INSTITUTE OF TECHNOLOGY
SCHOOL OF INFORMATICS

Module for Industrial Project I

Compiled By: Ayenew Yifru
Natnael Gonfa
31-Oct-17

Table of Contents

Preface

This document is a generic Final Project I (Industrial Project I) document for use by all first degree Computer Science, Information Science and Information Technology Students for their final project's. It provides guidance *and template material which is intended to assist the relevant outline to prepare documentations for software and network related final projects in the school. It describes detail outlines of contents and their descriptions with necessary examples that helps students to properly document their project progresses in their final year senior project. It is also useful background reading for anyone involved in developing or monitoring the final year student projects in Hawassa University or in other universities.*

SOFTWARE DEVELOPMENT OUTLINE

CHAPTER ONE

INTRODUCTION

Many students think that choosing their final project topic is the most exciting part of their studies in the university [3]. After all; this is something that they get to decide for themselves rather than having to complete a task decided by their tutors. We will stress in this chapter that it is important to choose something that will sustain your interest throughout the months that you will need to complete it. You may even decide to do some project work that is something that forms part of your leisure activities!

Before you start your final project, you need to have at least some idea of what you want to do. This is probably the most difficult, and yet the most important, part of your final year project. Up until now most of your studies have been concerned with answering questions that other people have set in any of courses you were taken already in the university. This chapter is concerned with how to formulate and clarify your final project (research) topic and your final project question. Without being clear about what you are, going to do in your final project it is difficult to plan how you are going to do it.

This chapter introduces the users as well as advisor and/or examiner about the general introduction about system[1]. The chapter can be organized with a number of sub-sections. It begins with background of the study then after objectives of the project (both general and specific objectives), and significance of the study are followed. The chapter also include scope and limitations of the study, methodology (data collection, system analysis and design, system development, testing and deployment methodologies), development and working environments, feasibility studies (economic, technical and operational feasibilities), cost breakdown and in the last subsection, it deals with the schedule for the system.

[1] Sometimes this chapter of the documentation can be considered as a proposal for Industrial projects.

Background of the Study

This is the most important component of your project (report). Put your biggest effort into getting this perfect. Most advisors and examiners read the Abstract, Background of the study (Introduction) and Conclusions chapters of a project work first, and then they dive into the main body text afterwards. This means that you have to be particularly careful in wording these sections, since there is some content overlap. If you just copy and paste text between them, people will, notice and it will not leave them with a very favorable impression[2].

There are some specific rules related to these project (or technical report) components that you must know about. There are guidelines that are useful to knowbefore writing background of the study [1]:

- Themain one being the advice is do not to cut and paste text.
- Another is that you write this part of your documentation at the last time of your proposal or documentation. (You can be stubborn and write them first if you like; just be prepared to do this twice or more, because you will find this has to be completely rewritten in the end anyway.)

The fact that these are written last generally means they are often the most poorly written since most people naturally start to burn out as they approach the end of such a large writing project. However, keep in mind that these are the sections that will get the most attention and scrutiny; so you absolutely have to make them your best content in the document. Here is a general overview of how to write this important section, presented in the typical order in which they are written.

What goes in your 'Introduction'?

A good technical final project (thesis) Introduction does four things:

1. It introduces the problem and motivation for the study.
    a. Tell the reader what the topic of the report is.
    b. Explain why this topic is important or relevant.

[2]Many people read technical reports in the same order – in fact, some people actually never read anything but the Abstract, Introduction and Conclusions!

2. It provides a brief summary of previous engineering and/or scientific work on the topic.

Here you present an overview what is known about the problem. You would typically cite earlier studies conducted on the same topic and/or at this same site, and in doing so, you should reveal the yawning void in the knowledge that your brilliant research will fill.

If you are writing a project, you're going to need a full-blown literature review with very specific details of all of the scientific or engineering work done on the topic to date. In the introduction, just present a brief overview, sufficient to establish the need for your work[3].

3. It outlines the purpose and specific objectives of the project.

These are linked to solving the problem or filling the knowledge gap identified above.Often, the specific objectives are listed in point form. Sometimes a numbered list is used[4].

4. It provides a 'road map' for the rest of the report.

This is so that the reader knows what's coming and sees the logic of your organization.Describe (in approximately one sentence each) the contents of each of the report/project chapters.

What doesn't go in your Introduction?

Never put any results or decisions in the Introduction. Just because you are writing it last doesn't mean you should give away the story.

Statement of the Problem

A problem statement is a brief overview of the issues or problems existing in the concerned area selected for the research (project) [3]. It is an explanation of the issues prevalent in a particular sector which drives the researcher to take interest in that sector for in-depth study and analysis, so as to understand and solve them. In other words, a problem statement is a short description of

[3] Sometimes writers intentionally or unintentionally omitted to site references that are accessed for industrial (final) projects, but this is not a good habit.
[4] In this section you are expected to state your general objective in one of your paragraph that is found in it.

the issues that need to be addressed by a problem solving team and should be presented to them (or created by them) before they try to solve a problem.

Components of problem statement

The word count of problem statement for a project work should be in range of 150-300 words (one or two pages). The problem statement in any project works includes four important segments:

- **Background of the Problem**: Here you can reflect on facts related to the problem to make the reader understand about the gravity of the problem.
- **Anchor**: How one needs to resolve this problem in the project work[5].
- **General problem**: How is impacts a larger population (i.e. the general problem in day-to-day business transaction in the organization).
- **Specific problem**: How it impacts your sample population (i.e. the list of specific problem that are addressed in the proposed system, it is clear and measurable compared with general problem stated in the previous section).

For example:

- **Background of the problem:** The high attrition rate in manufacturing organization is creating anxiety and fear among the employees and thus affecting the productivity of the organization as a whole.

Here you need to refer to previous research done in the past in the manufacturing sector to determine the key reasons for high attrition rate. It should stimulate the reader to read further.

- **Anchor**: This must include a statistical value to magnify and elucidates the problem.

Here you can present the attrition percentage within the manufacturing industry and compare it with the case company.

[5] Sometimes students missed to clearly distinguished the above four segments of statement of the problem in their final year projects. There main focuses target on how they can state problems that are currently found in the existing system.

- **General Problem:** The general business problem is to determine the financial lost to the organization.

The general business problem needs to just outline the problem.

- Specific Problem: Since high attrition rate is affecting the overall productivity of the employees it is in turn affecting the performance of the organization. In order to do so one needs to determine the relationship between employee productivity and organizational performance.

This is narrower in scope than the general business problem and focused around need of the study which allows easy transition to Need of the Study.

General problems with problem statement

Quite often students are not able to frame their problem statement properly as they miss out on one or the other component or get confused on what to include or not. Most common problems which are observed have been highlighted below which will improve your ability to write problem statement:

- Unable to clearly identify the research problem.
- Often confused with research questions of the study (sometimes the students only write the existing problem but it is expected to write project questions).
- The problem is not encouraging enough for the researcher to read further.
- Not data driven i.e. NO citations.
- Not focused with the research subject.

Objectives of the Project

A crucial task in any project work is defining its core objectives or questions. What is the central goal or purpose of the research? What research topics, questions or problems does the project intend to address, and why?[6]

General Objective (aim)

The term project general objective (aim) usually refers to the main goal or overarching purpose of a research project. Sentence(s) stating the aim of a project are usually quite brief and to the point.

For Example: The general objective of the project will be:

- To investigate factors associated with partner violence.
- To design and implement Human Resources Management System for Tokuma Software PLC.

Because of their generality, project general objective is almost always positioned at the very beginning of a statement of research aims and objectives (or questions). They are broad and introductory rather than specific and focused.

Specific objectives

A project general objective will usually be followed by a series of statements describing a project's specific objectives. Specific objectives indicate in more detail the specific researchtopics or issues the project plans to investigate, building on the main theme stated inthe research aim. Normally at least two or three specific objectives will be stated. Itis good practice to put these in a numbered list (bullet list) so they can be clearly identified laterin a proposal or report. Here is an example of a set of specific objectives:

Objective 1: To examine whether alcohol consumption is associated with increased partner violence.

[6]Many projects get into difficulty because not enough time and thought is devoted at the start to properly defining the project's research goals. As a result, precious time and resources can be wasted collecting irrelevant or unnecessary research data.

Objective 2: To examine whether labor force status (employment, unemployment, not in the labor force) is associated with variations in the incidence of partner violence.

Objective 3: To explore differences between couples with an extended history of partner violence and couples with only a brief, recent history of partner violence.

Objective 4: To investigate positive (negative) impacts of peer to increase partner violence.

Clearstatements indicating a project's research general objective, specific objectives or questions do notnormally spring forth fully formed in a sudden moment. They tend to emergeslowly, after considerable thought, and take time to develop and finalize.

When first designing a project, try to give yourself plenty of time to think throughyour general and specific objectives. Ideally, this thinking should not be done in a hurry or underpressure (i.e. with a limited time period or other conditions that puts the researcher in some sort pressures). Read around your subject. Analyze previous studies in the area. Look at howother researchers frame their general and specific objectives for their project works. What key technical terms or conceptsdo they employ for their project works? The better you understand the published literature (projects that are already completed) on your topic, themore likely you are to be able to effectively conceptualize your own general and specificobjectives.

Another important way to help clarify your research general and specific objectives is to writethem down and ask other people[7] to comment on them. Draft up an initial statement ofyour general and specific objectives. Revise it until you are satisfied with it as an opening or provisionalattempt to describe your goals. Show it to colleagues, supervisors, friends and family.Ask them what they think. Do they understand what you mean? Do they agree withthe particular objectives you have chosen? Do they regard them as feasible?

Use all the feedback you get from other people as a basis for critically assessing theclarity, relevance, logical consistency and practicality of your project general and specific objectives.This should help ensure your project starts off on the right foot and minimize the scope for you (or other people) to later become dissatisfied with your statedresearch goals.

[7] It is a common way in case of final project, since the student wrote their project general and specific objectives then submit to their advisor in order to receive comments from the advisor.

In order to evaluate the correctness of project objective is to follow the mnemonic S.M.A.R.T.:

- **Specific**: Define your objectives clearly, in detail, leaving no room for misinterpretation. Think of the five w's (who, what, when, where, and why).
- **Measurable**: State the measures and performance specifications you'll use to determine whether you've met your objectives.
- **Achievable or Attainable**: Choose objectives that the team has a reasonable expectation of successfully completing.
- **Realistic**: Set objectives the project team believes it can achieve. Relevant objectives align with group or company goals.
- **Time-bound:** Include the date or specific period by which you'll achieve the objectives.

Example of Poor and Good objectives:

- Poorly Written Objectives
  - Personal: Earn more money.
  - Personal: Go to university.
  - Nonprofit: Help children read better.
  - Business: Create a new app.
  - Business: Install a new system.
- Examples of Well Written Objectives
  - Personal: Your goal may be to buy a house, but your objective is to get a condominium for under $250,000 with two bedrooms by August.
  - Business: The goal is to build up the company in the near future. The specific objective is to increase sales of Super Widget by 10 percent by the second quarter of next year.
  - Business: The goal is to make existing customers happier. The specific objective is to improve customer satisfaction rates by 50 percent by June 30 through training of customer service team.

Common errors when writing project objectives[8]

- One quite frequent error is collapsing all the information on a project's general objective and specific objectives into a single paragraph. This makes it hard for readers to absorb the information and distinguish the project's overall general objective from its more specific objectives. A project's general objective and specific objectives should be clearly distinguished. Present them in separate sentences or paragraphs. Each research objective should be numbered (bulleted).

For example

This project aims to investigate factors associated with partner violence and in particularwhether alcohol consumption and labor force status (employment, unemployment, notin the labor force) is associated with increased partner violence, and whether there aredifferences between couples with an extended history of partner violence and coupleswith only a brief, recent history of partner violence.

- Another common error is phrasing research aims or objectives in such a way that their meaning is vague or ambiguous

For example

This project aims to investigate partner violence and:

  - Alcohol consumption in couples with a history of partner violence
  - The labor force status of couples engaged in partner violence
  - Couples with an extended history of partner violence and couples with only a brief, recent history of partner violence

[8]Another reasonably common error is confusing 'research objectives' with 'project objectives'. As indicated above, research objectives refer to the areas of knowledge the project is aiming to build on or advance. Project objectives are something quite different. They are the practical steps involved in getting the day-to-day work of the project completed. Examples of project objectives include completing fieldwork or interviews within a scheduled timeframe, writing a project report, or communicating project results to different audiences. But this problem is not a big issue for final projects because the intension of the projects is to develop software instead of adding new knowledge in the field of specialization.

- ❖ Occasionally, a mix of project objectives and project questions[9] will be presented as a single list. This can be quite confusing and difficult to follow. A better approach is to list only a series of project objectives or only a series of project questions.

This project objective to investigate factors associated with partner violence. More specifically, it aims to:

For Example

- o Examine whether alcohol consumption is associated with increased partner violence.
- o Is labor force status (employment, unemployment, not in the labor force) associated with variations in the incidence of partner violence?
- o Are there differences between couples with an extended history of partner violence and couples with only a brief, recent history of partner violence?

Scope of the Study

Any time that a task of doing a final project is to be undertaken, the essential guideline is to define explicitly the scope of the study. Therefore, this sub-section gives you a clear understanding to what is to be included in the scope of the study (project work) and what is the significance of the same.

The scope of the study basically means all those things that will be covered in the final project. It defines clearly the extent of content that will be covered by the means of the final project in order to come to a system that meets the users' requirements and give conclusive and satisfactory answers to them [4].

The scope[10] of the study has to be defined at a preliminary stage and that is very important. It cannot be done in the later phase of doing the projects as it creates a lot of ambiguity about the project goals. If the team (project team) fails to define the scope at the initial stage itself, it is

[9] Questions that are expected to found in the statement of the problem section of your project.

[10] Some scholar define scope as "The Scope of study is the area in an agreement where the work to be performed is described. The scope should contain any milestones, reports, deliverables, and end products (software, design document, hardware or others) that are expected to be provided by the performing party. The SOW should also contain a time line for all deliverables."

indicative that the project would eventually not meet the expectations set by the department final project committee.

A thorough understanding of the field of the study is very important so as to know specifically what the requirements from the project are. It is imperative for the student in the project team to have unquestionable project and writing skills.

In order to write a comprehensive and valuable scope, all the important things to be included are the outline of the limitations of the study, the specifications about the data that has been used for project and the various theories[11] that have been put to use in the project to properly develop a proposed system. The importance of the scope of the study is such that it explains the reasons why a certain kind of functionalities has been excluded from the project. For further information regarding scope of the project please see the checklist found in Appendix one.

Example 1:

- In the proposed study I will only investigate the usage of multimedia on students of average academic ability.
- In the proposal study I will only investigate the effect of plastic content on the modules of elasticity of steel.

Example 2:

This study will focus on developing a web-based help desk system using a problem tracking technique for Graduate School of Informatics (SoI) at Hawassa University. This system will let users (staff, students or lecturers) to freely interact with the technical support employees who have the ability to answer all problems related to the graduate services provided by the SoI-HU. In addition, SoI-HUgraduate students can navigate FAQ section which contains a groups of frequently asked questions that are related to SoI-HU Graduate programs at HU and they can find their problem or inquiry there. Moreover, the proposed system will help the students, lecturers, and staff at SoI-HUgraduate programs to see the latest SoI-HU graduate news and

[11] Sample projects done by other programmers related to your project, and final project documentation done by other students in different universities.

articles added by the technical support employees. The prototype to be developed is based on problem tracking technique.

Limitation of the Study

It is for sure that your final project will have some limitations[12] and it is normal. However, it is critically important for you to be striving to minimize the range of scope of limitations throughout the final project work and provide the acknowledgement of limitations in conclusions[13] chapter in an honest manner.

It is important to:

- Always acknowledge a study's limitations. It is far better that you identify and acknowledge your study's limitations than to have them pointed out by your advisor and/or examiner of your project.
- Keep in mind that acknowledgement of a project's limitations is an opportunity to make suggestions for further works.
- Acknowledgement of a study's limitations also provides you with an opportunity to demonstrate that you have thought critically about the project problem, understood the related projects done by others, and correctly code and demonstrate your project.
- Claiming limitations is a subjective[14] process because you must evaluate the impact of those limitations.

The most common kinds of limitations in the students' final projects:

- Lack of programming skill
- Fluency in a language
- Lack of references materials (books, sample documentations, and others) in the university related to the project.

[12] Functional and/or non-functional requirements of the proposed system can't be fully implemented due to a number of limitation.
[13] State all limitations that hinders you in order to me both functional and non-functional requirements of the proposed system.
[14] Most of the time limitation of the study can be evaluated by the viewers' personal judgements about the project. Therefore, the project members must have a clear justification for every limitation that was stated in their final project documentation.

- Lack of available and/or reliable data - a lack of data or of reliable data will likely require you to limit the scope of your project.
- Lack of prior project development activities – most students are not done projects like final project in other courses. Therefore, the students may face a number of problems in project management, coding and testing the project.
- Lack of budget to buy necessary hardware for the project.
- Lack of hardware and software resources in the university.
- Shortage of time[15].
- Infrequent contact to project advisor.
- Self-reported data -- whether you are relying on pre-existing data or you are conducting by your own sample data set (testing set) and gathering the data yourself, self-reported data is limited by the fact that it rarely can be independently verified.
- Access -- if your study depends on having access to people, organizations, or documents and, for whatever reason, access is denied or limited in some way, the reasons for this need to be described.
- Cultural and other type of bias.

When discussing the limitations of your research, be sure to:

- Describe each limitation in detailed but concise terms (i.e. it is better to state the limitations in bullets and the readers gives an attention to it);
- Explain why each limitation exists (i.e. justify your reason clearly for every limitation that can be found in your project);
- Provide the reasons why each limitation could not be overcome using the method(s) chosen to acquire or gather the data (i.e. justify your reasons if possible by citing other studies that had similar problems];
- Assess the impact of each limitation in relation to the overall project functional and non-functional requirements and conclusions of your study; and
- If appropriate, describe how these limitations could point to the need for further project works.

[15] Since students do their final project in the final year two semesters but they are also taking other courses.

For example:the second example in the previous section (i.e. scope of the study) states the scope of the proposed system for managing technical support for graduate students in the school. These are some of the limitations of this project:

- The study couldn't focus on technical support issues for undergraduate program users (i.e. staffs, students and lecturers) in the school.
- The study couldn't address technical support problems for other graduate and undergraduate programs in the university.
- Due to the methodology (i.e. problem tracking technique) the proposed system follows the study can't address problems all issues for graduate programs.

Methodology

Methodology is the systematic, theoretical analysis of the methods applied to a field of study (project). It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as data collection paradigm, theoretical model for analysis and design follows, phases in software development, testing and deployment.

The introduction to your methodology section should begin by restating the project work and underlying assumptions underpinning your study. This is followed by situating the methods you will use to gather, analyze, process information, development, testing and deployment within the overall "tradition" of your field of study and within the particular project design you have chosen to study the problem. If the method you choose lies outside of the tradition of your field, provide a justification for how your choice of methods specifically addresses the problem in ways that have not been utilized in prior studies.

Data Collection Methodology

Any project work is only as good as the data that drives it, so choosing the right technique of data collection can make all the difference.Likewise, there are a variety of techniques to use when gathering primary and secondary data for the project. Listed below are some of the most common data collection techniques.

Techniques	Key facts	Example
Interview	• Interviews can be conducted in person or over the telephone • Interviews can be done formally (structured), semi-structured, or informally • Questions should be focused, clear, and encourage open-ended responses • Interviews are mainly qualitative in nature	What are the influential bottlenecks to the organization ICT infrastructure?
Questionnaires and Surveys	• Responses can be analyzed with quantitative methods by assigning numerical values to Likert-type scales • Results are generally easier (than qualitative techniques) to analyze • Pretest/Posttest can be compared and analyzed	
Observation	• Allows for the study of the dynamics of a situation, frequency counts of target behaviors, or other behaviors as indicated by needs of the evaluation • Good source for providing additional information about a particular group, can use video to provide documentation • Can produce qualitative (e.g., narrative data) and quantitative data (e.g., frequency counts, mean length of interactions, and instructional time)	Conducting in site observation to evaluate the current working environment of the organization regarding to ICT infrastructure.
Focus Group	• A facilitated group interview with individuals that have something in common • Gathers information about combined	A group of staffs, clients of the organization and development team discuss issues related to the current

	perspectives and opinions • Responses are often coded into categories and analyzed thematically	status of company ICT infrastructure.
Ethnographies, Oral History, and Case Studies	• Involves studying a single phenomenon • Examines people in their natural settings • Uses a combination of techniques such as observation, interviews, and surveys • Ethnography is a more holistic approach to evaluation • Researcher can become a confounding variable	This data collection method is mostly used in social science researches.
Documents and records	• Consists of examining existing data in the form of databases, meeting minutes, reports, attendance logs, financial records, newsletters, etc. • This can be an inexpensive way to gather information but may be an incomplete data source	This approach mainly focuses on checking the existing documents that helps the process of software development.

Note: from the above data collection methods; developer (team) can select some of them based on the relevancy in process of collecting relevant information to the entire project. Most software developers use interview, observation, focus group discussion and document and record analysis methods to conduct the software development.

System Analysis and Design Methodology (only for software development)

When developing information systems, use a standard of steps called the systems development lifecycle (SDLC). SDLC[16] includes phases such as planning, analysis, design, implementation, and maintenance. At the heart of systems development, analysis and design are the second and third phases of SDLC.

[16] The common approach for final project documentation in the school is object oriented system development approach. Chapter three and four of this module can describe the detail contents and steps in this system development strategy.

The analysis phase[17]usually requires a careful study of the current system, which continues two sub phases: requirements determination and analysis study. Requirements determination process usually involves a careful study of the current manual and computerized systems that may be replaced or improved within the project. Analysis study process usually involves analysts to study the structural requirements according to the components interrelationships and eliminate redundancies.

In the design phase[18], analysts design all aspects of the system, provide physical specifics on the system from input and output screens to reports, databases, and computer processes. In the effort to improve the systems analysis and design processes, different approaches have been developed.

Some of the system development approaches are:

- The traditional waterfall approach focuses on compartmentalizing project into several phases.
- The agile approach focuses on self-adaptive processes with an emphasis on individual talents.
- The object-oriented approach focuses on combining data and processes into objects and shares the iterative development approach of the agile method.

These are most widely used system development approaches in the software industry. These approaches all have different advantages and disadvantages in a way that they could be used to fit and optimize different kinds of projects.Therefore, as a student you can select one of the system development approach[19] to your project work; this selection could be done by evaluating advantages and disadvantages each method with respect to your proposed work.

System Implementation (only for software development)

System implementation (coding) is a process of writing computer programs in order to meet the business functional requirements stated in system analysis phase and detail design requirements found in design phase. The team is responsible to select appropriate coding method by evaluating

[17] The detail activities are written in chapter three of this module.
[18] The detail activities are written in chapter four of this module.
[19] You can read references to know advantages and disadvantages of each system development approaches.

advantages and disadvantages of each approaches and consider the programming skills of the team member in the group. The most common coding approaches are:

- **Object-oriented programming[20] (OOP)** is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods. In OOP, computer programs are designed by making them out of objects that interact with one another.
- **Procedural programming** is a programming paradigm, derived from structured programming, based upon the concept of the procedure call. Procedures simply contain a series of computational steps to be carried out by computer. Any given procedure might be called at any point during a program's execution, including by other procedures or itself in order to process the entire procedure.

Testing[21] and Deployment Methodology

Software testing[22] is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs, and verifying that the software product is fit for use.

There are generally four recognized levels of tests: unit testing, integration testing, component interface testing, and system testing. Tests are frequently grouped by where they are added in the software development process, or by the level of specificity of the test.

The team can specify testing methods under each testing level (i.e. more than testing method)and testing dataset (i.e. the set of data used by programmer and/or testing team in the company from

[20] It is the most common kind of programming approach at this time due to different features to make the programming easy and efficient. In addition to this, many programming languages are belonging to this group.
[21] This task is done from the early beginning of software coding up to deploying the system in the clients working machine.
[22] "Software testing involves the execution of a software component or system component to evaluate one or more properties of interest."

the beginning to the end of software development) that could verify the successfulness of a software to meet the functional and non-functional requirements of the system.

Software deployment is all of the activities that make a software system available for use.The deployment process consists of several interrelated activities with possible transitions between them. These activities can occur at the producer side or at the consumer side or both. Because every software system is unique, the precise processes or procedures within each activity can hardly be defined.

Therefore, the team can prepare the software deployment plan (i.e. software deployment has a unique[23] feature between (among) organizations) for the successful integration of the new system with the existing business transactions in the organization.

Development Environment (Only for Software Development)

Software development environment refers to the collection of hardware and software tools a systemdeveloper (programmer) uses to build software systems [6]. As technology improves anduser expectations grow, an environment's functionality tends to change.

Softwaredevelopment environment is an environment that augments or automatesthe activities comprising the software development cycle, includingprogramming-in-the-large tasks such as configuration management andprogramming-in-the-many tasks such as project and team management.Wealso mean an environment that supports largescale, long-term maintenanceof software.

The team must specify both hardware and software requirements for developing the proposed project work.

For example:

- Software Requirement
  - Operating System: Window 10 x64 (64 bits)

[23] This condition can be happening due to different reasons in the organization; like staff computer skill, budget, related software availability in the organization and others.

- Programming language: NetBeans 8.1 (IDE[24] for writing programs using a Java Programming language)
- JDK 1.7
- Database: Microsoft SQL Server 2012 R2 (64 bits)

❖ Hardware Requirement

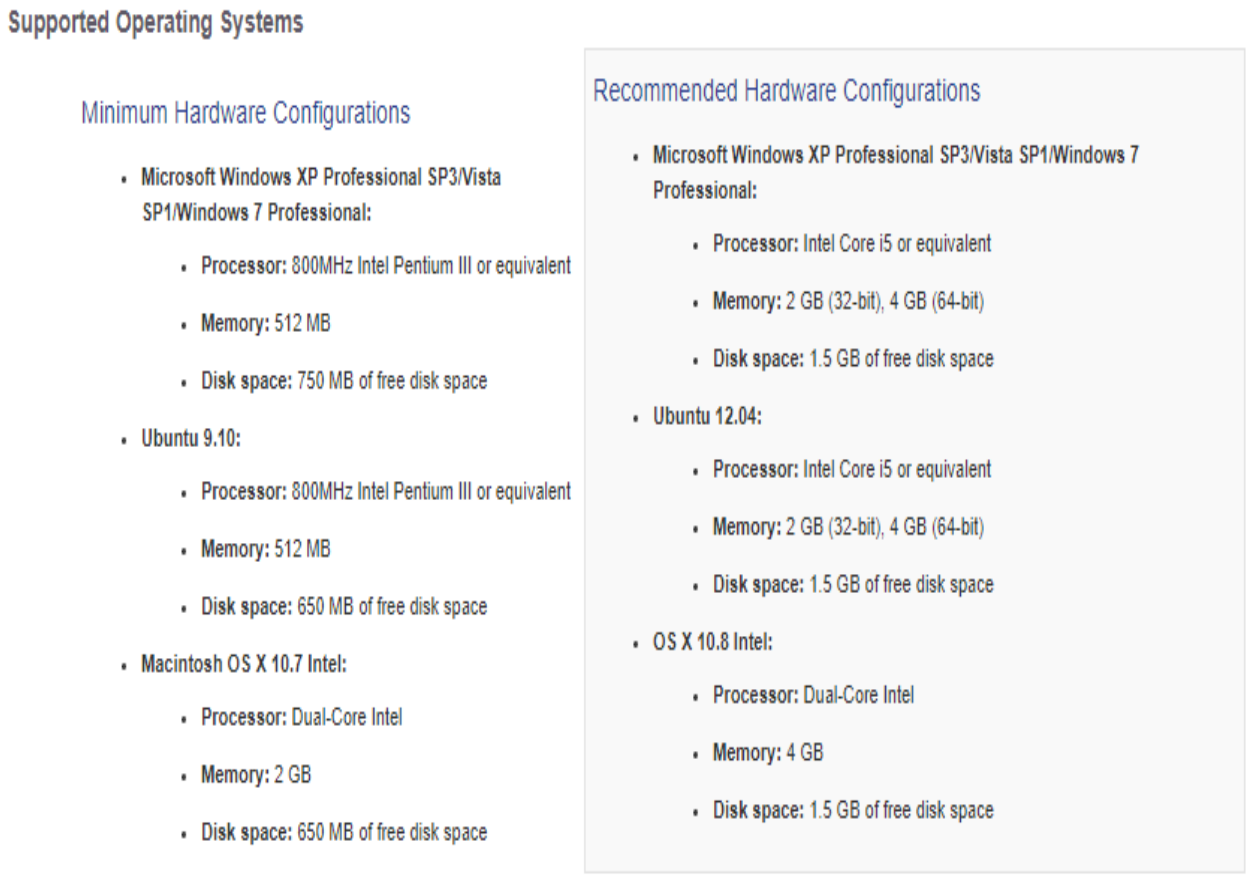
- Personal computer:
  - 500GB hard disk
  - 4GB RAM
  - Corei7 2.3 GHz
  - 14.1" screen
  - Built in Wi-Fi support
  - LAN support
- 2 TB External hard disc

System Requirement (Only for System Development)

To be used efficiently, all computer software needs certain hardware components (like architecture, processing power, memory, secondary storage, display adapter and peripherals) or other software resources (like platform, API, Drivers and Web Browsers) to be present on a computer. These prerequisites are known as system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum (i.e. the minimum software and hardware requirements in order to run the software) and recommended.

Example 1:system requirements for NetBeans IDE 8.1[25]

[24] Integrated Development Environment (IDE) is a software package that have modules for writing, compiling, executing and monitoring a computer program using different programming languages.
[25] The system requirement (minimum and recommended) varies based on the kind of Operating System the users use to install the IDE in their computer system.



Supported Operating Systems

Minimum Hardware Configurations

- Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional:
  - Processor: 800MHz Intel Pentium III or equivalent
  - Memory: 512 MB
  - Disk space: 750 MB of free disk space
- Ubuntu 9.10:
  - Processor: 800MHz Intel Pentium III or equivalent
  - Memory: 512 MB
  - Disk space: 650 MB of free disk space
- Macintosh OS X 10.7 Intel:
  - Processor: Dual-Core Intel
  - Memory: 2 GB
  - Disk space: 650 MB of free disk space

Recommended Hardware Configurations

- Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional:
  - Processor: Intel Core i5 or equivalent
  - Memory: 2 GB (32-bit), 4 GB (64-bit)
  - Disk space: 1.5 GB of free disk space
- Ubuntu 12.04:
  - Processor: Intel Core i5 or equivalent
  - Memory: 2 GB (32-bit), 4 GB (64-bit)
  - Disk space: 1.5 GB of free disk space
- OS X 10.8 Intel:
  - Processor: Dual-Core Intel
  - Memory: 4 GB
  - Disk space: 1.5 GB of free disk space

Example 2:system requirements[26] for Android Studio 2.0.

Criterion	Description
OS version	Windows 7 or later Mac OS X 10.9.5 or later GNOME or KDE desktop
RAM	3 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
Disk space	500 MB disk space for Android Studio, at least 1.5 GB for Android SDK, emulator system images, and caches
Java version	Java Development Kit (JDK) 8
Screen resolution	1280x800 minimum screen resolution

[26] This system requirement defines both hardware and software requirements to the minimum as well as the recommended system requirements.

Feasibility Study

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technical, legal (sometimes political), operational and scheduling factors [7]. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it.There are five types of feasibility study[27]; three of them discussed here and done by the students for their final project's:

Technical Feasibility

Thisassessment focuses on the technical resources (capabilities) available to the team members in the group. It helps to determine whether the team has a required technical resources meet the objective and whether the team is capable of converting the ideas into working system. Technical feasibility also involves evaluation of the hardware, software, and other technology requirements of the proposed system.

For example: the team will propose to design and implement an intelligent machine (robot). But due to shortage of required hardware and software in the university this system is technically infeasible.

Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.

Therefore, the team assess the operational feasibility of the project and manage the system development activities in the system life cycle in order to satisfy the operational feasibility of the project.

[27] Schedule and legal feasibilities are the remaining feasibility studies for any project.

Economic Feasibility

Thisassessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility helping decision makers determine the positive economic benefits to the organization that the proposed project will provide.

In order to correctly calculate the economic[28] feasibility of the project, the team can consider different issues that have a direct impact on the cost of the proposed system (like: hardware and software cost, training cost, installation and deployment cost) and running (operational[29]) cost to use the proposed system.

For example:

Type	Potential Costs	Potential Benefits
Quantitative	• Hardware/software upgrades • Fully-burdened cost of labor • Support costs for the application • Expected operational costs • Training costs for users • Training costs to train developers in new/updated technologies	• Reduced operating costs • Reduced personnel costs from a reduction in staff • Increased revenue from additional sales of your organizations products/services
Qualitative	• Increased employee dissatisfaction from fear of change • Negative public perception from layoffs as the result of automation	• Improved decisions as the result of access to accurate information • Raising of existing in the industry • Positive public perception that your organization is an innovator

[28] The team also finds some standard models that can be used to calculate the economic feasibility of the proposed system to the client.
[29] Operational cost of the software will includes; personnel salary, electric bill, cost of accessories, and other cost to use the proposed system successfully.

Cost Estimation and Schedule Breakdown

In this sub section of the chapter, the team members of the project are expected to calculate the total budget required to develop the proposed system, list out activities and propose the time frame (schedule) for each activity in the proposed system development process. For these tasks the team may use well known cost estimation and schedule models in software development projects[30].

For detail information and example concerning cost breakdown and schedule preparation for your final project please refer the sample cost breakdown and schedule using Gantt Chart in appendix two and three respectively.

[30]A Gantt chart is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project.

Chapter Two: DESCRIPTIONS OF EXSTING SYSTEM

Introduction of the Existing System

The team can spend enough time to ensure that the team could understand the functionality currently provided by the existing system(s), both manual and automated. This should not be exhaustive, as the existing system is being replaced [9]. This study mainly focuses to address the following main issues:

- Objectives
  - To understand how the requirements are currently being met within the existing system.
  - To document the flow, processing and use of information within the existing system.
  - To identify the problems with the existing system.
  - To define new requirements.
- Responsibilities
  - The team conduct interviews and obtain information about the way the manual and automated systems (within the scope of the project) currently operate in the given organization (company).
- Inputs
  - User, operations, and procedure manuals for existing systems
  - Source code listings
  - Sample input forms, reports, and screen layouts
  - System documentation
  - Dictionary reports

Note: Many systems will have little or no documentation regarding some of their tasks to run the business. This will make the job of identifying requirements more challenging for the team and require more time from the users.

- Method
  - Interviews may be used to gather information about the existing system Schedule these in advance beginning with user management.

- Before any interviews are conducted, the team members should study the available printed material and discuss the information. This helps to maximize the use of time in an interview and to establish a rapport with the user on first contact. The product of this initial research should be a developing understanding of the functional structure of the business under study.
- Conduct interviews, beginning with managers and working down to the level of the clerical or production staff who work most closely with the system(s).
- Prepare a synthesis of the interview notes.

❖ Working Documents
- Interview Notes (Off-the-Record items should not be made available in 'published' interview notes.)

❖ Deliverables
- Interview Notes
- List of Sources of Information

Consider the following issues at the time conducting interviews[31] to collect information from the users of the existing system.

❖ Interviews are used as the standard mechanism for determining how current business systems operate. This process is often time consuming and unproductive since users:
- do not always understand how their system(s) operate
- tend to focus on solutions (how) not structure or content (what)
- may not know why the current system(s) was set up as it was (the reasons may have completely disappeared)
- may have a limited perspective of the system(s), focusing only on the portion for which they are responsible
- do not always have an understanding of the cause of problems
- may be unable (or unwilling) to look at alternative approaches
- may have organizational or political reasons for resisting change
- may not have thought about what is required in the new system

[31] Sometimes interviewing different stakeholders of the existing system is not the only task done by the team members in order to write description of existing system. The team may also use other means of data collection before doing this task.

- may not be aware of what technology or different approaches could be applied to the design and implementation of their system(s)
- An alternative is to select key users from the organization and to involve them in the project from the beginning. Involve them (using their knowledge of the existing systems) in the definition of the requirements of the new system, and by-pass the documentation of existing systems.

Note: to write introduction to the existing system the team could answer all the above mentioned issues in the existing system in order to neatly describe the system to the users and other stakeholders. On the other hand, the introduction section will be write by the team members in few pages (maximum of three pages); it can't to strictly follows the format to answer above mentioned issues. For further information and example for description of existing system please refer appendix four.

Proposed System Description

Most organizations now operate in an environment that is rapidly changing. The relative strength of national economies around the nation can change dramatically and at short notice; the fortunes of large companies, which may be an organizations supplier, customers or competitors, can be transformed overnight; new technologies are introduced which change production processes, distribution networks and the relationship with the consumer; and governments (both regional and federal) will change dynamically. Some authors make the case for developing business strategies to cope with this turmoil. The process of replacement offers an opportunity to extend the capabilities of systems to take advantage of new technological developments, or to enhance their usefulness to management and workforce.

The team could clearly state how the proposed system answers all questions (issues) in the existing system and consider the future direction of the organization and technological changes in the IT industries. This task can be done after the detail understanding of the current system and reviewing other related works to the system; it helps the team to know how other researchers and programmers handle the same problem. In addition to this, the team can also analyze non-technical parameters (issues) that can be exist in system development; like budget, social and

political feasibilities of the study. For further information and example for proposed system description please refer appendix four and a reference found in the reference list.

Strength of Existing System[32]

Strengths describe the positive attributes, tangible and intangible, internal to your organization. They are within your control. The team can conduct an analysis to know the strength of the existing system in collaboration with different stakeholders in the company. The study clearly answers the following questions:

- What do you do well?
- Internal resources do you have? Think about the following:
  - Positive attributes of people, such as knowledge, background, education, credentials, network, reputation, or skills.
  - Tangible assets of the company, such as capital, credit, existing customers or distribution channels, patents, or technology.
  - Positive attributes of data handling, repository structure, file organization, reporting and other data processing activities in the organization.
- What advantages do you have over your competition?
- Do you have strong research and development capabilities? Manufacturing facilities?
- What other positive aspects, internal to your business, add value or offer you a competitive advantage?

For example: consider the strength of software company.

- Excellent Web design staff
- Low systems analyst turnover
- Recently upgraded network
- Well structure system development standard

[32] Some scholars prefer a SWOT (Strength, Weakness, Opportunities and Threat) analysis instead of identifying strength and weakness of the existing system. A SWOT analysis can focus on a specific product or project, an operating division, the entire company, or the mission statement itself. The overall aim is to avoid seeking goals that are unrealistic, unprofitable, or unachievable.
Strengths and weaknesses are internal to the company (think: reputation, patents, location). You can change them over time but not without some work. Opportunities and threats are external (think: suppliers, competitors, prices)—they are out there in the market, happening whether you like it or not. You can't change them.

Weakness of Existing System[33]

Weaknesses are aspects of your business that detract from the value you offer or place you at a competitive disadvantage [11, 12]. You need to enhance these areas in order to compete with your best competitor.The team can conduct an analysis to know the weakness of the existing system in collaboration with different stakeholders in the company. The study clearly answers the following questions:

- What factors that are within your control detract (like skills of workers, power fluctuation, incentives to your workers, and others) from your ability to obtain or maintain a competitive edge?
- What areas need improvement (like software usage, server and other resources) to accomplish your objectives or compete with your strongest competitor?
- What does your business lack (for example, expertise or access to skills or technology)?
- Does your business have limited resources?
- Is your business in a poor location?

For example: consider the strength of software company.

- Still using several legacy systems
- Budget increase was turned down
- Documentation needs frequent updating

[33] An enterprise SWOT analysis usually begins with these questions:
- What are our strengths, and how can we use them to achieve our business goals?
- What are our weaknesses, and how can we reduce or eliminate them?
- What are our opportunities, and how do we plan to take advantage of them?
- What are our threats, and how can we assess, manage, and respond to the possible risks?

Chapter Three: SYSTEM FEATURES

Introduction

This chapter introduces for students and advisors about proposed system major specification which includes functional and non-functional requirements and different analysis level models that depicts the application domain. Analysis results in a model of the system that aims to be correct, complete, consistent, and verifiable. You as developers should formalize the system specification during requirements elicitation. In object-oriented analysis, developers build a model describing the business domain.

For Example:

The analysis model of a watch describes how the watch represents time (e.g., Does the watch know about leap years? Does it know about the day of the week? Does it know about the phases of the moon?) The analysis model is then extended to describe how the actors and the system interact to manipulate the application domain model (e.g., How does the watch owner reset the time? How does the watch owner reset the day of the week?). The analysis model, together with the system specification (functional and non-functional requirements, is an input for preparing the architecture of the system or high-level design.

The term 'requirement' is not used consistently in the software industry. In some case, a requirement is simply a high-level, abstract statement of a service that a system should provide or a constraint on a system. On the other end, it is a detailed, formal definition of a system function. Based on this user requirement are classified in to two.

a. **User requirements** are statements, in a natural language plus diagrams, of what services the system is specified is expected to provide to system users and the constraints under which it must operate.
b. **System requirements:** are more detailed descriptions of the software system's functions, services, and operational constraints.[34]

[34] The system requirements document (sometimes called a functional specification) should define exactly what is to be implemented. It acts as contract between the users/customers and the software developers.

For Example:

Let's illustrates the distinction between user and system requirements. This example from a mental health care patient management system (MHC-PMS) [35] shows how a user requirement may be expanded into several system requirements.

User Requirement Definition

1. The MHC-PMS shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System Requirements Specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost, and the prescribing clinics shall be generated.
1.2 The system shall automatically generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed, and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g., 10 mg, 20 mg) separate reports shall be created for each dose unit.
1.5 Access to all cost reports shall be restricted to authorized users listedon amanagement access control list. separate reports shall be created for each dose unit.

Functional requirements

Functional requirements – a description of facility or feature required. Functional requirements deal with what the system should do provide for users. They include description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system.

[35] User requirements are quite in general whereas system requirements provide more specific information about the services and functions of the system that is to be implemented.

For example, here are examples of functional requirements for the MHC-PMS system, used to maintain information about patients receiving treatment for mental health problems.

1. A user shall be able to search the appointments lists for all clinics.
2. The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
3. Each staff member using the system shall be uniquely identified by his or her eight-digit employee number.

Non-Functional requirements

- Non-functional requirements are detail constraints, on the services or functions offered by the system. They include constraints, on the development process, and constraints imposed by standards. Non- functional requirements often apply to the system as a whole, rather than individual system or services. They are expected characteristics of software; which users make assumption of. They describe how well or to what standard a function should be provided. For example, levels of required service such as response times; security and access requirements; technical constraints; required interfacing with users' and other systems; and project constraints such as implementation on the organization's hardware/software platform. Service level requirements are measures of the quality of service required, and are crucial to capacity planning and physical design. Identify realistic, measurable target values for each service level. These include service hours, service availability, responsiveness, throughput and reliability. Security includes defining priority and frequency of backup of data, recovery, fallback and contingency planning and access restrictions. Access restrictions should deal with what data needs protected; what data should be restricted to a particular user role; and level of restriction required, e.g. physical, password, view only.

  Non-functional requirements may come from required characteristics of the software (product requirements), the organization developing the software (organizational requirements), or from external sources.

Product requirements: These requirements specify or constrain the behavior of the software. Examples include performance requirements on how fast the system must execute and how much memory it requires, reliability requirements that set out the acceptable failure rate, security requirements, and usability requirements.

- **Usability requirements**: here you should describe all of the functional requirements that affect usability such as ease of learning, task efficiency, ease of remembering, understandability, attractiveness, etc.
- **Performance requirements**: describe all of the functional requirements that affect performance such as speed, safety, precision, reliability and availability, capacity, scalability, etc.
- **Security requirements**: describe all of the functional requirements that affect security such as security audits, identification/authentication, privacy, facility access times, etc.
- **Interface requirements:** Describe all of the functional requirements that affect interfaces such as user navigation, presentation of application and associated functionality, screen location of interface elements, data display and manipulation, etc.

Organizational requirements: These requirements are broad system requirements derived from policies and procedures in the customer's and developer's organization. Examples include operational process requirements that define how the system will be used, development process requirements that specify the programming language, the development environment or process standards to be used, and environmental requirements that specify the operating environment of the system.

External requirements: This broad heading covers all requirements that are derived from factors external to the system and its development process. These may include regulatory requirements that set out what must be done for the system to be approved for use by a regulator, such as a central bank; legislative requirements that must be followed to ensure that the system operates within the law; and ethical requirements that ensure that the system will be acceptable to its users and the general public.

Example: If you remember we have introduced user requirements for MHC-PMS [36] earlier. Now we will list some of potential non-functional requirements for MHC-PMS based on the classification mentioned above.

- PRODUCT REQUIREMENT

  The MHC-PMS shall be available to all clinics during normal working hours (Mon–Fri, 08.30–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.
- ORGANIZATIONAL REQUIREMENT

  Users of the MHC-PMS system shall authenticate themselves using their health authority identity card.
- EXTERNAL REQUIREMENT

  The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

Analysis Models

Introduction:

This section introduces the proposed system with UML system models. System models describe the model of the envisioned system. For the new system model we will use case models, sequence diagrams, Analysis level class model, and state chart diagrams.

Use case Model

Use case identifies the actors involved in an interaction and names the type of interactions. This is then supplemented by additional information describing the interaction with the system. [37] The set of use cases represents all of the possible interactions that will be described in the system requirements. Actors in the process, who may be human or other systems, are represented as stick figures. Each class of interaction is represented as a named ellipse.

[36] MHC-PMS is short form for Mental Health Care Patient Management System
[37] Use cases are documented using a high-level use case diagram.

Functional Requirements are decomposed into a number of System Use cases. Use cases are of primary importance early in a project's requirements/analysis phase. Their purpose is to document the business process that the Application must support without bias to technology and implementation.

The use case model will contain the following:

- Use case diagram
- Use case documentation according to the system use case template
- One or more sequence diagrams for each of the use cases

The system use case code is again a combination of two parts – prefix and sequence number. The prefix for system use case is always "SUC" and the sequence number is a three digits' number. The uniqueness of the system use case id is only within the RAD document i.e. two system use cases in different documents can have the same code.

e.g. SUC-003

Regarding the system use cases naming the following guidelines can be followed:

- The use case name should be descriptive i.e. one should see the responsibility of the business use case just from its name
- Compact the use case name as much as possible, the number of characters should not exceed 50
- No space should be used in between words
- It has to start with active verb
- Every new word should start with upper case
- The use case diagram must come before the textual documentation of the use case

Use case Description Template

Use Case ID:	[Identifier of each of the use case. It is a running number to identify the bureau and the sequence of the use case. Format: XX(bureau identifier).XXXX (Sequence identifier) e.g. 08.0005]
Use Case Name:	[A concise, results-oriented (showing the goal of the use case) name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. The use case name should start with an active verb. Example: Maintain Land Requesters
Created By:	[The name of the person or the team that originally created the use case. Format dd/mm/yy]
Date Created:	[The date on which the use case is initially created]

Updated By	**Date of Update**
[The name of the person or the team that modifies the use case]	[Date of modification. Format dd/mm/yy]

Actors:	[An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case and any other actors who will participate in completing the use case.]

Description:	[Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case. Use not more than three statements.]
Trigger:	[Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.]
Preconditions:	[List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples: User's identity has been authenticated. All required files are loaded on the memory.]
Normal Flow:	[Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is done as a numbered list of actions performed by the actor, alternating with responses provided by the system.]
Postconditions:	[Describe the state of the system at the conclusion of the use case execution. Number each post condition. Examples: A file is opened for the customer Customer is registered.]
Alternative	[Document other, legitimate usage scenarios that can take place within

Flows:	this use case separately in this section. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form "X.Y", where "X" is the number in the normal flow and Y is a sequence number for the alternative flow. For example, "5.3" would indicate the third alternative flow line number 5 in the normal flow.]
Exceptions:	[Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception.
Priority:	[The relative priority of the service provided by the given use case. Three categories of priority values can be assigned: High, Intermediate or Low]
Frequency of Use:	[How frequently is the use case used – every five minutes, once in a day, every other day, once in a week, etc]
Business Rules:	[Write any business rule of the organization that directly or indirectly affect the use case]
Special Requirements:	[Identify any additional requirements, such as non-functional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.]
Assumptions:	[List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use

	case description.]
Notes and Issues:	[List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.]

For Example:

Let's consider Hawassa University Integrated Student Information System where students are enrolling in courses with the potential help of registrars. Teachers/Instructors input the marks students earn on assignments and registrars authorize the distribution of transcripts (report cards) to students. [38] Understanding the associations between actor and use cases is important which actually indicates the use case invoked by that particular actor.

However, understanding the relation don't represent flows of information [39]; they merely indicate an actor is somehow involved with a use case. Information is flowing back and forth between the actor and the use case, for example, students would need to indicate which seminars they want to enroll in and the system would need to indicate to the students whether they have been enrolled.

One potential use case in this system is Enroll in course/seminar where Student and Registrar actors interact with these use case. As you can see in the use case diagram below the line between the *Enroll in Seminar* use case and the *Registrar* actor has no arrowhead, indicating it is not clear how the interaction between the system and registrars start. Perhaps a registrar may notice a student needs help and offers assistance, whereas other times, the student may request help from the registrar, important information that would be documented in the description of the use case.[40]

[38]In some Use cases there is more than one actor involvedE.g. you can see in the figure Drop seminar use case has a relation with two actors (student and registrar).
[39] Information flow can be modeled using UML Activity diagram.
[40]Actors are always involved with at least one use case and are always drawn on the outside edges of a use case diagram.

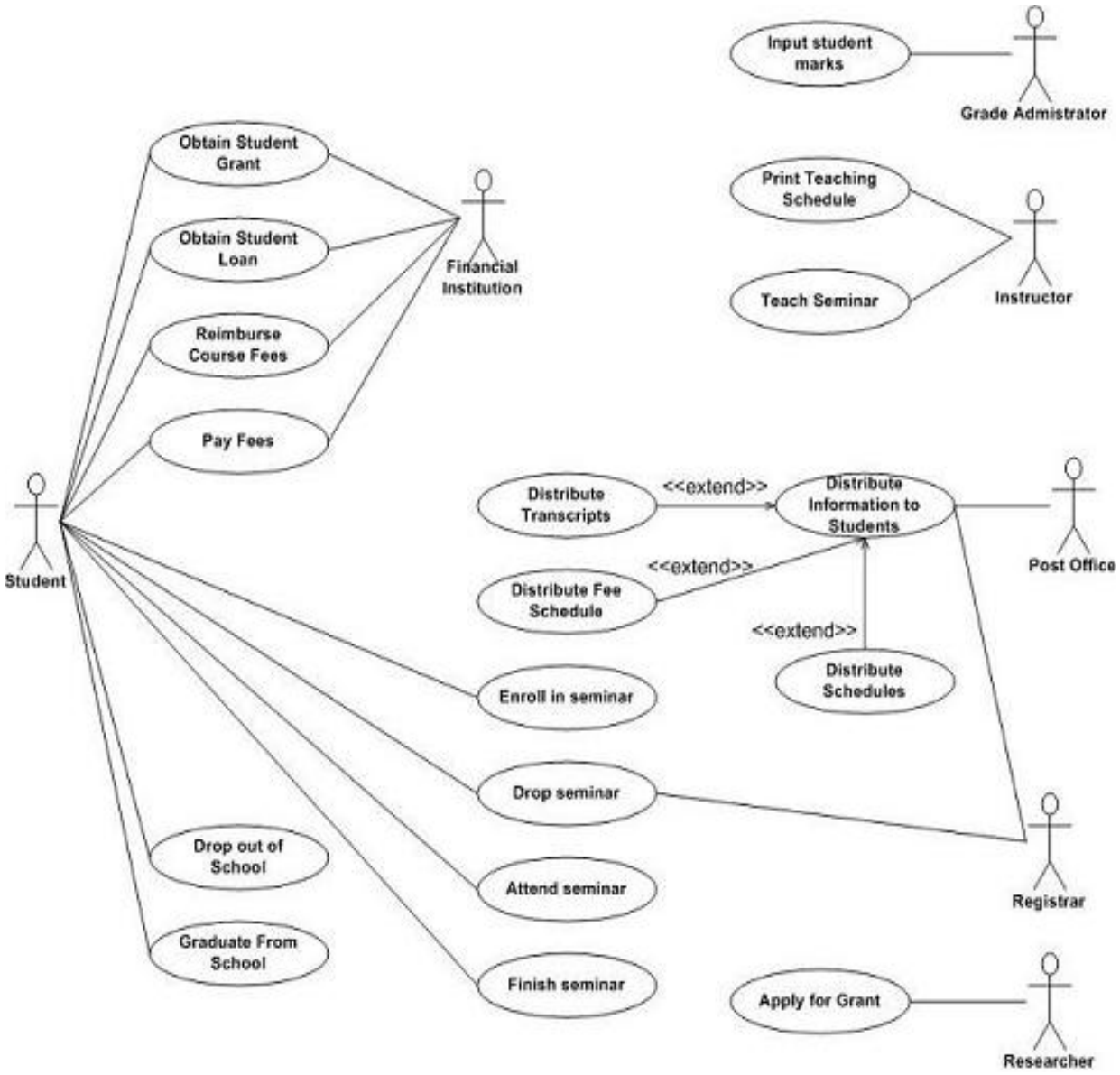


Figure System Use case diagram

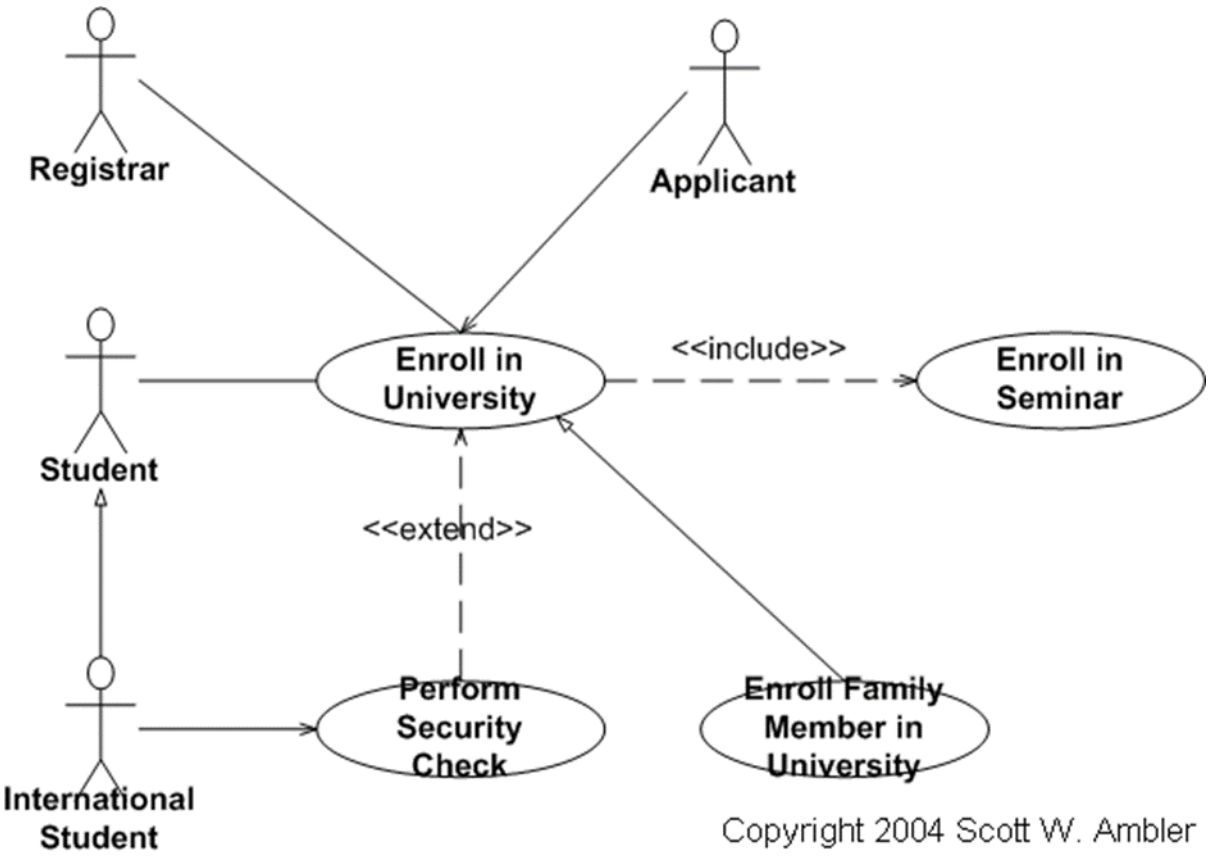
Creating Use Case Diagrams

I like to start by identifying as many actors as possible. You should ask how the actors interact with the system to identify an initial set of use cases. Then, on the diagram, you connect the actors with the use cases with which they are involved. If actor supplies information, initiates the use case, or receives any information because of the use case, then there should be an association between them. I generally do not include arrowheads on the association lines because my experience is that people confuse them for indications of information flow, not initial invocation.

As I begin to notice similarities between use cases, or between actors, I start modeling the appropriate relationships between them.

Reuse Opportunities

Figure 2 shows the three types of relationships between use cases -- extends, includes, and inheritance -- as well as inheritance between actors. I like to think of extend relationships as the equivalent of a "hardware interrupt" because you don't know when or if the extending use case will be invoked (perhaps a better way to look at this is extending use cases are conditional). Include relationships as the equivalent of a procedure call. Inheritance is applied in the same way as you would on UML class diagram[41],-- to model specialization of use cases or actors in this case.



[41] UML 2 Class diagramsare the mainstay of object oriented analysis. Class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association), and the operations and attributes of the classes.

Reuse in Use Case Models: <<extend>>, <<Include>>, and Inheritance

One of your goals during analysis is to identify potential opportunities for reuse, a goal you can work toward as you are developing your use case model. Potential reuse can be modeled through four generalization relationships supported by UML use case models: extend relationships between use cases, include relationships between use cases, inheritance between use cases and inheritance between actors.

Extend Associations between Use Cases

It is a generalization relationship where an extending use case continues the behavior of a base case. The extending use case accomplishes this by conceptually inserting additional action sequences into the base use case sequence. This enables an extending use case to continue the activity sequence of a base use case when the appropriate extension point is reached in the base use case and the extension condition is fulfilled. When the extending use case activity sequence is completed, the base use case continues. In the figure above you see that the use case "Enroll International Student in University" extends the use case "Enroll in University;" the notation for doing so is simply a normal use case association with the stereotype of <<extend>>.

Include Associations between Use Cases

It is a generalization relationship denoting the inclusion of the behavior described by another use case. It is the invocation of a use case by another one. In the figure above notice that the use case "enroll in university" includes the use case "Enroll in Seminar"; the notation for doing so is simply a normal use case association with the stereotype of <<include>>.

Inheritance

Use cases can inherit from other use cases, offering a third opportunity to indicate a potential reuse. In the figure above depicts an example of this, showing that "Enroll Family member in University" inherits from the "Enroll in University" use case.

Another opportunity for indicating potential reuse within use case models occurs between actors: An actor on a use case diagram can inherit from another actor. An example is shown in the above figure, where the "International Student" actor inherits from "Student". The standard

UML notation for inheritance, the open-headed arrow, is used and the advice presented about the appropriate use of inheritance still applies.

Use case Description Example

Use Case ID:	SUC-019
Use Case Name:	Enrol in University
Created By:	Natnael Gonfa
Date Created:	14/10/2017

Updated By	**Date of Update**
Natnael Gonfa	25/10/2017

Actors:	Registrar, Applicant
Description:	Enrol someone in the university
Trigger:	When someone/applicant wants to register in the university.
Preconditions:	• The Registrar is logged into the system. • The Applicant has already undergone initial checks to verify that they are eligible to enroll.
Normal Flow:	1. An applicant wants to enroll in the university. 2. The applicant hands a filled out copy of form UI13 University Application Form to the registrar. [Alternate Course A: Forms Not Filled Out] 3. The registrar visually inspects the forms. 4. The registrar determines that the forms have been filled out properly. [Alternate Course 4.1: Forms Improperly Filled Out]. 5. The registrar clicks on the Create Student icon. 6. The system displays UI89 Create Student Screen. 7. The registrar inputs the name, address, and phone number of the applicant. [Extension Point: UC34 Perform Security Check.

	Applicable to Step 17] 8. The system determines that the applicant does not already exist within the system according to BR37 Potential Match Criteria for New Students. [Alternate Course 8.1: Students Appears to Exist Within the System]. 9. The system determines that the applicant is on the eligible applicants list. [Alternate Course 9.1: Person is Not Eligible to Enroll] 10. The system adds the applicant to its records. The applicant is now considered to be a student. 11. The registrar helps the student to enroll in seminars via the use case UC 17 Enroll in Seminar. 12. The system calculates the required initial payment in accordance to BR16 Calculate Enrollment Fees. 13. The system displays UI15 Fee Summary Screen. 14. The registrar asks the student to pay the initial payment in accordance to BR19 Fee Payment Options. 15. The student pays the initial fee. [Alternate Course 15.1: The Student Can't Pay At This Time] 16. The system prints a receipt. 17. The registrar hands the student the receipt. 18. The use case ends.
Post conditions:	The Applicant will be enrolled in the university as a student if they are eligible.
Alternative Flows:	Alternate Course 4.1: Forms Not Filled Out 4.2. The Applicant asks for a set of forms. 4.3. The Applicant fills out the forms as appropriate. 4.4. The use case continues at step 2 in the basic course of action. Alternate Course 8.1: write the description here

	Alternate Course 9.1: write the description here Alternate course 15.1 : write the description here
Exceptions:	None
Priority:	High
Frequency of Use:	Once in a year
Business Rules:	
Special Requirements:	The system should be user friendly.
Assumptions:	
Notes and Issues:	

Sequence Diagram

UML Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.

Sequence diagrams are typically used to model:

Usage scenarios: A usage scenario is a description of a potential way your system is used. The logic of a usage scenario may be part of a use case, perhaps an alternate course. It may also be one entire pass through a use case, such as the logic described by the basic course of action or a portion of the basic course of action, plus one or more alternate scenarios. The logic of a usage scenario may also be a pass through the logic contained in several use cases. For example, a student enrolls in the university, and then immediately enrolls in three seminars.

The logic of methods. Sequence diagrams can be used to explore the logic of a complex operation, function, or procedure. One way to think of sequence diagrams, particularly highly detailed diagrams, is as visual object code.

The logic of services. A service is effectively a high-level method, often one that can be invoked by a wide variety of clients.

For Example

Let's consider the previous "Enroll in University" use case, taking a system-level approach where the interactions between the actors and the system are shown.

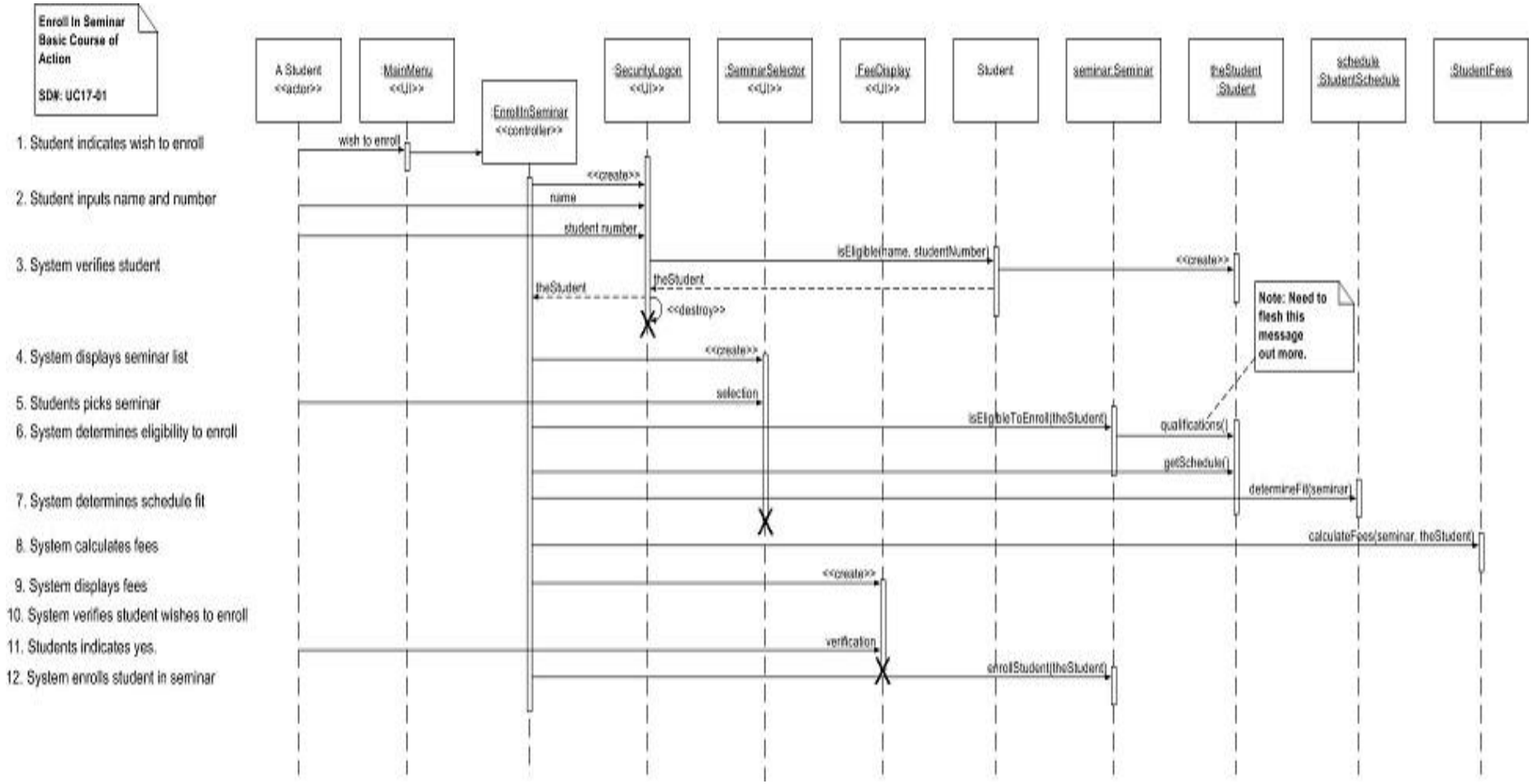
The figure shown below models the basic course of action for the "Enroll in Seminar" use case. The boxes across the top of the diagram represent classifiers or their instances, typically use cases, objects, classes or actors. Because you can send messages to both objects and classes, objects respond to messages through the invocation of an operation, and classes do so through the invocation of static operations, it makes sense to include both on sequence diagrams. Because actors initiate and take an active part in usage scenarios, they are also included in sequence diagrams. Objects have labels in the standard UML format "name: ClassName," where "name" is optional (objects with no name are called anonymous objects). Classes have labels in the format "ClassName," and actors have names in the format "Actor Name" – both UML standards as well. In the figureyou see the student actor has the name "A Student" is labeled with the stereotype <<actor>>.

The instance of the major UI element representing "UI32 Seminar Selection Screen," is an anonymous object with the name ":SeminarSelector" and the stereotype <<UI>>. The "Student" class is indicated on the diagram, the box with the name "Student," because the static message "isEligible(name,studentNumber" is sent to it.

The dashed lines hanging from the boxes are called object lifelines, representing the life span of the object during the scenario being modeled. The long, thin boxes on the lifelines are method-invocation boxes indicating that processing is being performed by the target object/class to fulfill a message. The X at the bottom of a method-invocation box is a UML convention to indicate that an object has been removed from memory, typically the result of receiving a message with the stereotype of <<destroy>>.

Messages are indicated by labeled arrows if the source and target of a message is an object or class however if either the source or target is a human actor, then the message is labeled with

brief text describing the information being communicated. The return values are indicatedby dashed and labeled arrows. For example, in the figure below“: EnrollInSeminar” object sends the message “isEligibleToEnroll(theStudent)” to the instance of “Seminar”. The return value “theStudent” is coming back from the “Student” class as the result of sending the message “isEligibleToEnroll(theStudent)” to “seminar.”



Sequence diagram for Enroll in seminar/course basic course of action

Activity Diagram

Activity diagram are typically used for business process modeling, for modeling the logic captured by a single use case or usage scenario, or for modeling the detailed logic of a business rule. 42they're used to describe business activities and software systems' functionality. You'll use a set of specialized symbols—including those for starting, ending, merging, or receiving steps in the flow—to build an activity diagram.

Stakeholders/Clients have many issues to manage, so it's important to communicate with clarity and brevity. Activity diagrams help people on the business and development sides of an organization come together.

Activity Diagram Components

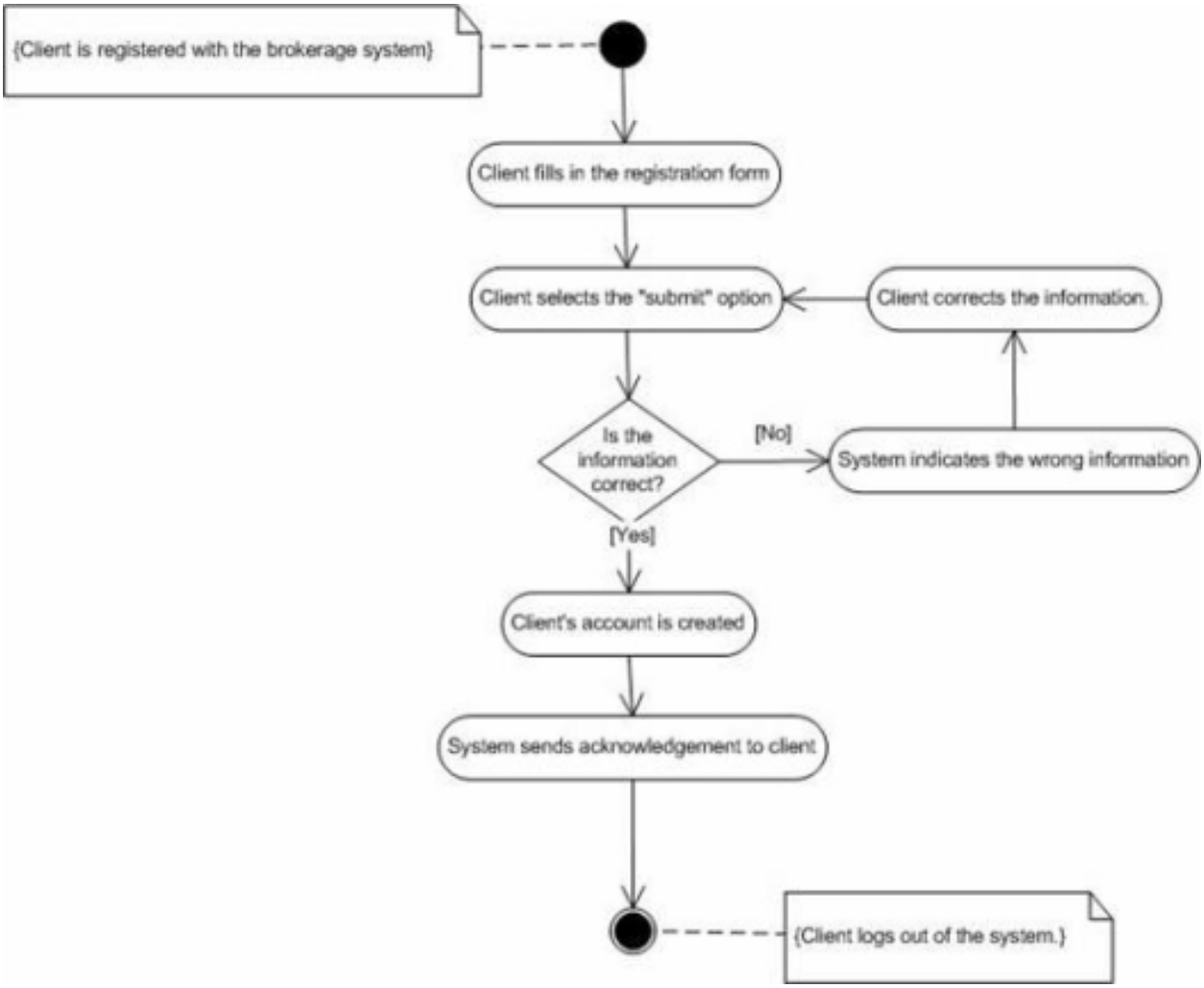
Some of the most common components of an activity diagram include:

- **Actions/Activity** - a step in the activity wherein the users or software perform a given task and it is symbolized with a round-edged rectangle.
- **Decision node** - a conditional branch in the flow that is represented with a diamond. It includes a single input and two or more outputs.
- **Control flows** - this is another name for the connectors that show the flow between steps in the diagram.
- **Start node** - symbolizes the beginning of the activity. This is represented with a black circle.
- **End node** - represents the final step in the activity. It's modeled with an outlined black circle.
- **Fork**. A black bar with one flow going into it and several leaving it. This denotes the beginning of parallel activity.
- **Join**. A black bar with several flows entering it and one leaving it. All flows going into the join must reach it before processing may continue. This denotes the end of parallel processing.

[42] UML Activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams(DFD) from structured development.

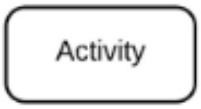
Activity Diagram for Course Registration System

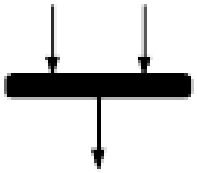
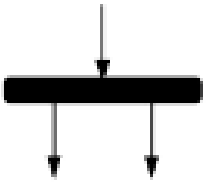
This activity diagram shows a typical event or class registration for a client. This diagram uses notes to give more details about the initial and final states. After filling out the registration form, the client submits the form to a validation loop that is represented as a decision in the flow. If the information is correct, the system creates an account for the client and lets the client know about the creation of the account.



Activity Diagram Symbols and Notation

Now that you've seen some examples, let's break down an activity diagram into its individual elements.

-  A black circle is the standard notation for an initial state before an activity takes place. It can either stand alone or you can use a note to further elucidate the starting point.
-  The black circle that looks like a selected radio button is the UML symbol for the end state of an activity. As shown in two examples above, notes can also be used to explain an end state.
- 

The activity symbols are the basic building blocks of an activity diagram and usually have a short description of the activity they represent.
- Arrows represent the direction flow of the flow chart. The arrow points in the direction of progressing activities.
-  A join combines two concurrent activities back into a flow where only one activity is happening at a time.
-  A fork splits one activity flow into two concurrent activities.
- 

Condition text is placed next to a decision marker to let you know under what condition an activity flow should split off in that direction.
-  A marker shaped like a diamond is the standard symbol for a decision. There are always at least two paths coming out of a decision and the condition text lets you know which options are mutually exclusive.

-  The final flow marker shows the ending point for a process in a flow. The difference between a final flow node and the end state node is that the latter represents the end of all flows in an activity.
-  The shape used for notes.

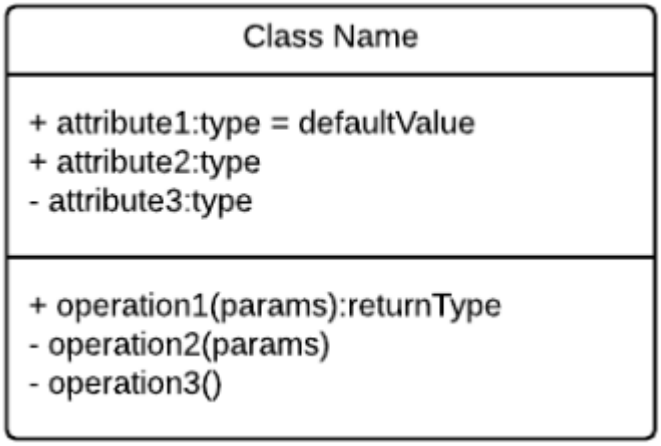
Analysis Level Class diagram

A class diagram is at the heart of UML. It represents the core purposes of UML because it separates the design elements from the coding of the system. UML was set up as a standardized model to describe an object-oriented programming approach. Since classes are the building block of objects, class diagrams are the building blocks of UML. The diagramming components in a class diagram can represent the classes that will actually be programmed, the main objects, or the interaction between class and object.

The class shape itself consists of a rectangle with three rows. The top row contains the name of the class, the middle row has the attributes of the class, and the bottom section expresses the methods or operations that the class may utilize. In a diagram, classes and subclasses are grouped together to show the static relationship between each object.

Basics of Class Diagrams

Class - Diagram Anatomy



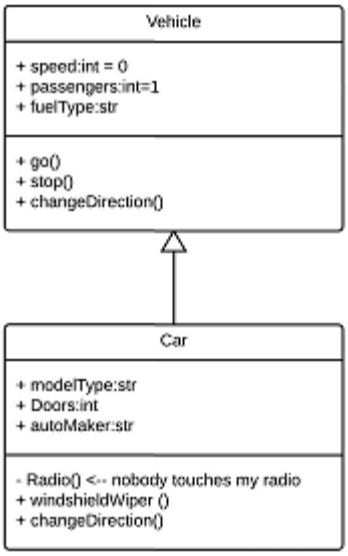
The class diagram is composed of three parts:

- Upper section - Name of the class - This section is always required whether you are talking about the classifier or an object.
- Middle Section - Attributes of the class - The attributes describe the variables that describe the qualities of the class. This is only required when describing a specific instance of a class.
- Bottom section - Class operations (methods) - Displayed in list format, each operation takes up its own line. The operations describe how a class can interact with data.

Object / Class Interactions in Class Diagrams

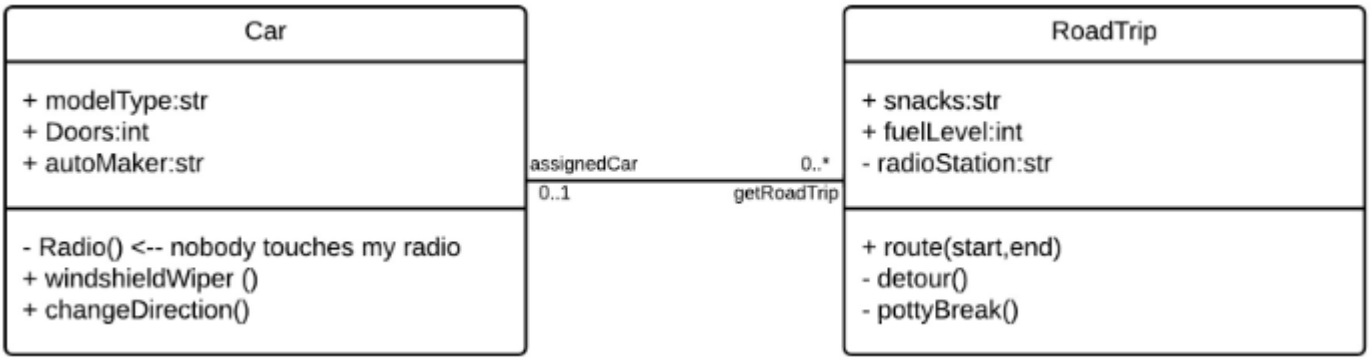
Interactions between objects and classes are an integral part of class diagrams.

Inheritance



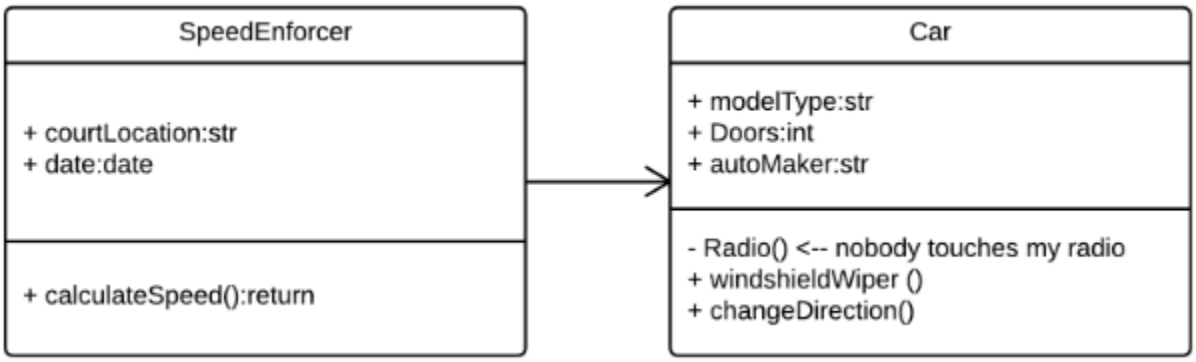
Inheritance is when a child object assumes all the characteristics of its parent object. For example, if we had the object vehicle, a child class Car would inherit all the attributes (speed, numbers of passengers, fuel) and methods (go(), stop(), changeDirection()) of the parent class in addition to the specific attributes(modelType, # of doors, autoMaker) and methods of its own class (Radio(), windshieldWiper(), ac/heat()). Inheritance is shown in a class diagram by using a solid line with a closed, hollow arrow.

Bidirectional Associations



Bidirectional associations are the default associations between two classes and are represented by a straight line between two classes. Both classes are aware of each other and of their relationship with each other. In the example above, the Car class and RoadTrip class are interrelated. At one end of the line the Car takes on the association of "assignedCar" with the multiplicity value of 0..1 which means that when the instance of RoadTrip exists, it can either have one instance of Car associated with it or no Cars associated with it. In this case, a separate Caravan class with a multiplicity value of 0..* is needed to demonstrate that a RoadTrip could have multiple instances of Cars associated with it. Since one Car instance could have multiple "getRoadTrip" associations-- in other words, one car could go on multiple road trips--the multiplicity value is set to 0..*

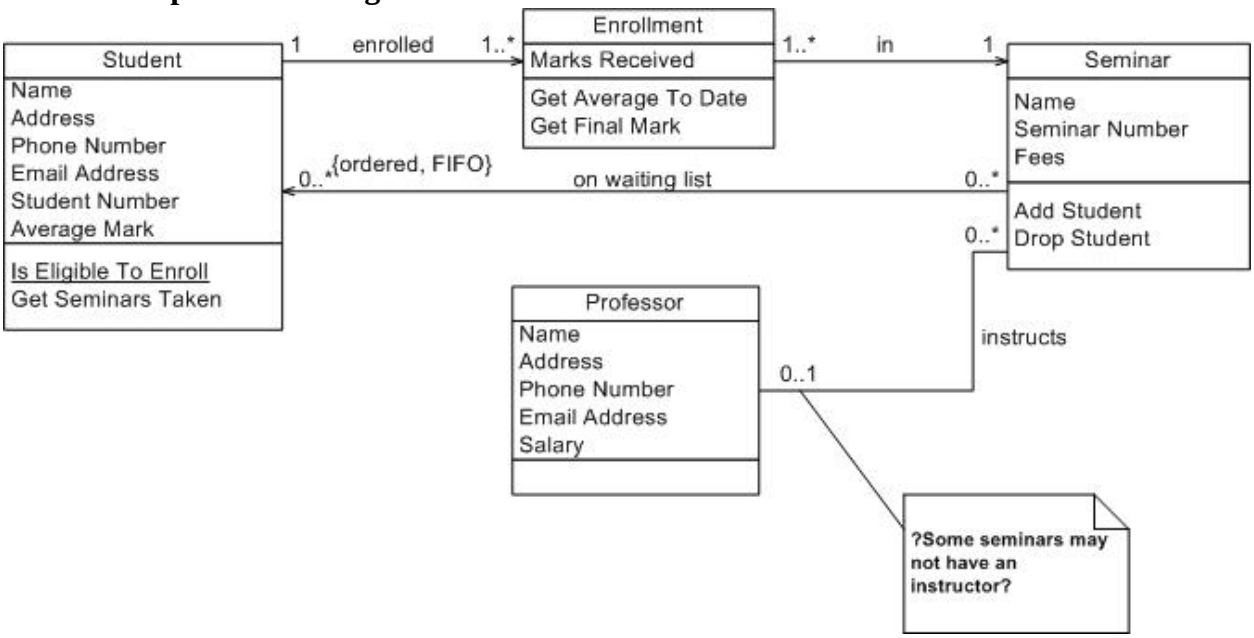
Unidirectional Association



A unidirectional association is drawn as a unbroken line with an open arrowhead pointing from the knowing class to the known class. In this case, on your road trip through Arizona you might run across a speed trap where a speed cam records your driving activity, but you won't know

about it until you get notification in the mail. It isn't drawn in the image but in this case the multiplicity value would be 0..* depending on how many times you drive by the speed cam.

Initial conceptual class diagram for Student Enrollment



User Interface Design

User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. UI brings together concepts from interaction design[43], visual design[44], and information architecture[45].

Choosing Interface Elements

Users have become familiar with interface elements acting in a certain way, so try to be consistent and predictable in your choices and their layout. Doing so will help with task completion, efficiency, and satisfaction.

[43] Interaction design focuses on creating engaging interfaces with well thought out behaviors. Understanding how users and technology communicate with each other is fundamental to this field.

[44] Visual design focuses on the aesthetics of a site and its related materials by strategically implementing images, colors, fonts, and other elements. A successful visual design does not take away from the content on the page or function. Instead, it enhances it by engaging users and helping to build trust and interest in the brand.

[45] Information architecture (IA) focuses on organizing, structuring, and labeling content in an effective and sustainable way. The goal is to help users find information and complete tasks. To do this, you need to understand how the pieces fit together to create the larger picture, how items relate to each other within the system.

Interface elements include but are not limited to:

- **Input Controls**: buttons, text fields, checkboxes, radio buttons, dropdown lists, list boxes, toggles, date field
- **Navigational Components**: breadcrumb, slider, search field, pagination, slider, tags, icons
- **Informational Components**: tooltips, icons, progress bar, notifications, message boxes, modal windows

Best Practices for Designing an Interface

Everything stems from knowing your users, including understanding their goals, skills, preferences, and tendencies. Once you know about your user, make sure to consider the following when designing your interface:

- **Keep the interface simple.** The best interfaces are almost invisible to the user. They avoid unnecessary elements and are clear in the language they use on labels and in messaging.
- **Create consistency and use common UI elements.** By using common elements in your UI, users feel more comfortable and are able to get things done more quickly. It is also important to create patterns in language, layout and design throughout the site to help facilitate efficiency. Once a user learns how to do something, they should be able to transfer that skill to other parts of the site.
- **Be purposeful in page layout.** Consider the spatial relationships between items on the page and structure the page based on importance. Careful placement of items can help draw attention to the most important pieces of information and can aid scanning and readability.
- **Strategically use color and texture.** You can direct attention toward or redirect attention away from items using color, light, contrast, and texture to your advantage.
- **Use typography to create hierarchy and clarity.** Carefully consider how you use typeface. Different sizes, fonts, and arrangement of the text to help increase scalability, legibility and readability.

- **Make sure that the system communicates what's happening.** Always inform your users of location, actions, changes in state, or errors. The use of various UI elements to communicate status and, if necessary, next steps can reduce frustration for your user.
- **Think about the defaults.** By carefully thinking about and anticipating the goals people bring to your site, you can create defaults that reduce the burden on the user. This becomes particularly important when it comes to form design where you might have an opportunity to have some fields pre-chosen or filled out.

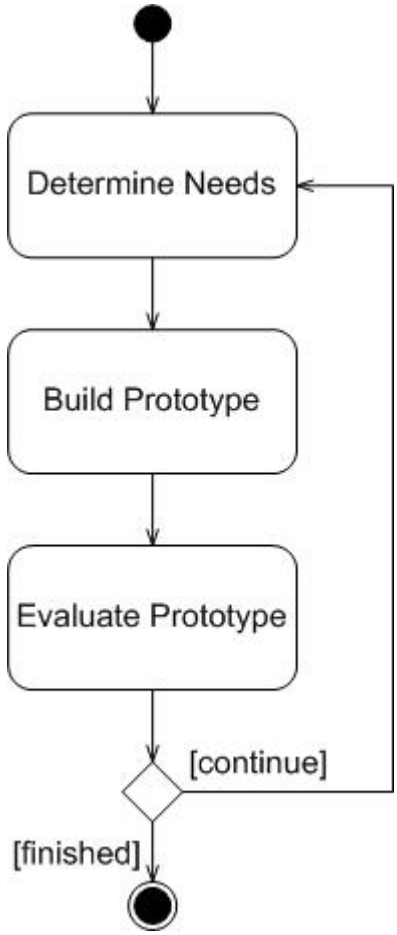
User Interface Prototyping

User interface (UI) prototyping is an iterative analysis technique in which users are actively involved in the mocking-up of the UI for a system. UI prototypes have several purposes:

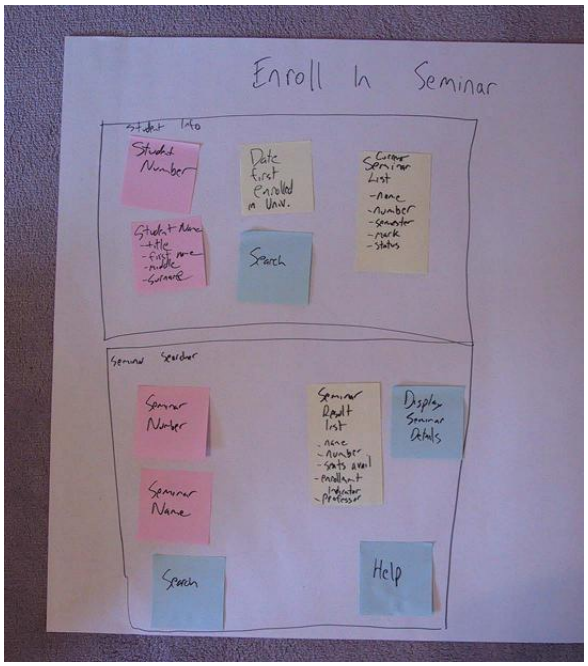
- As an analysis artifact that enables you to explore the problem space with your stakeholders.
- As a requirements artifact to initially envision the system.
- As a design artifact that enables you to explore the solution space of your system.
- A vehicle for you to communicate the possible UI design(s) of your system.
- A potential foundation from which to continue developing the system (if you intend to throw the prototype away and start over from scratch then you don't need to invest the time writing quality code for your prototype).

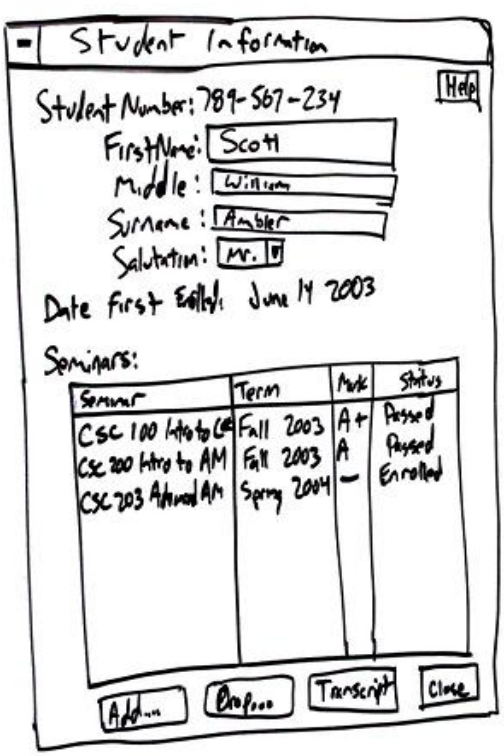
As you see in the activity diagram depicted in the Figure below there are four high-level steps in the UI prototyping process. The first step is to analyze the user interface needs of your users. User interface modeling moves from requirements definition into analysis at the point you decide to evolve all or part of your essential user interface prototype into a traditional UI prototype. This implies you convert your hand-drawings, flip-chart paper, and sticky notes into something a little more substantial. You begin this process by making platform decisions which in effect is an architectural decision. For example, do you intend to deploy your system so it runs in an Internet browser, as an application with a windows-based graphical user interface (GUI), as a cross-platform Java application, or as a mainframe-based set of "green screens?" Different platforms lead to different prototyping tools, for a browser-based application, you need to use an HTML-

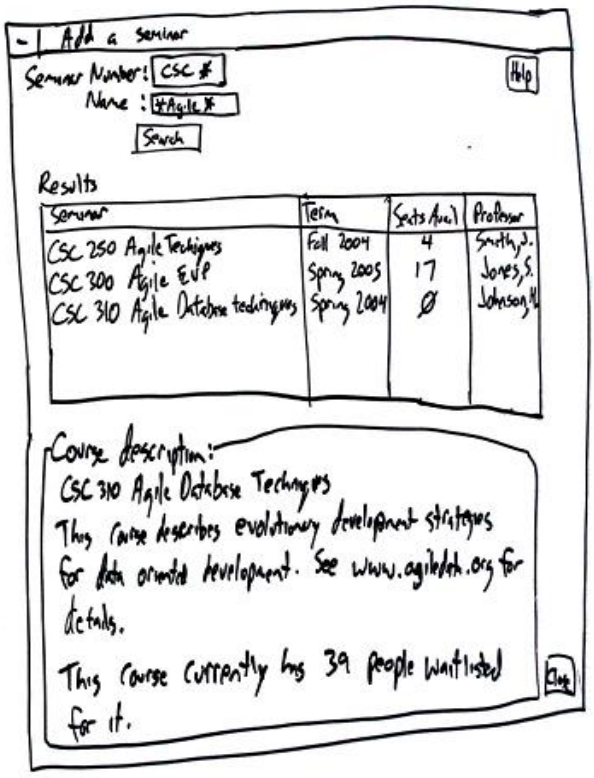
development tool, whereas a Java-based application would require a Java development tool and a different approach to the user interface design.



While you're determining the needs of your stakeholders you may decide to transform your essential user interface prototypes, if you created them to begin with, with sketches. The first figure shown below depicts an essential UI and the latter figure shows a sketch of two potential screens or HTML pages based on that prototype.


Enroll In Seminar
Student Info
Student Number
Date first enrolled in UNW.
Current Seminar List
- name
- number
- semester
- mark
- status
Search
Seminar Search
Seminar Number
Seminar Name
Seminar Result list
- name
- number
- seats avail
- enrollment
- instructor
- professor
Display Seminar Details
Search
Help


Student Information
Help
Student Number: 789-567-234
FirstName: Scott
Middle: William
Surname: Amber
Salutation: Mr.
Date First Enrolled: June 14 2003
Seminars:
Seminar
Term
Mark
Status
CSC 100 Intro to CE
CSC 200 Intro to AM
CSC 203 Advanced AM
Fall 2003
Fall 2003
Spring 2004
A+
A
—
Passed
Passed
Enrolled
Add...
Drop...
Transcript
Close


Add a Seminar
Seminar Number: CSC #
Name: *Agile*
Search
Help
Results
Seminar
Term
Seats Avail
Professor
CSC 250 Agile Techniques
Fall 2004
4
Smith, J.
CSC 300 Agile EVP
Spring 2005
17
Jones, S.
CSC 310 Agile Database techniques
Spring 2004
0
Johnson, M.
Course description:
CSC 310 Agile Database Techniques
This course describes evolutionary development strategies for data oriented development. See www.agiledata.org for details.
This course currently has 39 people waitlisted for it.
Close

Chapter Four: SYSTEM DESIGN

Introduction

This chapter will introduce for students about the proposed system design[46]specification. It describes the system at the architectural level, including subsystems and their services, hardware mapping, data management, access control, global software control structure, and boundary conditions[47].The system design specification documents the high-level system design and the low-level detailed design specifications.

Purpose of the System Design Document(SDD)

The SDD tracks the necessary information required to effectively define the proposed system architecture and it should give guidance on the architecture of the system to be developed.

Scope

Here you have to describe your design scope which is actually done in the requirement engineering. Just in case if you minimize the scope you have to reason out.

Architectural Design

Architectural design is the process of defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security, and manageability. It involves a series of decisions based on a wide range of factors, and each of these decisions can have considerable impact on the quality, performance, maintainability, and overall success of your system.

Software architecture encompasses the set of significant decisions about the organization of a software system including the selection of the structural elements and their interfaces by which the system is composed; behavior as specified in collaboration among those elements;

[46]System Design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements.

[47] The system design specification should define a virtual machine that implements all requirements in the analysis, and it should provide a foundational guide for further implementation details all the way to an executable solution.

composition of these structural and behavioral elements into larger subsystems; and an architectural style that guides this organization. Software architecture also involves functionality, usability, resilience, performance, reuse, comprehensibility, economic and technology constraints, tradeoffs and aesthetic concerns.

Architectural goals and constraints

In this section, you have to describe the software requirements and objectives that have some significant impact on the architecture, for example, safety, security, privacy, use of an off-the-shelf product, portability, distribution, and reuse. It also captures the special constraints that may apply: design and implementation strategy, development tools, team structure, schedule, legacy code, and so on.

For example:

Let's consider once again Hawassa University Integrated system (Online Portal).There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

1. The existing course system at HU must be accessed to retrieve all course information for the current semester.
2. All student, instructor, and registrar functionality must be available from both local campus PCs and remote PCs with internet dial up connections.
3. The online portal must ensure complete protection of data from unauthorized access. All remote accesses are subject to user identification and password control.
4. The page itself has server side security which is actually encrypted before transmitting over the internet to avoid Man in the middle attack[48].
5. The online portal will be implemented as a client-server system. The client portion resides on PCs and the server portion must operate on the HU- Windows Server.

[48] Man in the middle attack is like an eavesdropping. Data is sent from point A (computer) to point B (server/website), and an attacker can get in-between these transmissions. They then set up tools programmed to "listen in" on transmissions, intercept data that is specifically targeted as valuable, and capture the data.

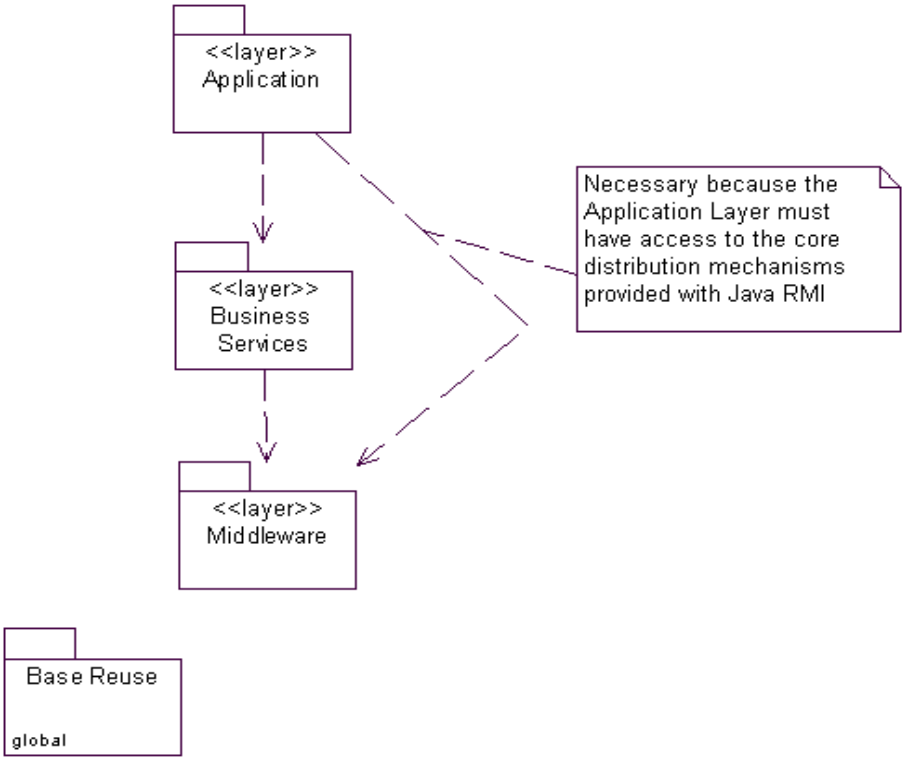
Logical View of the Architecture

Here you have to describe the most important classes, their organization in service packages and subsystems, and the organization of these subsystems into layers. Also describes the most important use-case realizations, for example, the dynamic aspects of the architecture. Design level class diagrams have to be included also to illustrate the relationships between architecturally significant classes, subsystems, packages and layers.

For example:

The logical view of the HU course registration system is comprised of three main packages: User Interface, Business Services, and Business Objects. The User Interface package contains classes for each of the forms that the actors use to communicate with the system. Boundary classes exist to support login, maintaining of student and instructor info, assign courses, submitting grade and viewing student status report cards.

- The Business Services Packages contains control classes for interfacing with the controlling student registration, and managing the student evaluation.
- The Business Objects Package includes entity classes for the university artifacts (i.e. course offering, schedule) and boundary classes for the interface with the Course Catalog System.



Application layer

This application layer has all the boundary classes that represent the application screens that the user sees. This layer depends upon the Process Objects layer; that straddles the separation of the client from mid-tier.

Business Services layer

The Business Services process layer has all the controller classes that represent the use case managers that drive the application behavior. This layer represents the client-to-mid-tier border. The Business Services layer depends upon the Process Objects layer; that straddles the separation of the client from mid-tier.

Middleware layer

The Middleware layer supports access to Relational DBMS and OODBMS.

Base Reuse

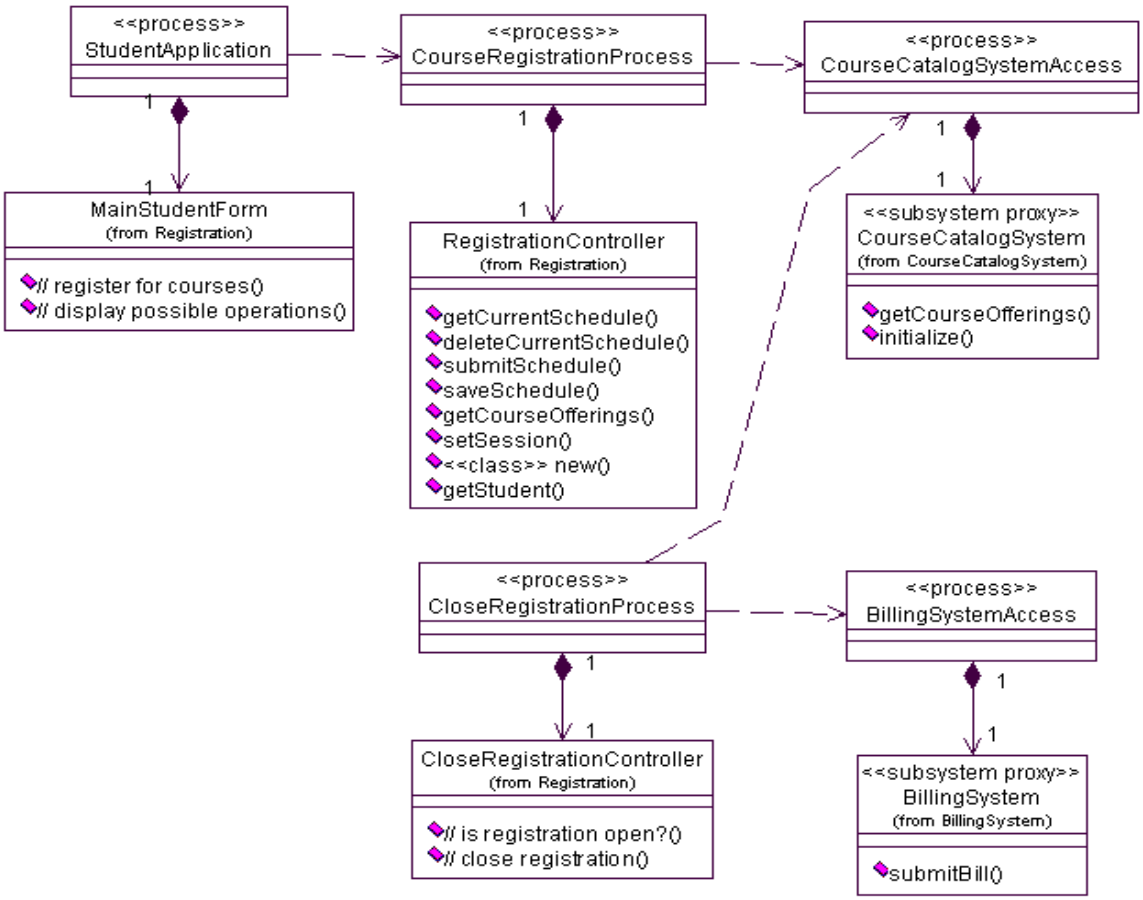
The Base Reuse package includes classes to support list functions and patterns.

Process View

Here you have to describe a process view of the architecture. It describes the tasks (procedures and threads) involved in the system's execution, their interactions and configurations. It also describes the allocation of objects and classes to tasks.

The Process Model illustrates the course registration classes organized as executable processes. Processes exist to support student registration, instructor functions, registration closing, and access to the external Billing System and Course Catalog System.

Processes



Process View depicted Using Design Level Class diagram

1. **CourseCatalogSystemAccess**

This process manages access to the legacy Course Catalog System. It can be shared by multiple users registering for courses. This allows for a cache of recently retrieved courses and offerings to improve performance.

Analysis Mechanisms:

Legacy Interface

Requirements Traceability:

- Design Constraints: The system shall integrate with existing legacy system (course catalog database).

2. **CourseCatalog**

The entire catalog of all courses and course offerings offered by the university including those from previous semesters.

This class acts as an adapter. It works to makes sure the CourseCatalogSystem can be accessed through the ICourseCatalog interface to the subsystem.

3. ***CourseRegistrationProcess***

There is one instance of this process for each student that is currently registering for courses.

4. ***RegistrationController***

This supports the use case allowing a student to register for courses in the current semester. The student can also modify or delete course selections if changes are made within the add/drop period at the beginning of the semester.

Analysis Mechanisms:

- Distribution

5. ***StudentApplication***

Manages the student functionality, including user interface processing and coordination with the business processes.

There is one instance of this process for each student that is currently registering for courses.

1. **MainStudentForm**

Controls the interface of the Student application. Controls the family of forms that the Student uses.

2. ***BillingSystemAccess***

This process communicates with the external Billing System to initiate student billing.

3. ***CloseRegistrationProcess***

The Close Registration process is initiated at the end of the registration time period. This process communicates with the process controlling access to the Billing System.

4. ***BillingSystem***

The Billing System supports the submitting of student bills for the courses registered for by the student for the current semester.

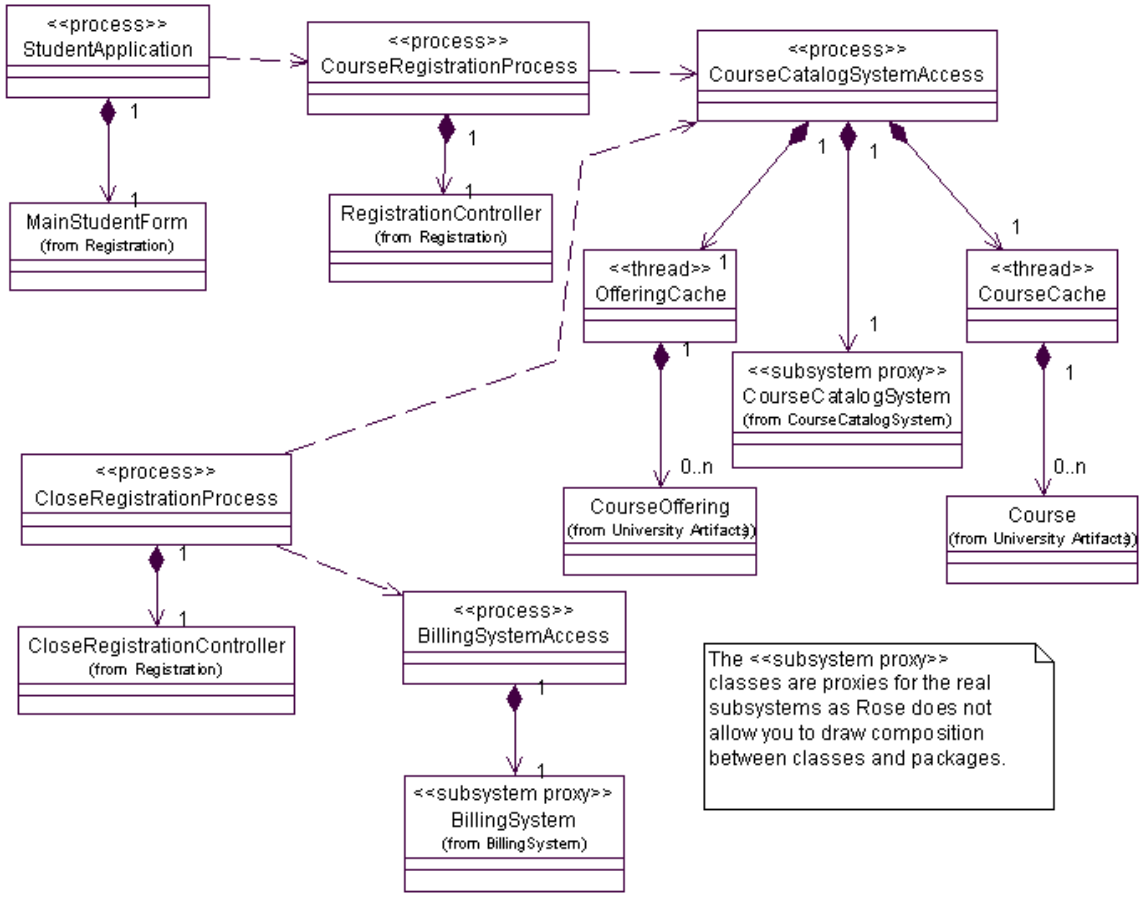
Analysis Mechanisms:
- Legacy Interface

5. ***CloseRegistrationController***

The Close Registration Controller controls access to the Billing System.

Analysis Mechanisms:
- Distribution

2.4 Process to Design Elements



I. CourseCache

The Course Cache thread is used to asynchronously retrieve items from the legacy Course Catalog System.

II. OfferingCache

The OfferingCache thread isused to asynchronously retrieve items from the legacy Course Catalog System.

III. Course

A class offered by the university.

Analysis Mechanisms:

- Persistency

- Legacy Interface

IV. CourseOffering

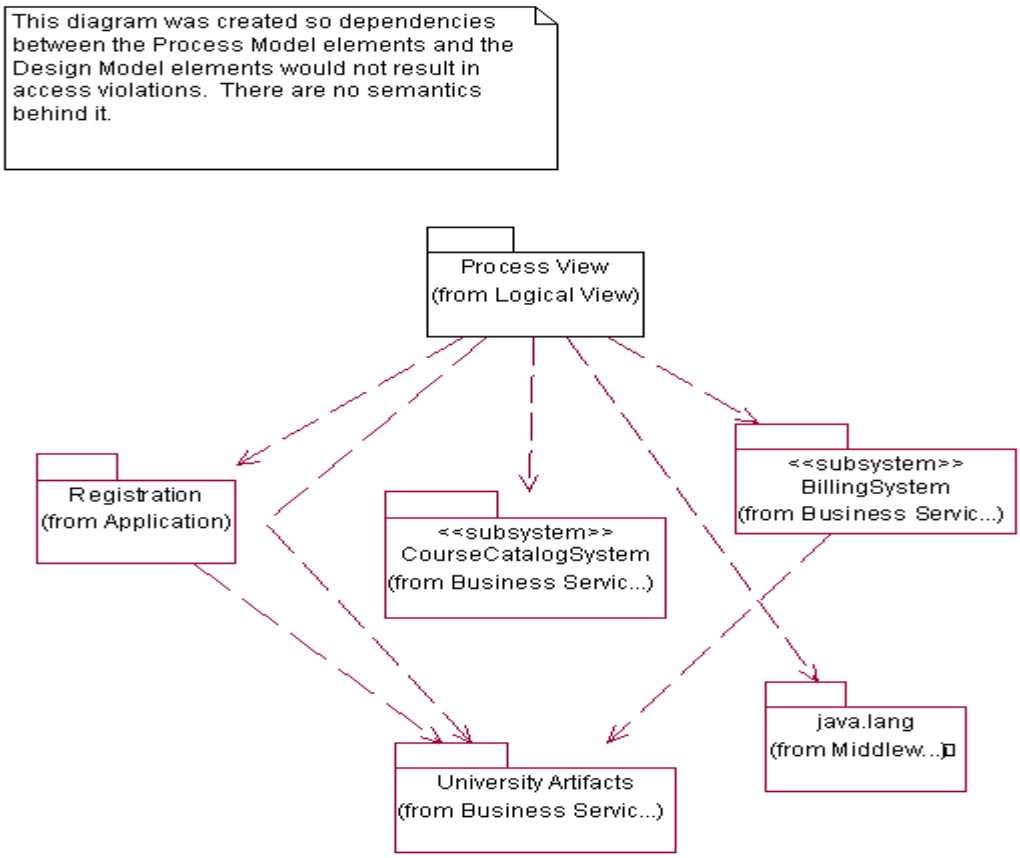
A specific offering for a course, including days of the week and times.

Analysis Mechanisms:

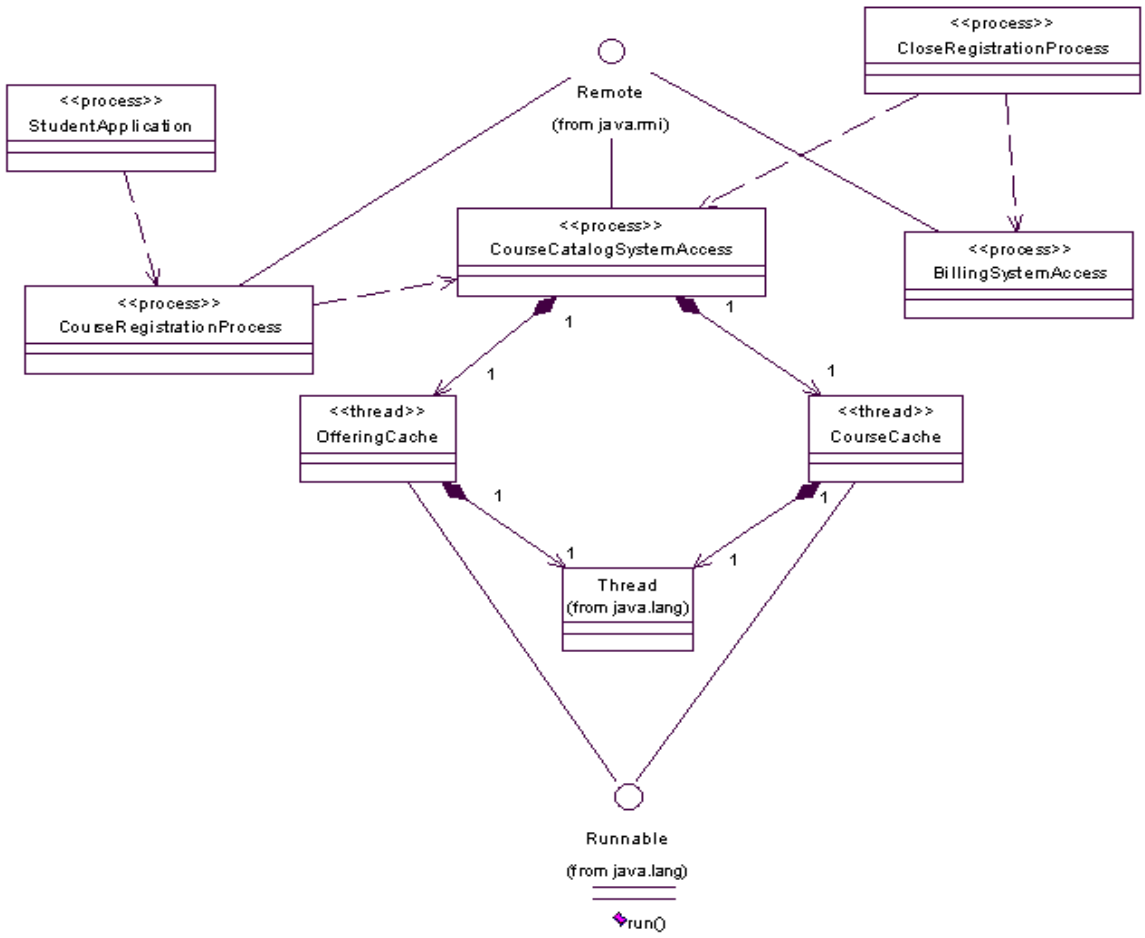
- Persistency

- Legacy Interface

2.5 Process Model to Design Model Dependencies



2.6 Processes to the Implementation



Remote

- The Remote interface serves to identify all remote objects. Any object that is a remote object must directly or indirectly implement this interface. Only those methods specified in a remote interface are available remotely.
- Implementation classes can implement any number of remote interfaces and can extend other remote implementation classes.

Runnable

- The *Runnable* interface should be implemented by any class whose instances are intended to be executed by a thread. The class must define a method of no arguments called *run*.

- This interface is designed to provide a common protocol for objects that wish to execute code while they are active. For example, *Runnable* is implemented by class *Thread*.
- Being active simply means that a thread has been started and has not yet been stopped.

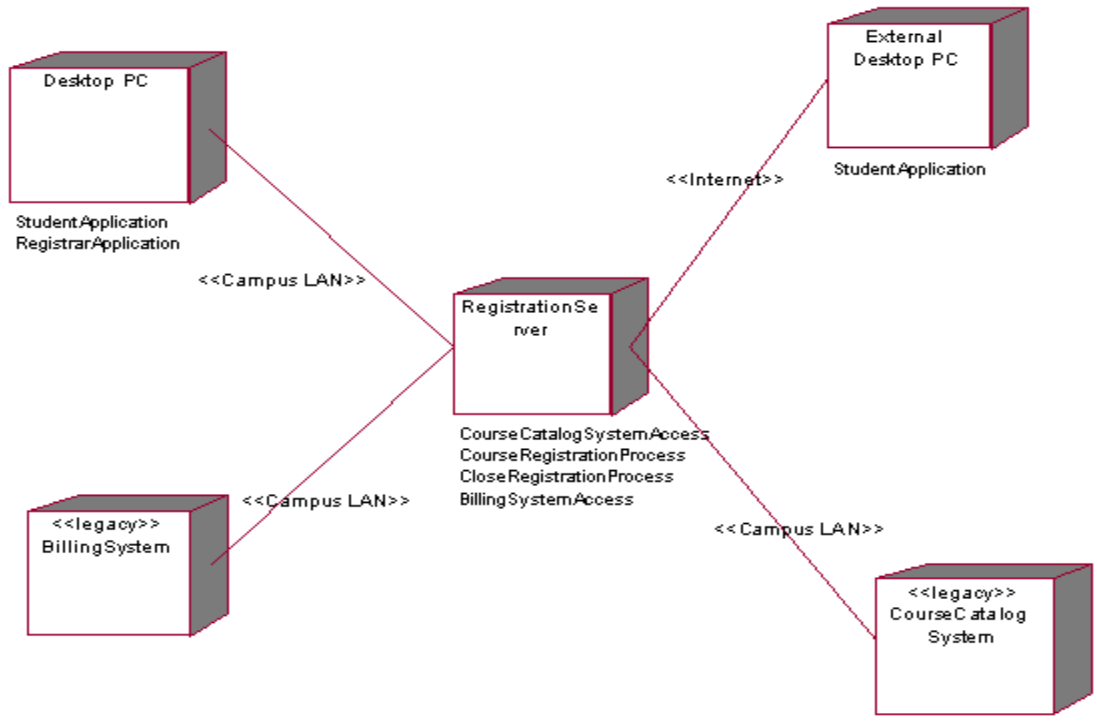
Thread

- A *thread* is a thread of execution in a program. The Java Virtual Machine allows an application to have multiple threads of execution running concurrently.
- Every thread has a priority. Threads with higher priority are executed in preference to threads with lower priority. Each thread may or may not also be marked as a daemon. When code running in some thread creates a new Thread object, the new thread has its priority initially set equal to the priority of the creating thread, and is a daemon thread if and only if the creating thread is a daemon.

Deployment View

A description of the deployment view of the architecture Describes the various physical nodes for the most typical platform configurations. Also describes the allocation of tasks (from the Process View) to the physical nodes.

This section is organized by physical network configuration; each such configuration is illustrated by a deployment diagram, followed by a mapping of processes to each processor.



External Desktop PC

Students register for courses using external desktop PCs which are connected to the College Server via internet dial up.

Desktop PC

Students register for courses via local Desktop PCs that are connected directly to the College Server via LAN. These local PCs are also used by professors to select course and submit student grades. The Registrar uses these local PCs to maintain student and professor information.

Registration Server

The Registration Server is the main campus UNIX Server. All faculty and students have access to the Server through the campus LAN.

Course Catalog

The Course Catalog System is a legacy system that contains the complete course catalog. Access to it is available via the College Server and LAN.

Billing System

The Billing System (also called the Finance System) is a legacy system that generates the student bills each semester.

Size and Performance

The chosen software architecture supports the key sizing and timing requirements, as stipulated in the Supplementary Specification [15]:

- The system shall support up to 2000 simultaneous users against the central database at any given time, and up to 500 simultaneous users against the local servers at any one time.
- The system shall provide access to the legacy course catalog database with no more than a 10 second latency.
- The system must be able to complete 80% of all transactions within 2 minutes.
- The client portion shall require less than 20 MB disk space and 32 MB RAM.

The selected architecture supports the sizing and timing requirements through the implementation of a client-server architecture. The client portion is implemented on local campus PCs or remote dial up PCs. The components have been designed to ensure that minimal disk and memory requirements are needed on the PC client portion.

Quality

The software architecture supports the quality requirements, as stipulated in the supplementary specification:

- The desktop user-interface shall be Windows 7/8/10 compliant.
- The user interface of the HU portal course registration System shall be designed for ease-of-use and shall be appropriate for a computer-literate user community with no additional training on the System.

- Each feature of the HU portal course registration System shall have built-in online help for the user. Online Help shall include step by step instructions on using the System. Online Help shall include definitions for terms and acronyms.
- The HU portal course registration system shall be available 24 hours a day, 7 days a week. There shall be no more than 4% down time.
- Mean Time Between Failures shall exceed 300 hours.
- Upgrades to the PC client portion of HU portal course registration shall be downloadable from the Windows Server over the internet. This feature enables students to have easy access to system upgrades.

Database Design

A well designed database gives users access to essential information. By the following database design principles, you can design a database that performs well and adapts to future needs.

The database design process

A well-structured database

- Saves disk space by eliminating redundant data.
- Maintains data accuracy and integrity.
- Provides access to the data in useful ways.

Designing an efficient, useful database is very crucial in software development. To achieve that you have include the following phases:

Requirement analysis, or identifying the purpose of your database

Understanding the purpose of your database will inform your choices throughout the design process. Make sure you consider the database from every perspective. For instance, if you were making a database for a student course registration system, you'd want to consider the ways in which both students and instructors would need to access the data.

Here are some ways to gather information before creating the database:

- Interview the people who will use it

- Analyze the business forms, such as student registration form
- Explore any existing data systems (including physical and digital files)

Organizing data into tables

The next step is to lay out a visual representation of your database. Within a database, related data are grouped into tables, each of which consists of rows (also called tuples) and columns, like a spreadsheet. To convert your lists of data into tables, start by creating a table for each type of entity, such as students, courses, instructors, and semester. Each row of a table is called a record. Records include data about something or someone, such as a particular student. By contrast, columns (also known as fields or attributes) contain a single type of information that appears in each record, such as the addresses of all the instructors listed in the table.

First Name	Last Name	Age	Postal Code
Natnael	Gonfa	30	34760
Ayenew	Yifru	32	97453
Mulugeta	Mehari	30	64829

To keep the data consistent from one record to the next, assign the appropriate data type to each column. Common data types include:

- CHAR - a specific length of text
- VARCHAR - text of variable lengths
- TEXT - large amounts of text
- INT - positive or negative whole number
- FLOAT, DOUBLE - can also store floating point numbers
- BLOB - binary data

Some database management systems also offer the Autonumber/auto-increment data type, which automatically generates a unique number in each row.

Specifying primary keys and analyzing relationships

With your database tables now converted into tables, you are ready to analyze the relationships between those tables. Cardinality refers to the quantity of elements that interact between two

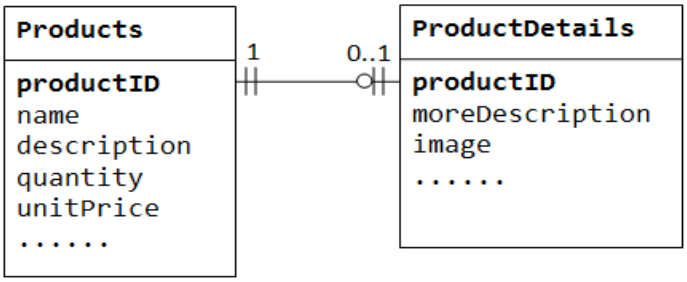
related tables. Identifying the cardinality helps make sure you've divided the data into tables most efficiently. Each entity can potentially have a relationship with every other one, but those relationships are typically one of three types:

One-to-one relationships

When there's only one instance of Entity A for every instance of Entity B, they are said to have a one-to-one relationship (often written 1:1). You can indicate this kind of relationship in an ER diagram with a line with a dash on each end:

For example,

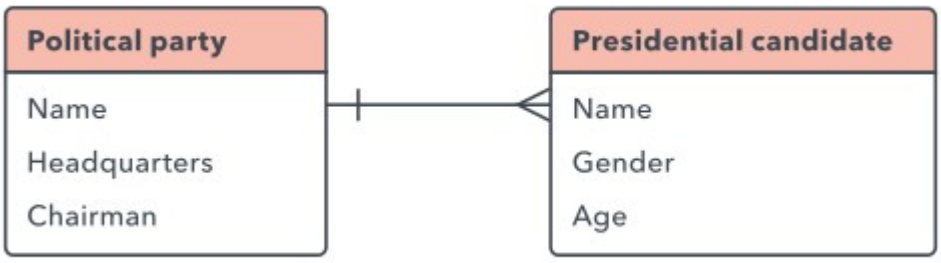
In a "product sales" database, a product may have optional supplementary information such as image, moreDescription and comment. Keeping them inside the Products table results in many empty spaces (in those records without these optional data). Furthermore, these large data may degrade the performance of the database. Instead, we can create another table (say `ProductDetails`, `ProductLines` or `ProductExtras`) to store the optional data. A record will only be created for those products with optional data. The two tables, `Products` and `ProductDetails`, exhibit a *one-to-one relationship*.



One-to-many relationships

These relationships occur when a record in one table is associated with multiple entries in another. For example, a single customer might have placed many orders, or a patron may have multiple books checked out from the library at once:[49]

[49] One-to-many (1:M) relationships are indicated with what's called "Crow's foot notation," as in the example above:

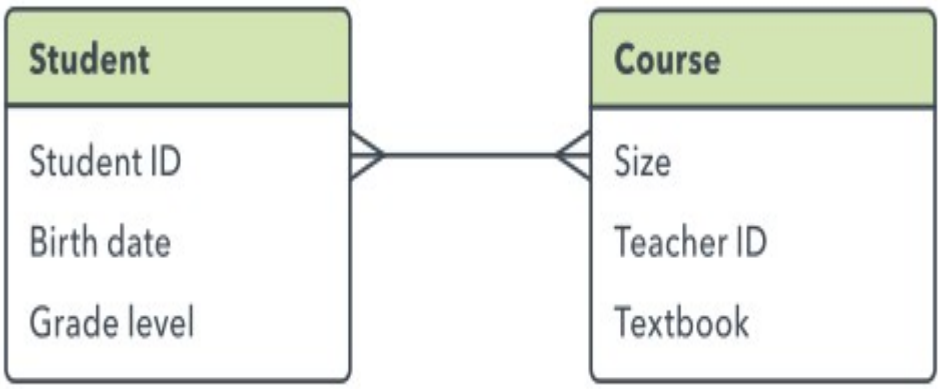


To implement a 1:M relationship as you set up a database, simply add the primary key from the "one" side of the relationship as an attribute in the other table. When a primary key is listed in another table in this manner, it's called a foreign key. The table on the "1" side of the relationship is a considered a parent table to the child table on the other side.

Many-to-many relationships

When multiple entities from a table can be associated with multiple entities in another table, they are said to have a many-to-many (M:N) relationship. This might happen in the case of students and classes, since a student can take many classes and a class can have many students.

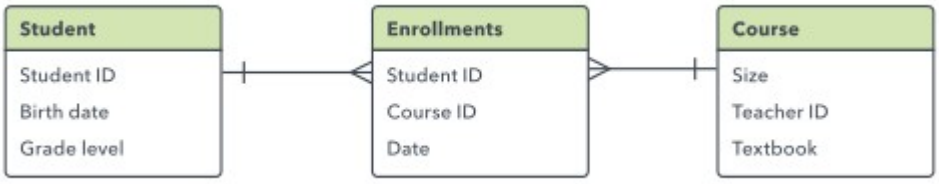
In an ER diagram, these relationships are portrayed with these lines:



Unfortunately, it's not directly possible to implement this kind of relationship in a database. Instead, you have to break it up into two one-to-many relationships. To do so, create a new entity between those two tables. In the diagram mentioned there exists many to many relationship

between student and course. So create a new entity; you might call it "Enrollments"[50] that link the two entities.

Each record in the link table would match together two of the entities in the neighboring tables (it may include supplemental information as well). For instance, a link table between students and classes might look like this:



Normalizing to standardize the tables

Once you have a preliminary design for your database, you can apply normalization rules to make sure the tables are structured correctly. Think of these rules as the industry standards.[51] Each form, or level of normalization, includes the rules associated with the lower forms.

First normal form

The first normal form (abbreviated as 1NF) specifies that each cell in the table can have only one value, never a list of values.

Let's consider the University Enrollment database:

When thinking about the logical normalization process we first look at all of the data required to accomplish a task. Now consider the following object (data file or table), named ENROL, that contains the data fields (attributes) required to enroll the student in a class.

ENROLL ***(Course_Code, Course_Description, Student_Number, Student_Name, Address, City, Zipcode, Course_level, Course_Start_Date, Course_End_Date, Class_Room_Number,***

[50] The new entity "Enrollments" shows the data of students registered for a course. The entity is also called associative entity.
[51] Not all databases are good candidates for normalization. In general, online transaction processing (OLTP for short) databases, in which users are concerned with creating, reading, updating, and deleting records, should be normalized.

Building_Number, Building_Name, Building_Address, Lecturer_Number, Lecturer_Name, Department_Code, Department_Name)

Consider the following (part of a) database:

Enroll			
Course Code	**Course Description**	**Student Number**	**Student Name**
CoSc 2071	**Fundamental of database system**	**CS/0001/07** **CS/0006/07** **CS/0003/07**	**Abebe** **Ermiyas** **Daniel**
CoSc 2072	**Advanced database system**	**CS/0005/07** **CS/0006/07** **CS/0007/07**	**Gebeyehu** **Ermiyas** **Melat**

Each course code can have any number of students in it, so the students' information makes a group what is called a repeating group. Data cannot be stored or processed in a database when it is in this form. What we must have is one record containing ***all the data*** for each student who is enrolled in a class. There can be no "gaps" in the data when stored in a file.

The following table (data file) illustrates the data in First Normal Form (1NF)

Enroll			
Course Code	Course Description	Student Number	Student Name
CoSc 2071	Fundamental of database system	CS/0001/07	Abebe
CoSc 2071	Fundamental of database system	CS/0006/07	Ermiyas
CoSc 2071	Fundamental of database system	CS/0003/07	Daniel
CoSc 2072	Advanced database system	CS/0005/07	Gebeyehu
CoSc 2072	Advanced database system	CS/0006/07	Ermiyas
CoSc 2072	Advanced database system	CS/0007/07	Melat

Converting to 1NF basically requires that we "flatten" the database table given above so that each row (record) contains no repeating groups. Only one value per field can be entered and no "gaps" exist in the data.

Second normal form

The problem with 1NF is that there is redundancy with respect to entering all of the data into a computer for each and every course in which you enroll. For example, your name, address, etc., will have to be entered for each course that you take. If you take four courses, your name will have to be entered four times. Developing a logical method of avoiding the entry of your name four times leads to the definition of what is called Second Normal Form (2NF).

We must next introduce the concept of a "**KEY**" field. A key field is the one that is used to ***uniquely identify*** **each** record in a data file.

For example, the Student_Number field can be used to uniquely identify each student's record in a student table. However, since one student may be enrolled in more than one class each quarter, the Student_Number field alone is not sufficient to uniquely identify each record in the ENROL file illustrated above. The combination of the Student_Number field and the Course_Code field forms a unique combination and can therefore be considered as the key field for the ENROLL table.

A relation is in 2NF if, and only if, it is in 1NF and every non-key attribute (field) is fully functionally dependent upon the key field.[52]For example, we could remove the name, address, etc. columns into a relation named STUDENT and remove them from the ENROLL object. The result will give two relations (tables):

STUDENT (Student_Number, Student_Name, Address, City, State, Zipcode)

ENROLL (Student_Number, Course_Code, Course_Description, Course_level, Course_Start_Date, Course_End_Date, Class_Room_Number, Building_Number, Building_Name, Building_Address, Lecturer_Number, Lecturer_Name, Department_Code, Department_Name)

[52]If a relation is in 2NF it means that all data attributes (fields) that are not used to *uniquely* identify records or rows in a table should not be entered into the database more than once.

Here we see that the student name, address, etc., are functionally dependent upon the student number in the STUDENT file, and that the course description, start date, building name, etc., are functionally dependent upon the Student Number and the Course Code in the ENROLL file.

Third normal form

While getting the data files into 2NF is better than 1NF, there are still some problems with the form. For example, if the location of the course (class room) changes buildings, *all* records in the ENROLL file for that course will have to be updated. The building name and address are "transitively dependent" upon the building number. Resolving the "transitive dependency" leads us to Third Normal Form (3NF).

A relation is in 3NF if, and only if, it is in 2NF and no non-key fields are transitively dependent upon the key field(s). That is, one non-key attribute cannot be functionally dependent upon another non-key field.

Our example is clearly not in 3NF since the building name (non-key field) depends upon the building number (non-key field). The relation can be resolved into 3NF by dividing it into component relations, each meeting 3NF form.

Not only this, even the class description, start time, and start date are transitively dependent upon the class code, which is not considered a key field here because it forms only part of the key field for the ENROL table. We can also recognize that Lecturer name is functionally dependent upon the Lecturer code, which is not a key field. The building code and Lecturer code fields are not key fields because they are not used to uniquely identify each record in the ENROLL file.

ENROLL ***(Student_Number, Course_Code)***
BUILDING ***(Building_Number, Building_Name, Building_Address)***
COURSE ***(Course_Code, Course_Description, Course_level, Course_Start_Date, Course_End_Date, Class_Room_Number, Building_Number, Lecturer_Code)***
LECTURER ***(Lecturer_Code, Lecturer_Name, Department_Code, Department_Name)***
PROGRAM ***(Program_Code, Program_Description)***
STUDENT ***(Student_Number, Student_Name, Address, City, State, Zipcode)***

Note also that the LECTURER relation is not in 3NF since the Department Name is transitively dependent upon the Department Code. We resolve this into:

LECTURER ***(Lecturer_Code, Lecturer_Name, Department_Code)***
DEPARTMENT ***(Department_Code, Department_Name, Dept_Head)***

Chapter Five: CONCLUSION AND RECOMMENDATION

Conclusion

The conclusion[53] is intended to help the reader understand why your project work should matter to them after they have finished reading the paper[54]. A conclusion is not merely a summary of the main topics covered or a re-statement of your project problem, but a synthesis of key points. For most projects, one well-developed paragraph is sufficient for a conclusion, although in some cases, a two or three paragraph conclusion may be required.

When you write a conclusion for project, always remember that you have to make a summary of the content as well as the purpose that you have in mind without looking way too wooden or dry. Most conclusions of project tend to have a couple key elements. Of course there are some tactics that you should be using if you want to write a conclusion that is effective. Plus, there are some things that you should avoid as well.

When writing the conclusion to your paper, follow these general rules:

- State your conclusions in clear, and simple language. Re-state the purpose of your study then state how your project resolve issues in the existing system and how your system is different from other related projects.
- Do not simply reiterate your results or the activities done in the entire software development life cycle.
- Indicate opportunities for future project works[55]if you haven't already done so in the design and implementation section of your project work. Highlighting the need for further project works provides the reader with evidence that you have an in-depth awareness of the existing problem.

[53]Just as your introduction is the first impression your reader will have of your writing, your conclusion is the last. A good conclusion will show that you have successfully answered the question or completed the task set.
[54]The Conclusion section of a technical report gives the author one final chance to emphasize his/her most important points. Many readers know that authors often tend to hold back their findings until they get to the Conclusion, so the reader will just skip everything else and go right to the end of the paper. Your conclusion should be a “Mini-Me” version of your entire paper. It should also examine the greater significance of what you have done.
[55] This idea is required if and only if your documentation doesn’t have a recommendation section.

A good conclusion should:

- Be more than just a summary. It should be a thoughtful end to a piece of writing; for example, by applying what you have written to the outside world.
- Emphasize or reinforce your main ideas, but with your ideas restated in a fresh way: don't use the same language again. You should refer back to your introduction, either with key words or parallel concepts and images.
- Fit in with the rest of the assignment. Different types of writing require different types of conclusion. A short pie will probably not require extensive restatement of your main points, whereas a longer piece probably will.
- Perhaps include a provocative question or two;
- If appropriate, suggest results or consequences or make a call for some sort of action.
- Make predictions or suggest solutions, again if appropriate.

A good conclusion should NOT:

- Include completely new ideas. If they are important, include them in your main text.
- Be confident with what you say: avoid phrases such as: "I may not be an expert." or "At least this is my opinion"
- Focuson minor points.
- Qualifythe impact of any previous points.

The following mentioned are few tips on how to write a good conclusion for project work.

- Stick with a basic synthesis of information
- Bring things full circle: tie your project work together by directly linking your introduction with your conclusion.
- Close with logic
- Pose a question[56]: instead of handing the reader the conclusion, you are asking the reader to form his or her own conclusion.
- Make a suggestion

[56] This kind of conclusion is not always suitable for every project works.

Recommendation

Sometimes recommendations are included with a report's conclusion, although they serve different purposes[57]. Whereas a conclusion offers you the opportunity to summarize or review your report's main ideas, recommendations suggest actions to be taken in response to the findings of a report. You can regard recommendations as a prompt to action for your readers. As you have seen from your planning, your report structure should lead up to the recommendations and provide justification for them.

What makes a good recommendation? Effective recommendations:

- Describe a suggested course of action to be taken to solve a particular problem that can't addressed yet in the project;
- Are written as action statements without justification[58];
- Arestated in clear, and specific language;
- Shouldbe expressed in order of importance (i.e. it is better to start from the most important recommendation and proceeded to others based on their significance to the clients or users of the system);
- Arebased on the case built up in the body of the report; are written in parallel structure to the developed system.

A word of caution about writing recommendations: you should always consider your relationship with the reader first. If you have no authority to make recommendations, the reader may be hostile to their presence.

Example 1.In the data preprocessing step, we explored an approach for spell checking based on clustering but were not able to integrate that into the final recommendations. Such a step would be helpful in reducing the number of unique training queries.

Example 2. Given a new query we find the closest query in the training set based on modified string distance and base our models on that. For some queries, this does not work so well

[57] Whereas in your project documentation conclusion and recommendation are explicitly defined. Therefore, you can follow the format given by the school to write your documentation.
[58] These recommendations are forwarded to the project owners to properly implement the new software system to the organization.

because we are not able to find a suitably close query (preprocessed). Hence there is scope for trying different approaches to model the query strings (based on keywords) and try to make recommendations using the given SKU attributes.

Example 3: The benefits of the project risk management are enormous. A company can gain both financial and non-financial benefits if uncertain projects are handled proactively. Following are the recommendations to improve the performance towards a better software project risk management:

- Making risk management part of the project
- Identify risk management early in the project
- Communicate about risks
- Consider both threats and opportunities
- Priorities Risks
- Plan and implement risk response and
- Register project risk

References

A citation is a reference[59] to a published or unpublished source that you consulted and obtained information from while writing yourproject documentation. The way in which you document your sources depends on the writing style manual your school (university) offers to the documentation.

Citations show your readers where you obtained your material, provides a means of critiquing your study, and offers the opportunity to obtain additional information about the research problem under investigation. The act of citing sources is also a defense against allegations of plagiarism.

Properly citing the works of others is important because:

- Proper citation allows readers to locate the materials you used. Citations to other sources helps readers expand their knowledge on a topic.
- Citing other people's words and ideas indicates that you have conducted a thorough review of the literature on your topic and, therefore, you are operating from an informed perspective. This increases your credibility as the author of the work.
- Other researcher's ideas can be used to reinforce your arguments, or, if you disagree with them, can act as positions from which to argue an alternative viewpoint.
- Outside university, ideas are considered intellectual property and there can serious repercussions if you fail to cite where you got an idea.
- Ithelps avoid charges of plagiarism (academic theft) because it makes clear when you are using someone else's ideas and words.

There are two principal components to citing references

- The way you acknowledge, cite the source in your text.

[59] References usually come at the end of a text and should contain only those works cited within the text. Therefore, use the term 'References' to cover works cited, and 'Additional Bibliography' to refer to works read as general background.
A Bibliography is any list of references at the end of a text, whether cited or not. It includes texts you made use of, not only texts you referred to in your paper, but your own additional background reading, and any other articles you think the reader might need as background reading.

- The way you list your sources at the end of your work to enable identification(reference).

There are wide varieties of formatting to write referencesand cite them on the final project works and/or researches. The one selected from the school is API (American Psychological Association) style and every group must follow it as a standard reference list-writing format for thefinal project documentation.

In order to site the reference from the reference list in your documentation, you can use either the number (i.e. the reference number) in the list or name of the author (i.e. when the number of authors are greater than or equal to two you could be use the name of the first author) for the reference material. In this module, the team preferred to use reference list number to cite a reference[60] (s) in the module.

General Guide to Formatting a Bibliography:

- For a book:

Author[61] (s)[62]. Title of the book. City: Publisher, Edition, Date of publication.

EXAMPLE:James F. Kurose, Keith W. Ross. ComputerNetworking a Top Down Approach: Pearson, sixth Ed., 2013.

- For an encyclopedia:

Encyclopedia Title, Edition Date. Volume Number, "Article Title," page numbers.

EXAMPLE:The Encyclopedia Britannica, 1997. Volume 7, "Gorillas," pp. 50-51.

- For a scientific article:

Author, "Article Title." Name of magazine. Volume number, (Date): page numbers.

EXAMPLE:M. Blumenthal, D. Clark, “Rethinking the Design of the Internet: theEnd-to-end Arguments vs. the Brave New World,” ACM Transactions on InternetTechnology, Vol. 1, No. 1 (Aug. 2001), pp. 70–109.

- Internet (Published article):

[60] The number of references in one citation is more than one you could write all reference numbers by separating a comma.
[61] If the number of authors are, two you can write the names the two authors separating with comma.
[62] If the number of authors are more than two you can write the first author and follow with the text et. al. (meaning all the other) instead of listing all authors of the book and/o the article.

Author of message, (Date). Subject of message. Electronic conference or bulletin board (Online). Available e-mail: LISTSERV@ e-mail address

EXAMPLE:P. Biddle et.al.[63] (Nov. 2002,Washington, D.C.) "The Darknet and the Futureof Content Distribution," 2002 ACM Workshop on Digital Rights Management, http://crypto.stanford.edu/DRM2002/darknet5.doc.

- World Wide Web (online resources):

Title of the page, URL (Uniform Resource Locator or WWW address). Author (or item's name, if mentioned), date.

EXAMPLE: Advisory 2001–09: Statistical Weaknesses in TCP/IP InitialSequence Numbers,http://www.cert.org/advisories/CA-2001-09.htm, CERT, accessed on Oct 15, 2017.

Note: For more examples, please refer the reference section of this module.

[63] Authors of this article are P. Biddle, P. England, M. Peinado, B. Willman instead of writing all the authors we prefer P. Biddle et.al.

Appendix

An appendix contains supplementary material that is not an essential part of the text itself but which may be helpful in providing a more comprehensive understanding of the project problem or it is information that is too cumbersome to be included in the body of the paper. A separate appendix should be used for each distinct topic or set of data and always have a title descriptive of its contents.

The project must be able to stand alone without the appendices, and the paper must contain all information including tables, diagrams, and results necessary to understand the problem. The key point to remember when including an appendix is that the information is non-essential; if it were removed, the reader would still be able to comprehend the significance, validity, and implications of your work.

It is appropriate to include appendices for the following reasons:

- Including this material in the body of the paper that would render it poorly structured or interrupt the narrative flow;
- Information is too lengthy and detailed to be easily summarized in the body of the paper;
- Inclusion of helpful, supporting, or useful material would otherwise distract the reader from the main content of the paper;
- Provides relevant information or data that is more easily understood or analyzed in a self-contained section of the paper;
- Can be used when there are constraints placed on the length of your paper; and,
- Provides a place to further demonstrate your understanding of the project problem by giving additional details about a new or innovative method, technical details, or design protocols.

When considering whether to include content in an appendix, keep in mind the following points:

- It is usually good practice to include your raw data in an appendix
- Any tables and figures included in the appendix should be numbered as a separate sequence from the main paper.

- If you have more than three appendices, consider listing them on a separate page at the beginning of your paper.
- The appendix can be a good place to put maps, photographs, diagrams, sample codes and other images.
- An appendix should be streamlined and not loaded with a lot information

Never include an appendix that is not referred to in the text. All appendices should be summarized in your paper where it is relevant to the content. Appendices might also be arranged sequentially by the order they referenced in the text.

There are few rules regarding what type of material can be included in an appendix, but here are some common examples:

- Correspondence: if your project work included collaborations with others or outreach to others, then correspondence in the form of letters, memorandums, or copies of emails from those you interacted with could be included.
- Interview Transcripts: interviewing respondents is often used to gather information. The full transcript from an interview is important so the reader can read the entire dialog between project team and respondent[64].
- Non-textual elements
- Questionnaires or surveys
- Raw statistical data
- Project special hardware devices[65]
- Sample calculations
- Sample codes from the project

[64] When you conduct an interview questions and questioner in local languages, you could attach both the original and translated once.

[65] If you use special (system specific) hardware, it is better to state some description about the device in the appendix section

COMPUTER NETWORK DESIGN AND IMPLEMENTATION

Chapter I. Introduction (Similar to Software Development Module)

Chapter II. Network Needs Analysis

This chapter focuses on analyzing the needs of designing a new computer network infrastructure or modifying the existing one in the organization. This chapter is organizing as follows; it begins with defining the data types in the first section, next to this data sources, and Numbers of Users and Priority Levels are discussed. The chapter also includes storage, transmission-speed, security and reliability requirements of the existing and proposed network infrastructures. Finally, the chapter discussed about load variation estimation and description of existing network infrastructure[66] (if exist) in the organization. The chapter gives a clear introduction to the team members about contents included in the documentation with supported examples. Then the team uses this understanding to analyze the network requirement for the proposed system for their final project.

Data Types

This section of the chapter defines what kind of data to be expected in the network to be stored, processed and transferred from one station to the other. Therefore, the team can define types of data that are processed, stored and transferred in the company network infrastructure. The types of data served by the network will be reports (annual, quarter, monthly and daily), bulletins, accounting information, student information, personnel profiles, medical history, web pages and other.

The majority of the data will be vary company to company depending on the business domain of the company; in mean while every company have a text (ASCII and non-ASCII) data, graphics, audio files and possibly huge amount of voice and video (primarily for PC-based teleconferencing, web files, and other files) data.

For example: let consider the ICT infrastructure in Tokuma IT Solutions (i.e. the company targeted to develop software for different clients and other IT related business in Ethiopia). By considering this, the company have the following data sources for the local network in the organization vicinity:

[66] Measuring the existing network infrastructure of the company by all network need assessment (analysis) measures to decide whether the existing network require advancement or not.

- Detail company's clients information (i.e. business requirements, term of agreements, contract letter and others)
- Human resource information of the organization
- Previous developed software to different organizations
- Inventory information of the company
- Materials that supports software development (audio files, textbooks, web files, video files and other files that supports programmers in the software development processes in the company)

Data Sources

A data source is simply the source of the data. It can be a file, a particular database on a DBMS, or even a live data feed. The data might be located on the same computer as the program, or on another computer somewhere on a network.

Data will be created as well as used at all end stations on the network (depending on the size of the organization, the organization can have a dedicated server to be a main data source for its users). The data will be produced by software applications in different operating systems and platforms, web services, different programming languages and IDEs (Integrated Development Environment) and office application software (like Microsoft Office 2016, Office365, OpenOffice and other), audio and video editing software (media player, adobe Photoshop, and other software). Other data sources to be supported on at least a limited basis will Window10 Accessories (Paint, 3D paint, Notepad, etc.), Video conferencing, and NetMeeting (Viber, Facebook, WeChat etc.). Therefore, the team can clearly define the data sources of the proposed network infrastructure in the organization.

Numbers of Users and Priority Levels

The Priority is derived from the impact and the urgency of accessing information in the network, based on the context of an organization. The allocation of a priority code determines how the user is being taken care of by the tool, the support staff and access to different services in the company network.

Therefore, the team can categorize users of the infrastructure into different groups based the rules and regulations of the organization and define their priority[67] based on the organization structure, privilege level and resources in the organization.

For Example: consider the network infrastructure in the Hawassa Senior Secondary and Preparatory School. In the school network, the users will be administrators (i.e. School principal and vice-principal), secretaries, teachers, students, students'-family and other workers. The maximum estimated users on the network at any given time are 260[68](i.e. 200 students, 40 teachers and 20 other users in the school).

Three priority levels will be supported: administrators and teachers (top priority), students and other workers (medium priority), and students' family (low priority). Note that these designations do not correspond to administrative levels in the school; rather, they are network service levels (i.e. services delivered to the staffs and students). Network management processes (a packages in Simple Network Management Protocol) will receive top priority service; most network processes will receive medium-priority service; a few processes (e.g., e-mail transfers, backup, web access (WWW), FTP etc.) will be given low-priority service. It should be noted that network management will usually consume a small amount of the available bandwidth; this means that management and user processes will usually enjoy identical support. Background processes will also usually receive more than adequate service, but they will be delayed as needed to maintain support for management and user services.

Transmission Speed Requirements

Since the network is transparent to all users of the organization, the team is expected to calculate the network bandwidth requirements for both LAN[69] and WAN in the organization. To estimate the bandwidth requirements of the organization properly, the team can consider the a number of

[67] The most common way of defining a priority is defining the priority matrix for all users of the network by stating their duties and responsibilities in the organization and then specify a number of priority levels and assign users to each group.
[68] The number of active users can be calculated in the school by considering maximum of twenty percent of users are active at any working hour in the school.
[69] Most of the time the transmission speed of the LAN is not a big issue, because latest cables or WIFI support huge amount of data transfer in the LAN.

issues that have a direct impact to the network infrastructure of the organization. Some of the issues are listed below:

- Number of users
- Applications that can be accessed through connection (i.e. could the company access any forms of cloud services)
- File transfers
- Quality of service
- Voice, video and data transmissions percentage
- Live data transmission facilities and services
- Frequency to access remote servers for executing a business transaction (i.e. transaction dependencies on remote servers).

For example: consider network infrastructure in the Hawassa Referral Hospital (suppose the hospital have one thousand students, three hundred staffs both academic and administrative, researchers and clients of the hospital). The hospital could provide email, video and audio file accesses, web accesses, social network sites accesses, communication, application download, uploads, accesses hospital specific system[70] and other services.

By considering the number of staffs, researchers and students of the organization and services that are deliver to them, the capacity of the sole internet service provider in the country and budget of the organization. The following bandwidth requirement is enough to properly meet the needs the hospital community:

Every student and instructors (staffs) require 0.1 mbps and 0.5mbps of WAN data transmission rate respectively to access an internet service in the hospital to access all services in well-suited manner. This network estimation is done by considering staffs areexist in a highest priority level to access all the above-mentioned services. Therefore, thetotal bandwidth requirement to the

[70] Applications that are designed in order to provide a better health services to the nation run on LAN and/or WAN from the central data hubs.

hospital will be the multiply of individual requirement with the total number of students and staffs in the hospital (**i.e. 1000*0.1mbps +300*0.5mbps=350 mbps**[71]).

Load Variation Estimates

Loadforecasting (estimation) plays a key role in helping the company to advance a network serviceand tomake important decisions on network users and its priority, load switching, network re-configuration, infrastructuredevelopment, purchasing and installing new hardware and software to enhance a network services to its clients. In order to determine the network load it is better to study for long period (i.e. to know the network access history), accesses the organization ICT policy to know the current stand and to propose the future network (bandwidth) requirement of the organization.

Therefore, theteam can define the network load (usage) variation in the organization in different day and time interval. In order to determine the users' characteristics (network access) the team can conduct a serious of studies (i.e. collect necessary data to reach some conclusion about load variation) in the organization. The team may also conduct interviews, uses networking tools in order to know users access history and observing the LAN of the organization at different days and time intervals.

For Example: Let consider network infrastructure in the Hawassa City Municipality. The following cases can be considered:

- The data indicate that the highest average traffic volume will occur from 8:30 a.m. to 5:00 p.m., Monday to Friday (since it is the normal working hours of the Municipality offices in the city).
- The network traffic reaches a peak volume at two times during the working days of the organization: 9:00 a.m. to 12:00 p.m. and 2:30 p.m. to 5:00 p.m.
- At nighttime and weekends, the network traffic is minimal except for the daily backups of the PCs to the LAN servers in the districts and several batch data transfers anticipated from the State Office.

[71] Since the bandwidth requirement is calculated all users, but it is obviously true the network speed is better than the minimum bandwidth stated here. Because the number of users is always less than the total number staffs and students of the hospital.

The data indicate the following network design parameters:

- The average required throughput on any LAN during working hours (8:00 a.m. to 6:00 p.m.) will be only about 10 mbps.
- The average required throughput on the WAN during working hours (8:00 a.m. to 6:00 p.m.) will be only 0.5 mbps.
- The peak expected traffic load on any LAN would be about 100 mbps.
- The peak expected traffic load on the WAN would be about 2.5 mbps.

Of course, to avoid user complaints, the network will be designed for the average of peak traffic loads and the average throughput requirements of the organization.

Storage Requirements

Storage requirement is all about knowing and determining the requirements, or making your best educated guess when you cannot know for sure[72].Factors to consider include potential data storage requirements of future projects and tier(s) of storage. Tiered storage allows less frequently accessed items to be moved to a lower (and less costly) tier, freeing faster (and more expensive) tiers for those applications that need it. Techniques like data deduplication and file compression can help reduce the amount of storage capacity needed by eliminating extra copies of the same file and reducing the amount of space needed to store files and images [13]. The following issues could also be taking into consideration in predicting organization storage requirement [14]:

- Business Continuity Planning Process (BCP) /Disaster Recovery (DR): so that technical recovery strategies are tightly coupled to the business continuity objectives, and hence enterprise risk management.
- Record keeping and archival management: so that storage solutions are optimized in parallel with a long-term view of information management in the organization.
- Operations management: storage solutions can consider the memory requirements to run day-to-day business transaction in the organization.

[72] Storage requirement planning is the practice of assessing and forecasting future storage requirements so that just enough disk space can be purchased to meet the needs of users and applications.

Consequently, the team can estimate storage requirements to the proposed network infrastructure to meet the organizational objectives and users' requirements by considering the above-mentioned issues in storage requirement prediction for the companies. In addition, the team also defines the memory technologies (i.e. level of tiers, storage types and technologies) and individual storage device sizes that can be configured in the network to meet the objectives.

Example: consider the network infrastructure in the university, the storage requirements need to be large enough to store all student, instructors, and other required data(note: student data are data about students, not data generated by students). By considering, the current trends in the storage requirements for holding students and teachers' data indicate that teachers and students will need an average of 500 MB and 100MB of server space respectively. In addition, the university storage requirement can also consider for Departments, Faculties, college and Institutes, registrar and other offices in the university. The maximum estimated server-side storage requirement for any user cannot exceed 1 GB.

Additionally, the network operating system will occupy about 50GB on each LAN server. Considering price-performance issues, each PC will have a minimum storage capacity of 500GB; each LAN server will have a minimum storage capacity of 12TB. A main data server in the registrar Office will have a 36 TB capacity. It is easy to calculate the total memory requirements for the entire network (i.e. multiplying total number of students and instructors with their storage assignment in the network). Nevertheless, the memory requirement cannot only dependent on the current students; but it also considers the future expansion of the university and university alumni records too.

Reliability[73] Requirements

Reliability is an attribute of any computer-related component (software, or hardware, or a network, for example) that consistently performs according to its specifications. It has to be considering one of three related attributes (i.e. reliability, availability, and serviceability) that must be taken into consideration when making, buying, or using a computer product or component.

[73] In theory, a reliable product is free of technical errors; in practice, however, vendors frequently express a product's reliability quotient as a percentage.

Since, network reliability plays a major role in developing network functionality. The network monitoring systems and network devices are necessary for making the network reliable in the organization. The network monitoring systems detects and identifies the network problems (i.e. there are a number of tools and hardware devices used for this purpose). The network devices ensure that the data reaches the appropriate destination with a predefined reliability set by the organization IT experts.

The reliability of the network is measured by following factors:

- Frequency of failure - determines how frequently the network fails. This issue mainly focuses on the repetition of network/service failures in the organization network infrastructure due to many reasons.
- Recovery time - it is the time taken by a device or network to recover from the failure. It also includes the human capital of the organization that maintains the network facilities in the company.
- Catastrophe - network must be protected from the disasters such as fire, earthquake and other natural/manmade disasters.

Therefore, the team is could clearly specifies the minimum network reliability requirements for both LAN and WAN based on the above criteria's and consider other national and international network reliability standards. This activity requires a deep understanding of the company ICT policy (if exist), its business requirements and their future expansion.

Example: Tokuma Broadcasting Organization wants to upgrade the existing network facility in the organization. The Bid document specifies that, the organization wants to broadcast its Radio Programs in live streaming to its audiences throughout the globe. The following minimum network requirement is needed to meet the organization objectives:

- The LANs are expected to operate at 99 % and the WAN is expected to operate at 95% uptime and an undiscovered error rate of 1% and 2% for LAN and WAN respectively.
- Both LAN and WAN networks could works properly in the organization broadcasting hours (i.e. the broadcasting organization may have at least two modes of data accesses to the Internet Service Provider).

- The network could have a replica servers in other places to keep the company data save and reliance to any kind of catastrophes.

Security Requirements

Network security[74] consists of the policies and practices adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs; conducting transactions and communications among businesses, government agencies and individuals. Network security is involved in organizations, enterprises, and other types of institutions. The most common and simple way of protecting a network resource is by assigning it a unique name and a corresponding password.

Therefore, the team first studies the existing network security standards, policies and guidelines in the organization. Then, the team can briefly specifies what kind of network security measures to be implemented in the newly network infrastructure by considering current security trends and organizational business objectives. Some of the security measures that can be used in the company are:

- A firewall will be used to control threats from the outside world; therefore, the internal network is entirely secure from some unauthorized users (i.e. hackers and crackers).
- Part of the security will be Users accounts and passwords that will give limited access. There will be different access capabilities for network managers and users.
- Implement physical security (like biometric identification systems, installing alarm and fire controlling systems, CCTV system to manage activities in the centers etc.) strategies for data centers and server rooms in order to secure the core services from any kind of physical security challenges.

[74] CISCO also defines "Network security is any activity designed to protect the usability and integrity of your network and data. It includes both hardware and software technologies. Effective network security manages access to the network. It targets a variety of threats and stops them from entering or spreading on your network."

CISCO, a prominent network appliance manufacture, proposes a general security checklist that can be taken into consideration in order to design and implement secure network infrastructure in the organization. For detail, please refer the appendix five.

General Security Planning Tips [15]

The following tips can help you develop and win support for an effective network security plan:

- Focus on return on value rather than return on investment. Consider the harm a network security breach could do to your business, such as lost revenue or customer litigation (legal actions).
- Never assume that network attacks will come only from outsiders.
- Do not be tempted to confront security concerns with a gradually approach rather than a single, unified strategy that protects your whole network.
- Work with others in your company to develop and roll out security strategies, focusing on technology, training, and physical site security with tools like surveillance cameras.
- Find the right balance between security and usability.

Existing Network

The team can study the status of computer network infrastructure in the organization. When the organization has a network infrastructure, the team is expected to specify why the new network design is required regarding different parameters that need change in the company network infrastructure. Like:

- Changes in services in the organization: the company could provide new services to its customers to meet the interest.
- Changes in computer networking and infrastructure technologies (i.e. advancements in wireless and mobile phone needs changes the access ways and interfacing facilities to the existing system)
- Service year of the entire network: the network facility may require an upgrade due to it is the planned service is over.

- Clients (users) satisfaction: the measures of user's satisfaction changes dramatically time to time. Therefore, it is necessary make changes on the company network infrastructure to meet user's need from the system.
- Number of users' increases: companies design the network infrastructure by considering its users and its budget. Therefore, the company could upgrade the network infrastructure in order to provide acceptable services to clients based on national and international standards[75].
- Advancements in network security: since security is one of a critical issue in network services. Therefore, companies can adapt changes in security to secure their day-to-day business transaction and secure their clients information.
- Policy changes: either national[76] or organizational level ICT policy changes due to different reasons have a great impact to upgrade the existing network infrastructure in the company.

[75] For example, Ethio Telecom upgrades its network infrastructure in the nation every year. This task is done in order to meet the nation goal to expand the communication services for more number of users (i.e. at this time Ethio Telecom have 40 million clients).

[76] National Bank of Ethiopia (NBE), proposes different policy changes in bank sector in the country in order to stabilize the financial institutions, one of the change is in the banks ICT policies. Therefore, every bank could implement the new policies forwarded by the NBE in order to secure their day-to-day business transactions.

Chapter III. Logical Network Design

This chapter[77] is focus on activities required in order to perform a logical network design. Designing a network topology is the first step in the logical design phase of the top-down network design methodology[78]. To meet a customer's goals for scalability and adaptability, it is important to architect a logical topology before selecting physical products or technologies. During the topology design phase, you identify networks and interconnection points, the size and scope of networks, and the types of internetworking devices that will be required, but not the actual devices.

The chapter also provides guidelines for assigning addresses and names to internetwork components, including networks, subnets, routers, servers, and end systems. The section mainly focuses on Internet Protocol (IP) addressing and naming devices in the organization network.

The third section of this chapter is to help you select the right switching and routing protocols for your network design customer. The selections you make will depend on your customer's business and technical goals. In order to select appropriate switching and routing protocols you could consider the following attributes: network traffic characteristics, bandwidth, memory, and CPU usage, the approximate number of peer routers or switches supported, the capability to adapt changes in an internetwork and the capability to authenticate route updates for security reasons.

Developing security strategies that can protect all parts of a complicated network while having a limited effect on ease of use and performance is one of the most important and difficult[79] tasks related to network design. To help you handle the difficulties inherent in designing network security for complex networks, this fourth section of this chapter teaches a systematic, top-down

[77] The descriptions stated in this chapter and the preceding chapters is useful if and only if the students have a prior knowledge to configuring networking devices (i.e. configuring switches, routers, servers and other devices found in the network).
[78] A network design methodology suggested by the prominent network devices manufacturer and service supplier CISCO.
[79] Security design is challenged by the complexity and porous nature of modern networks that include public servers for electronic commerce, extranet connections for business partners, and remote-access services for users reaching the network from home, customer sites, hotel rooms, Internet cafes, and so on.

approach that focuses on planning and policy development before the selection of security products.

The last section of the chapter concludes the discussion of logical network design. Network management is one of the most important aspects of logical network design. Management is often overlooked during the design of a network because it is considered an operational issue rather than a design issue. If you consider management from the beginning, however, you can avoid scalability and performance problems that occur when management is added to a design after the design is complete.

Designing a Network Topology

CISCO advices a network engineers to use a hierarchical model in order to design a network topology. Campus[80] network design topologies should meet a customer's goals for availability and performance by featuring small bandwidth domains, small broadcast domains, redundancy, mirrored servers, and multiple ways for a workstation to reach a router for off-net communications. The networks should be designed using a hierarchical, modular approach so that the network offers good performance, maintainability, and scalability.

The network can consist of access, distribution, and core layers; these layers are critical in order to meet organizational objectives and it has specific role in the network[81].

Guidelines to design a network topology:

- Split the entire network into different virtual LANs groups
- Identify areas that are suitable for wireless LAN
- Positioning a wireless access point for maximum coverage
- Check areas that requires redundant wireless access points and/or wired LANs
- Check server redundancies for company critical services to guaranty the availability of the services to the clients

[80] The prominent networking devices manufacturer Cisco uses this naming to refer any company/organization that is a client for network infrastructure development.
[81] The core layer provides optimal transport between sites. The distribution layer connects network services to the access layer and implements policies regarding security, traffic loading, and routing. Moreover, the access layer consists of the routers at the edge of the networks. The access layer provides switches or hubs for end user access.

- Planning for physical security[82]
- Meeting security goals with firewall topologies
- If necessary implement Virtual private networks (VPN), it use advanced encryption and tunneling to permit organizations to establish secure, end-to-end, private network connections over a third-party network.
- Redundant WAN Segments: since WAN links can be critical pieces of an enterprise internetwork, redundant (backup) WAN links are often included in an enterprise edge network topology.

By applying the above guidelines, the team designs a network topology for the company by considering the company's business objectives (i.e. stated in the first chapter of this documentation) and budget.

Designing Models for Addressing and Numbering

It is critical issue to use structured model for network layer addressing and naming. Without structure, it is easy to run out of addresses, waste addresses, introduce duplicate addresses and names, and use addresses and names that are hard to manage. To meet a customer's goals for scalability, performance, and manageability, you should assign addresses and names systematically. Therefore, as network designers, the group could develop policies[83] and procedures for addressing and naming for the proposed network infrastructure. The policies and procedure could consider the guidelines given below for network layer addresses and naming devices in the network.

Guidelines for Assigning Network Layer Addresses:

Network layer addresses should be planned, managed, and documented. Although an end system can learn its address dynamically, no mechanisms exist for assigning network or subnet numbers

[82] You should start working with your design customer right away to make sure that critical equipment will be installed in computer rooms that have protection from unauthorized access, theft, vandalism, and natural disasters such as floods, fires, storms, and earthquakes. Physical security is not really an aspect of logical network design

[83] Policies often involve a plan for distributing authority for addressing and naming to avoid one department having to manage all addresses and names. A central authority can assign blocks of addresses and names in a hierarchical fashion to departments and branch offices.

dynamically. The following list provides some simple rules for network layer addressing that can help you architect scalability and manageability into a network design.

- Design a structured model for addressing before assigning any addresses.
- Leave room for growth in the addressing model.
- Assign blocks of addresses in a hierarchical fashion to foster good scalability and availability.
- Assign blocks of addresses based on the physical network, not on group membership, to avoid problems when groups or individuals move.
- If the level of network management expertise in regional and branch offices is high, you can delegate authority for addressing regional and branch-office networks, subnets, servers, and end systems.
- To maximize flexibility and minimize configuration, use dynamic addressing for end systems.
- To maximize security and adaptability, use private addresses with Network Address Translation (NAT) in IP environments.

Guidelines for Assigning Names[84]:

- To maximize usability, names should be short, meaningful, unambiguous, and distinct.
- A good practice is to include in a name some sort of indication of the device's type. For example, you can prefix or suffix router names with the characters *rtr*, switches with *sw*, servers with *svr*, and so on.
- Using meaningful prefixes or suffixes decreases ambiguity for end users and helps managers, more easily extract device names from network management tools.
- Names can also include a location code.
- Try to avoid names that have unusual characters, such as underscores, ampersands, asterisks, and so on, even if the naming protocol allows these characters.
- It is also best if names are not case-sensitive because people usually cannot remember which case to use.
- You should also avoid spaces in names.

[84] Names play an essential role in meeting a customer's goals for usability. Short, meaningful names enhance user productivity and simplify network management.

- If a device has more than one interface and more than one address, you should map all the addresses to one common name.

Selecting Switching and Routing Protocols

This section describe the characteristics to be considered in order to select an appropriate switching and routing protocols for the proposed network. The selections you make will depend on your customer's business and technical goals. To match options with goals, you can make a decision table, a decision table that matches routing protocols to a fictional customer's business and technical goals.

For example: let see the following decision table for routing algorithms and organization objectives.

	Critical Goals			Other Goals		
Protocols	Adaptability[85]	Must scale to a large size	Industry standard[86]	Not create a lot of traffic	Run on inexpensive routers	Easy to configure and manage
BGP	X	X	X	8	7	7
OSPF	X	X	X	8	8	8
IS-IS	X	X	X	8	6	6
IGRP	X	X				
EIGRP	X	X				
RIP			X			

After a decision has been made, you should troubleshoot the decision. Ask yourself the following:

- If this option is chosen, what could go wrong?
- Has this option been tried before (in other customers)? If so, what problems occurred?
- How will the customer react to this decision?
- What are the contingency plans if the customer does not approve of the decision?

[85] Adaptability must adapt to changes in a large internetwork within seconds

[86] Must be an industry standard and compatible with existing equipment

This decision-making process can be used during both the logical and physical network design phases. You can use the process to help you select protocols, technologies, and devices that will meet a customer's requirements. Therefore, the team develops a company specific decision table (i.e. company business objectives with routing algorithms) for selecting an appropriate switching and routing protocols for the proposed infrastructure. For the detail information about routing algorithms, please see appendix six that shows the comparison of different routing algorithms mostly used by routers.

Developing Network Security Strategies

Following a structured set of steps when developing and implementing network, security will help you address the varied concerns that play a part in security design. Many security strategies have been developed in a haphazard way and have failed to actually secure assets and to meet a customer's primary goals for security. Breaking down the process of security design into the following steps will help you effectively plan and execute a security strategy[87]:

- Identify network assets.
- Analyze security risks.
- Analyze security requirements and tradeoffs.
- Develop a security plan.
- Define a security policy.
- Develop procedures for applying security policies.
- Develop a technical implementation strategy.
- Achieve buy-in from users, managers, and technical staff.
- Train users, managers, and technical staff.
- Implement the technical strategy and security procedures.
- Test the security and update it if any problems are found.
- Maintain security.

Performing required tasks in every step by considering organizational objectives and security tradeoff (i.e. each security methods have its own tradeoffs) with respected to the current business

[87] The team goes through each step in the security design steps until the proposed security document meet the organization security objectives and satisfy the clients.

transaction of the organization. The successful completion each steps in security design, starting from identifying network assets, becomes an input for the proceeding steps in the processes, so as a team, careful design and operation of tasks is necessary. Some of the activities have an output documents that supports how the system to be successfully accomplished in the ongoing tasks done by the organization.

Before designing a security policy document to the organization, tasks like identify network assets, analyze security risks, analyze security requirements, tradeoffs, and developing a security plan designed could be accomplished successfully. Because the activities that are done in those steps are used as inputs for the proper security policy designs. In addition, security design document is also including the following security parameters:

- An access policy that defines access rights and privileges of users in the network. The access policy should provide guidelines for connecting external networks, connecting devices to a network, and adding new software to systems. An access policy might also address how data is categorized (for example, confidential, internal, and top secret).
- An accountability policy that defines the responsibilities of users, operations staff, and management. The accountability policy should specify an audit capability and provide incident-handling guidelines that specify what to do and whom to contact if a possible intrusion is detected.
- An authentication policy that establishes trust through an effective password policy and sets up guidelines for remote-location authentication.
- A privacy policy that defines reasonable expectations of privacy regarding the monitoring of electronic mail, logging of keystrokes, and access to users' files.
- Computer-technology purchasing guidelines that specify the requirements for acquiring, configuring, and auditing computer systems and networks for compliance with the policy.

This security policy document[88] will be a benchmark to perform a security related tasks in the organization to meet the company objectives. This document also considers the possible security

[88] The security policy document must match with the ICT policy of the organization. That means, the document could briefly specify security strategies stated in the ICT policy document of the organization. Therefore, the team must access the ICT policy document (if exists) before preparing security design document.

mechanisms[89] to be used in order to achieve security goals of the client in the proposed network infrastructure to the organization.

Security experts promote the security defense in depth principle. This principle states that network security should be multilayered, with many different techniques used to protect the network. Because there is, a security mechanism can be guaranteed to withstand every attack. Therefore, each mechanism should have a backup mechanism[90].

As part of implementing security defense in depth, security design should be modular just like the network design. Multiple security methods should be designed and applied to different parts of the network, whether it is the internet connection, the wireless infrastructure, server farms, user services or the remote-access component.

Developing Network Management Strategies

Network management is one of the most important aspects of logical network design. A good network management design can help an organization achieve availability, performance, and security goals. Effective network management processes can help an organization measure how well design goals are being met in working environment and adjust network parameters if these goals are not being met.

Network management also facilitates meeting scalability goals because it can help an organization analyze current network behavior, apply upgrades appropriately, and troubleshoot any problems with upgrades. The goal of this section is to help the team to design network management strategies and to help the team to select the right tools and products to implement the strategies.

Most clients have a need to develop network management processes (it also true for the team network project) that can help them manage the implementation and operation of the network, diagnose and fix problems, optimize performance, and plan enhancements. The International

[89] Security mechanism includes; physical security, authentication, authorization, accounting (auditing), data encryption (i.e. private or public key encryptions), packet filters, firewall, and Intrusion Detection and Prevention Systems of networks.
[90] The backup security mechanisms are critical issues in the design of a secure systems, because the attacker and attacking mechanism in the network is changing dynamically.

Organization for Standardization (ISO) defines five types of network management processes, which are often referred to with the FCAPS acronym:

- **Fault management**: refers to detecting, isolating, diagnosing, and correcting problems. It also includes processes for reporting problems to end users and managers, and tracking trends related to problems. In some cases, fault management means developing workarounds until a problem can be fixed. Monitoring tools are often based on the Simple Network Management Protocol (SNMP) and Remote Monitoring (RMON) standards.
- **Configuration management**: helps a network manager keep track of network devices and maintain information on how devices are configured. With configuration management, a network manager can define and save a default configuration for similar devices, modify the default configuration for specific devices, and load the configuration on devices. It also lets a manager maintain an inventory of network assets and do version-logging[91].
- **Accounting management**: facilitates usage-based billing, whereby individual departments or projects are charged for network services. Even in cases in which there is no money exchange, accounting of network usage can be useful to catch departments or individuals who "abuse" the network. The abuse could be intentional or unintentional.
- **Performance management**: allows the measurement of network behavior and effectiveness. It includes examining network application and protocol behavior, analyzing reachability, measuring response time, and recording network route changes. It facilitates optimizing a network, meeting service-level agreements (SLA), and planning for expansion.
- **Security management**: lets a network manager maintain and distribute passwords and other authentication and authorization information. It includes processes for generating, distributing, and storing encryption keys. It can also include tools and reports to analyze a group of router and switch configurations for compliance with site security standards. It is a process for collecting, storing, and examining security audit logs.

[91] Version-logging refers to keeping track of the version of operating systems or applications running on network devices.

After you have discussed high-level network management processes with your customer, and developed network management architecture[92], you can make some decisions on which network management tools and protocols to recommend to your customer.

When selecting management tools, consider the flexibility of the tools and the varied audiences that may interface with them. Network management tools[93] should provide an intuitive user interface that can react quickly to user input. In many cases, having both a browser interface and command-line interface (CLI) is beneficial.

If the tools allow dynamic configuration of devices, configuration changes should take effect without requiring a reboot of the device. Management software should also check the validity of any configuration changes and automatically restore operation to the last known configuration or software image in case of error. Management software that supports the dynamic configuration of devices should require authentication to avoid an unauthorized user making changes.

After you have determined which management protocols will be used, you can estimate the amount of traffic caused by network management. When the management protocol is selected, you should determine which network and device characteristics will be managed.

[92] Network management architecture is a software platform could be used to manage a network infrastructure in the company. It consists of three major components: a managed device, an agent and a network management system (NMS).

[93] Popular network management tools are Simple Network Management Protocol (SNMP), Remote Monitoring (RMON), Cisco Discovery Protocol (CDP), and Cisco Netflow Accounting.

Chapter IV. Physical Network Design

Physical network design involves the selection of LAN and WAN technologies for campus (organization) network design. During this phase of the network design process, choices are made regarding cabling, physical and data link layer protocols, and internetworking devices (such as switches, routers, and wireless access points). A logical design, which Chapter III, "Logical Network Design," covered, forms the foundation for a physical design. In addition, business goals, technical requirements, network traffic characteristics, and traffic flows influence a physical design.

A network designer/team has many options for LAN and WAN implementations. No single technology or device is the right answer for all circumstances (i.e. an appropriate device and configuration selection is always dependent on the organization business objectives). The goal of this chapter is to give you (team) information about the scalability, performance, affordability, and manageability characteristics of typical options, to help you make the right selections for your particular client.

An effective design process is to develop organization network solutions first, followed by remote access and WAN solutions. After you have designed a client's network, you can more effectively select WAN and remote-access technologies based on the bandwidth and performance requirements of traffic that flows from one junction in the network to the other junctions.

This chapter begins with a discussion of LAN cabling-plant design, including cabling options for building and organization networks. The chapter then provides information about LAN technologies. The next section provides some selection criteria you can use when selecting switches, routers, and wireless access points for a network design. The chapter concludes with an example of a campus network design that was developed for the Wandering Valley Community College.

LAN Cabling Plant Design

It is important to design and implement the cabling infrastructure carefully, keeping in mind availability and scalability goals, and the expected lifetime of the design. In many cases, your network design must adapt to existing cabling. Documenting cabling specification could include the following issues:

- Campus- and building-cabling topologies
- Types and lengths of cables between buildings
- Types and lengths of cables for vertical cabling between floors
- Types and lengths of cables for horizontal cabling within floors
- Types and lengths of cables for work-area cabling going from telecommunications closets to workstations

Cabling Topologies

- A centralized cabling scheme terminates most or all of the cable runs in one area of the design environment. A star topology is an example of a centralized system. This cabling topology is suitable for small networks.
- A distributed cabling scheme terminates cable runs throughout the design environment. Ring, bus, and mesh topologies are examples of distributed systems. These cabling topologies are suitable for medium and large network.

Company's network implementations use three major types of cables; these are shielded copper (including shielded twisted-pair (STP), coaxial (coax), and twin axial (twinax) cables), unshielded copper (typically UTP) cable and fiber-optic cables. The cable selection could consider properties of network cables, budget of the organization, geographic span of the network and future expansion of the organization.

LAN Technologies

After analyzing the business constraints and objectives of the client organization you could know the issues that could be considered in order to select an appropriate networking (LAN) topology

(topologies) that meets the needs of the organization. The following issues could affect LAN topology selection:

- Biases (technology religion)
- Policies about approved technologies or vendors
- The customer's tolerance to risk
- Technical expertise of the staff and plans for staff education
- Budgeting and scheduling

In addition, to the above mentioned issues in LAN topology selection the team also consider technical goals of the organization. Technical goals have a direct impact on how the proposed LAN can meet scalability, availability, manageability, adaptability, affordability, and other technical issues.

The team should consider clients strict requirements about throughput, delay, and delay variation for any network applications. The team should also consider the types of applications the customer plans to run on the network. Applications that allow users to share videos, collaborate with desktop sharing, watch high-definition television, and so on are more bandwidth hungry and delay sensitive than text-based applications.

Ethernet Technology Choices

Ethernet is a scalable technology that has adapted to increasing capacity requirements. The following options for implementing Ethernet networks are available:

- Half- and full-duplex Ethernet
- 100-Mbps Ethernet
- 1000-Mbps (1-Gbps or Gigabit) Ethernet
- 10-Gbps Ethernet[94]
- Metro Ethernet
- Long-Reach Ethernet (LRE)
- Cisco EtherChannel

[94] Please refer appendix seven for further information about 10-Gbps Ethernet.

Each of these technologies is a possibility for the access, distribution, or core layers of a campus topology, although usually the higher speeds, such as Gigabit Ethernet, are reserved for the core layer.

The choice of an Ethernet technology for the access layer depends on the location and size of user communities, bandwidth and Quality of Service (QoS) requirements for applications, broadcast and other protocol behavior, and traffic flow. The choice of an Ethernet technology for the distribution and core layers depends on the network topology, the location of data stores, and traffic flow.

Selecting Internetworking Devices for a Organization Network Design

After team has designed a network topology and made some decisions about the placement and scope of shared, switched, and routed network segments, you should then recommend actual switches, bridges, and routers from various vendors. Please refer a review of the major differences between internetworking devices that can be used to connect network segments in appendix eight.

General Criteria for Selecting Internetworking Devices: selecting internetworking devices general criteria include the following:

- Number of ports
- Processing speed
- Amount of memory
- Amount of latency introduced when the device relays data
- Throughput in packets per second
- Ingress/egress queuing and buffering techniques
- LAN and WAN technologies supported
- Autosensing of speed (for example, 10 or 100 Mbps)
- Auto-detection of half- versus full-duplex operation
- Media (cabling) supported
- Ease of configuration
- Cost

- Manageability (for example, support for Simple Network Management Protocol [SNMP] and Remote Monitoring [RMON], status indicators)
- Mean time between failure (MTBF) and mean time to repair (MTTR)
- Support for packet filters and other security measures
- Support for hot-swappable components
- Support for in-service software upgrades
- Support for redundant power supplies
- Support for optimization features
- Support for QoS features
- Availability and quality of technical support
- Availability and quality of documentation
- Availability and quality of training (for complex switches and routers)
- Reputation and viability of the vendor
- Availability of independent test results that confirm the performance of the device

These criteria are applicable for any kind of interconnecting devices (i.e. router, switch, hub or bridge). However, each interconnecting devices have their own specific specification that can be taking in to consideration to select an appropriate device to meet the objective of the organization.

Therefore, the team could analyze the advantages and disadvantages of internetworking devices with respected to the organizational objectives and budget before deciding which devices can be purchased and installed in the proposed network.

Chapter V. Conclusion and Recommendation

REFERENCES

1. How to Write your Introduction, Abstract and Summary, https://thesistips.wordpress.com/ 2012/03/25/how-to-write-your-introduction-abstract-and-summary/, accessed on Oct 04, 2017.
2. David Thomas, Ian D Hodges. Designing and Managing Your Research Project: Sage Publication, 2010.
3. Mark Saunders et. al. Research Methods for Business Students:Pearson Education Limited, fifth edition, 2009.
4. Scope of the Study, http://www.phdthesis.in/scope-of-the-study/ , accessed on Oct 06, 2017.
5. Organizing Your Social Sciences Research Paper, http://libguides.usc.edu/writingguide/citingsources , accessed on Oct 06, 2017.
6. IBM Rational Technical White Paper Development Environment Definition: IBM publication, January 2011.
7. Why a Feasibility Study is Important in Project Managementhttps://www.simplilearn.com/feasibility-study-article, accessed on Oct 10, 2017.
8. Project Scope Checklist, http://www.axia-consulting.co.uk/html/project_scope_checklist.html, accessed on Oct 12, 2017.
9. Systems Analysis: Document Existing System: Gather Data, http://it.toolbox.com/ blogs/enterprise-solutions/systems-analysis-document-existing-systems-gather-data-43880, accessed on Oct13, 2017.
10. Abstract, Existing & Proposed System, http://wirelesscms01.blogspot.com/2012/02/abstract-existing-proposed-system.html, accessed on Oct 13, 2017.
11. Rosenblatt. Systems Analysis and Design: Shelly Cashman Series, Tenth Edition, 2013.
12. Alan Dennis et. al. System Analysis & Design: An Object -Oriented Approach with UML: Wiley Publishing, Fifth Edition, 2015.
13. Storage Capacity Planning, http://searchstorage.techtarget.com/definition/storage-capacity-planning, accessed on Oct 23, 2017.

14. Understanding Storage Requirements, http://www.enterprisestorageforum.com/management/features/article.php/3365271/Understanding-Storage-Requirements.html a, accessed on Oct 13,2017.
15. Network Security Checklist, https://www.cisco.com/c/en/us/solutions/small-business/resource-center/secure-my-business/network-security-checklist.html, accessed on Oct 26, 2017
16. Priscilla Oppenheimer. Top–down Network Design: CISCO Press, Third Edition, 2011.
17. G

APPENDIX

1. Project Scope Checklist

No	Project Scope Checklist	Yes	No	Don't know
	Does your scope definition include:			
1	- the need(s) or issue(s) to be resolved by the project?			
2	- the outcome(s), objective(s), or deliverable(s)?			
3	- the new system(s) to be implemented?			
4	- key functionality required within the new system(s)?			
5	- technical infrastructure of the new system(s)?			
6	- data requirements?			
7	- key business processes to be reengineered?			
8	- old system(s) to be replaced?			
9	- technologies to be changed?			
10	- who it is for and who is affected – in terms of business functions, areas and locations?			
11	- the expected activities / work to be performed?			
12	- resources to be used / anticipated team size?			
13	- expected duration / time scales?			
14	- deliverable acceptance criteria?			
15	Has your scope been quantified wherever possible?			
16	Does your scope statement clearly list what is included within the project and specify the project boundaries?			
17	Does your scope statement list what is specifically excluded from the project?			
18	Do you have an agreed process for dealing with areas / items which you are unsure whether they are included or excluded?			
19	Have your scope assumptions been documented?			
20	Are the assumptions realistic?			

21 Does your scope statement provide a sufficiently detailed baseline – so as to identify whether a future proposed change is within or outside the project scope?

22 Is your scope definition sufficiently detailed? i.e. is it at least a few pages long?

23 Has your scope definition been carefully checked / verified?

24 Can you confirm - there is nothing else which should be included within your project scope?

25 Is there a common understanding of your project scope by all involved?

26 Has your project team worked with the users / stakeholders to define the project scope?

27 Is your project scope agreed by all i.e. users, stakeholders, project sponsor, project manager and project team?

28 Have all signed-off their agreement to the project scope statement?

29 Do users / stakeholders understand 'scope management' and the potential problems caused by scope creep?

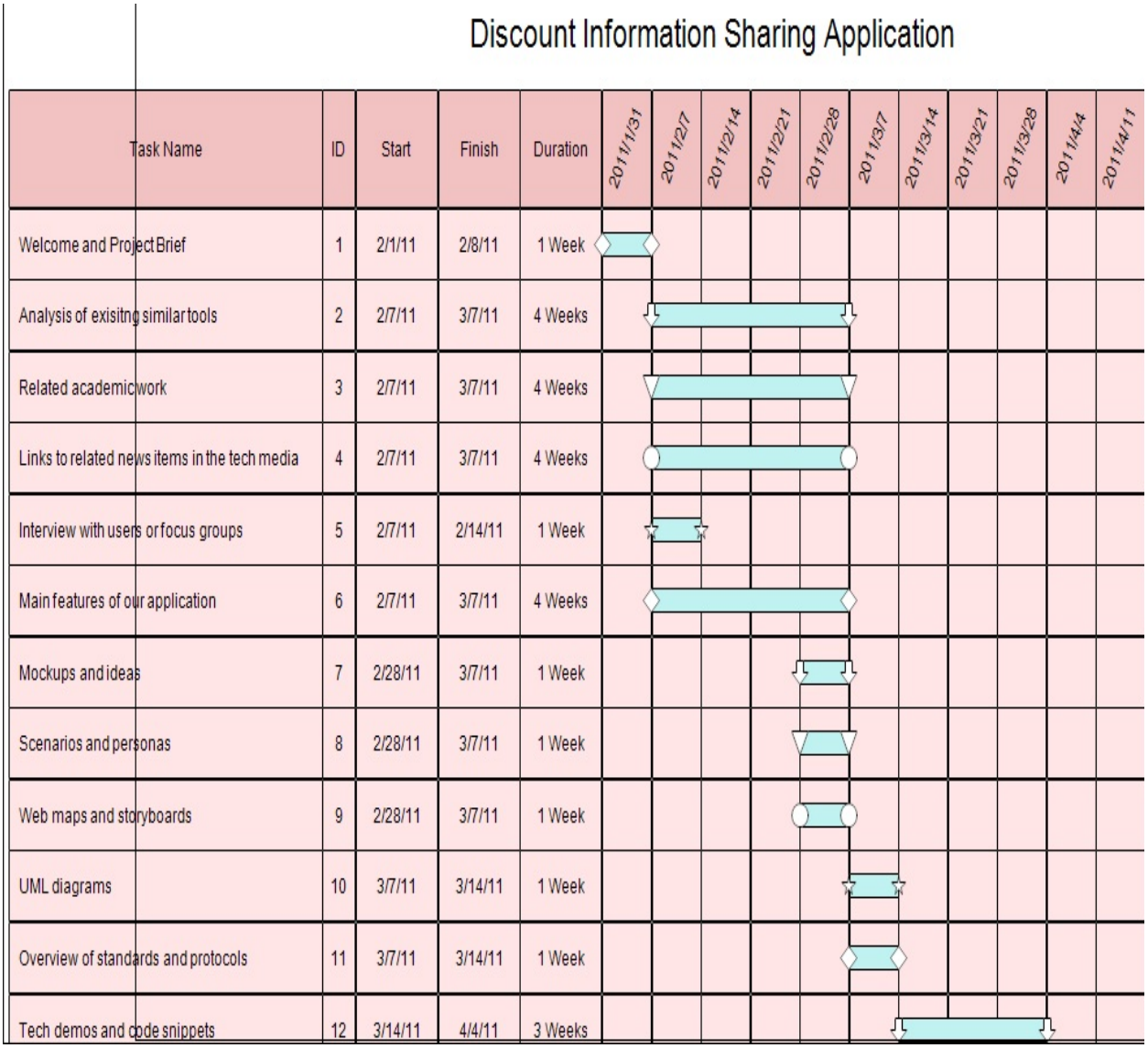
30 Do you have a formal process for managing scope changes, which everyone understands and will use?

2. Sample cost breakdown for final project

Ref	Project Expenditures	**Cost**
1	**Software costs**	
1.1	Application software user licenses	10000
1.2	Software modifications	0
1.3	Additional licenses e.g. Visual Studio 2017	10000
1.4	Database user licenses	50000
1.5	Operating system	0
2	**Hardware costs**	
2.1	Servers (new or upgraded, dedicated or shared)	200000
2.2	PC's (new or upgrades)	10000

2.3	Additional memory	2000
2.4	Printers	10000
2.5	Back up devices	10000
2.6	Disk storage	5000
3	**Network costs**	
3.1	Cabling or wireless LAN, WAN or other network	50000
3.2	Leased or dedicated lines	50000
3.3	Communications software	20000
3.4	Internet access e.g. ADSL / broadband, satellite	12000
4	**Training**	
4.1	Technical training for implementation team (by vendor)	7000
4.2	Training key users (by vendor)	4000
4.3	Temporary internal training facilities	3000
4.4	Training remainder of users (internally or by vendor)	3000
4.5	Training materials	2000
4.6	Travel and expenses to attend training	8000
5	**Contingency**	
5.1	Estimated 10% of total implementation costs	46600
	Total cost	512600

3. Sample Gantt chart for projects



4. Sample for existing and proposed system

Existing system [10]

A Content Management System (CMS) works by storing pictures, audio text and other forms of information into a database. When a webpage is requested, the CMS system accesses the database and renders the webpage. Because the data is separated from the code, changes to the data can be made using a web interface that requires no knowledge of Hypertext Markup Language (HTML) or File Transfer Protocol (FTP).

The disadvantage of dynamic content management systems is that they are resource hungry. The servers they run on need to be maintained by technicians, and content management requires more memory, CPU power and software maintenance. Small setting changes in the server can cause the entire CMS to fail and as a consequence the website will not be viewable or return errors.

Proposed system

In the present project work we put forth a design for a Wireless Content Management System (WiCoM) architecture to operate on a mobile device. WiCoM is a wireless application aimed at helping the general administration of cyber contents while being on the move.

It is a wireless cyber content management software running on a java enabled mobile device having General Packet Radio Service (GPRS[95]) connectivity (i.e. a mobile communication for 2G and 3G mobile phones). The main purpose of this application is to enable secure cyber content administration using any smart mobile device (java enabled and having GPRS connectivity), which helps "Any Where Any Time" administration possible.

This application provides interface for the user to securely add in various useful data, which can be interactively displayed on the web in real-time. Both textual as well as multimedia contents can be managed.

It finds application in news reporting agency to administer news site in real-time. A reporter arriving at the site of the event can record the news of the current scenario from the various sources.

Here we can upload pictures, audios and videos, right now to the web-server making it available to the world in no time. There are options to edit/delete and thus provide various content management related features.

[95]General Packet Radio Service (GPRS) is a packet oriented mobile data service on the 2G and 3G cellular communication system's global system for mobile communications (GSM).

5. Network Security Checklist

Policy [15]

Every business should have a written network security plan in place. A thorough policy will cover topics such as:

- Acceptable use policy, to specify what types of network activities are allowed and which ones are prohibited
- E-mail and communications activities, to help minimize problems from e-mails and attachments
- Antivirus policy, to help protect the network against threats like viruses, worms, and Trojan horses.
- Identity policy, to help safeguard the network from unauthorized users
- Password policy, to help employees select strong passwords and protect them
- Encryption policy, to provide guidance on using encryption technology to protect network data
- Remote access policy, to help employees safely access the network when working outside the office

Inventory Your Current Security Technologies

Do you have any of the following?

- Firewall, to keep unauthorized users off your network
- Virtual private network (VPN), to give employees, customers, and partners secure access to your network
- Intrusion prevention, to detect and stop threats before they harm your network
- Content security, to protect your network from viruses, spam, spyware, and other attacks
- Secure wireless network, to provide safe network access to visitors and employees on the go
- Identity management, to give you control over who and what can access the network
- Compliance validation, to make sure that any device accessing the network meets your security requirements

Identify Your Most Important Digital Assets and Who Uses Them

- Exactly what are your company's digital assets (i.e. intellectual property and customer records)?
- What are they worth?
- Where do those assets reside?
- Who has access to these assets, and why? Can all employees access the same assets?
- Do you extend access to business partners and customers?
- How do you control that access?

What Would a Security Breach Do to Your Business?

- What is the potential financial impact of a network outage due to a security breach?
- Could a security breach disrupt your supply chain?
- What would happen if your Website went down?
- Do you have e-commerce features on your site? How long could the site be down before you lost money?
- Are you insured against Internet attacks, or against the misuse of your customers' data? Is this insurance adequate?
- Do you have backup and recovery capabilities to restore information if necessary after a security breach?

Consider Your Current and Future Needs

- How do you expect your business plan to evolve over the next few years?
- How recently have you updated your network equipment? Software? Virus definitions?
- What type of security training do you provide to your employees?
- How will growth affect your digital assets and their value to your business as a whole?
- In the future, are you likely to have a greater need for remote employees, customers, or partners to access those digital assets?

6. Comparison of Routing Algorithm

Protocol	Distance Vector or Link State	Interior or Exterior	Classful or Classless	Metrics Supported	Scalability	Convergence Time	Resource Consumption	Supports Security? Authenticates Routes?	Ease of Design, Configuration, and Troubleshooting
RIPv1	Distance vector	Interior	Classful	Hop count	15 hops	Can be long (if no load balancing)	Memory: low CPU: low BW: high	No	Easy
RIPv2	Distance vector	Interior	Classless	Hop count	15 hops	Can be long (if no load balancing)	Memory: low CPU: low BW: high	Yes	Easy
IGRP	Distance vector	Interior	Classful	BW, delay, reliability, load	255 hops (default is 100)	Quick (uses triggered updates and poison reverse)	Memory: low CPU: low BW: high	No	Easy
EIGRP	Advanced distance vector	Interior	Classless	BW, delay, reliability, load	1000s of routers	Very quick (uses DUAL algorithm)	Memory: mod CPU: low BW: low	Yes	Easy
OSPF	Link state	Interior	Classless	Cost (100 mil. divided by BW on Cisco routers)	A hundreds routers per area, a few hundred areas	Quick (uses LSAs and Hello packets)	Memory: high CPU: high BW: low	Yes	Mod
BGP	Path vector routes)	Exterior	Classless	Value of path attributes and other configurable factors	1000s of routers	Quick (uses update and keep alive packets, and withdraws	Memory: high CPU: high Bandwidth: low	Yes	Mod
IS-IS	Link state	Interior	Classless	Configured path value, plus delay, expense, and errors	Hundreds of routers per area, a few hundred areas	Quick (uses LSAs)	Memory: high CPU: high BW: low	Yes	Mod

Where:

BW-Bandwidth, mod- moderate

7. 10-Gbps Ethernet Implementations

Implementation	Wavelength	Medium	Minimum Modal Bandwidth	Operating Distance
10GBASE-LX4	1310 nm	62.5-micron multimode (mm) fiber	500 MHz/km	2–300 m
10GBASE-LX4	1310 nm	50- mm fiber	400 MHz/km	2–240 m
10GBASE-LX4	1310 nm	50- mm fiber	500 MHz/km	2–300 m
10GBASE-LX4	1310 nm	10-micron single-mode fiber	Not applicable	2–10 km
10GBASE-S	850 nm	62.5-micron multimode fiber	160 MHz/km	2–26 m
10GBASE-S	850 nm	62.5-micron multimode fiber	200 MHz/km	2–33 m
10GBASE-S	850 nm	50-micron multimode fiber	400 MHz/km	2–66 m
10GBASE-S	850nm	50-micron multimode fiber	500 MHz/km	2–82 m
10GBASE-S	850 nm	50-micron multimode fiber	2000 MHz/km	2–300 m
10GBASE-L	1310 nm	10-micron single-mode fiber	Not applicable	2–10 km
10GBASE-E	1550 nm	10-micron single-mode fiber	Not applicable	2–30 km
10GBASE-CX4	Not applicable	Twinax	Not applicable	15 m
SFP+ Direct Attach	Not applicable	Twinax	Not applicable	10 m
10GBASE-T	Not applicable	UTP or STP	Not applicable	100 m

8. Comparing Hubs, Bridges, Switches, and Routers

Device	OSI Layers Implemented	How bandwidth (Collision) Domains Are Segmented	How Broadcast Domains Are Segmented	Typical Deployment	Typical Additional Features
Hub	1	All ports are in the same bandwidth domain.	All ports are in the same broadcast domain.	Connects individual devices in small LANs	Autopartitioning to isolate misbehaving nodes
Bridge	1–2	Each port delineates a bandwidth domain.	All ports are in the same broadcast domain.	Connects networks	User-configured packet filtering
Switch	1–2	Each port delineates a bandwidth domain.	All ports are in the same broadcast domain unless VLANs are used.	Connects individual devices or networks	Filtering, cell-relay capabil-ities, cut-through processing, multimedia (multicast) features
Router	1–3	Each port delineates a bandwidth domain.	Each port delineates a broadcast domain.	Connects networks	Filtering, firewalling, high-speed WAN links, compressi-on, advanced queuing and forwarding processes, multimedia (multicast) features