: <https://www.techtarget.com/searchsoftwarequality/definition/spiral-model>

[Accessed: 3 January 2023].

Difference between Prototype Model and Spiral Model - GeeksforGeeks. 2020.

Available at: <https://www.geeksforgeeks.org/difference-between-prototype-model-and-spiral-model/>

[Accessed: 3 January 2023].

The History of Agile. [no date].

Available at: <https://www.planview.com/resources/guide/agile-methodologies-a-beginners-guide/history-of-agile/>

[Accessed: 3 January 2023].

Geneca 2020. What is Agile Software Development? Agile Principles and Values ★ Geneca.

Available at: <https://www.geneca.com/what-is-agile-software-development/>

[Accessed: 3 January 2023].

What is AGILE? - What is SCRUM? - Agile FAQ's | Cprime. [no date].

Available at: <https://www.cprime.com/resources/what-is-agile-what-is-scrum/>

[Accessed: 3 January 2023].

Scrum Testing Methodology Tutorial. 2020.

Available at: <https://www.guru99.com/scrum-testing-beginner-guide.html>

[Accessed: 3 January 2023].

Arvind 2019. What is Agile methodology | Agile Model and methodology | Edureka.

Available at: <https://www.edureka.co/blog/what-is-agile-methodology/>

[Accessed: 3 January 2023].

What Kind Of Projects Can Be Executed Using Scrum - Weekly Sharing - ZenTao. [no date].

Available at: <https://www.zentao.pm/blog/what-kind-of-projects-executed-using-scrum-1051.html>

[Accessed: 3 January 2023].

What Is Scrum Methodology? & Scrum Project Management. 2019.

Available at: <https://www.digitel.com/agile/scrum-methodology/>

[Accessed: 7 January 2023].

Scrum Project Management: Advantages and Disadvantages. 2013.

Available at: <https://www.simplilearn.com/scrum-project-management-article>

[Accessed: 7 January 2023].

Extreme Programming - Introduction. [no date].

Available at:
https://www.tutorialspoint.com/extreme_programming/extreme_programming_introduction.htm

[Accessed: 7 January 2023].

Software Engineering | Extreme Programming (XP) - GeeksforGeeks. 2018.

Available at: <https://www.geeksforgeeks.org/software-engineering-extreme-programming-xp/>

[Accessed: 7 January 2023].

Kukhnavets, P. 2018. Disadvantages and Advantages in Extreme Programming.

Available at: <https://hygger.io/blog/disadvantages-and-advantages-of-extreme-programming/>

[Accessed: 7 January 2023].

Agile Methodology: Advantages and Disadvantages. 2022.

Available at: <https://ccaps.umn.edu/story/agile-methodology-advantages-and-disadvantages>

[Accessed: 7 January 2023].

Advantages and Disadvantages of Agile Project Management [Checklist] · ActiveCollab. 2017.

Available at: <https://activecollab.com/blog/project-management/Agile-project-management-advantages-disadvantages>

[Accessed: 7 January 2023].

@appmaster_io [no date]. Traditional vs Agile Project Management Methods - Which One fits You? | AppMaster.

Available at: <https://appmaster.io/blog/traditional-vs-agile>

[Accessed: 7 January 2023].

admin 2019. 11 Advantages & 9 Disadvantages of Traditional Project Management.

Available at: <https://wadic.net/traditional-project-management-advantages-disadvantages/>

[Accessed: 7 January 2023].

Advantages and Disadvantages of Agile Project Management [Checklist] · ActiveCollab. 2017.

Available at: <https://activecollab.com/blog/project-management/Agile-project-management-advantages-disadvantages>

[Accessed: 7 January 2023].

admin 2019. 11 Advantages & 9 Disadvantages of Traditional Project Management.

Available at: <https://wadic.net/traditional-project-management-advantages-disadvantages/>

[Accessed: 9 January 2023].

Putnam, E. 2022. What is a Business Case and Why Would You Need One? (Video).

Available at: <https://edison365.com/what-is-a-business-case-and-why-would-you-need-one/>

[Accessed: 9 January 2023].

Bridges, J. 2022. How to Write a Business Case.

Available at: <https://www.projectmanager.com/blog/how-to-write-a-business-case>

[Accessed: 9 January 2023].

Gaille, B. 2016. 14 Pros and Cons of a Business Plan.

Available at: <https://brandongaille.com/14-pros-and-cons-of-a-business-plan/>

[Accessed: 9 January 2023].

Desirability, viability and feasibility diagram: What does it mean? - UX Design Institute. 2022.

Available at: <https://www.uxdesigninstitute.com/blog/desirability-viability-and-feasibility/>

[Accessed: 9 January 2023].

What is Feasibility Study in Project Management and Its Types? | Simplilearn. 2012.

Available at: <https://www.simplilearn.com/feasibility-study-article>

[Accessed: 10 January 2023].

Bridges, J. 2021. What Is a Feasibility Study? How to Conduct One for Your Project.

Available at: <https://www.projectmanager.com/training/how-to-conduct-a-feasibility-study>

[Accessed: 10 January 2023].

Top 4 Techniques of Preliminary Survey and Feasibility Study. 2015.

Available at: <https://www.yourarticlerepository.com/management/mis-management/top-4-techniques-of-preliminary-survey-and-feasibility-study/70409>

[Accessed: 10 January 2023].

Limited, D. [no date]. Docular - Legal documents.

Available at: <https://docular.net/>

[Accessed: 10 January 2023].

Parker, J. 2012. Business, User, and System Requirements - Enfocus Solutions Inc.

Available at: <https://enfocussolutions.com/business-user-and-system-requirements/>

[Accessed: 10 January 2023].

Kashun, M. [no date]. Requirement Gathering Methods.

Available at: https://www.academia.edu/36844343/Requirement_Gathering_Methods

[Accessed: 22 January 2023].

6.3Fundamental Requirements Gathering Techniques – Wachemo University e-Learning Platform. [no date].

Available at: <https://wachemo-elearning.net/courses/system-analysis-and-design/lessons/chapter-6-gathering-user-requirements/topic/6-3-fundamental-requirements-gathering-techniques/>

[Accessed: 22 January 2023].

Communication, T.B. 2013. Advantage and disadvantage of interview.

Available at: <https://thebusinesscommunication.com/advantage-and-disadvantage-of-interview/>

[Accessed: 22 January 2023].

ReQtest 2012. How to Use Interviews to Gather Requirements | ReQtest.

Available at: <https://reqtest.com/requirements-blog/how-to-use-interviews-to-gather-requirements/>

[Accessed: 22 January 2023].

What is the Difference Between Structured and Unstructured Interviews. [no date].

Available at: <https://intromagic.io/difference-between-structured-and-unstructured-interviews/>

[Accessed: 22 January 2023].

JAD Survey Method: Requirements Gathering Techniques. [no date].

Available at: <https://www.directutor.com/content/survey-method-requirements-gathering-techniques>

[Accessed: 22 January 2023].

Learnovate, M. 2022. Open Ended vs Closed Ended Questions - MIM Learnovate.

Available at: <https://mimlearnovate.com/open-ended-vs-closed-ended-questions/>

[Accessed: 22 January 2023].

Corrales, E. 2021. Trello Review: Features, Pros, and Pricing | Developer.com.

Available at: <https://www.developer.com/project-management/trello-review/>

[Accessed: 22 January 2023].

Jafari, L. 2020. What Are Non-Functional Requirements? Types and Examples - WINaTALENT | Blog.

Available at: <https://winatalent.com/blog/2020/05/what-are-non-functional-requirements-types-and-examples/>

[Accessed: 22 January 2023].

Non-Functional Requirement Examples. [no date].

Available at: <https://www.linkedin.com/pulse/non-functional-requirement-examples-dan-hughes>

[Accessed: 22 January 2023].

Participant Observation and Non-Participant Observation. 2015.

Available at: <https://www.yourarticlerepository.com/social-research/data-collection/participant-observation-and-non-participant-observation/64510>

[Accessed: 30 January 2023].

Advantages & Disadvantages of a Focus Group. [no date].

Available at: <https://smallbusiness.chron.com/advantages-disadvantages-focus-group-784.html>

[Accessed: 30 January 2023].

<https://www.umsl.edu/~sauterv/analysis/F2015/Requirement%20Gathering%20Methods.html.htm>. [no date].

Available at: <https://www.umsl.edu/~sauterv/analysis/F2015/Requirement%20Gathering%20Methods.html.htm>

[Accessed: 30 January 2023].

Wasik, R. 2017. The Advantages & Disadvantages of Prototyping - Rapids Reproductions.

Available at: <https://rapidsrepro.com/advantages-disadvantages-prototyping/>

[Accessed: 30 January 2023].

Agile Project Management Vs. Traditional Project Management. [no date].

Available at: <https://www.knowledgehut.com/blog/agile/agile-project-management-vs-traditional-project-management>

[Accessed: 30 January 2023].

Logical Databases- Advantages and Disadvantages. [no date].

Available at: http://www.stechies.com/logical-database_1/

[Accessed: 30 January 2023].

Borgida, A., Casanova, M.A. and F. Laender, A.H. [no date]. Logical Database Design: from Conceptual to Logical Schema.

Available at: https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_645

[Accessed: 30 January 2023].

Hill, A. 2020. UML Class Diagrams – Definition, Attributes, Benefits and Process - My Chart Guide.

Available at: <https://mychartguide.com/uml-class-diagrams/>

[Accessed: 30 January 2023].

Holland, K. 2018. What are the Advantages of Use Case Diagrams?

Available at: <https://medium.com/@katie.holland.runs.fast/what-are-the-advantages-of-use-case-diagrams-2f9b47387574>

[Accessed: 30 January 2023].

How a UML use case diagram can benefit any process - Learn Wordpress. [no date].

Available at: <https://nulab.com/learn/software-development/how-a-uml-use-case-diagram-can-benefit-any-process/>

[Accessed: 30 January 2023].

What is Data Flow Diagram? [no date].

Available at: <https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/>

[Accessed: 30 January 2023].

What is a Flowchart. [no date].

Available at: <https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial>

[Accessed: 30 January 2023].

An introduction to Flowcharts - GeeksforGeeks. 2017.

Available at: <https://www.geeksforgeeks.org/an-introduction-to-flowcharts/>

[Accessed: 30 January 2023].

Unified Modeling Language (UML) | Activity Diagrams - GeeksforGeeks. 2017. Available at:

<https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/>

[Accessed: 30 January 2023].

Sequence Diagrams - What is a Sequence Diagram? [no date].

Available at: <https://www.smartdraw.com/sequence-diagram/>

[Accessed: 30 January 2023].

Oliver, K. 2022. The Importance & benefits of Wireframing.

Available at: <https://www.zebedeecreations.com/blog/the-importance-of-wireframing/>

[Accessed: 30 January 2023].

Prototype Design Pattern - GeeksforGeeks. 2017.

Available at: <https://www.geeksforgeeks.org/prototype-design-pattern/>

[Accessed: 30 January 2023].

Athuraliya, A. and posts by Amanda Athuraliya →, V. all 2019. The Easy Guide to Prototyping | Prototyping Types and Process.

Available at: <https://creately.com/blog/diagrams/what-is-prototyping-techniques-process/>

[Accessed: 30 January 2023].