Modern University for Technology & Information
Faculty of Computers & Artificial Intelligence
AY 2023/2024

Fall 2023

Graduation Project Guide (GPG)

CS 492

Prof. Dr. Hafez S. Abdel Wahab

Prof. Dr. Mohamed A. Assal

Prof. Dr. Hanafy M. Ismail

Prof. Dr. Hesham M. El-Deeb

Dr. Rasha M. Saeed

رؤيـة كلية الحاسبات والذكاء الاصطناعي

تسعى كلية الحاسبات والذكاء الاصطناعي لأن تكون صرحاً تعليمياً وبحثياً متميزاً في مجال علوم الحاسب وتكنولوجيا المعلومات يعمل وفقاً لمعايير الجودة العالمية بما يمكن من تحقيق المنافسة والتميز على كل من المستويات المحلية والإقليمية والعالمية.

Vision of Faculty of Computers & Artificial Intelligence

Faculty of Computers and Artificial Intelligence aims to be a pioneer educational institution and research center in the field of Computer Science and Information Technology. This makes it possible to achieve competition and uniqueness at the local, regional, and international levels.

رسالـة كلية الحاسبات والذكاء الإصطناعي

تلتزم كلية الحاسبات والذكاء الاصطناعي بتقديم خدمات تعليمية أكاديمية تؤدي إلى خريج متميز قادر على المنافسة محلياً وإقليمياً وعالمياً وعلى كفاءة عالية في مجالات الحاسبات وتكنولوجيا المعلومات ويتمتع بمهارات وأخلاقيات مهنية رفيعة وقادر على التعلم الذاتي لمواكبة التطورات السريعة في مجال تخصصه وتقديم تعليم مستمر وخدمات تدريبية واستشارية لتحقيق التنمية القومية والارتقاء بالبيئة المحلية وإجراء البحوث العلمية في مختلف مجالات علوم الحاسب وتكنولوجيا المعلومات.

Mission of Faculty of Computers & Artificial Intelligence

Faculty of Computers and Artificial Intelligence is committed to:

- providing academic educational services that lead to a distinguished graduate capable of competing locally, regionally, internationally, and be highly qualified in the field of computers and information technology. In addition, graduates will have high professional skills and ethics and will be capable of self-learning to cope with rapid developments in his/her field of specialization.

- Providing continuous education, training and consulting services to achieve national development and improve the local environment.

- Conducting scientific research in various fields of computer science and information technology.

Modern University for Technology & Information

Faculty of Computers & Artificial Intelligence

Course Specifications

Course Name: Graduation Project

Course Code: CS492

A. Affiliation and Basic information:

- ☒ **Relevant programs: B.Sc. in Computer Science**
- ☒ **Department offering the program: Computer Science**
- ☒ **Department offering the course: Computer Science**
- ☒ **Academic Level: Fourth**
- ☒ **Semester in which Course is offered: Academic Year**
- ☒ **Course Pre-requisite(s): Student complete at least 99 credit hours**
- ☒ **Credit Hours: 6**
- ☒ **Date of Specification Approval**: July 2013

Lecture	Tutorial / Practical
6	24

B. Professional information:

Overall Aims of Course

Forming team work, selecting topics, project management and time management, problem analysis, problem solving and selection of solutions, project design, literature search, system design, project development, project verification and validation, error detection and modification, evaluation of sources and analysis, documentation

1. Course Learning Objectives:

1- To manage and execute a substantial project in a limited time.
2- To identify and learn whatever new skills are needed to complete the project.
3- To apply design and engineering skills in the accomplishment of a single task.

In this context the skills mentioned may be in the general area of design and engineering in its broadest sense, or may be very specifically related to particular tools.

2. Intended Learning Outcomes (ILOS)

A. Knowledge and understanding:

a1. Knowledge of the essential concepts and majors principles relevant to system analysis and development.(A1)

a2. Understand the importance of wide range of software and hardware used in development of computer systems and information technology products.(A2,A3,A10,A11,A19)

a3. Understand the principles of software project management. (A8)

a4. Understand the importance of software documentation.(A1)

a5. Modeling organizational processes and data, defining and implementing technical and process solutions, managing projects, and integrating systems (A9,A11,A19).

B. Intellectual skills:

On completion of this course the successful student will be able to:

b1. Analyze a wide range of systems and provide solutions through suitable designs, structures, diagrams, and other appropriate analysis and design methods. (B1,B4)

b2. Identify a range of solutions and critically evaluate them and justify proposed design and development solutions.(B3)

b3. Design and implement practical software systems.(B5)

b4. Transform user requirements into system requirements.(B10,B18)

b5. Design and evaluate Man-Machine interfacing methodologies.(B6)

b6. Plan, execute and complete a significant design and, as appropriate, implementation within the time budget available. (B7,B18,B19)

C. Professional and practical skills :

On completion of this course the successful student will be able to:

c1. Plan and undertake a major individual / group systems analysis project in the area of computer science.(C1,C2,C4,C7)

c2. Prepare and deliver coherent and structured written technical reports.(C2,C12,C18,C19)

c3. Give technical systems analysis presentations suitable for the time, and audience.(C5,C10,C18,C19)

C4. Use appropriate computer-based design support tools.(C6,C9)

D. General and transferable skills:
On completion of this course the successful student will be able to:

d1. Use an appropriate mix of tools and aids in preparing and presenting reports for a range of audiences, including management, technical, users, industry or the academic community. (D1,D2,D3,D4,D7)
d2. Reveal communication skills, public speaking and presentation skills, and delegation, writing skills, oral delivery, and effectively using various media for a variety of audiences. (D6)
d3. Demonstrate an appreciation of the need to continue professional development in recognition of the requirement for life-long learning.(D8)

Course Contribution in the Program ILO's

ILO's		Program ILO's
A	Knowledge and understanding	A1,2,3,8,9,10,11,19
B	Intellectual skills	B1,3,4,5,6,7,10,18,19
C	Professional and practical skills	C1,2,4,5,6,7,9,10,12,18,19
D	General and transferable skills	D1,2,3,4,6,7,8

3. Contents

Design Phase
Implementation Phase
Testing Phase
Documentation
Prepare for finale presentation

Project Classifications:

****

How to Choose a Project?

The list of projects will be available for students to choose from. This list will contain the projects title and names of supervisors. The main selection and allocation of students to projects was made at the beginning of the year.
Usually each project is suitable for more than one student (normally 5 students). Therefore, groups of 5 students should be arranged by students themselves. Each group of 5 students should make their choices of projects on the selection form obtained from the GPC.
Usually each project is suitable for more than one student (normally 5 students). Therefore, groups of 5 students should be arranged by students themselves. Each group of 5 students should make their choices of projects

on the selection form. Obtained from the (Graduation Project Committee) GPC.
GPC will decide which Project groups will take according to their project desire which will comply with their average GPA of the group. Students are strongly encouraged to see the associated members of staff for projects they are interested in, to find out more about the projects.

4. Project Timetable

You are expected to be in regular attendance working on your project. You must co-operate in maintaining regular contact with your supervisor. It is an attendance requirement that you see your supervisor every week during term time. The formal project deliverables are a demonstration with discussion, and a written report.
Not contacting your supervisor regularly is equivalent to absence from lectures which shall not exceed 25%. Students who exceed the 25% limit without an emergency excuse acceptable to and approved by the Faculty Counsel , of the relevant faculty shall not be allowed to take the final presentation of the project and shall receive a mark of zero for the course. If the excuse is approved by the Faculty Counsel, the student shall be considered to have withdrawn from the course.
The project lifecycle should follow a sensible methodology and include the various stages identified in any Software Engineering course.
Work on the project itself, in particular use of equipment and computing facilities, must finish at the end of the 16th week of the semester. In some cases, this can be extended to another semester.
The project report and the Auxiliary Appendix together with any relevant discs, logic circuit and wiring diagrams etc., must be handed in to the Graduation Project Committee after being signed by the supervisor by the end of the 16th week of the second semester. The GPC will announce a time table for all project discussions. It forms a number of discussion committees, where each consists of two staff members and discusses one project.
The formal demonstrations and project discussion take place within one week after the submission of the report. The demonstration and discussion will contribute to the assessment of the "Quality of the project work".

5. Assessment methods:

The subdivision of marks within the project is:

- The supervisor evaluation: 40%
- The Project committee mark: 60% divided into
- Project Quality 42%
- Presentation and Oral Discussion,18%

6. Assessment Timing Grading:

Assessment Method	Timing	Grade(points)
Semester Work : seminars and reports	Weeks 14, 28	40
Final Exam (Demonstration, Discussion, Report)	week30	160
Total		200

Demonstration:

The demonstration is an informal presentation of the results of the project to one of the project discussion committee. The students will say briefly, what the aims of the project are, and will then demonstrate the results for example by running the program or using the equipment constructed.
The duration is about 40 minutes. See Guidance on demonstrations below for more information.

Report

The report is a formal written report on the project. This must be word processed. The report must follow a set of standards, given below, to facilitate its inclusion in the library and its usefulness for subsequent readers. Besides these, student will find it useful to read the slides of the talk given on writing, which is given in the lecture.
Copies of previous graduation project reports are available for reference in the Department.
Project documentation may be prepared on the PCs and printed on a laser printer. Students should hand in three soft cover copies of the report. After the discussion with the discussion committee ,students should make all the correction that are suggested by the committee within the specified period of time under the supervision of their supervisors, then they should handed in three Blue color hard cover copies of the project. The title of the project, the University, Faculty, Department names, and students' names are all written in Golden color.

Data Show:

Students are expected to make reasonable use of the Data show using power point presentation on the day of their demonstrations.

Guidance on Demonstrations:

A demonstration lasts about 40 minutes. The group of students should aim to spend no more than 25 minutes summarizing what their project is designed to achieve and showing what it currently does achieve. The rest of the time is spent in answering questions.

Note: Students should not attempt to demonstrate on the computer every last thing their program can do. A demonstration of its basic operation plus one or two highlights should suffice.
The mark given for the demonstration is based on the quality and quantity of the work attempted and the final state of achievement.
Students should have their working documents to hand and appropriate reference material, design workings, reasonably up-to-date listings, examples, tests, etc. Obviously, the kinds of things that are sensibly shown in a demonstration vary from project to project. If students are in doubt as to what to show, they should ask their Supervisors.
The discussion committee consists of two staff members. In general the supervisor of the project is not present.
In general, students should be available and ready to start their demonstrations within one week of their submission of the project.

Report Standards

1. The report is a formal written account of the project, satisfying certain standards for inclusion in a library. Students must hand in all relevant work on the project by the end of the 11th week of the second semester. In addition to the report, this includes program listings, discs, detailed logic and wiring lists, etc. It is important to meet this deadline. When students hand this to their supervisors it must be accompanied by a signed version of a form supplied by the GPC. In the case of programming projects, program listings must be submitted in some bound form in an ``Auxiliary Appendix'' that does not need to satisfy any particular standard apart from being neat and tidy. It is suggested however that an economical listing would be double-sided on A4.
   Here is a suggested structure for a report. Some projects may be rather different from others, and therefore have good reasons for not following these suggestions exactly. Supervisor guidance should anyway be sought!

- Introduction (1st chapter). What is the overall aim of the project. Why is it worth doing? Who will benefit from it? If the overall aim can be split into a number of sub goals, this is a possible place to do it. Finish with a chapter by chapter overview of the rest of the report.
- Background (2nd chapter). Analyze the background to the project. This should mention any previous work, here or elsewhere, and explain its relevance to the project. This could be an appropriate place to justify the choice of platform/software etc. used in the project.
- Description of the student's own work: Design and Implementation (a chapter each).

The structure of these chapters may reflect the project lifecycle, but do not write a diary of progress. The design should be clearly described and justified. Supporting diagrams should be used where appropriate and helpful. Keep your design description fairly high level. When describing implementation, confine yourself to the important, difficult, or interesting bits. Do not include large chunks of code.

Figures may well be useful.

- Results (1 chapter). What is the resulting system like to use. Include screen shots as appropriate.
- Testing and Evaluation (1 chapter). What testing was done? How confident are student that everything works correctly, and what evidence can they produce to support this claim? Have students evaluated the system against its aims? How did they make this evaluation?
- Conclusions (last chapter). What conclusions can students draw from the whole project? This should include a clear statement of what has been achieved overall, and will normally continue by suggesting areas of further related work, which could be done.

2. The report must be on paper of A4 size (210 x 297 mm). Only one side of paper should be used except in the Auxiliary Appendix.

3. The report must be produced using word processing facilities. The body of the report should be suitably divided into chapters and sections. Chapters, sections, pages, figures and appendices should all be numbered. Chapters, sections and appendices should have a heading. Each chapter should start on a new page. The body of the report should be preceded by a temporary title page, an abstract and a list of contents, and it should be followed by the references and then any appendices.

4. Straightforward and peripheral aspects of the work done should be mentioned only briefly, and description and explanation concentrated on important and interesting aspects. No extra credit is gained by writing a long report and excessive length is detrimental. More detailed description should be placed in appendices to the report. The appendices and/or the Auxiliary Appendix should contain any further documentation. Only the report itself will be held in the Department. Therefore, where important material is not included in it, e.g. because it is not convenient to produce it in A4 format, or it would be too bulky, it may sometimes be appropriate to include extracts in the report.

7. Copyright

In general, it is an infringement of copyright to reproduce any material, except short acknowledged quotations, from a published book or journal without the written permission of the publisher.
Except for the copying of material that is clearly from internal documents of the Department, any copying of books, journals, or documents required for the report should be checked with the supervisor before it is carried out.
Any material that is copied must be acknowledged as such. Attempting to present material written by others as your own is plagiarism and a serious disciplinary offence, as described in the University guidelines in the Undergraduate Handbook.

8. Avoiding Plagiarism

1. Unacknowledged direct copying from the work of another person, or the close paraphrasing of somebody else's work, is called plagiarism and is a serious offence, equated with cheating in examinations. This applies to copying both from other students' work and from published sources such as books, reports or journal articles.
2. Paraphrasing, when the original statement is still identifiable and has no acknowledgement, is plagiarism. A close paraphrase of another person's work must have an acknowledgement to the source. It is not acceptable for you to put together unacknowledged passages from the same or from different sources linking these together with a few words or sentences of your own and changing a few words from the original text: this is regarded as over-dependence on other sources, which is a form of plagiarism.
3. Direct quotations from an earlier piece of your own work, if not attributed, suggest that your work is original, when in fact it is not. The direct copying of one's own writings qualifies as plagiarism if the fact that the work has been or is to be presented elsewhere is not acknowledged.
4. Plagiarism is a serious offence and will always result in imposition of a penalty. In deciding upon the penalty the Department will take into account factors such as the year of study, the extent and proportion of the work that has been plagiarized, and the apparent intent of the student. The penalties that can be imposed range from a minimum of a zero mark for the work (without allowing resubmission) through caution to disciplinary measures (such as suspension or expulsion).

9. List of references:

- ***Required books:*** C. W. Dawson. The Essence of Computing Projects, A Student's Guide. ISBN 0-13-021972-X, Prentice Hall 2000.

  The project list and notes for guidance in carrying out a project are available in the Graduation Project Committee.

- ***Recommended books:***
- ***Periodicals, Web sites, etc.***

10. Facilities required for teaching and learning:

- Data Show or Smart Board

Course coordinator: Associate Prof.
Head of the Department: Prof. Hanafy Ismael

CONTENTS

CHAPTER 1

AN OVERVIEW ON THE GRADUATION PROJECT

By: Prof. Dr. Hafez S. Abdel-Wahab

One of the primary goal of academic training is to learn how to learn, i.e. to learn how to continuously absorb new knowledge. This is increasingly important in rapidly changing areas such as computer science and information systems. The process of exploring the unknown, studying and learning new things, building new knowledge about things that no one has understood before – that is what we think of as performing research. Undertaking a graduation project is one step towards an increased understanding of how to study, how to learn about complex phenomena, and towards learning how to build new knowledge about the world around us.

The graduation project builds and tests the skills and knowledge acquired during your education and training to become professionals. The graduation project is different from a traditional course in several ways; in its size, in its goals, in the form of examination, in the form of supervision and in the form of communication (personal dialogue, as opposed to lectures). A project represents a significantly larger workload than a single course. While traditional courses include lectures and lab work, where the focus is on acquiring knowledge in a specific subject area, the graduation project focuses on deepening your understanding of a subject. But above all, it should give you training in carrying out projects independently, at an advanced level.

This introductory chapter sets the scene for the book, discusses the characteristics of graduation projects, and explains how best to use this book in order to complete your project successfully.

1.1 Purpose of the Book

Computer science and information systems are two areas spanning a wide range of topics, for example, artificial intelligence, CASE-tools, database systems, human-computer interaction, information systems assessment, programming languages, operating systems, and web-based information systems.

The wide range of areas within computer science and information systems means that it is not always easy to formulate a problem that is suitable for a project, to choose the appropriate research method, or to develop a structure for a written report. Furthermore, many students experience uncertainty as to what to expect from a project, how to complete it within the given time frame, and how to attain the goals of the project. This is understandable since most students will have had no prior experience of a project as complex and as broad in scope as a graduation project. It is difficult to envisage what it will be like. These concerns are due partly to the lack of suitable textbooks and the lack of references specifically targeting students doing projects in computer science and information systems.

Moreover, the project is probably the biggest project you, as a student, will have undertaken in your academic life, and maybe even in your life. The supervisors, according to their deep knowledge and experience, will publish a list of projects (group Projects) to all students

that they can choose from. They also make an advisory session to explain to students the idea of each project & the background needs to start the project. Then according to students' choice, the graduation project is allocated to them (see below).

This book focuses on the process of carrying out a project, with a particular emphasis on the roles and responsibilities of the student, the supervisor and the examiner. The aim of the book is to bridge the gap between different research methods and describe the general process of carrying out a project in the computing disciplines. In this book we identify a series of actions that should be of assistance to you when planning and carrying out your project.

1.2 Objectives of Graduation Projects

Firstly, the graduation project is a group project (see below 1.5). What characterizes a project is the fact that it is something that is planned, has a specific purpose, lasts only for a limited time with a clear start and finish, and is undertaken with finite resources with respect to personnel, money, and equipment.

The Graduation Project combines a strong theoretical background with hands-on practice. It provides the students with lifelong learning capabilities, the necessary skills to promote their intellectual growth and optimal benefit from academic modules. These skills include outstanding computer skills, group (team) work, project management, project presentation techniques, and language proficiency, research skills through the Internet as well as data gathering and analysis. Students are trained to adequately use these skills in handling and solving problems.

You can view the graduation project as serving several (sometimes overlapping) purposes:

- **Learning more.** The project is an opportunity for studying a subject in more depth.
- **A stepping stone towards finding and securing a job.** You may view the project as preparation for working life, by practicing your skills and knowledge on real- world problems.
- **A stepping stone towards graduate studies.** You may use the project as preparation for graduate studies, by exploring a research problem and learning about the research process.

In addition, your university probably sees your project as serving two further purposes. Typically, these can be captured by the following goals, shared by most projects, and which emphasize the educational motivation and the research motivation.

The first goal is the **educational** part of the project. This can be viewed as a test to show that you have mastered previously attained knowledge and skills, and know how they can be applied to a problem that is more realistic than those normally presented in courses. In detail, the "educational" part has the following set of learning goals. The project should:

1. develop your critical thinking;
2. enhance your ability to work independently within group of colleagues;
3. increase your understanding of how to use and appreciate scientific methods as tools for problem solving; and

4. develop your presentation skills, oral as well as written.

With “critical thinking” we mean the ability to approach something new in a systematic and logical way, and to use creative and diverse, yet systematic ways to approach and solve a problem. Further, to support opinions with trustworthy evidence, data and logical reasoning; and also the ability to decide how a problem fits into a larger context. Those who have become comfortable with thinking in this way can often apply these acquired skills also in everyday life.

The second goal is the **research** part of the project, in which you will deepen your understanding of the subject area, and contribute to the common knowledge and understanding of the subject area. The research part of the project introduces students to the fascinating world of science and makes the latest knowledge available to them. In fact, you should expect that faculty studies, wherever appropriate and possible, incorporates the findings of the latest research. It is generally considered that incorporating elements of research methodology and giving students the opportunity to undertake their own research related project, or to be involved in a bigger research project, facilitates individual students’ development towards becoming independent and critically thinking people. It is our opinion that thesis projects offer excellent opportunities for closing the gap between research and teaching.

Of course, the link benefits you as a student in many ways. Not only are teachers able to pass on their knowledge and experience in the topic area, as well as research methods, but it also gives an excellent opportunity for you to become involved in leading-edge research

activities. You will notice that you will be working more closely with faculty members, and very often in a more collegial and informal way than is normally the case in traditional courses. As mentioned before, the project itself gives insights into what research is and how it is performed, and is a good preparation for post-graduate studies, since it includes initial training in using research methods.

It should be noted that:

- Research is accomplished by the students so it should be a major effort and readiness commensurate with the requirements of the project.
- Professors hold a formal seminar for students who are assigned/chosen to the project to guide them and provide them with advice and guidance on the scientific origins of scientific research.
- The project is completed by students under the guidance of a faculty members in the specialty and under his/her supervision, and through periodic scheduled with the supervisor.
- The supervisor provides the students with the scientific advice and guidance, as well as some scientific references if necessary to prevent students from falling into the curriculum or subject errors or related to both of them or give inadequate interpretation to the conclusion and thoroughly review the results.

The project is evaluated in public discussion to highlight the value of the objective of the project and instill confidence in the student and also see his capability to speak in public. The important standards (as will be

seen in the assessment section) are taken into consideration in the process of evaluation as: -

- Degree of importance of the addressed problem.
- The validation of the proposed solutions.
- The impact of the results.
- The significance of recommendations regarding problem conceptualization.
- Ensuring the use of authentic references and follow scientific methods in the search.
- Granting the integrity of research hypotheses and the reasonable conclusions.
- To verify that the research aspects and objectives are completely covered.
- To ensure the professional context formulation and expression and the integrity of approached results.
- The concern of Scientific integrity and copyrights
- To prove his linguistic capabilities in the formulation of the results and conclusions.

1.3 Research Topics/Directions for CS & IS

Computer science and information systems have a wide range of topics. For example, Computer-Aided Software Engineering (CASE) technologies are tools that provide automated assistance for software development. The goal of introducing CASE tools is the reduction of the time and cost of software development and the enhancement of the quality of the systems developed. Also here are some summaries of these topics' tracks:

- Data Management & Analytics
- Intelligent Systems & Optimizations
- Machine Learning & Intelligence
- Pervasive Sensing & Systems
- Multimedia
- Human-Computer Interaction
- Software Engineering & Systems
- Information Security & Cybersecurity
- Information Systems Management
- Analytics & Decision Support Practice
- IoT Internet of things
- Cloud computing
- Internet information system

These research topics can be divided (grouped) into the following categories:

- **AI and Machine Learning**

  We primarily focus on applying AI techniques such as natural language processing, machine learning, multi-agent, and intelligent tutoring to a variety of applications including knowledge management (knowledge acquisition, filtering, organization, and reuse, etc.), text mining, information extraction, ontology development, and Web-based learning.

- **Data Science**

  The area includes both fundamental and applied research in database management, data mining, and data warehousing. Faculty

research includes database and data warehouse; data mining methodologies and applications, specifically privacy preserving data mining, anomaly detection, spatial data mining, and data mining for digital government; data integration; and mobile databases. Research has been conducted in application areas such as cyber security, GIS, healthcare, and digital government.

- **Health Informatics**

  Health Informatics is the interdisciplinary study of the design, development, adoption, and application of IT-based innovations in health services delivery, management and planning. HIT plays a crucial role in terms of improving the quality of care, reducing health care costs, and enabling better health outcomes.

- **Human-Computer Interaction**

  A number of our faculty conduct research in the area of human-computer interaction (HCI). Our faculty investigates HCI from a broad variety of perspectives. As a result, HCI research addresses a diverse collection of interrelated research questions centered on the design, implementation, and evaluation of highly usable interactive systems. The two core areas are:

  - Accessible Computing — broadly defined to include issues associated with disabilities, age, culture, as well as context-aware computing, among others.
  - Human-Information Interaction — studies information behavior and the design of user interaction methods to support that behavior.

- **Software Engineering**

  Systems and Software Group analyzes and studies real life systems, and the processes of adopting, designing, developing, testing, and maintaining software solutions to improve organizational productivity, efficiency, and outcomes. The importance of aligning the organizational missions and goals with those of software projects is well recognized which prompts a research focus on the people and organizational issues on both systems and software sides in addition to the technical ones.

- **Transportation Technologies**

  Transportation technologies address the issues of safety and traffic jams. There are quite many solutions suggested. One of the solutions is using shared vehicle phone applications. It allows reducing the number of private cars on the roads. On the other hand, if more people start preferring private vehicles, it may cause even more traffic issues.

1.4 Actors in the Project

The three main actors in the project are you (the student), the supervisor and the examiner(s). Of the three actors, you are the most important, since you are the one who moves the project forward. You focus on solving some well-defined problem in a specific area, and thereby increase your understanding of the area. But you also learn methods that can be used to approach, structure and solve complex problems.

The supervisor is your ally. He or she should not only give you advice to help you achieve success in your project but will also critically point out strengths and weaknesses. Normally he or she is a domain expert in the area in which you are doing your project. The dialogue between you and the supervisor serves as a com- pass for establishing directions when exploring new areas.

In contrast, the examiner is the person who critically evaluates your work, and recommends or decides the grade. The examiner is not necessarily a domain expert in the specific topic of your work, but normally has a good understanding of the subject area generally. More importantly, the examiner has significant experience, enabling him or her to review your work with respect to both content and method.

A positive interaction between these three actors is vital for the successful completion of a project. Note that while these are three distinct roles and are usually performed by three different people, the roles of supervisor and examiner may sometimes be carried out by the same person. However, there are many advantages of keeping the roles strictly separate.

1.5 Group (Team) Project

All graduation projects for your Faculty are designed to be group (team) projects. By accomplishing group tasks, students learn to listen, trust and support each other, while developing life skills such as communication and collaboration – skills that can't be learned from a textbook, interactive or not. Learning to get along with peers, for example, isn't something you can pick up through memorization.

Many Groups (team)-building activities incorporate such skills as active listening, questioning assumptions, giving clear directions, problem-solving or learning how to ask effective questions."

Group projects can help students develop a host of skills that are increasingly important in the professional world. Positive group experiences, moreover, have been shown to contribute to student learning, retention, and overall college success.

Properly structured, group projects can reinforce skills that are relevant to both group and individual work, including the ability to:

- Break complex tasks into parts and steps
- Plan and manage time
- Refine understanding through discussion and explanation
- Give and receive feedback on performance
- Challenge assumptions
- Develop stronger communication skills.

Group projects can also help students develop skills specific to collaborative efforts, allowing students to...

- Tackle more complex problems than they could on their own.
- Delegate roles and responsibilities.
- Share diverse perspectives.
- Pool knowledge and skills.
- Hold one another (and be held) accountable.
- Receive social support and encouragement to take risks.

- Develop new approaches to resolving differences.
- Establish a shared identity with other group members.
  - Find effective peers to emulate.
  - Develop their own voice and perspectives in relation to peers.

While the potential learning benefits of group work are significant, simply assigning group work is no guarantee that these goals will be achieved. In fact, group projects can – and often do – backfire badly when they are not designed, supervised, and assessed in a way that promotes meaningful teamwork and deep collaboration.

1.6 Allocation

The Project Allocation contains the title of the project, group members of students, date of allocation, and expected date for submission. Description of the task of the graduation group project (including objective, stages for developing the project task for each member of the group). It also includes time schedule for each subtasks & finally the method and weight of the assessment.

Points to remember when reading your allocation of the group project. Here are some tips:

- Choose a part of project that you are interested in. The research process is more relevant if you care about your topic.
- Narrow your topic to something manageable.
  - If your topic is too broad, you will find too much information and not be able to focus.
  - Background reading can help you choose and limit the scope of your topic.

- Review the guidelines on topic selection outlined in your allocation. Ask your professor or TA for suggestions.
- Talk about research ideas with a friend. S/he may be able to help focus your topic by discussing issues that didn't occur to you at first.

Here is a template for the allocation of the graduation project as adopted by the Faculty of Computers and Artificial Intelligence. The Next pages show a complete example of the allocation of the graduation project for a project for computer science department. It contains a very comprehensive details about the group project assigned to a group of students.

Modern University for Technology and Information

The Faculty of Computers & Artificial Intelligence

Department: Computer Science

ALLOCATION OF FINAL PROJECT

A- Basic Information:

1) Field:

2) Title of Project:

3) Semester: Fall …. (Ac. Year …. – …….)

4) Student Group members:

(1) Student 1 (2) Student 2

(5) …..

5) Supervisors:

Prof. .....

T.A. .....

6) Allocation received on:

7) Expected Date of Submission:

B-Description Of The Task :

C-Time Schedule:

No.	Date	Task
Fall Semester		
1	Wk: 1 – 4	
2	Wk: 5 – 6	
3	Wk: 7 – 9	
4	Wk: 10 – 13	
Spring Semester		
5	Wk: 1 – 3	
6	Wk: 4 – 9	
7	Wk: 10 - 11	
8	Wk: 12 – 14	

D-Method Of Assessment:

1) Year Work (20% for each Semester) 40%
2) Report quality, presentation, and oral discussion 60%

( )	( )	( )
Supervisor	**Head Of Department**	**Dean Of Faculty**

Modern University for Technology and Information
The Faculty of Computers & Artificial Intelligence
Department: Computer Science

ALLOCATION OF FINAL PROJECT

A- Basic Information:

1. Field:

Signal Processing, Machine Learning, Deep Learning

2. Title of Project:

Prediction of Cardiac Arrhythmia using ANN

3. Semester: Fall 2020 (Ac. Year 2020 – 2021)

4. Student Group members:

(1) List
(2) Of Students
(3) ..
(4) ..
(5) ….. (9)

5. Supervisors:

Prof. Eng. Hafez M.S. Abdel-Wahab
T.A. Fatma Refaat
T.A. Reham Fares

6. Allocation received on: Sept. 2020

7. Expected Date of Submission: June 2021

B- Description Of the Task:

The electrocardiogram (ECG) has become a useful tool for the diagnosis of cardiovascular diseases as it is fast and noninvasive. It has been reported that about 80% of sudden cardiac deaths are the result of ventricular arrhythmias or irregular heartbeats. While an experienced cardiologist can easily distinguish arrhythmias by visually referencing the morphological pattern of the ECG signals, a computer-oriented approach can effectively reduce the diagnostic time and would enable the e-home health monitoring of cardiovascular disease. However, realizing such computer-oriented approaches remains challenging due to the time-varying dynamics and various profiles of ECG signals, which cause the classification precision to vary from patient to patient, as even for a healthy person, the morphological pattern of their ECG signals can vary significantly over a short time.

Artificial Neural Network (**ANN**) are useful tools that have been used in pattern recognition applications, such as the classification of handwriting and object recognition in large archives. Convolutional neural networks (**CNN**) is a very widely used ANN where the connectivity between neurons is similar to the organization of the visual cortex in animals, which makes CNNs superior to other methods in the recognition of the pattern and structure of items. CNN also provide a number of advantages over conventional classification techniques in biomedical applications. CNNs are a specialized type of neural network with an inherent grid-like

topology for processing input data in which nearby entries are correlated, such as those in a two-dimensional (2D) image.

Hands on the **CNN** to learn how it works, how to build, use, and control. Demonstrating students' capabilities of understands this technology & apply their gained knowledge through their studying the undergraduate courses. The objective of this group project is *to present an overview of a state-of-the-art* to design & implement a full-featured system using Deep Learning capable for detecting Heart abnormalities, specifically Prediction of Cardiac Arrhythmia.

All students should read and document a complete survey, of the previous work related to the required task. They should collect papers, recent as available, covering different techniques and algorithms to solve their problem.

Students choose their group leader. Group leader divide tasks of the project to all students as necessary to accomplish the required tasks of the project. Generally, with some details:

- All students learn very precisely the tools to use in development of the project
- Tools recommended to use in this project Python with Tensorflow & /or Keras library.
- Students starts this my Hands on the tools (including setup of the platform carefully & efficiently). **This should be finished by the end of first month (Hands-On)**
- Students starts by developing simple example using this platform.
- Deep learning stages to follow consists of:

  1) Understanding the problem in terms of AI (concept)

  2) Survey about similar problem used with AI

  3) Choose the suitable model for the tasks

4) Design the chosen model with necessary explanation

5) Training, verifying & testing the model.

6) Choosing suitable data set to be used in (1) & (2)

7) Evaluating the designed Model

8) Understanding & using the concept of pretrained model (transfer learning)

The project should be written in a conclusion report complying with the faculty standards.

C- Time Schedule:

No.	Date	Task	Assigned to
Fall Semester			
1	Wk: 1 – 4	Literature review on prediction of Arrhythmia using ANN.	All Students
2	Wk: 5 – 6	Understanding the nature of Cardiac signal & the main features that is most suited to fulfill the required tasks. Defining the objective of the required **predictor** system. Preparing the **dataset** (Wav file).	All Students
3	Wk: 7 – 8	Literature review for different approaches for both feature extraction phase & classifier phase (Training). Complete review of **CNN** & is application is the predictor.	All Students According to the Description Above
4	Wk: 9 – 10	Extracting features required that are to be used in **SV** system with experimentation to measure its effectiveness. (**ML approach**)	
5	Wk: 11 – 13	Analysis & detailed description of the system including the System block diagrams, function diagrams, main features used, and the	All Students According to

No.	Date	Task	Assigned to
		classification algorithm used. This system is based of **Deep Learning** Models (CNN).	the Description Above
Spring Semester			
6	Wk: 1 – 3	Implementation of selected algorithms of the required systems with verification of the results of training phase.	All Students According to the Description Above
7	Wk: 4 – 6	Running the test phase for the implemented system with appropriate testing scheme with the available dataset. Analyzing the results with conclusions.	
8	Wk: 7 – 8	Study one or more methods to enhance system performance (e.g. combining different features over one classifier or two classifiers) and using **pretrained** Models.	
9	Wk: 9 - 11	GUI implementation and testing system robustness.	All Students
10	Wk: 12 – 14	Revising the Project Document & Presentation	

D- Method Of Assessment:

3) Year Work (20% for each Semester) 40%
4) Report quality, presentation, and oral discussion 60%

( )	( )	( )
Supervisor	**Head of Department**	**Dean of Faculty**
Prof. Hafez M.S. Abdel-Wahab	***Prof. Hanafy Ismael***	***Prof. M. El Mayiah***

1.7 Assessment

When doing a project, it is important to familiarize yourself with the criteria and expected standards defined by your faculty and/or department. In the process outlined here, the assessment of projects involves a set of following criteria.

- **General**
  - Relevance and fulfillment of the tasks in allocation
  - Significance of findings & results analysis
  - Degree to which the work is the student's own work.
- **Report**
  - Clarity of presentation
  - Consistency between different parts of the report
  - Degree of insight apparent from the arguments presented to support the decisions made in the project
  - Ability to handle references and citations
  - General stylistic impression
- **Defense**
  - Degree of insight apparent from the arguments presented to support claims and conclusions
  - Degree of insight apparent from discussion in response to relevant questions Other
  - Fulfilment of deadlines and other formal requirements

Assessing group projects can be broken into assessment of both individuals and groups (teams) by assessing their products and the learning process. Individual students can be assessed using a self-assessment of

individual contributions, an individual product, and a peer assessment. The Group can be assessed with a group product, a group reflection about the process of working in a team and a team effectiveness questionnaire.

The group graduation project assessment is divided into three main categories: (1) term activity assessments (by project supervisor); (2) report quality; and (3) presentation and oral discussion.

1. **Supervisor assessment report**

   The report includes evaluation of each student in the group (individually). The task assigned to him/her, how it was accomplished, rate of knowledge gained, understanding the task, analysis & design phases of the project and implementation & testing. Here are sample of such supervisor report. Supervisors have 400 Points: Independent worker:200, Meetings:200

Modern University for Technology and Information

The Faculty of Computers & Information

Department: Information Systems

Year Work Evaluation of The Final Project

Department: ***Information Systems***

Student Name: ***Student (1) Name***

Title of Project: ***Online Reservation System (Web-Based & IPhone Based)***

Semester: Fall 2015 (academic year 2015/ 2016)

Supervisor Comments: *The objective of this group project is to design & implement a full-featured Online Reservation System (Web-Based & IPhone Based). The project is an integrated web-based (with native iPhone-iPad apps) solution for booking service with table reservation. The means that the back-end (data storage) will be the same for the two parts. The logic stage will carry the required tasks for each part while the presentation layer (front-end) may differ for each.* ***Student(1) ,*** *is responsible for the* ***development of the Native iPhone parts****. With other students in the group, the integration of their parts to produce the required system, fully working & tested. Full Details description of the tasks assigned for him is given in the Allocation of the Project handled to student at the beginning of the Fall 2015*

No	Task	Date	Supervisor Comments
1	Literature review of different techniques & method for creating web-based reservation system. Survey of Web-based vs native (mobile type) applications. Survey of the main features & benefits of reservation system.	Wk: 1 – 4	*Very Good review*
2	Defining the objective of the proposed Application including the key features & tasks required for this application. Studying of similar application, their features, architecture (As per Allocation).	Wk: 5 – 6	*Very good understanding of the objective & tasks required*
3	Full description and analysis of the system as a whole and each part of it (As per Allocation)	Wk: 7 – 9	*Good description & Analysis but done with some delay*
4	Analysis & detailed description of the proposed system including the System Architecture & other OOAD Diagrams. The application as described in Allocation.	Wk: 10 – 13	*Good, detailed description of the system with diagrams*
5	The Design of the required software application (as per allocation) with full details & description with necessary OO diagrams. Web-based screens, menus, forms, reports, …etc.	Wk: 1 – 3	*Excellent Design with Very good description*
6	Implementation of the application mentioned in the Allocation with full features & controls.	Wk: 4 - 9	*Very good software implementation*
7	Testing the application software.	Wk: 10-11	*Very good bug & functional Test with required auditing*
8	Revising the Project Document & Presentation	Wk: 12 - 14	*Very Good Report*

Year Work Evaluation (400 Points) : Independent worker:190 /200, Meetings: 182/200, Total : 372/400

General Comments: *Regular meeting & discussion with student showed very good understanding of the tasks. Very good implementation of the required task that shows a very good knowledge & practice of toughed materials, good skills & experience. Shelf dependent with excellent work.*

Supervisor Name: Prof. Eng. Hafez M.S. Abdel-Wahab

Signature: (……)

Date: 20th June 2016

Modern University for Technology and Information
The Faculty of Computers & Artificial Intelligence
Department: Computer Science

Year Work Evaluation of The Final Project

Department: ***Computer Science***

Student Name: ***Student (1) Name***

Title of Project: ***Prediction of Cardiac Arrhythmia Using Artificial Neural Networks***

Semester: Fall 2020 (academic year 2020/2021)

Supervisor Comments: *Hands on using ANN for Cardiac Arrhythmia. Demonstrating students' capabilities of understands this technology & apply their gained knowledge through their studying the undergraduate courses. The objective of this group project is* to present an overview of a state-of-the-art *methods used to design & implement a full-featured system using Deep Learning. He was the project leader. All algorithms were implemented using tensor-flow with* ***Python****. Detail description of the tasks assigned for him/her is given in the Allocation of the Project handled to student at the beginning of the Fall 2020.*

No.	Task	Date	Supervisor Comments
1	Literature review on prediction of Arrhythmia using ANN with a good comprehensive research survey on different Artificial Neural Network techniques for Detection.	Wk: 1 – 4	*Excellent review*
2	Understanding the nature of Cardiac signal & the main features that is most suited to fulfill the required tasks. Defining the objective of the required predictor system. Preparing the dataset (Wav file).	Wk: 5 – 6	*Thorough understanding of the tasks required*
3	Full description and analysis of the system and each part of it (As described per allocation).	Wk: 7 – 8	*Excellent review analysis done with no delay*
4	Analysis & detailed description of the system including the System block diagrams function diagrams, main features used, and the classification algorithm used. This system is based of **Deep Learning** Models (CNN). The application as described per project allocation.	Wk: 9 – 13	*Good description of the system using Python*
5	Implementation of selected models & algorithms of the required systems with verification of the results of training phase.	Wk: 1 – 5	*V.G. Design of required Algorithms*
6	Design & construct the test phase for the implemented system with appropriate testing scheme.	Wk: 6 – 8	*Excellent Implementation of required Algorithms*
7	Testing the application performance & robustness. Analyzing the results with conclusions.	Wk: 9 – 11	*Complete Test with results analysis*
8	Revising the Project Document & Presentation	Wk: 12 – 14	*Excellent Report*

Year Work Evaluation (400 Points): Independent worker:195 /200, Meetings:196/200, Total: 391/400

General Comments: Regular meeting & discussion with student showed excellent understanding of the tasks. Excellent & neat implementation of the required task that shows excellent & clear knowledge & practice of toughed materials, excellent skills & experience.

Supervisor Name: Prof. Eng. Hafez M.S. Abdel-Wahab

Signature: (…….)

Date: 12th July 2021

2. Quality of the report

This part contains the following items with the corresponding marks (420 out of 1000)

- Quality of Processes: (210)
  - Clear introduction / definition of project
  - Detailed analysis / methodical design
  - Thorough implementation / testing
- Quality of Conclusion: (30)
  - Conclusions summarized with acumen
- Quality of Report: (90)
  - Sensibly organized subdivided material
  - Thorough, clearly presented material
  - Relevant, clearly presented, valid material
- Quality of product: (90)
  - How good is the product?
  - Substantial work completion level.

3. Presentation & oral discussion

This part contains the following items with the corresponding marks (180 out of 1000)

- Well-organized presented material (100)
- Relevant, clear answer to questions (80)

Check the following Marking Sheet. The scale in this sheet is per the Marking Categories shown in the next table. (1-6) scale.

Ministry of Higher Education
Modern University for Technology and Information
Faculty of Computers & Artificial Intelligence

Final Project Marking Sheet

Student Full Name: ……
Project Title: ……

Tick the prompts and use these in deciding a mark.

Supervisor Evaluation:

	1	2	3	4	5	6				
Student Performance							See Attached Detailed Sheet	400		Student Performance / 400

Project Quality:

	1	2	3	4	5	6				
Project area not clearly defined							Clear introduction / definition of project	45		Quality of Processes: / 210
Scant analysis of problem/design							Detailed analysis / methodical design	90		
Poor implementation / testing							Thorough implementation / testing	75		
Poor, shallow conclusion							Conclusions summarized with acumen	30		Quality of Conclusion: / 30
Poorly organized report							Sensibly organized subdivided material	30		Quality of Report: / 90
Poorly presented literature review/ project management / self-evaluation							Thorough, clearly presented material	30		
Poor referencing / tables / diagrams							Relevant, clearly presented, valid material	30		
Poor theory / design / software							A very good product	45		Quality of product: / 90
Insignificant work completed							Substantial work completed	45		
										/ 420

Presentation and Oral Discussion

	1	2	3	4	5	6				
Poorly presented material							Well-organized presented material	100		Presentation and Oral Discussion / 180
Poor, irrelevant answer to questions							Relevant, clear answer to questions	80		

Total	( ) / 1000
Overall Mark	%

General Comments: (e.g. weak/strong points, justification of final mark in relation to above categories)

Supervisor

Name:	Signature:……

Committee

First Member Name:	*Signature:*……
Second Member Name:	*Signature:*……
Head Name:	*Signature:*……

Date: / /

Ministry of Higher Education
Modern University for Technology and Information
Faculty of Computers & Artificial Intelligence

Final Project Marking Categories

	(1) Fail	(2) 3rd	(3) 2/2	(4) 2/1	(5) 1 st	(6) good 1st
	< 40	40 - 50	50 - 60	60 - 70	70 - 80	> 80
	< 50%	50% - 64%	65% - 74%	75% - 84%	85% - 94%	95% and above
Total Mark	Poor accuracy. Little relevance. Unclear arguments. Insufficient evidence of background reading and research.	Mostly satisfactory. Reasonably coherent. Shows basic understanding. Attainable objectives. Adequate software engineering / analytical methods.	Satisfactory to good. Adequate understanding. Clear and attainable objectives. Good software engineering / analytical techniques.	Good to very good work. Broad understanding and good knowledge of subject. Relevant conclusions. Some ability to apply techniques to unfamiliar topics.	Very good to excellent work. Thorough understanding of topic. Good quality critical appraisal. Excellent software engineering / analytical techniques.	Exceptional work. Nearly flawless.

CHAPTER 2

PROJECT MANAGMENT

By: Prof. Dr. Hanafy Ismail

2.1 Introduction

This chapter explains the process of managing a project. Project management (PM) may be the most important aspect of systems development. Effective PM helps to ensure:

- The meeting of project expectations.
- The satisfying of budget and time constraints.

The nature of projects has changed from custom development to implementing packaged software such as ERP and data warehousing.

The following are key terms to in project management that you should be aware of:

- **Project**

  A planned undertaking of related activities to reach an objective that has a beginning and an end.

- **Project management**

  A controlled process of initiating, planning, executing, and closing down a project

- **Project manager**

  A systems analyst with a diverse set of skills—management, leadership, technical, conflict management, and customer relationship—who is responsible for initiating, planning, executing, and closing down a project

  The project manager needs to have both technical and organizational skills, and should be a good communicator with a strong sense of time management.

- **Deliverable**

  The end product of an SDLC phase.

  Each SDLC phase will have some deliverable. Planning's deliverable is typically a feasibility report, a budget, and a preliminary schedule. The Analysis phase usually results in a set of functional requirements. Logical design results in system specifications and physical design delivers actual programs and databases. So, each phase has a deliverable.

2.2 Representing and Scheduling Project Plans

Gantt Charts

Network Diagrams

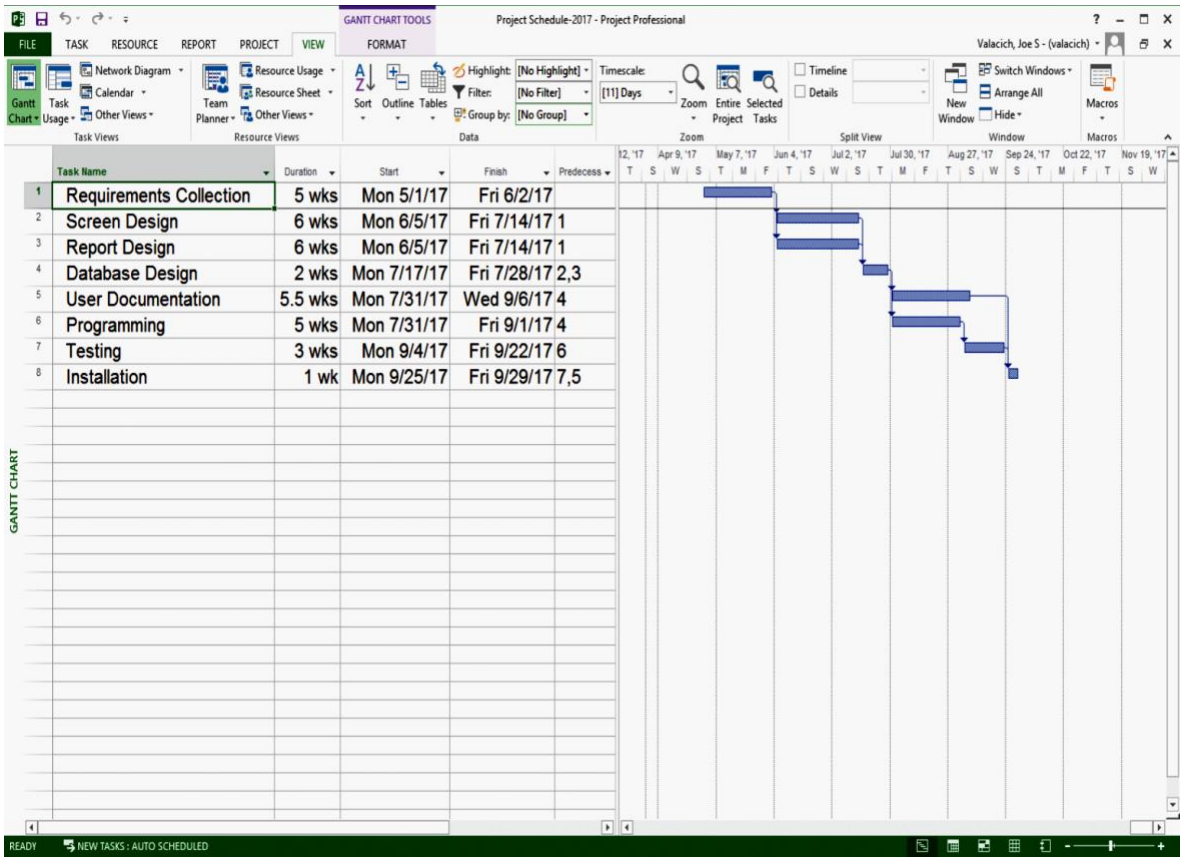
PERT Calculations

Critical Path Scheduling

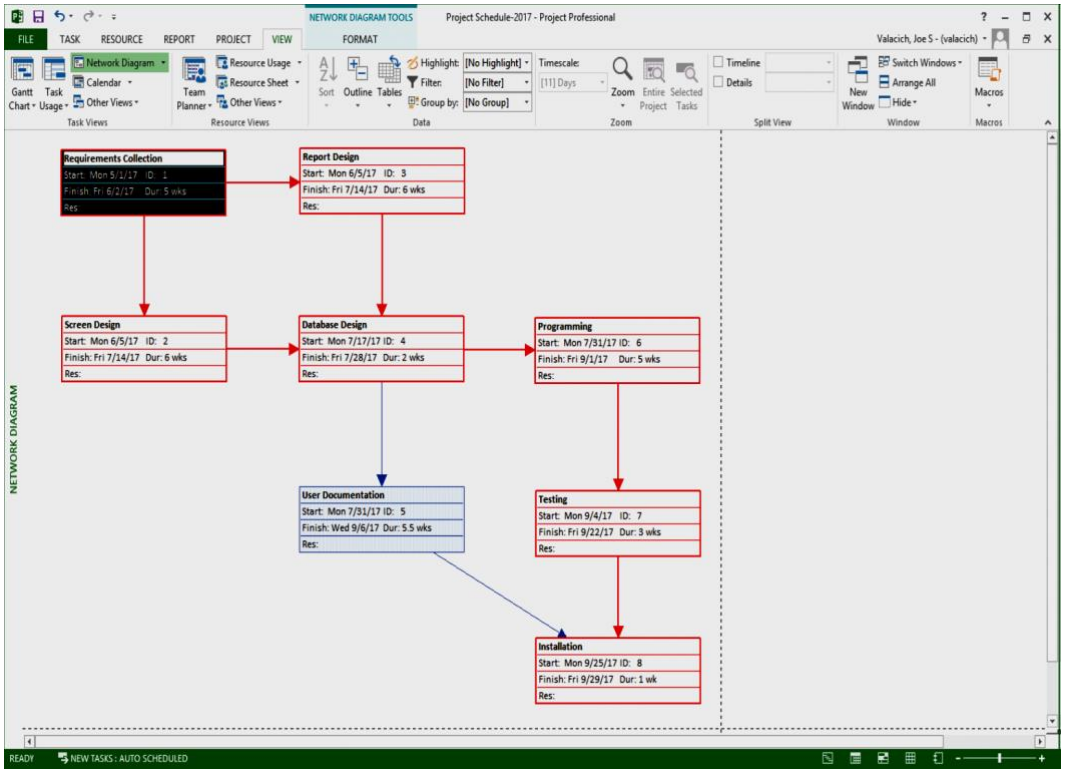
Project Management Software

2.2.1 Gantt Charts vs. Network Diagrams

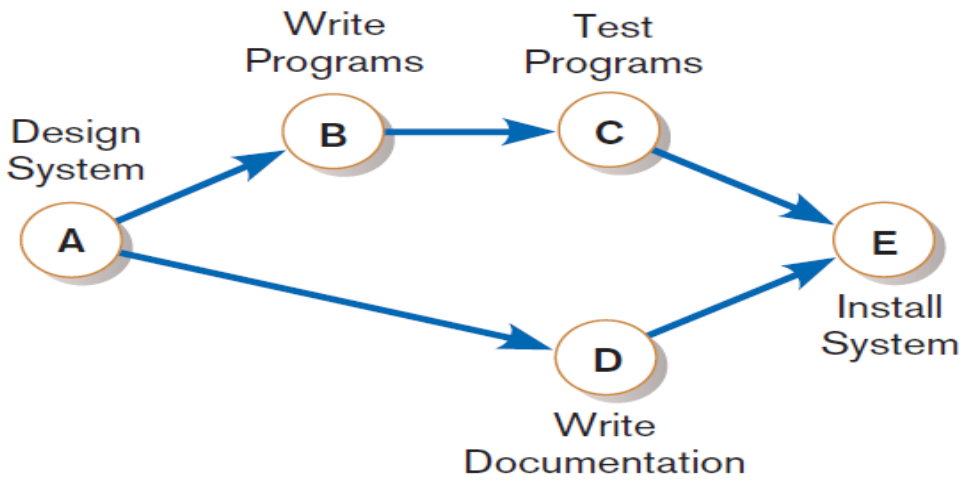
- **Gantt charts**
  - Show task durations.
  - Show time overlap.
  - Show slack time in duration.



- **Network diagrams**
  - Show task dependencies.
  - Do not show time overlap, but show parallelism?
  - Show slack time in boxes.



A network diagram showing activities (represented by circles) and sequence of those activities (represented by arrows)



This diagram shows that task D can take place at the same time that B and C are going. But B needs to complete before C can begin. And, both C and D must be completed in order for E to begin.

2.2.2 PERT: Program Evaluation Review Technique

The technique that uses optimistic (*o*), pessimistic (*p*), and realistic (*r*) time estimates to determine expected task duration

Formula for Estimated Time:

ET = (*o* + 4*r* + *p*)/6

Here is an example of PERT Analysis

ACTIVITY	TIME ESTIMATE (in weeks) *o*	*r*	*p*	EXPECTED TIME (ET) $\frac{o + 4r + p}{6}$
1. Requirements Collection	1	5	9	5
2. Screen Design	5	6	7	6
3. Report Design	3	6	9	6
4. Database Design	1	2	3	2
5. User Documentation	2	6	7	5.5
6. Programming	4	5	6	5
7. Testing	1	3	5	3
8. Installation	1	1	1	1

2.2.3 Critical Path Scheduling

A scheduling technique whose order and duration of a sequence of task activities directly affect the completion

- *Critical path:* the shortest time in which a project can be completed
- *Slack time:* the time an activity can be delayed without delaying the project

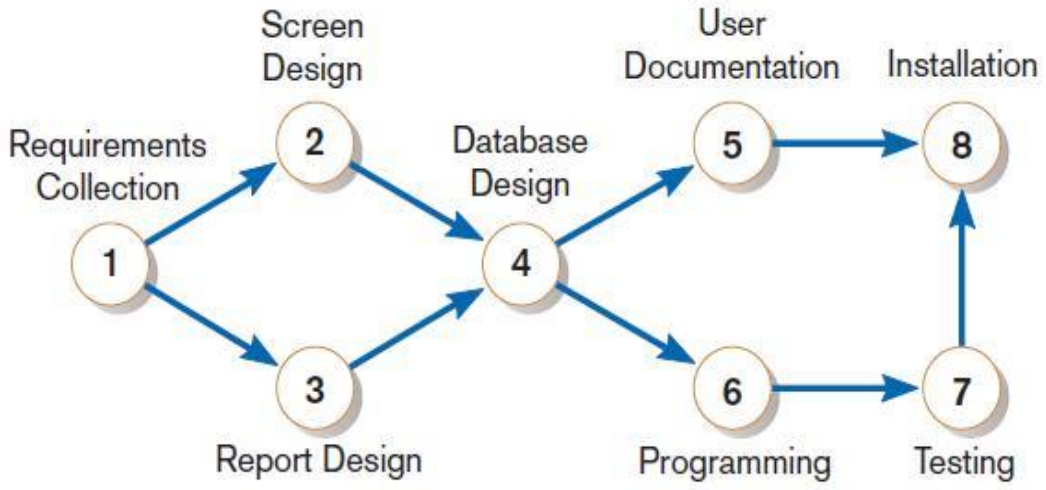
Here is an example of Critical Path:

(Dependencies between tasks)

PRECEDING ACTIVITIES indicate the activities that must be completed before the specified activity can begin.

ACTIVITY	PRECEDING ACTIVITY
1. Requirements Collection	—
2. Screen Design	1
3. Report Design	1
4. Database Design	2,3
5. User Documentation	4
6. Programming	4
7. Testing	6
8. Installation	5,7

Network diagram shows dependencies. A network diagram that illustrates the activities (circles) and the sequence (arrows) of those activities



- **Determining the Critical Path**

  Calculate the earliest possible completion time for each activity by summing the activity times in the longest path to the activity. This gives total expected project time.

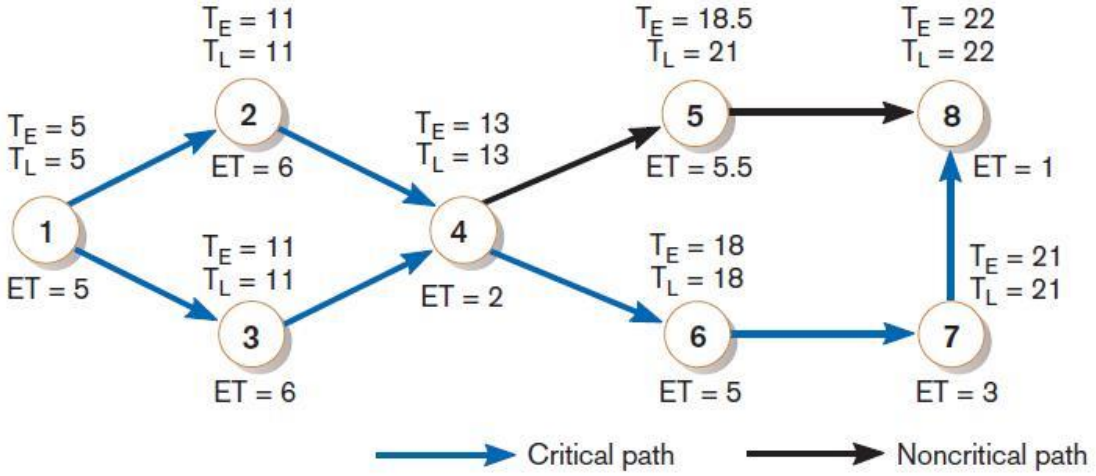
  Calculate the latest possible completion time for each activity by subtracting the activity times in the path following the activity from the total expected time. This gives slack time for activities.

  Critical path contains no activities with slack time.

  The key here is to determine slack time. Tasks that are not on the critical path may have some slack time, but no task on the critical path has any slack time. The critical path identifies the “bottleneck” in the project.

- **Critical Path Calculation**

  A network diagram for the SPTS project showing estimated times for each activity and the earliest and latest expected completion time for each activity



Early and late time calculations are determined and critical path established. (Note: Activity #5 can begin late without affecting project completion time.)

You can see that the task with slack is 5 (user documentation). That makes sense, because this happens while 6 and 7 happen. The total expected duration for user documentation (task 5) is 5.5weeks whereas the total duration for programming and testing (tasks 6 and 7) is eight weeks. So, task 5 has slack and is therefore not in a critical path.

It turns out that there are two critical paths:

1,2,4,6,7,8

1,3,4,6,7,8

This was a very simple example. A real project may have dozens or even hundreds of tasks, so it’s good to have an automated way of calculating the critical path.

Activity slack time calculations for the project; all activities except number 5 are on the critical path

ACTIVITY	T_E	T_L	SLACK $T_L - T_E$	ON CRITICAL PATH
1	5	5	0	✓
2	11	11	0	✓
3	11	11	0	✓
4	13	13	0	✓
5	18.5	21	2.5	
6	18	18	0	✓
7	21	21	0	✓
8	22	22	0	✓

Once you know of which tasks are critical and which are not, you can use this to make other scheduling and resource allocation decisions.

2.2.3 Using Project Management Software

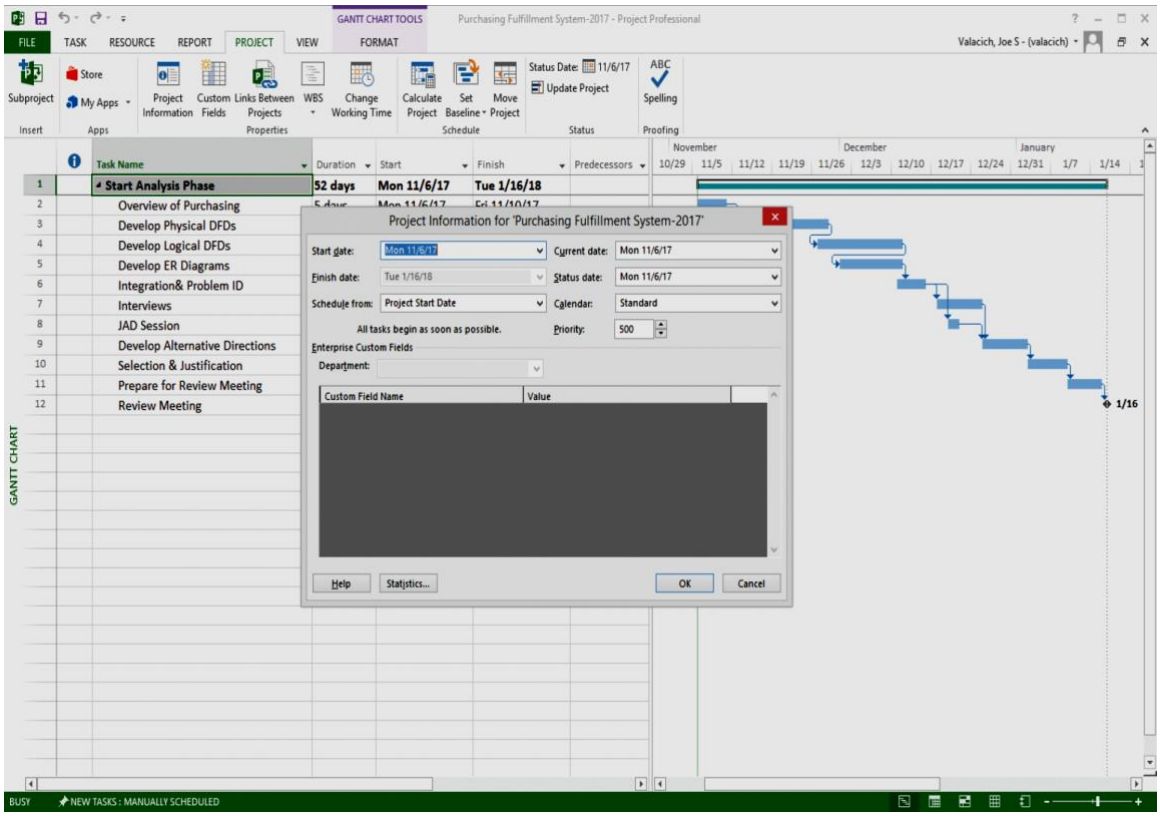
Many powerful software tools exist for assisting with project management.

Example: Microsoft Project can help with

- Entering project start or end date.
- Establishing tasks and task dependencies.
- Viewing project information as Gantt or Network diagrams.

- **Project Start Date**

  MS project allows you to specify all the task dependencies and durations, and then to change the start date if you want, which will result in a different end date for the project (as well as different dates for each of its tasks). You can also instead specify a hoped-for end date in order to determine the latest that a project can start,

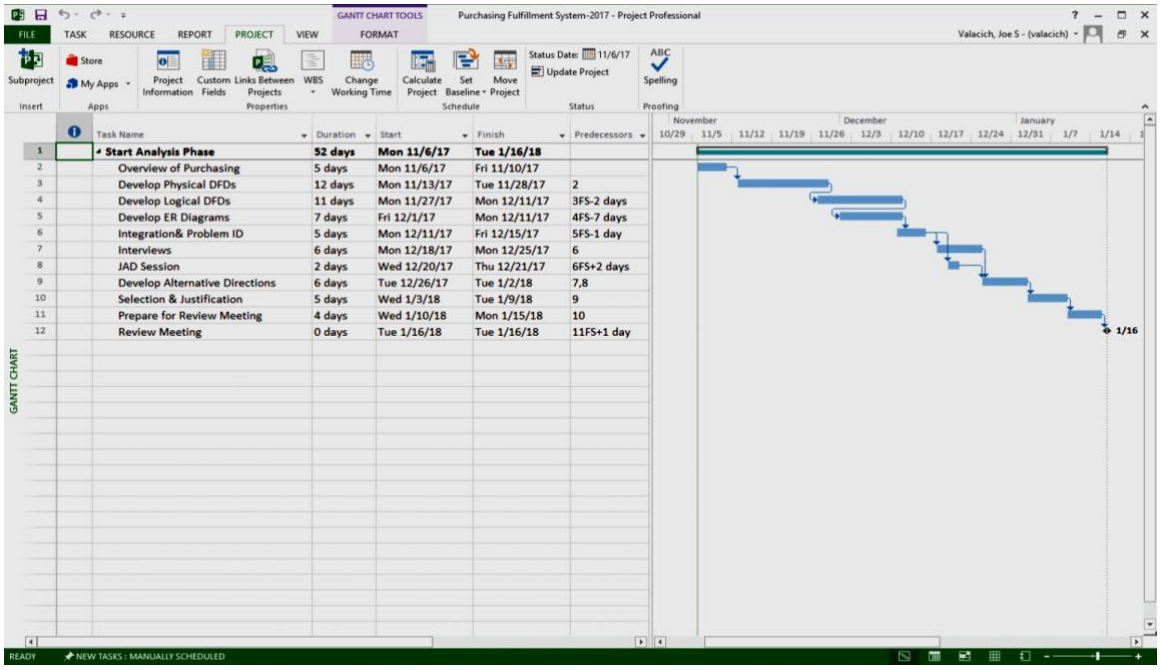


- **Entering Tasks**

  Entering tasks and assigning task relationships in Microsoft project

  In the work breakdown structure on the left, you can see many properties of each task in the project plan. Here, we see duration,

start and finish dates, and predecessors. There are many other properties you could see if you wanted, and there are a variety of WBS “themes”.



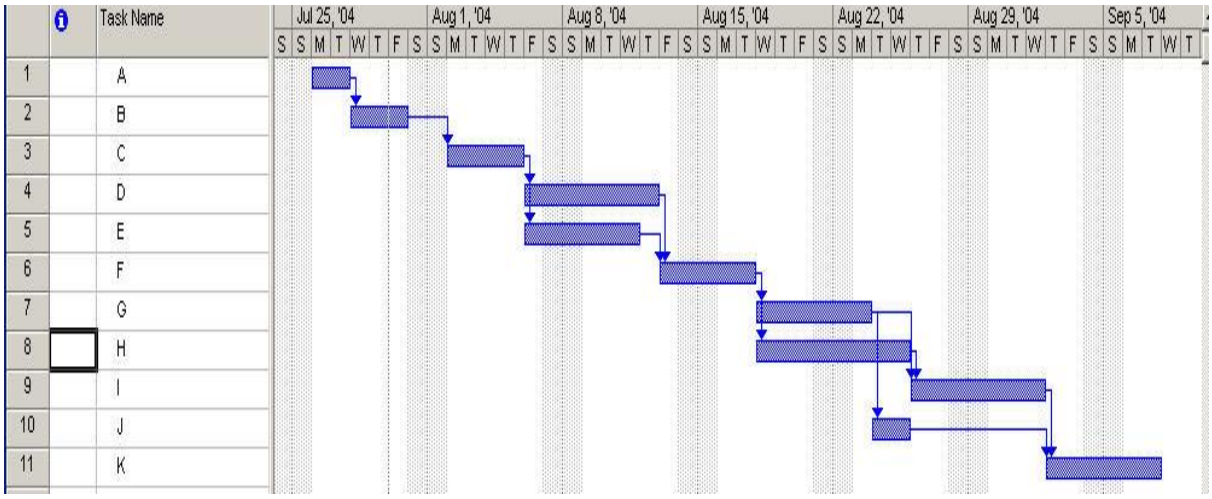
- **Solved Example**

  Assume you have a project with 11 activities labeled A-K (below). Derive the earliest finish time (EF), latest finish time (LF), and slack for each of activity (begin at time = 0). Which activities are on the critical path? Draw both Gantt chart and Network diagram for these activities Discuss the effect of changing the duration of activity E from 7 to 9 weeks, the duration of activity J from 5 to 7 weeks, and the duration of activity I from 8 to 10 weeks.

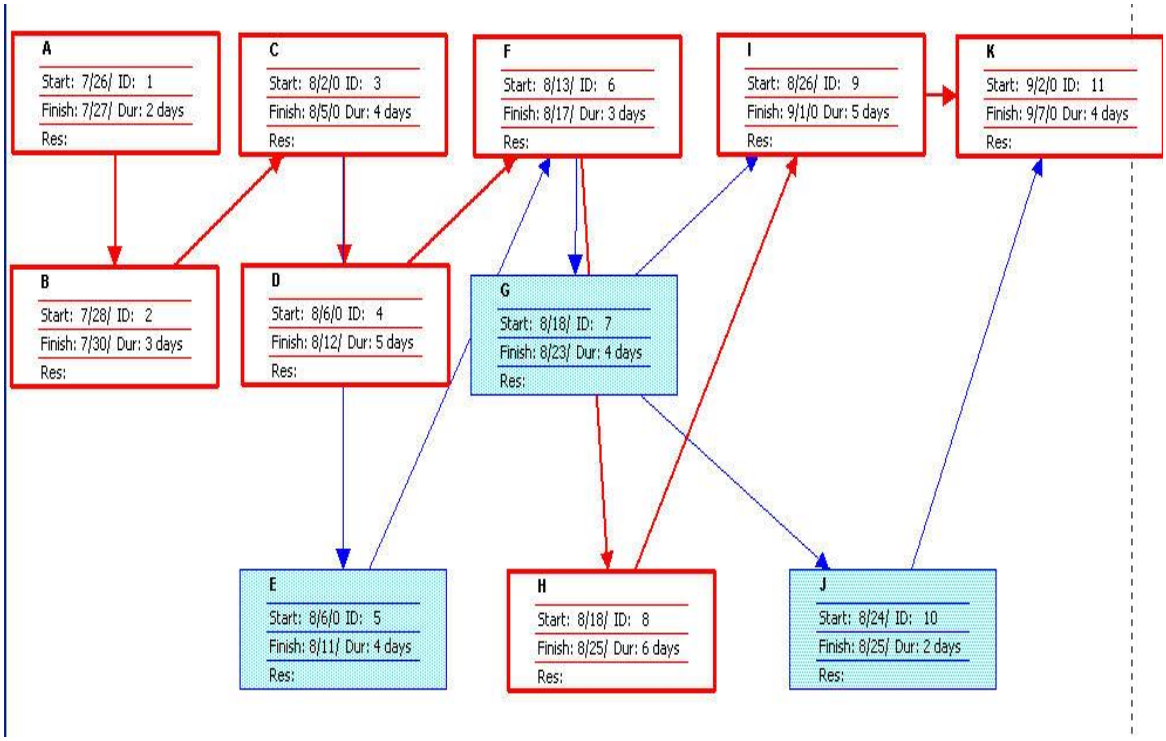
Activity	1	2	3	4	5	6	7	8	9	10	11
Preceding Event	-	1	2	3	3	4, 5	6	7	7,8	7	9,10
Duration	5	6	7	8	7	6	7	9	8	5	5

Activity	**Time**	**Immediate Predecessors**	**Early Finish**	**Late Finish**	**Slack**	**Critical Path?**
A	5	-	**5**	**5**	**0**	**Yes**
B	6	A	**11**	**11**	**0**	**Yes**
C	7	B	**18**	**18**	**0**	**Yes**
D	8	C	**26**	**26**	**0**	**Yes**
E	7	C	**25**	**26**	**1**	**No**
F	6	D, E	**32**	**32**	**0**	**Yes**
G	7	F	**39**	**39**	**0**	**Yes**
H	9	G	**48**	**48**	**0**	**Yes**
I	8	G, H	**56**	**56**	**0**	**Yes**
J	5	G	**44**	**56**	**12**	**No**
K	5	I, J	**61**	**61**	**0**	**Yes**

- **Gantt chart**


Task Name
Jul 25, '04
Aug 1, '04
Aug 8, '04
Aug 15, '04
Aug 22, '04
Aug 29, '04
Sep 5, '04
A
B
C
D
E
F
G
H
I
J
K

- **Networking diagram**



Changing Activity J from 5 week to 7 weeks does not mean much. Activity 10 is off the critical path, as is Activity K, its successor. There is enough slack time for Activities J and K so that this change does not affect the project's completion time. Changing Activity I from 8 week to 10 weeks affects the project's completion time. Activity I is on the critical path. Changing Activity E from 7 week to 9 weeks affects the project's completion time. Activity I is off the critical path but there is no enough slack time.

2.3 Graduation Project Example

Section 2.3.1 describes the project tasks and its description. Section 2.3.2 shows the project management. Section 2.3.3 describes the Software tool that's used to build up the planning which consists of Project Subsystem, Gantt chart, and the network diagram of the project.

2.3.1 Project Tasks

The project tasks and their description include:

- **Problem definition:** Identify a clear problem that is the focal point of our project that the project will address and improve. Where defining the problem is the first step towards a successful project.

- **Collecting data**: Survey is a collecting information about related system of our project and knowing how they work, their advantages and disadvantages and compared these systems. to configure feature idea of our project.

- **Specifying the required UML diagrams for the application:** System analysis include diagrams such as (use case diagram, sequence diagrams of use cases, and class diagram) to the analysis of the application.

- **Designing database classes:** Design the tables of database class and add the data of each table.

- **Designing analysis classes:** Design the analysis class definition including its name, attributes and behavior but without including any details yet on its software implementation.

- **Designing system menus and data forms:** Design all the menus that shows the data to user.

- **Designing transaction forms:** Design the forms of transaction between the user and the system.

- **Designing query forms:** Design the query form for maintaining the data (add, delete, update), of the user's request.

- **Designing report:** The written record of the project and generally is the only record that lives once the design team disbands at the end of the project. Appendices to the report contain supporting information with the details needed by a reader who wishes to fully understand the design.

- **System classes implementation:** The implementation part of a class definition captures object creation, data representation, and operation implementation.

- **System menu and data forms (Implementation):** Is a collection of menus used to present the functions or information about the application program or software system which contains a form contains numerous fields, or spaces to enter data. Each field holds a field label so that any user who views the form gets an idea of

its contents. A form is more user friendly than generating queries to create tables and insert data into fields.

- **Transaction forms implementation:** The implementation part of the transactions that happens in the system.

- **Query forms implementation:** The interface of a search engine. In the form, you place the search terms and choose the operators in order to formulate the query. Basic search –type form contains one box where you type the whole query. Operators and parentheses, if needed, must be typed by yourself.

- **Reports implementation:** The implementation part of the Reports that comes out of the system.

- **System testing:** Is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is a series of different tests whose sole purpose is to exercise the full computer-based system.

- **System documentation:** The collection of documents that describes the requirements, capabilities, limitations, design, operation, and

maintenance of a system, such as a communication, computing, or information processing system.

- **System demo:** A lost and found system demo is a freely distributed piece of an upcoming or recently released app. Demos are typically released by the system publisher to help consumers get a feel of the system before deciding whether to by the full version.

- **System presentation:** Representing the final work and everything that was done project by explaining what was learned and what was done by whom and how long it took to be done.

2.3.2 Project Management

The following table shows duration time and assigned students of each project tasks and its duration.

Task	Expected Time (weeks)	Assigned Students
1	4	All
2	1	All
3	5	All
4	1	Student 1, student 3
5	1	Student 2 , student 4
6	1	Student 5 , student6
7	1	Student 1, student 3
8	2	Student 2 , student 4
9	1	Student 5 , student6
10	2	Student 1, student 3
11	2	Student 2 , student 4
12	2	Student 5 , student6
13	1	Student 1, student 3

14	1	Student 2 , student 4
15	4	Student 5 , student6
16	4	All
17	2	All
18	5	All

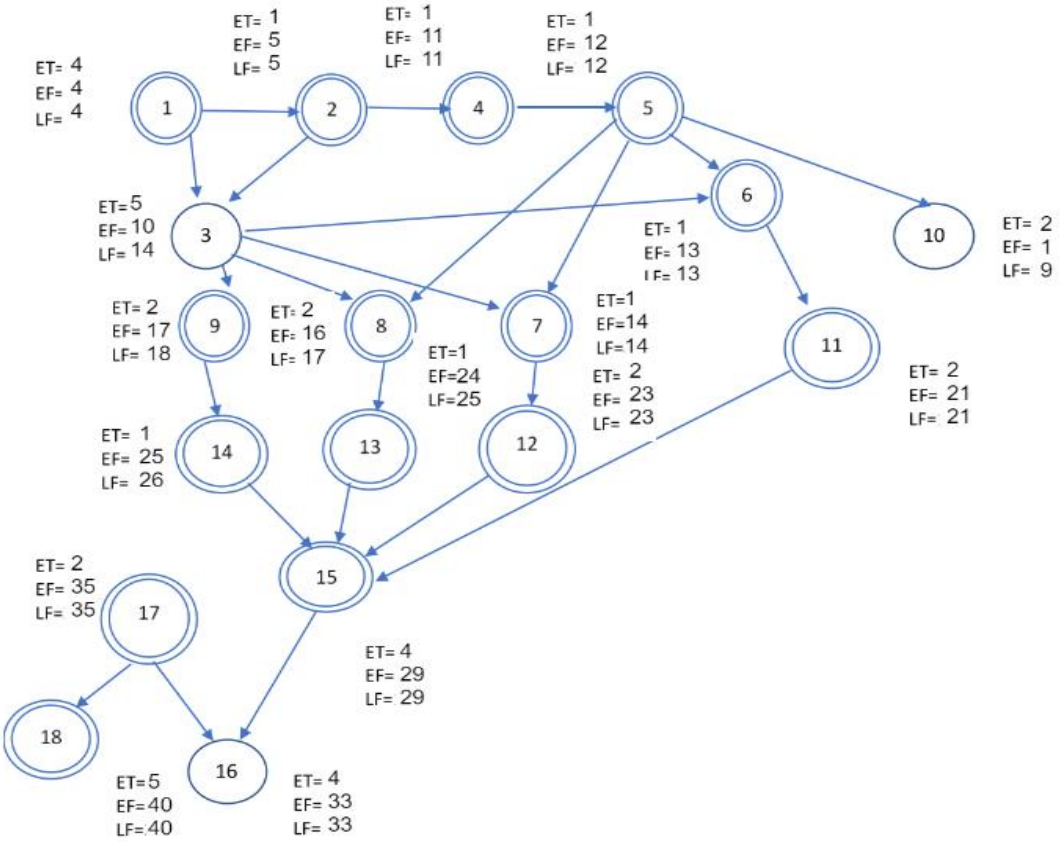
- **Activity slack time calculations for the projects**; all activities except number 3,8,9,10,13,14 are on the critical path.

The following table shows project activity slack management

Task	Preceding Event	Expected duration (week)	EF	LF	Slack |LF – EF|	Critical path
1	-	4	4	4	0	✓
2	1	1	5	5	0	✓
3	1,2	5	10	14	4	×
4	2	1	11	11	0	✓
5	4	1	12	12	0	✓
6	3,5	1	13	13	0	✓
7	3,5	1	14	14	0	✓
8	3,5	2	16	17	1	×
9	3,5	1	17	18	1	×
10	5	2	19	20	1	×
11	6	2	21	21	0	✓
12	7	2	23	23	0	✓
13	8	1	24	25	1	×
14	9	1	25	26	1	×
15	10,11,12,13,14	4	29	29	0	✓
16	15	4	33	33	0	✓
17	15	2	35	35	0	✓
18	17	5	40	40	0	✓

- **Gantt chart** showing project tasks, duration times for these tasks and predecessors: A network diagram for the project

showing estimated times for each activity and the earliest and latest expected completion time for each activity as shown in the following figure:



- **Software tool for project planning**

Task made by (Microsoft project planning)

Microsoft Project has the ability to

- Draw the action plan and represent it on both the Network Diagram and the Gantt chart.

- Allocate and organize resources for each activity.
- Follow-up progress of the project.
- Project budget management, and workload analysis.
- The possibility of programming work - as in all programs Microsoft Office package

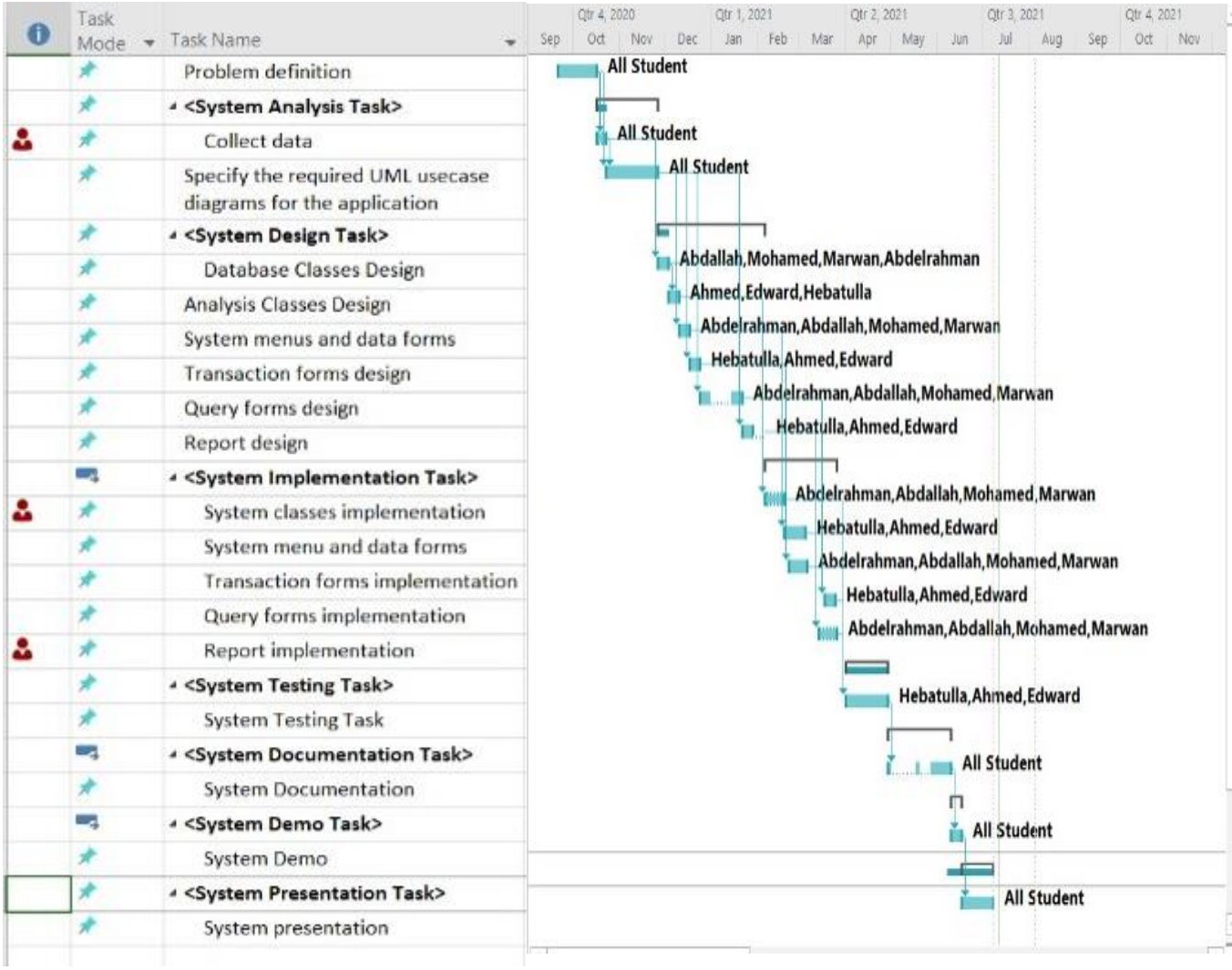
- **Project Subsystems**

  Group of interconnected and interactive parts that performs an important job or task as a component of a larger system. A subsystem, while a system in itself, is also wholly contained within a larger system. The system would be subsystems, such as ERD, transactions, and GUI.

Task Mode	Task Name	Duration	Start	Finish	Predecesso	Resource Names
	Problem definition	4 wks	Mon 9/21/2(	Fri 10/16/20		All Student
	<System Analysis Task>	**30 days**	**Sat 10/17/2(**	**Thu 11/26/20**		
	Collect data	1 wk	Sat 10/17/20	Thu 10/22/20	1	All Student
	Specify the required UML usecase diagrams for the application	5 wks	Fri 10/23/20	Thu 11/26/20	3,1	All Student
	<System Design Task>	**51 days**	**Fri 11/27/20**	**Fri 2/5/21**		
	Database Classes Design	1 wk	Fri 11/27/20	Thu 12/3/20	3	Abdallah,Mohamed,Marwan,
	Analysis Classes Design	1 wk	Fri 12/4/20	Thu 12/10/20	6	Ahmed,Edward,Hebatulla
	System menus and data forms	1 wk	Fri 12/11/20	Thu 12/17/20	6,4	Abdelrahman,Abdallah,Moha
	Transaction forms design	1 wk	Fri 12/18/20	Thu 12/24/20	6,4	Hebatulla,Ahmed,Edward
	Query forms design	2 wks	Fri 12/25/20	Thu 1/21/21	6,4	Abdelrahman,Abdallah,Moha
	Report design	1 wk	Fri 1/22/21	Fri 2/5/21	6,4	Hebatulla,Ahmed,Edward
	<System Implementation Task>	**34 days**	**Sat 2/6/21**	**Thu 3/25/21**		
	System classes implementation	2 wks	Sat 2/6/21	Thu 2/18/21	7	Abdelrahman,Abdallah,Moha
	System menu and data forms	2 wks	Fri 2/19/21	Thu 3/4/21	8	Hebatulla,Ahmed,Edward
	Transaction forms implementation	2 wks	Sun 2/21/21	Fri 3/5/21	9	Abdelrahman,Abdallah,Moha
	Query forms implementation	1 wk	Thu 3/18/21	Wed 3/24/21	10	Hebatulla,Ahmed,Edward
	Report implementation	2 wks	Sun 3/14/21	Thu 3/25/21	11	Abdelrahman,Abdallah,Moha
	<System Testing Task>	**20 days?**	**Thu 4/1/21**	**Wed 4/28/21**		
	System Testing Task	4 wks	Thu 4/1/21	Wed 4/28/21	13,14,15,16,1	Hebatulla,Ahmed,Edward
	<System Documentation Task>	**30 days**	**Thu 4/29/21**	**Wed 6/9/21**		
	System Documentation	3 wks	Thu 4/29/21	Wed 6/9/21	19	All Student
	<System Demo Task>	**5 days**	**Thu 6/10/21**	**Wed 6/16/21**		
	System Demo	1 wk	Thu 6/10/21	Wed 6/16/21	21	All Student
	<System Presentation Task>	**15 days**	**Thu 6/17/21**	**Wed 7/7/21**		
	System presentation	3 wks	Thu 6/17/21	Wed 7/7/21	23	All Student

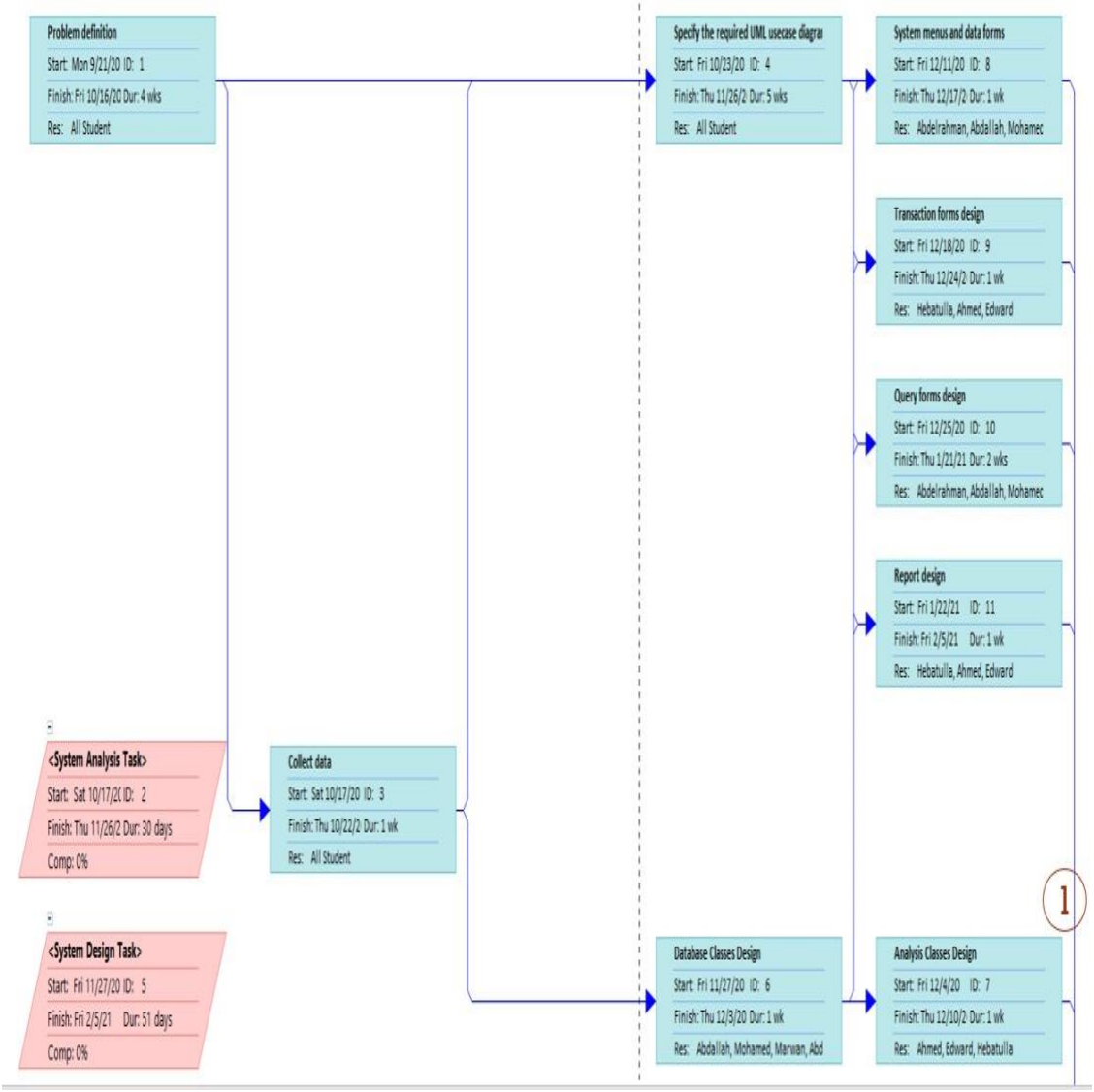
- **Gantt Chart**

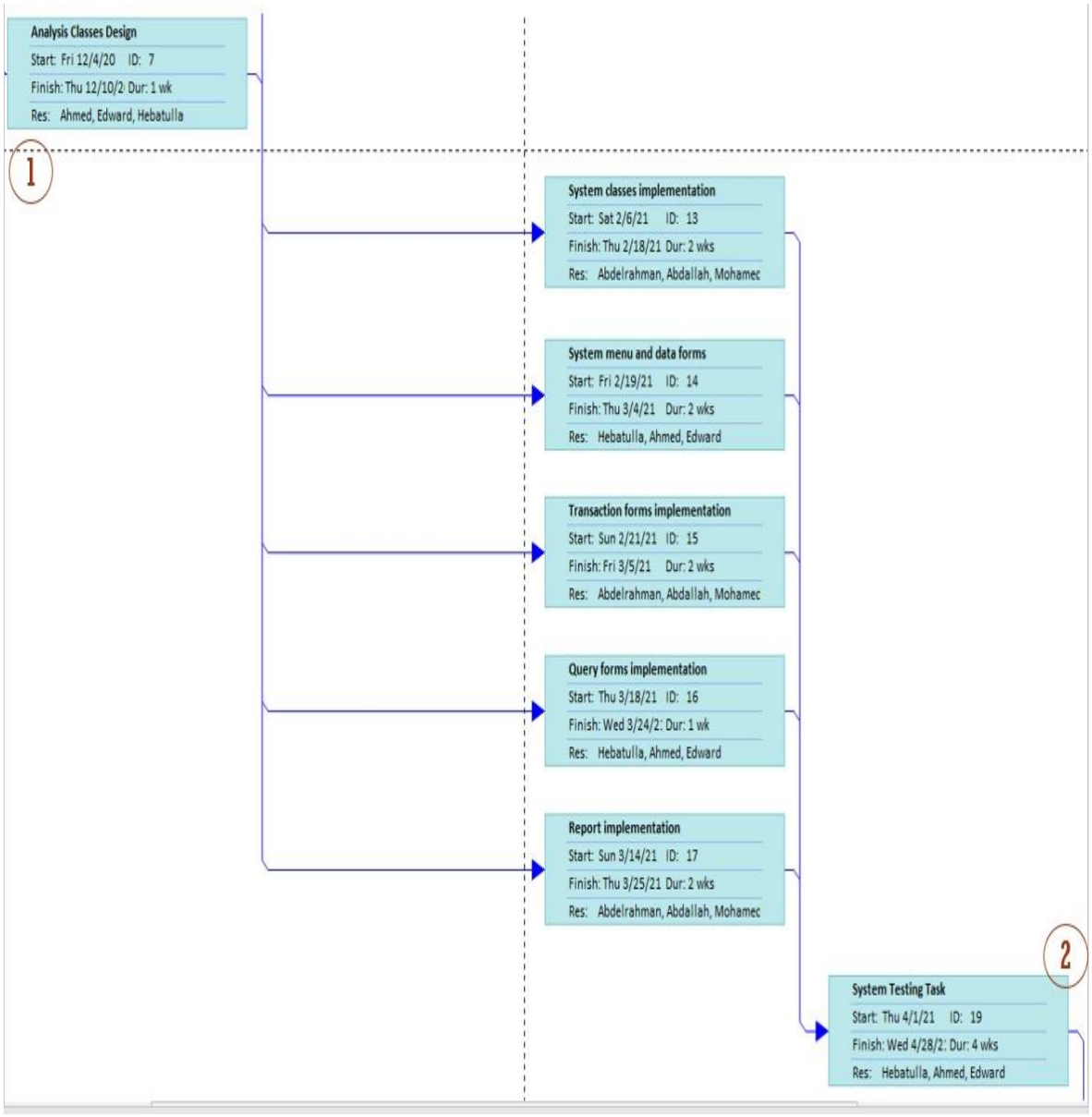
A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity, the following figure shows that the Gantt chart for system tasks:

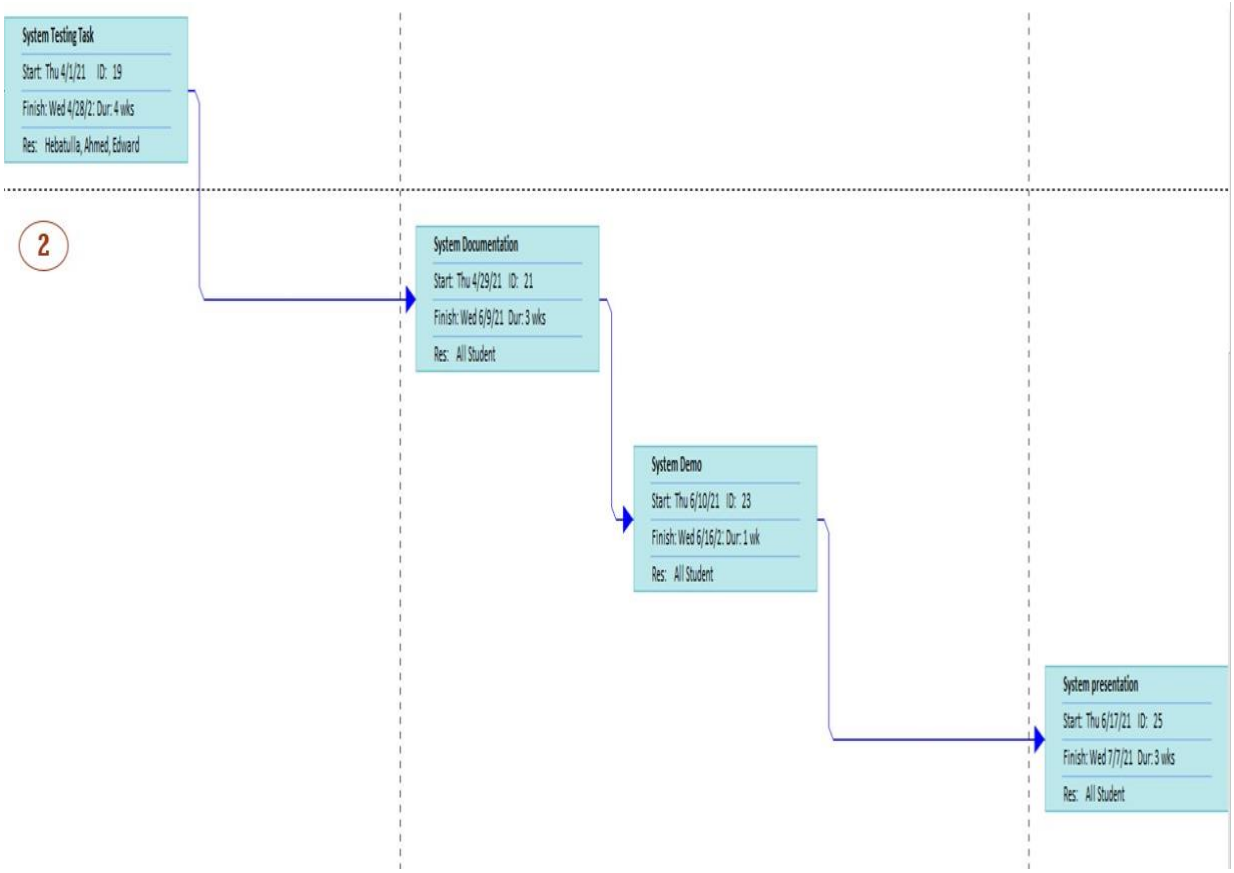


- **Network Diagrams**

A Network Diagram is a visual representation of a project's schedule. Well known complements to network diagrams. A network diagram in project management is useful for planning and tracking the project from beginning to finish. It represents a project's critical path as well as the scope for the project, The following figure shows that the Network Diagram.




Analysis Classes Design
Start: Fri 12/4/20 ID: 7
Finish: Thu 12/10/2 Dur: 1 wk
Res: Ahmed, Edward, Hebatulla
1
System classes implementation
Start: Sat 2/6/21 ID: 13
Finish: Thu 2/18/21 Dur: 2 wks
Res: Abdelrahman, Abdallah, Mohamec
System menu and data forms
Start: Fri 2/19/21 ID: 14
Finish: Thu 3/4/21 Dur: 2 wks
Res: Hebatulla, Ahmed, Edward
Transaction forms implementation
Start: Sun 2/21/21 ID: 15
Finish: Fri 3/5/21 Dur: 2 wks
Res: Abdelrahman, Abdallah, Mohamec
Query forms implementation
Start: Thu 3/18/21 ID: 16
Finish: Wed 3/24/2: Dur: 1 wk
Res: Hebatulla, Ahmed, Edward
Report implementation
Start: Sun 3/14/21 ID: 17
Finish: Thu 3/25/21 Dur: 2 wks
Res: Abdelrahman, Abdallah, Mohamec
2
System Testing Task
Start: Thu 4/1/21 ID: 19
Finish: Wed 4/28/2: Dur: 4 wks
Res: Hebatulla, Ahmed, Edward



For any successful project, we must make a planning to define the goal we want to reach, the beginning and end of the project, and the tasks we will do throughout the year to avoid missed deadline the delivery date and delivering the project on time. Without careful planning, project performance is almost certainly guaranteed to suffer. Planning enhances team members' confidence and prevents any worry about them being late to the project. project planning divided into the tasks for us in this chapter and determined the expected time for each task and identified the critical tasks (1,2,4,6,11,12,15,16,17,18) that must end in their duration and we used Microsoft project planning to generate gantt chart and network diagram for show the flow of the project.

CAPTER 3

MTI PROJECT RESOURCES

By: Prof. Dr. Mohamed Anwar Assal

3.1 MTI Laboratory Resources

Faculty of Computers and Artificial Intelligence hosts 8 Laboratories as follows:

- 6 x General Purpose Computer Laboratories
- 1 x CISCO Networking Laboratory
- 1 x Microsystem Laboratory

3.1.1 General Purpose Computer Laboratories Specifications

29 PCs with the following configurations:

Item	Description
Processor	Intel Core i7 3.4 GHz
RAM	4 GB
Hard Drive	500 GB
Mother Board	Gigabyte H61M-S2-B3
Video Adapter	Intel HD Graphic 4000
Monitor	Samsung LED 21”
Printer	2 x LaserJet Printers

Software:

- Microsoft Windows 8.1
- Microsoft Office
- Microsoft Visual Studio
- Microsoft SQL Server
- LARP
- Adobe Photoshop
- Typing Master
- Packet Tracer

- MATLAB
- Net Support School
- R Studio

3.1.2 CISCO Networking Laboratory

- Cisco CCNA Certified Laboratory
- 17 PCs with the following configurations

Item	Description
Processor	Intel Core i7 3.4 GHz
RAM	4 GB
Hard Drive	500 GB
Mother Board	Gigabyte H61M-S2-B3
Video Adapter	Intel HD Graphic 4000
Monitor	Samsung LED 21”
Printers	2 x Laserjet

Rack 2M With the following configurations:

Item	Description
Patch Panel	ICS 24 Port Cat 5e
Switch	Cisco 24 Ports 10/100
3 Switches	Catalyst 2960 Series
3 Routers	Cisco 2800 series

LCD Monitor Samsung 46”

3.1.3 Microsystems Laboratory

The microcontroller laboratory contains 8 setups, each consisting of a PC that is used to develop the microcontroller code using a suitable software/application. The PC is connected to a PIC microcontroller programmer board through a communication link (e.g., RS323 or USB). The microcontroller is inserted into the socket on the board and can be programmed through the computer.

Procedure

Developing a PIC microcontroller-based project simply takes no more than five or six steps, as follows:

1. Type the program into a PC

2. Assemble (or compile) the program
3. Optionally simulate the program on a PC
4. Load the program into PIC's program memory
5. Design and construct the hardware
6. Test the project.

3.2 MTI IT Center

The explosion of technology in the workplace, combined with a competitive global marketplace, created a demand for employees with high-tech skills, critical thinking, an ability to learn and adapt, and the ability to cope with the latest technology applications. As a result, MTI University had established on-campus information technology center, IT Center, which houses the state-of-the-art computing facilities and a technology-enabled active learning environment.

The IT Center has a particular interest in providing students with advanced vocational education programs that combine academic courses within practical contexts addressing the specialist technical knowledge

and skills that employers need at the associate professional and higher technician level. The IT Center offers a wide range of courses including ICDL, graphic design, web design and development, 3D modeling and networking.

The IT Center has reached academic cooperation agreements with Microsoft, CISCO and ICDL Egypt to promote education quality and innovations. The academic cooperation will help students to get up-to-date knowledge and skills from the first years of their education. The IT Center is currently an authorized Microsoft IT Academy and CISCO Networking Academy, consequently the center is authorized to hold CISCO and Microsoft certification exams. As a part of the agreements, MTI students and academic staff have access to the most updated technologies and official course materials for work, research and education purposes.

3.2.1 Microsoft Courses

Microsoft Certified Professional Developer (MCPD)	Course Duration
The MCPD is an umbrella certification that contains 2 sub-certifications. This is to allow the owner to demonstrate that they are an expert in one of the following categories:	
1. MCPD: Windows Developer	**120 Hours**
Build rich client applications for the Windows Forms platform by using the Microsoft .NET Framework 4 and Visual Studio 2010.	
I. TS: Developing Windows Applications	**40 Hours**
II. TS: Developing Windows Communication Foundation Solutions	**24 Hours**
III. TS: Developing Data Access Solutions	**40 Hours**
IV. PRO: Designing and Developing Windows Applications	**16 Hours**

2. MCPD: Web Developer		120 Hours
Build interactive, data-driven ASP.NET applications by using ASP.NET 4 for both intranet and Internet uses.		
I.	TS: Developing Web Applications	40 Hours
II.	TS: Developing Windows Communication Foundation Solutions	24 Hours
III.	TS: Developing Data Access Solutions	40 Hours
IV.	PRO: Designing and Developing Web Applications	16 Hours

Microsoft Certified Solution Developer (MCSD)		Course Duration
The MCAD is a certification that demonstrates that the owner is an advanced developer who, using Microsoft development tools, designs/develops enterprise solutions.		
Earning an MCSD: Windows Store Apps certification will qualify you for such jobs as software developer, web developer, and quality engineer.		
1. MCSD: Windows Store Apps Using C#		120 Hours
Demonstrate your expertise at designing and developing fast and fluid Windows 8 apps. There are two paths to achieving this certification using C#.		
I.	Programming in C#	40 Hours
II.	Essentials of Developing Windows Store Apps Using C#	40 Hours
III.	Advanced Windows Store App Development Using C#	40 Hours
2. MCSD: Web Applications		200 Hours
Get recognized for your expertise in creating and deploying modern web applications and services.		
I.	Programming in HTML5 with JavaScript and CSS3	40 Hours
II.	Developing ASP.NET MVC 4 Web Applications	80 Hours
III.	Developing Windows Azure and Web Services	80 Hours

Microsoft Certified IT Professional (MCITP)		Course Duration
These certifications prove that you have the comprehensive set of skills to perform a particular IT job role, such as database administrator or enterprise messaging administrator.		
Server Administrator on Windows Server		120 Hours
Demonstrate your expertise at designing and developing fast and fluid Windows 8 apps. There are two paths to achieving this certification using C#.		
I.	Windows Server Active Directory, Configuring	64 Hours
II.	Windows Server Network Infrastructure, Configuring	40 Hours
III.	Windows Server, Server Administrator	40 Hours

Microsoft Certified Solutions Associate (MCSA)		Course Duration
These certifications prove that you have the comprehensive set of skills to perform a particular IT job role, such as database administrator or enterprise messaging administrator.		
1. MCSA: Windows Server		**144 Hours**
Validate your skills and differentiate yourself as better able to work with Windows Server 2008 in a real-world business context.		
I.	**Windows Server Active Directory, Configuring**	**64 Hours**
II.	**Windows Server Network Infrastructure, Configuring**	**40 Hours**
III.	**Windows Server, Server Administrator**	**40 Hours**
2. MCSA: Windows Server		**240 Hours**
Get recognized for your expertise in creating and deploying modern web applications and services.		
I.	**Installing and Configuring Windows Server**	**80 Hours**
II.	**Administering Windows Server**	**80 Hours**
III.	**Configuring Advanced Windows Server Services**	**80 Hours**

Microsoft Project		
Course levels	**Prerequisites**	**Course Duration**
2		
Course Contents		
Level No. 1: core	Basic & Intermediate	**15 hrs.**
Level No. 2: Advanced	Advanced	**30 hrs.**

Microsoft Word		
Course levels	**Prerequisites**	**Course Duration**
2		
Course Contents		
Level No. 1: core	Basic & Intermediate	**10 hrs.**
Level No. 2: Advanced	Advanced	**20 hrs.**

Microsoft PowerPoint		
Course levels	**Prerequisites**	**Course Duration**
2	-	
Course Contents		

Level No. 1: core	Basic & Intermediate	**10 hrs.**
Level No. 2: Advanced	Advanced	**20 hrs.**

Microsoft Excel		
Course levels	**Prerequisites**	**Course Duration**
2	-	
Course Contents		
Level No. 1: core	Basic & Intermediate	**15 hrs.**
Level No. 2: Advanced	Advanced	**30 hrs.**

Microsoft Office Specialist (MOS) Master	**Course Duration**
	120 Hours
Demonstrate that you have the deepest level of skills needed to proficiently use Office programs by earning a Microsoft Office Specialist Master certification.	
I. Microsoft Office Word Expert	**30 Hours**
II. Microsoft Office Excel Expert	**30 Hours**
III. Microsoft Office PowerPoint Expert	**30 Hours**
IV. Microsoft Office Access Expert	**30 Hours**

3.2.2 CISCO Courses

Cisco Certified Network Associate (CCNA)	**Course Duration**
	125 Hours
The Cisco CCNA network associate certification validates the ability to install, configure, operate, and troubleshoot medium-size routed and switched networks, including implementation and verification of connections to remote sites in a WAN. International exams available for CCNA Certification	
Level 1: Network Fundamentals	**50 Hours**
Level 2: Routing Protocol and Concepts	**25 Hours**
Level 3: LAN Switching and Wireless	**25 Hours**
Level 4: Accessing the WAN	**25 Hours**

3.2.3 Photoshop

Photoshop	Course Duration
	30 Hours
Adobe Photoshop has long helped graphic designers bring their visions to life in advertisements, magazine layouts, billboards, and virtually every other medium. Whether you create images professionally or just want professional results, you can count on Adobe Photoshop for the ultimate in creative control. At the end of this course, the learner will be able to create and edit images, retouch and correct adjustments.	
Work Environment	2 Hours
Working With Selections	3 Hours
Transforming	8 Hours
Understanding Layers	8 Hours
Retouching & Repairing	9 Hours

3.2.4 Graphics Course

Graphic Course		
Course levels	**Prerequisites**	**Course Duration**
3	Basic computer and mouse skills	
Course Contents		
Level No. 1	**Adobe Photoshop Mastering the Essentials**	**18 Hours**
Level No. 2	Adobe Illustrator Mastering the Essentials	**18 Hours**
Level No.3	Adobe After effects After effects fundamentals	**18 Hours**
Level No.4	Adobe Premier	**24 Hours**
Course Target	**Graphic designer**	

3.2.5 Web Design

Web Design	Course Duration
	50 Hours
This learning path presents the learner with the advanced concepts and skills necessary for website design. You are brought step-by-step through the process of building and preparing to publish a website.	
HTML	20 Hours
CSS	10 Hours
JavaScript	20 Hours

3.2.6 3D Modeling using Maya Course

3D Modeling using Maya Course		
Course levels	**Prerequisites**	**Course Duration**
2	Basic computer and mouse skills	
Course Contents		
Option 1: Training only		
Level No. 1: Foundation	Modeling, Texture, Curves, Camera	**25 Hours**
Level No. 2: Advanced	Lights, Animation, Paint Effect & Rendering	**25 Hours**
Option 2		
Level No. 1: Foundation	Modeling, Texture, Curves, Camera	**25 Hours**
Level No. 2: Advanced	Lights, Animation, Paint Effect & Rendering	**25 Hours**
Workshop	Learn how to print a 3D model using 3D printer	**6 Hours**
Course Target	**3D modeling professional**	

3.2.7 Introduction to embedded systems with Arduino

Introduction to embedded systems with Arduino		
Course levels	**Prerequisites**	**Course Duration**
1	Basic knowledge of C/C++	
Course Contents		
Core	**Option 1:** Training + Components provided by the student	**20 Hours**
	Option 2: Training + Components provided by the center	
Course Target	1. **Get Familiar with Arduino environment** 2. **Programming Arduino with simplified C++** 3. **Building Projects using Arduino**	

3.2.8 Programming & Interfacing with PIC Microcontrollers

Programming& Interfacing with PIC Microcontrollers		
Course levels	**Prerequisites**	**Course Duration**
2	Electronics Basic	
Course Contents		
Core		**25 Hours**
Advanced		**20 Hours**
Course Target	**1. Get familiar with PIC Microcontrollers** **2. Program PIC Microcontroller with C** **3. Interfacing between PC & PIC** *** Student will provide components required for the Training**	

CHAPTER 4

SELECTING THE APPROPRIATE SOFTWARE TOOLS

By: Prof. Dr. Hesham El-Deeb

4.1 Introduction

It is important to appreciate where you are starting from and to understand the gap that you intend to close with the new project management software. The following pointers will help.

- Define the current situation and anticipated future in terms of numbers and types of projects and resources that you will be managing.

- Re-assess your current tools and identify their shortcomings. Describe in detail the gap that the new project management software will need to close, in terms of functionality, ease of use, and environmental requirements, etc.

- Do an honest assessment of your current project management maturity level, and get agreement on the improvements that you are aiming for. The new project management software can facilitate these improvements, by effectively supporting the proposed changes to the business process.

- Appreciate that good project managers are essentially anarchists. After all, you hire them to get things done and overcome the hurdles within your organization. This does not make them meek and mild when it comes to adhering to

detailed business process steps. They need to be involved and take ownership of the process improvements that you have in mind.

Before you can start the project management software selection process, you will need two detailed documents:

- **The system requirements document**

  This should be a comprehensive list and description of the key requirements. Classify these by:

  - Software function or capability.
  - Ease of Use - give specific requirements, e.g. just one click from saving project data to opening a view of the plan.
  - Environmental. Most software will not operate in every environment, e.g. on both Macs and PCs, Unix or Microsoft, etc.
  - Any specific pricing models or contract terms that you require.

- **Product evaluation criteria**

  So that you can consistently score each product and objectively assess the best option.

The requirements doc should be sent to likely vendors, with an invitation to tender. This should request a detailed, point by point response to each requirement listed, with pricing and implementation proposals. Be prepared to meet with vendors at this

stage, as the more information they have, the more relevant their proposal will be.

Applying the evaluation criteria should lead to a short list of vendors to be invited in for a lengthy demonstration and Q&A session, with a number of your colleagues.

As the list of likely contenders narrows, you can ask the candidates for further information, in terms of:

- Arranging a client reference visit or phone call. The vendor should be able to find a relevant client and have enough confidence not to attend your visit himself.

- A Proof of Concept Workshop.

- Ask the likely candidates to configure their products to represent aspects of your process steps, using data that you supply. This will enable you to see how readily their out of the box offering can match your requirements. Good vendors will offer a further evaluation period, provided that you accept some product training and hand holding.

As a result of this process a clear preference should emerge, for final negotiation and agreement.

4.2 Android Development Tools List

Starting out in Android development can be a daunting task – there's so much information out there, so many tutorials and so

many resources it can be hard to navigate. That's why we've compiled this big list of Android development resources and tools so you can find all you might need in one place.

From IDEs to learning tutorials, consoles to libraries, we've covered the very first steps a budding developer can take on this path, to useful libraries and plug-ins for the more experienced users.

The sections in this guide are:

- Editors and IDEs
- Language Resources
- Libraries
- Plugins

Each section has a brief description of what it is before delving into the best tools within that category. If you decide that you don't want to stick to *just* Android, but would like to go cross-platform, you can check out our list of cross-platform development resources, and if you're thinking of making a game then you should probably read our guide on game specific resources.

Let's jump in. We start with Android Editors and IDEs. What are IDEs you may ask?

Android Editors and IDEs

IDE stands for "integrated development environment" – effectively a piece of software that allows you to create other software, in our case a mobile application. According to Techopedia: An Integrated Development Environment (IDE) is an application that facilitates application development.

IDEs can contain a compiler, which translates the language you are coding in to Android's Java, and a debugging console, all usable through the same graphical interface.

This is where you'll be doing most of the work; creating, designing, testing and polishing your mobile app.

As well as "the originals", that is to say Eclipse, IntelliJIDEA and of course Android Studio, we've chosen a couple of other platforms here to include C, C++ and PhoneGap, though generally, it is recommended that you stick to Java and HTML5.

Android Studio – The official IDE, based on the community-created IntelliJIDEA (see below).

Eclipse – Before Android Studio, this was the official Android development environment. Used to code Java but can be expanded to other languages via plugins, it is still a powerful tool.

IntelliJIDEA – Android Studio is based on this, and this IDE is not only extremely useful, but has a massive amount of community-created plugins, making it highly customizable.

DroidEdit – An Android text and code editor to use on Android platforms.

Android-IDE – A complete web and Android development environment, it also allows you to edit Java and PhoneGap apps.

Cordova – Mobile apps with HTML, CSS and JS, its one of the best tools if you want to create hybrid apps. Free and open source.

Corona – A 2D-development platform with a specific focus on games but can be used to create other types of mobile apps too. One of the best for cross-platform development and 100% free. Build Games for Mobile, TV, and Desktop using Corona SDK

Titanium – One of the lesser-known platforms, it allows for the creation of native apps for iOS, Android and Windowsphone and runs off a single JavaScript codebase.

Xamarin – Widely featured in the press and a very impressive IDE for native Android, iOS and Windows applications. Open source and free with two further price plans, it uses C# as its language

CppDroid – Allows you to code, edit compile and execute C and C++ code. Packed full of features including practice programs and syntax highlighting.

Experimenting with an IDE can be a good way to learn mobile app development, but if you want a more structured approach, we've got some tutorials for you.

Android Language Resources

There're a number of languages you could use to develop Android applications, but the king of them all is Java, which is the staple for app development.

We've included tutorials for this below, as well other languages such as LUA and Cordova (for hybrid apps), C and C++. A word of warning for C and C++: you can use them with the Android Native Development Kit to create Android apps, but while this

always means an increased app complexity, it doesn't always have the payoff of better performance.

Java – Straight to the source, if you're developing in Android, Java is probably the language you want to be using. Has it's own development kit, but there are plenty of other SDKs out there too.

Codecademy – One of the premier code-learning resources online, it has been used by thousands of people to get into Java coding, as well as other languages and frameworks. An interactive, learn-as-you-code format.

Team treehouse – Another e-learning website, but well known for the strength of its Java courses.

Udemy – Online learning can't go without mentioning Udemy, which features dozens of both highly specific and generic Java learning courses.

New Boston – YouTube tutorials to learn how to develop in Android – currently has over 5 million views. Covers everything from setting up the SDK to XLM Layouts. 200 videos in total.

Android Application Development Tutorials 1

Ryan Park Apps resource list – Ryan Parks taught himself how to code in Java and published, among others, a personal finance application. This is the list of resources he used.

Oracle Java Tutorials – Both general and specialized Java tutorials by IT giants Oracle, starts from the very basic concepts and overview.

Cave of Programming – Covers both Java and C++, comes with exercises and tests: also sometimes offers paid-for courses for free, pending approval by the creator of the site, John.

Mkyong – Very specific tutorials and guides on Java development. More for the skilled coder who knows what they're looking for.
Programming by Doing – Points-based tutorials refined over 15 years which can also be bought as a book. Final exercise is creating your own Paint clone, and it has its own support subedits.
Java Design Patterns – Github repository of Java implemented design patterns.

C++ and C

LearnCPP – Free and extensive website to learn coding C++. This is really the long way around to Android development, but C++ does give you further options beyond mobile apps.

SoloLearn – Offers courses on all sorts of languages beyond C++ too – over 80 lessons going from the basics to classes and objects, structured much like a game.

LUA with Corona

Corona Learning Centre – The Corona SDK (the builder is further down in this list) creates Android apps using the LUA language, and is one of the most used languages after Java for the platform. Highly recommended.

Tyler Neylon – "Learn LUA in 15 Minutes" is perhaps a bit ambitious, especially for the novice developer, but it's about one of the clearest explanations you can find for LUA online.

Programming in LUA – The most extensive resource for learning LUA. Written in 2003 but still very relevant, and available as a book too.

PhoneGap/Cordova/Others

TutorialsPoint on PhoneGap – Not really a language, but basically Cordova plus additions by Adobe to develop using HTML, CSS and JavaScript. You can get PhoneGap here.

Coenraets on Cordova – Very comprehensive tutorial on building mobile apps with the free and open-source engine managed by Apache. You can download Cordova here and find PhoneGap tutorials on the Coenraets website too.

Kotlin – Fairly new programming language made for JVM and Android, claims to be fully interoperable with Java. Comes with its own tutorials and knowledge base and was made with Gradle in mind.

Getting Started with Kotlin in 2 minutes

This next section is for those who are perhaps a bit more experience, but there's nothing stopping you from reading through it to find out what libraries are. You'll need them sooner or later!

Android Libraries

A library, in the software development world, is a collection of data which can do any number of things: set rules for app behavior,

graphic effects, pre-written code, templates, text, communication protocols and much more.

A lot of these libraries are organized by their topic or activity – you can find libraries for translating objects from one language to another, or for managing the way certain visuals move and behave on screen. The dev community tends to be quite open with a lot of their work, and chances are someone has made the library you need, but you can always pick and choose bits of code that are useful to you and put them into your own library.

Universal Image Loader – Does what it says on the tin with out-of-the-box loading and caching of images. Highly customizable.
GSon – Serializing and desterilizing Java objects in JSON.

Retrofit – Described as an “elegant solution for organizing API calls”.

Awesome Java – A list of some of the best Java frameworks and libraries.

AndroidView Animations – Library with very simple syntax to get regular View animations working smoothly.

EventBus – Aimed at making communication between parts of your application as smooth and easy as possible.

ButterKnife – Very lightweight library which streamlines various wordy Android syntax issues by using annotations to create boilerplate template code.

Picasso – Specifically useful when download images for apps. Just inputting the image's URL will download the image, store as bitmap and cache it.

Awesome Android Libraries – A curated list of Android development libraries, organized by functionality, name and license.

Android Arsenal Libraries – A good resource all round, but in this case a collection of quality (and free) Android dev libraries.

Now, if you want to customize your coding environment, look no further than plug-ins.

Android Plug-ins

Snippets of code and "add-ons" for improving quality-of-life while coding; from efficiency, syntax and other functionalities, plugins can be your best friend when customizing your developing environment.

Once again the dev community tends to share plugins, and there are an absolutely huge amount out there waiting for you to discover them.

Plugin collection for IntelliJ – The main repository for IntelliJ plugins, an absolute treasure-trove of handy tools for the IntelliJ IDE.

A curated list of IntelliJ Plugins

The above repository is absolutely huge, so to help you get started and find some gems, here's a curated list of the best IntelliJ plugins.

Import Drawables – For IntelliJ, allows importing of drawables at different resolutions and other image-based functionalities.

GenyMotion – One of the biggest and most reliable testing and emulation tools for Android apps – employed by BlaBla Car among other high-profile names.

Genymotion Android Emulator

Boilerplate Code Generation – For IntelliJ, generates parcelable boilerplate code.

Android Holo Colors – Generates all necessary XML to have editext and color spinners in your Android app.

Butterknife Injections – Works with the ButterKnife library to generate boilerplate code injections.

H.A.X.M. – Stands for Hardware Accelerated Execution Manager, developed by Intel, it executes Android apps faster for those who use SDK emulators.

Robotium Recorder – test framework for both native and hybrid applications, both on emulators and Android devices.

ADB-IDEA – A quality-of-life and productivity plug-in to increase the speed at which you develop Android apps. Compatible with both Android Studio and IntelliJ IDEA.

IDEA Vim – Vim emulation plug INS made by Jetbrains and based on the IntelliJ platform.

Folding Plugin – Highly recommended file grouping plugin

CodeGlance – Creates a “code minimap” within your IDE for easier navigation, integrates with your pre-existing syntax highlighting.

Android Material Design Icon – Now built-in to Android Studio 1.4, one of the easiest ways to create icons at just the right size and resolution.

KeyPromoter – Can’t get the hang of keyboard shortcuts? This plug in will constantly remind you how much faster you could be doing things.

4.3 Software Platforms for Website Applications

Best web development tools in 2021: software platforms for website developers

- Sketch
- InVision Cloud
- Sublime Text
- Foundation
- Chrome DevTools
- Visual Studio Code

4.4 Selected Software Tools Case Study

The big question that is frequently asked by students: what are the software tools that we may use in our project? The perfect answer for such a question is via a finished project as a case study.

A Chabot Admission Academic System with a problem defined as follows:

1. some of college enquiry chat bot systems do not answer adequately for students who is willing to enroll in one of the colleges available within the university.
2. One of the greatest disadvantages of some chatbots is that they have been designed to handle first-level questions only. They may not be able to solve complex queries.
3. Some chatbots may not be appropriate for questions.
4. Some chatbots have not interactive UI and does not support handled devices.

The appropriate software tools for that project was as follows:

1. Microsoft project 2016 **(Planning activities: Tasks duration, Gantt chart, Network diagram)**
2. Creately **(Context diagram, DFD, ERD)**
3. Microsoft office 2020 **(Drawing Data dictionary, Process specification, presentation)**
4. Diagrams.net **(block diagram, design Report)**
5. LARP **(Flowchart & Pseudocode for processes**)
6. Adobe Photoshop **(logo & photos)**

7. Adobe XD (**GUI design**)
8. Visual Studio 2019 (**implementation**)
9. Visual Studio code (**implementation**)
10. SQL (**Database implementation**)
11. Postman (**Test API**)
12. Adobe after effect (**Demo**)
13. Adobe Photoshop (**Demo**)
14. Adobe premiere (**Demo**)

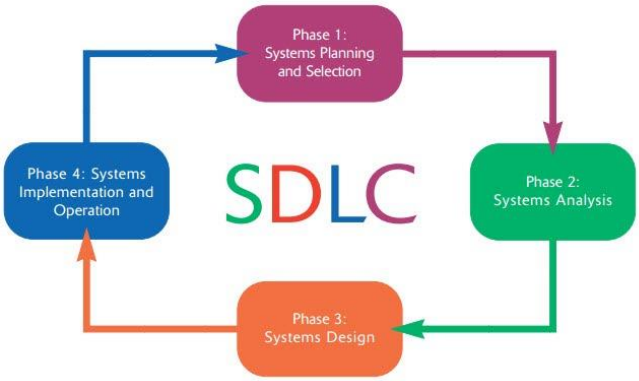
CHAPTER 5

UNIFIED MODELING LANGUAGE (UML)

By: Prof. Dr. Hesham El-Deeb

5.1 Introduction

The Unified Modeling Language (UML) is UML is a standardized graphical display format for the visualization, specification, design and documentation of software systems. It offers a set of standardized diagram types with which complex data, processes and systems can easily be arranged in a clear, intuitive manner. The usage of the UML will be essential in the different phases of the software life cycle (SDLC).



The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design. It was developed at Rational Software in 1994–1995, with further development led by them through 1996.

In 1997, UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005, UML was also published by the International Organization for Standardization (ISO) as an

approved ISO standard. Since then the standard has been periodically revised to cover the latest revision of UML. In software engineering, most practitioners do not use UML, but instead produce informal hand drawn diagrams; these diagrams often include elements from UML however

5.1.1 UML Urgency

UML offers a way to visualize a system's architectural blueprints in a diagram, including elements such as:

- Any activities (jobs);
- Individual components of the system; and how they can interact with other software components;
- How the system will run;
- How entities interact with others (components and interfaces);
- External user interface.

Although originally intended for object-oriented design documentation, UML has been extended to a larger set of design documentation (as listed above), and been found useful in many contexts.

5.1.2 Advantages of UML

The use of UML as a “common language” leads to an improvement in cooperation between technical and non-technical competencies like project leaders, business analysts, software/hardware architects, designers and developers. It helps in the better understanding of systems, in revealing simplification and/or recoverability options, and in the easier recognition of

possible risks. Through early detection of errors in the analysis and design phase of a project, costs can be reduced during the implementation phase. The advantages associated with Round-Trip Engineering offer developers the ability to save a great deal of time. Although UML was initially developed for the modeling of software systems, it can also be implemented for any hardware or organizational project. By allowing processes to be visualized, they can subsequently be analyzed and improved. Developers of embedded systems or real-time systems may use non-object-oriented programming languages – applying UML makes sense in this case also.

5.1.3 UML Versions

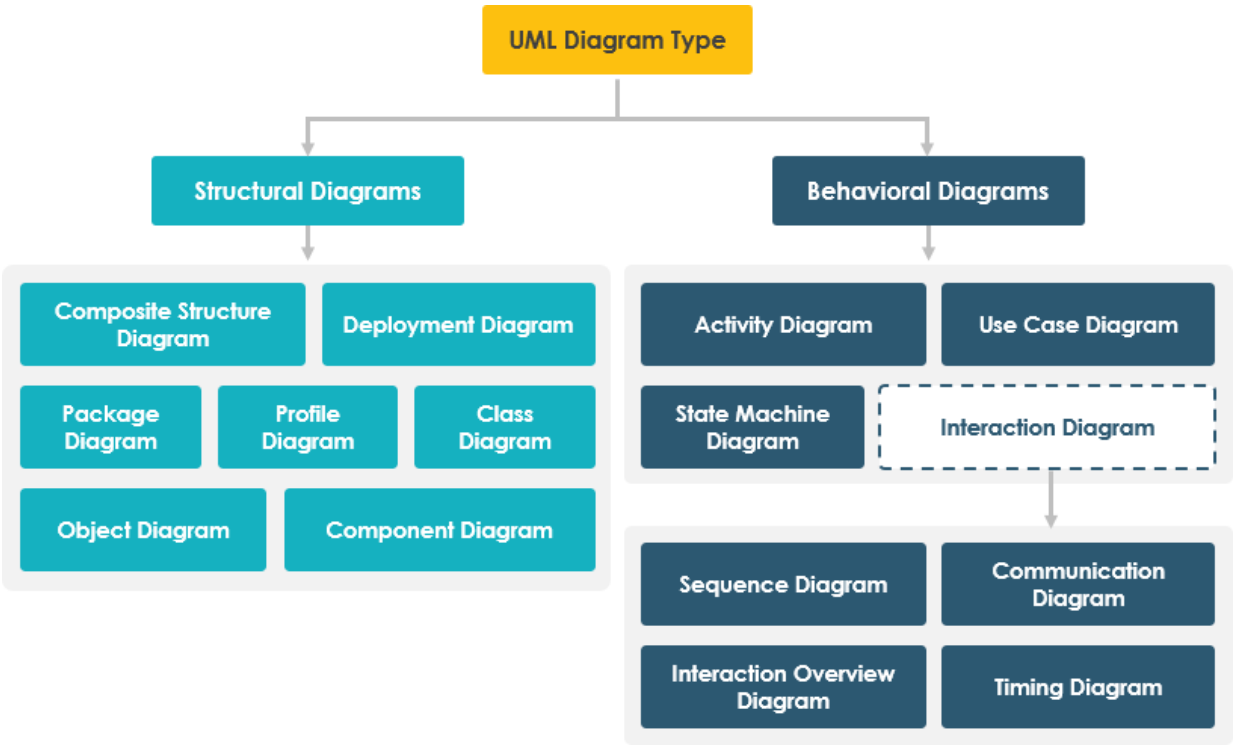
UML was adopted by Object Management Group. This was the first version of UML and the journey continue until 2017 having the last version of the UML as shown in the following table.

VERSION	ADOPTION DATE	URL
2.5.1	December 2017	https://www.omg.org/spec/UML/2.5.1/
2.4.1	July 2011	https://www.omg.org/spec/UML/2.4.1/
2.3	May 2010	https://www.omg.org/spec/UML/2.3/
2.2	January 2009	https://www.omg.org/spec/UML/2.2/
2.1.2	October 2007	https://www.omg.org/spec/UML/2.1.2/
2.0	July 2005	https://www.omg.org/spec/UML/2.0/
1.5	March 2003	https://www.omg.org/spec/UML/1.5/

VERSION	ADOPTION DATE	URL
1.4	September 2001	https://www.omg.org/spec/UML/1.4/
1.3	February 2000	https://www.omg.org/spec/UML/1.3/
1.2	July 1999	https://www.omg.org/spec/UML/1.2/
1.1	December 1997	https://www.omg.org/spec/UML/1.1/

5.2 UML Diagram Types

So what are the different UML diagram types? There are two main categories; **structure diagrams** and **behavioral diagrams** as shown in the following figure.



Structure diagrams show the things in the modeled system. In a more technical term, they show different objects in a system. **Behavioral diagrams** show what should happen in a system. They describe how the objects interact with each other to create a functioning system.

The following table to figure out the differences between various UML diagrams. These diagrams could be used in different phases with different perspectives according to the following table:

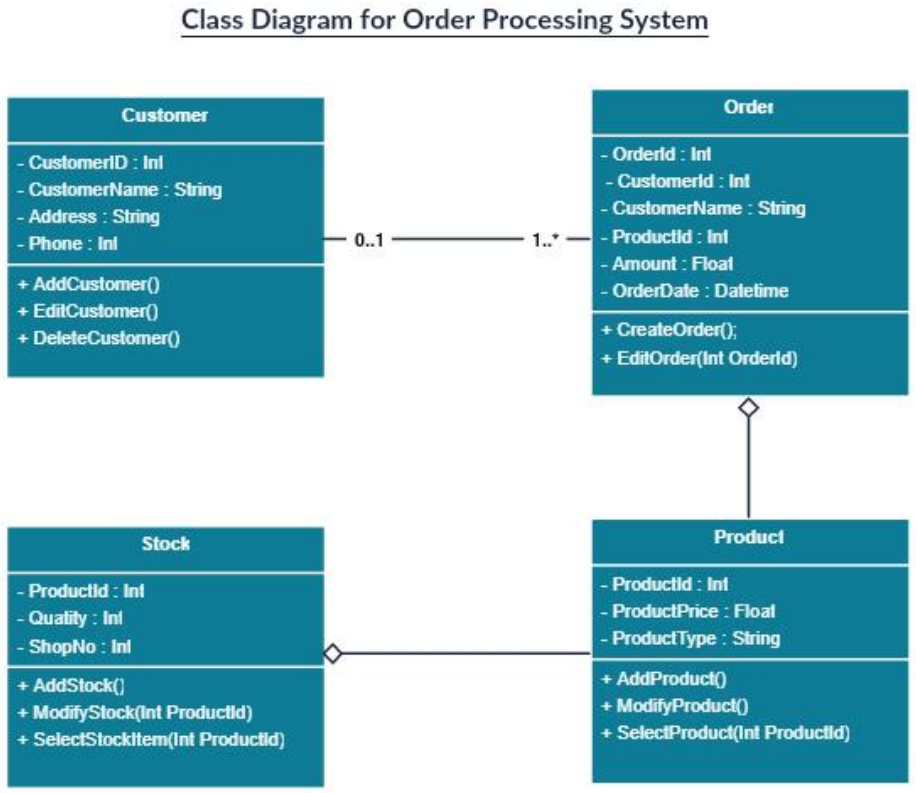
Diagram Name	Used to	Primary Phase
Structure Diagrams		
Class	Illustrate the relationships between classes modeled in the system.	Analysis, Design
Object	Illustrate the relationships between objects modeled in the system. Used when actual instances of the classes will better communicate the model.	Analysis, Design
Package	Group other UML elements together to form higher level constructs.	Analysis, Design, Implementation
Deployment	Show the physical architecture of the system. Can also be used to show software components being deployed onto the physical architecture.	Physical Design, Implementation
Component	illustrate the physical relationships among the software components.	Physical Design, Implementation
Composite Structure	Illustrate the internal structure of a class, i.e., the relationships among the parts of a class.	Analysis, Design
Behavioral Diagrams		
Activity	Illustrate business workflows independent of classes, the flow of activities in a use case, or detailed design of a method.	Analysis, Design
Sequence	Model the behavior of objects within a use case. Focuses on the time-based ordering of an activity.	Analysis, Design
Communication	Model the behavior of objects within a use case. Focuses on the communication among a set of collaborating objects of an activity.	Analysis, Design
Interaction Overview	Illustrate an overview of the flow of control of a process.	Analysis, Design
Timing	Illustrate the interaction that takes place among a set of objects and the state changes in which they go through along a time axis.	Analysis, Design
Behavioral State Machine	Examine the behavior of one class.	Analysis, Design
Protocol State Machine	Illustrates the dependencies among the different interfaces of a class.	Analysis, Design
Use-Case	Capture business requirements for the system and to illustrate the interaction between the system and its environment.	Analysis

5.2.1 Class Diagram

Class diagrams are the main building block of any object-oriented solution. It shows the classes in a system, attributes, and operations of each class and the relationship between each class.

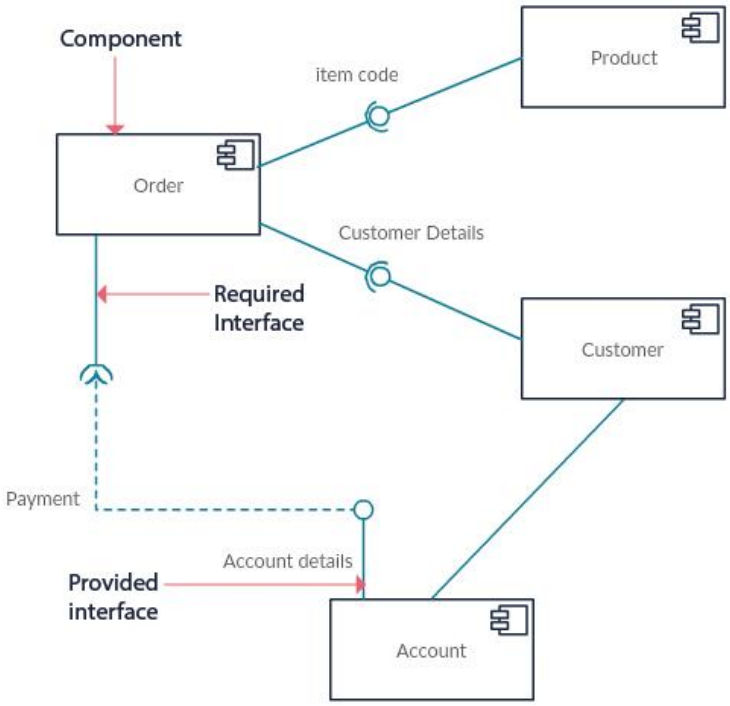
In most modeling tools, a class has three parts. Name at the top, attributes in the middle and operations or methods at the bottom. In a large system with many related classes, classes are grouped together to create class diagrams. Different relationships between classes are shown by different types of arrows.

Below is an image of a class diagram. Follow the link below for more class diagram examples or get started instantly with our class diagram templates.



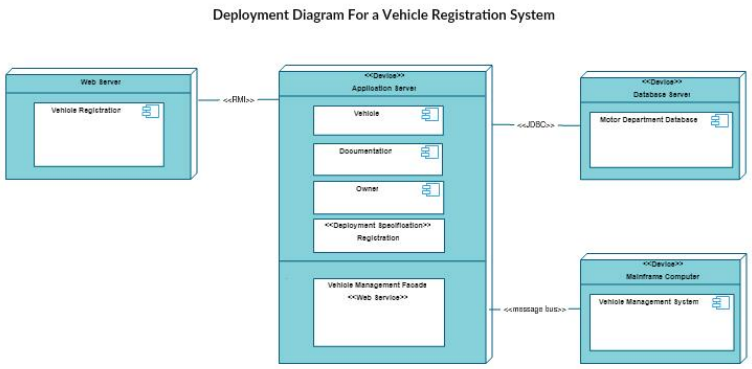
5.2.2 Component Diagram

A component diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems with many components. Components communicate with each other using interfaces. The interfaces are linked using connectors. The image below shows a component diagram.



5.2.3 Deployment Diagram

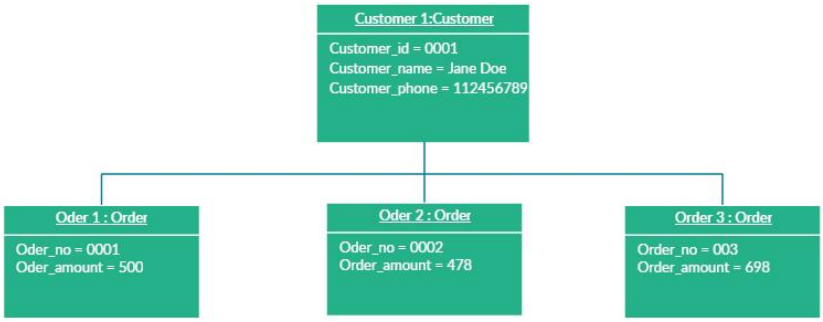
A deployment diagram shows the hardware of your system and the software in that hardware. Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration. Below is an example deployment diagram.



5.2.4 Object Diagram

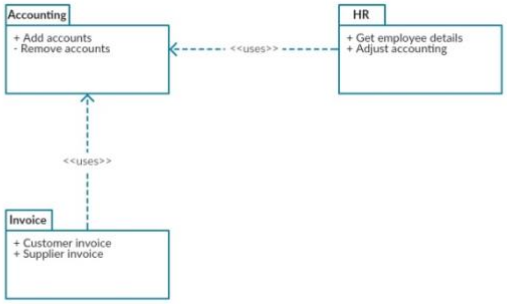
Object Diagrams, sometimes referred to as Instance diagrams are very similar to class diagrams. Like class diagrams, they also show the relationship between objects but they use real-world examples.

They show how a system will look like at a given time. Because there is data available in the objects, they are used to explain complex relationships between objects.



5.2.5 Package Diagram

As the name suggests, a package diagram shows the dependencies between different packages in a system. Check out this wiki article to learn more about the dependencies and elements found in package diagrams.

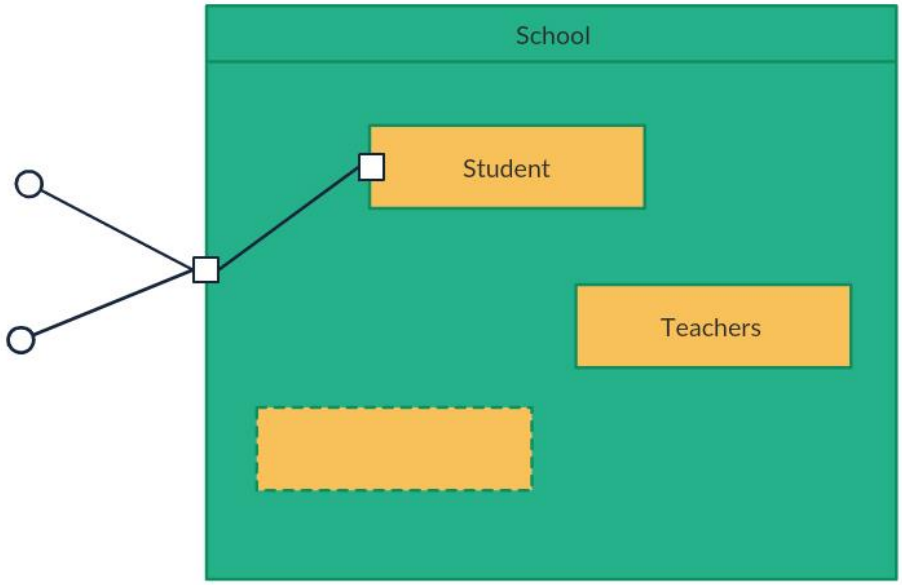


5.2.6 Profile Diagram

Profile diagram is a new diagram type introduced in UML 2. This is a diagram type that is very rarely used in any specification. For more profile diagram templates, visit our diagram community.

5.2.7 Composite Structure Diagram

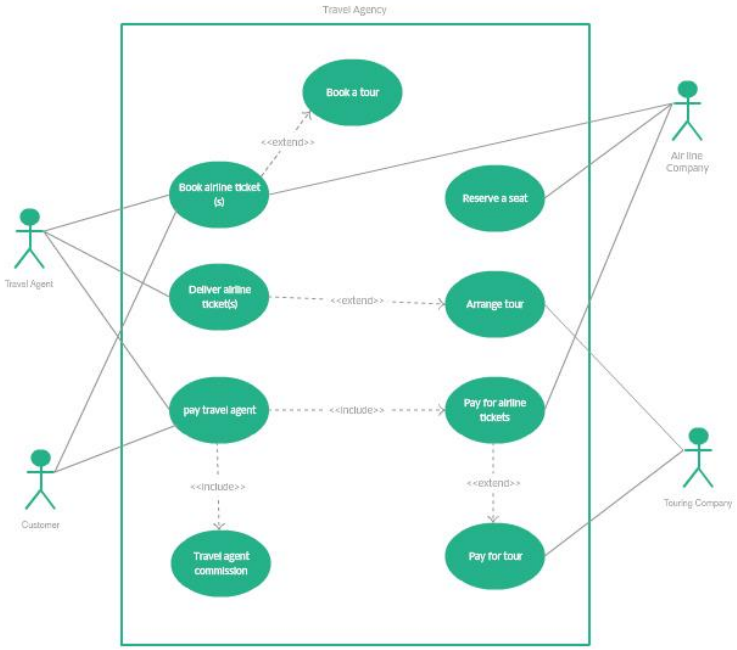
Composite structure diagrams are used to show the internal structure of a class. Some of the common composite structure diagrams.


School
Student
Teachers

5.2.8 Use Case Diagram

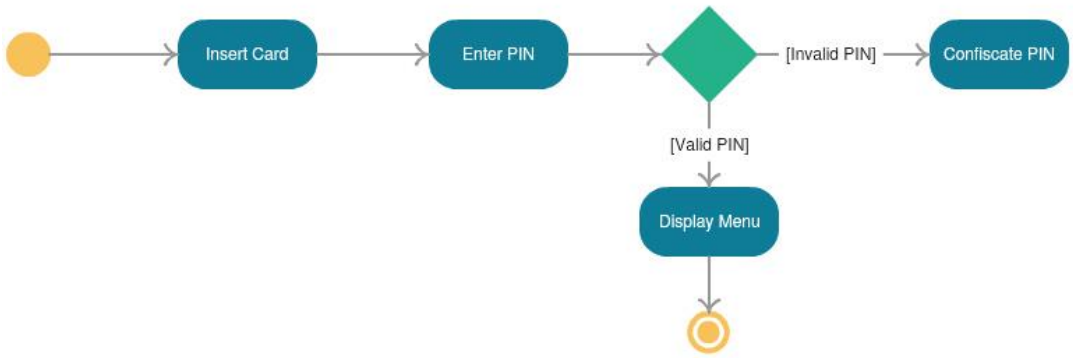
As the most known diagram type of the behavioral UML types, Use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions interact.

It's a great starting point for any project discussion because you can easily identify the main actors involved and the main processes of the system. You can create use case diagrams using our tool and/or get started instantly using our use case templates.



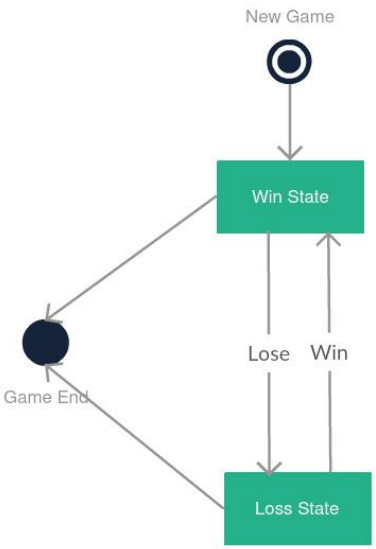
5.2.9 Activity Diagram

Activity diagrams represent workflows in a graphical way. They can be used to describe the business workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State machine diagrams.



5.2.10 State Machine Diagram

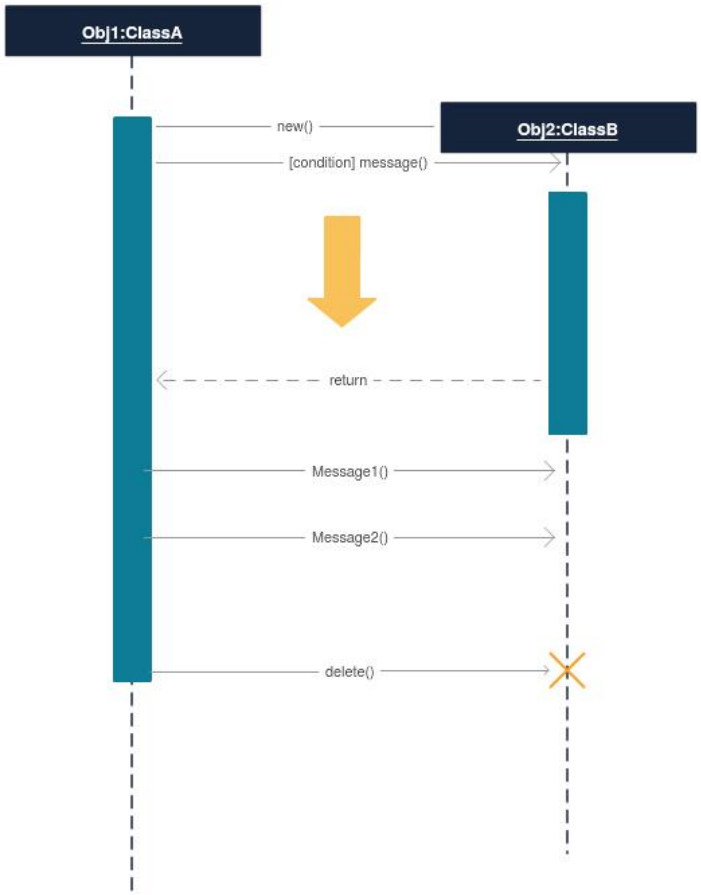
State machine diagrams are similar to activity diagrams, although notations and usage change a bit. They are sometimes known as state diagrams or state chart diagrams as well. These are very useful to describe the behavior of objects that act differently according to the state they are in at the moment. The State machine diagram below shows the basic states and actions.



State Machine diagram in UML, or State chart diagram

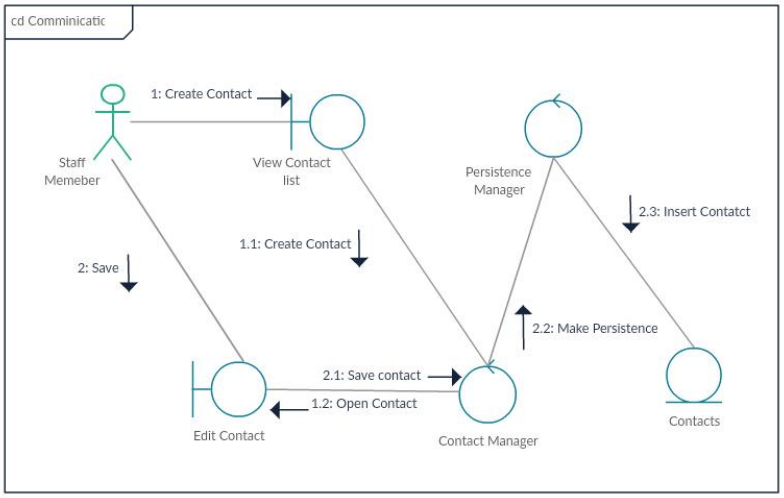
5.2.11 Sequence Diagram

Sequence diagrams in UML show how objects interact with each other and the order those interactions occur. It's important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are shown as arrows. This article explains the purpose and the basics of Sequence diagrams. Also, check out this complete Sequence Diagram Tutorial to learn more about sequence diagrams.



5.2.12 Communication Diagram

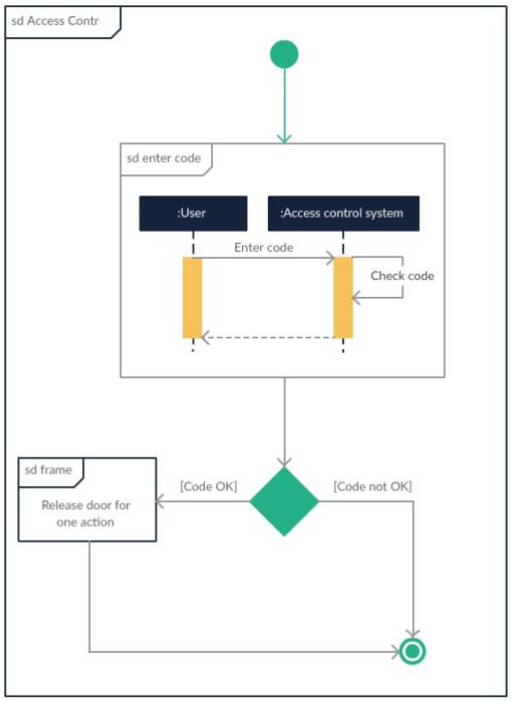
In UML 1 they were called collaboration diagrams. Communication diagrams are similar to sequence diagrams, but the focus is on messages passed between objects. The same information can be represented using a sequence diagram and different objects. Click here to understand the differences using an example.



5.2.13 Interaction Overview Diagram

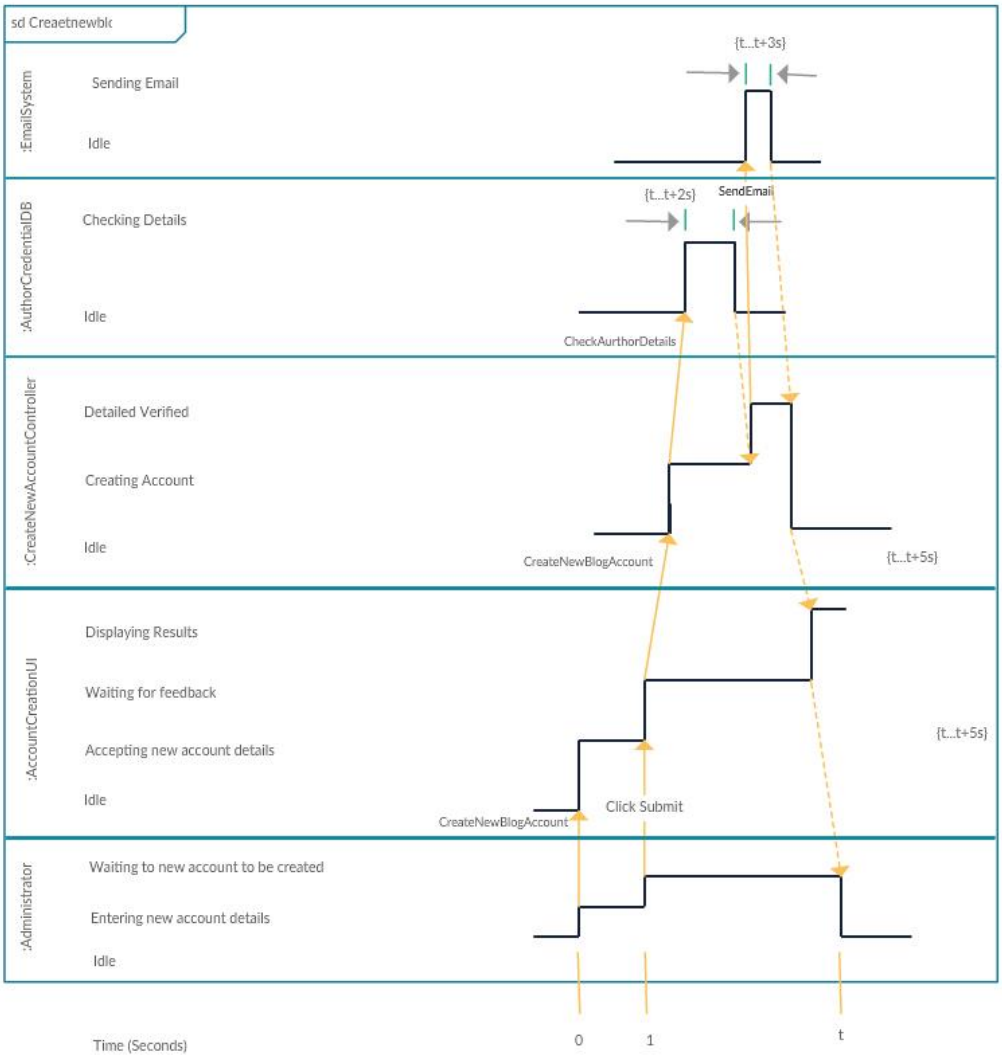
Interaction overview diagrams are very similar to activity diagrams. While activity diagrams show a sequence of processes, Interaction overview diagrams show a sequence of interaction diagrams.

They are a collection of interaction diagrams and the order they happen. As mentioned before, there are seven types of interaction diagrams, so any one of them can be a node in an interaction overview diagram.



5.2.14 Timing Diagram

Timing diagrams are very similar to sequence diagrams. They represent the behavior of objects in a given time frame. If it's only one object, the diagram is straightforward. But, if there is more than one object is involved, a Timing diagram is used to show interactions between objects during that time frame.



Mentioned above are all the UML diagram types. UML offers many diagram types, and sometimes two diagrams can explain the same thing using different notations.

5.3 Order of Drawing the UML Diagrams

After reading Martin Fowler's book on UML: the following order is recommended to follow:

1. Analyze the actors first (a UML diagram on the actors and how they interact with the system)

2. Use-Case Diagram.
3. Analyze external factors that would impact a system (not really a diagram just including for information sake)
4. Analyze risks in the systems and estimating their impact (again not really a diagram)
5. Sequence Diagram. (With data-flow in them)
6. Class Diagram.

The rest of the UML diagrams are preferred to be drawn in parallel for documentation purpose.

Generally speaking, to develop a project using UML; you have to do the following steps:

- Construct a use case diagram for your application according its functionality
- Design a class diagram for your application which include data classes only
- Write the scenario (written use case) for each use case in your use case diagram
- Draw a sequence diagram for each written use case
- Derive methods required in each data class from sequence diagrams
- Derive the additional I/O interface classes and Control classes from the sequences diagrams
- Construct your application Analysis class diagram from all data, I/O interface, and control classes

To learn more and get OneNote, visit www.onenote.com

CHAPTER 6

WRITING COLLABORATIVELY

By: Dr. Rasha M. Saeed

6.1 Introduction

You have been introduced to collaborative writing in the *Technical Report Writing* course you studied in your first year of enrollment in college. Now that you made it to the fourth year and registered for your graduation project, it is the time you practice actual collaboration on a large scale. This may be the first time you and your team mates work together over an extended period of time to produce a document.

6.2 The Planning Stage

Planning allows collaborators to develop an effective approach and reach agreement before investing a lot of time and resources. Planning prevents small problems from becoming big problems with a deadline looming.

In the first meeting with your supervisor, it is essential that one or two group members serve as **recorders** (i.e. secretaries) to capture the thoughts before they are lost, either in the form of written notes (aka. *minutes of meeting*) or audio recording. It helps if the minutes and recording are made available for everyone to check later. Your supervisor will discuss the nature of academic research, general idea of the project, allocation, objectives and excepted outcomes. He/she may engage with you in brainstorming, assign roles and leadership, introduce different sources

of information, and familiarize you with regulations set by Faculty concerning the assessment and evaluation of the final product. In short, your supervisor will guide you on how to embark on your research journey.

While in the planning stage, a group should take several steps that can save a lot of hassle and bother later on:

1. **Choose a leader/ coordinator** from among the group. This person serves as the link between the team and the supervisor. In choosing a leader/coordinator, group members should avoid both passive and dominating personality types. The group should give the leader/coordinator the authority to enforce deadlines, call meetings, and shepherd the group through the collaborative process.

2. **Set deadlines for completed work** and stick to them. An accurate schedule helps collaborators plan ahead, allocate their time, and meet deadlines. These deadlines are also called *milestones*. Deadlines should be realistic and co-relate with allocation set by your supervisor. Deadlines should also allow sufficient time for the revising and editing stages. It is always useful to set the deadlines backward. For example, if your graduation project defense is scheduled mid June, it is wise to set a deadline for the editing and proofreading stage during the first week of June, a deadline for the revision process during the third week May, and so on. Take into consideration the timings for final exams, mid-terms exams, mid-year recess and any other academic activity that may cause inevitable delays to the writing process.

3. **Schedule frequent review sessions**. In these sessions, group members should offer support and encouragement, check the workloads, and see if group goals are being met. Usually these sessions are scheduled weekly in your semester timetable. It is an excellent chance for you to meet with other members and your supervisor.

4. **Make up a style sheet** for everyone to follow. Agree on rubrics as these:

   - Font type and size and spacing for text
   - Line length and margins
   - Highlighting (boldface, italics, and so forth)
   - Placement and style of page numbers
   - Format for headers and footers
   - Placement, caption style, and identification (numbers or letters) for figures and tables
   - Format for at least three levels of headings, to include font size and type, grammatical structure, capitalization, placement, and spacing
   - Format for lists and informal tables [bullets (•), numbers, spacing, indentation, and so forth]
   - Documentation for both text and graphics

   At any point of time, whenever group members are asked to submit a report on their work, different segments will fall together in harmony and will naturally look unified, hence the document will need less revision.

6.3 The Drafting Stage

The actual drafting of the graduation project report starts in the second semester of registration; however, you and your team members will inevitably practice initial drafting in the first semester as you collect your data from different sources of information and develop the literature review. Therefore, it is essential now to be familiar with the several possible approaches of collaborative drafting. You can adopt all of these methods, or combinations of them:

6.3.1 Dividing the Work

Since your graduation project report is a **lengthy document**, and different team members are assigned different tasks in the allocation and hence will be assessed individually, the most common drafting procedure is to divide the drafting among the group. Each member of the group takes responsibility for a segment of the organizational plan and writes a draft based on the allocation of the project.

However, even when a group has agreed on the design features, there will be **many stylistic differences** in the first drafts. So, be prepared to spend a good deal of time revising and polishing to get a final product in which all the segments fit together smoothly.

6.3.2 Drafting in Collaboration

For **short important chapters**, such as the introduction and the conclusion, a group may want to draft the document in collaboration. Members write together, may be in front of a screen; generally, one person will control the keyboard, but all collaborators can read the screen and

provide immediate feedback as changes are made to the document. This can also take place online in a videoconference session on MS Teams or Zoom, for example.

6.3.3 One Person Doing the Drafting

It may be the case that after every member finalizes his/her part of the initial draft, the group assigns one lead writer from the group to put the segments together, blending the parts into a stylistic uniformity. However, it should be clear to team members and the supervisor which segment belongs to which member – since this affects the assessment of individual members.

6. 4 The Revising Stage

Revising your graduation project report will take place in the final stages, as your defense approaches. Indeed, your supervisor will have scrutinized and revised the document carefully; however, team members should have worked to present a revised draft to the supervisor. In revising, concern yourself primarily with content, organization, style, and tone. Be concerned with how well a draft fits purpose and organization. There are several approaches to collaborative revision.

6.4.1 Criterion-Based Comments

Criterion-based comments measure the draft against some standards, may be in the form of checklist. Criteria of assessment can be distilled from the *Final Project Marking Sheet* presented to you in the first chapter of this book. Criteria include items such as document design, organization, use of tables and graphs, referencing, spelling and grammar, etc.

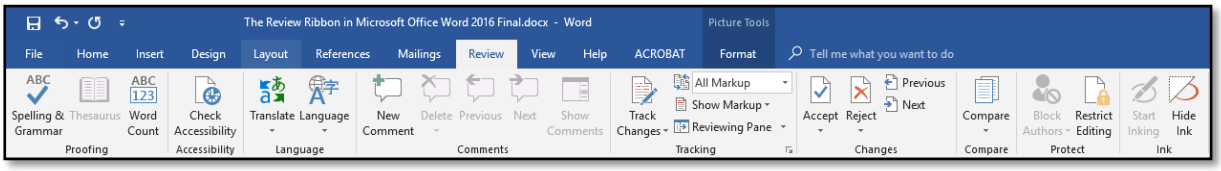
6.4.2 Reader-Based Comments

Reader-based comments are simply your reaction as a reader to what is before you. Again, to ensure objectivity of revision, it is essential to be guided by the *Final Project Marking Sheet*.

6.4.3 Word Processing

MS Word offers two powerful features in the *Review* tab that you will find useful in collaborative work:

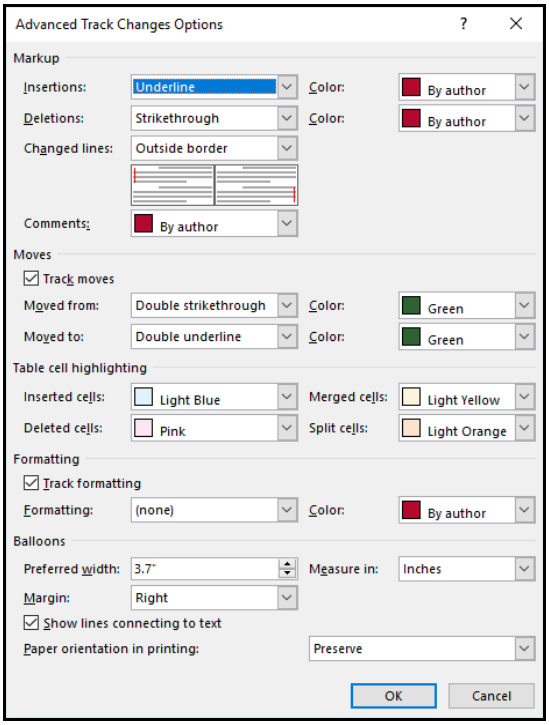
- **comment feature** lets readers add electronic comments to a file.
- **revision feature** lets readers mark up a text by deleting, revising, and adding words and tracks who made which suggested changes.



You can distribute your document to readers electronically, compare two versions, allow or restrict editing without ever having to print your document. Therefore, it is worthwhile to educate yourself on how to take advantage of these indispensable features. Several tutorials are available on *YouTube*. Here are quick tips to **electronically review** a document.

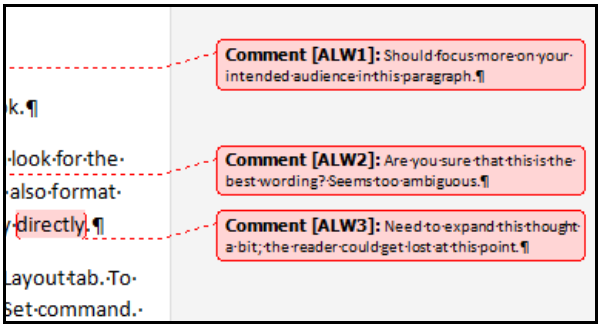
1. Highlight the relevant text
2. Select the **New Comment** button in the **Comments** group to write comments in a bubble in the margin.
3. Select the **Track Changes** button to distinguish between revised text and original text.

4. On the **Home** tab in the **Font** group, select the **Text Highlight** button to emphasize a particular passage.
5. To change the color or design of comment bubbles or markup, open the **Track Changes Options** dialog box



To revise a document that has already been commented on by reviewers, you can do the following:

1. Use the **Tracking** group to change how the document is displayed.
2. Select buttons in the **Changes** group to see the **previous** or **next** comment and to **accept** or **reject** a change.
3. Select the **Reviewing Pane** button to review all comments and changes.



6.5 The Editing and Proofreading Stage

Make editing and proofreading a separate process from revision. This is the next-to-final step before you submit your report to its intended audience (initially to the examiners during the defense, and later to Faculty after defense). You should make sure that your report is as mechanically perfect as possible and that it meets the requirements of **standard English** and the **format** requirements determined by Faculty.

The editing and proofreading stage is a particularly essential since drafting has been divided among the group. Make sure **you have typed what you meant to type**. All the equal headings should look alike; margins and spacing should be consistent, etc. In other words, **the final product should be seamless.** That is, no one should be able to tell where Sarah's work leaves off and John's begins. The editing and proofreading stage involves the following:

- **Checking mechanics**, i.e. spelling and grammar. *MS Word* spelling and grammar checkers will significantly help.
- **Checking style sheet**. Be sure that the entire document adopts Harvard Style Sheet in both in-text citations and reference list.
- **Checking illustrations**, i.e. tables and graphs. Make sure they are well-placed in the report, they are numbered, and they match the list of tables and list of figures.
- **Checking document design**. Make sure that your design makes your document readable and accessible for your readers. Also, check if your table of contents is complete and accurate, and if it matches the headings you have used. Also, check if your headers and footers accurately portray your material.

6.6 Conducting Meetings

Collaboration involves meetings / conferences. In a troubled group conference, the climate is defensive. Conferees feel insecure, constantly fearing a personal attack and preparing to defend themselves. Nevertheless, in a productive group conference, the climate is permissive and supportive. Members truly listen to one another. Whether you are meeting live in a room on campus or using videoconferencing tools, the aspects of meetings discussed in this section can help you use your time productively and produce the best possible document.

6.6.1 Listening Effectively

Participating in a meeting involves listening and speaking. Unlike hearing, which involves receiving and processing sound waves, listening involves understanding what the speaker is saying and interpreting the information. If you listen carefully to your team members and to your supervisor, you will understand what they are thinking and you will be able to speak knowledgeably and constructively. Follow these steps to improve your effectiveness as a listener.

- **Pay attention to the speaker**. Look at the speaker, and don't let your mind wander.
- **Don't get emotionally involved with the speaker's ideas**. Even if you disagree, continue to listen. Keep an open mind. Don't stop listening in order to plan what you are going to say next.
- **Ask questions to clarify what the speaker said**.

- **Provide appropriate feedback**. The most important feedback is to look into the speaker's eyes. You can nod your approval to signal that you understand what he or she is saying. Appropriate feedback helps assure the speaker that he or she is communicating effectively.

6.6.2 Communicating Diplomatically

Because collaborating can be stressful, it can lead to interpersonal conflict. People can become frustrated and angry with one another because of personality clashes or because of disputes about the project. If the project is to succeed, however, team members have to work together productively and harmoniously. When you speak in a team meeting, you want to appear helpful, not critical or bossy.

The following suggestions for communicating diplomatically will help you communicate effectively:

- **Listen carefully, without interrupting.**

- **Give everyone a chance to speak.** Don't dominate the discussion.

- **Avoid personal remarks and insults.** Be tolerant and respectful of other people's views and working methods.

- **Don't overstate your position.** A modest qualifier such as "I think" or "it seems to me" is an effective signal to your listeners that you realize that everyone might not share your views.

- **Don't get emotionally attached to your own ideas.** When people oppose you, try to understand why. Digging in is usually unwise – unless it's a matter of principle – because, although it's possible that you are right and everyone else is wrong, it's not likely.

- **Ask relevant questions.** Bright people ask questions to understand what they hear and to connect it to other ideas. Asking questions also encourages other team members to examine what they hear.

6.6.3 Critiquing Constructively

Most people are very sensitive about their writing. Following the following three suggestions for critiquing the work of a group member will increase the chances that your colleague will consider your ideas positively.

- **Start with a positive comment.** Even if the work is weak, say, "You've obviously put a lot of work into this, John. Thanks." Or, "This is a really good start. Thanks, John." Whenever possible, support other members of the group with **compliments.**

- **Discuss the larger issues first.** Begin with the big issues, such as organization, development, logic, design, and graphics. Then work on smaller issues, such as paragraph development, sentence-level matters, and word choice. Leave editing and proofreading until the end of the process.

- **Talk about the document, not the writer.** Your goal is to improve the quality of the document you will submit, not to evaluate the writer or the draft. Offer constructive suggestions. Study the following example:

RUDE	Why didn't you include the context diagram here, as you said you would?
↓	
BETTER	I wonder if the report would be stronger if we included the context diagram here, as we agreed earlier.

CHAPTER 7

DEVELOPING A LITERATURE REVIEW

By: Dr. Rasha M. Saeed

7.1 Introduction

Research falls into two categories: primary research and secondary research. *Primary research* involves discovering or creating technical information yourself. *Secondary research* involves finding information that other people have already discovered or created. This chapter presents secondary research because you will probably do secondary research first. To design an application that goes into primary research, you need a thorough understanding of the information that already exists about your subject by reading scholarly literature in academic journals and books.

In the first semester of registering for your graduation project, you are required to collect and develop a *Literature Review*. It usually falls after the introduction and before the research methods section of your final graduation report that you submit by the end of the second semester of registration; its main purpose is to provide a context for your research point. So, make sure your sources are well-chosen and your research is thorough.

A *Literature Review* is a survey of scholarly articles, books, dissertations, conference proceedings, and/or other published material on a topic discusses those sources in conversation with each other (also called *synthesis*). The review provides a summary, description, and critical

evaluation of a topic, issue, or area of research – in your case it is the specific topic of your graduation project. It usually includes:

- Overview of the subject and the objective(s) of the review
- Analysis of works in favor, works against, and works with neutral views on the subject. These should be clearly divided
- Explanations of the similarities and differences between the works
- Comparison of different views held by other authors
- Critique of the methodology
- Examination of gaps in the research
- Evaluation of how each study contributes to the argument in question
- Conclusion that summarizes the literature review

The most interesting literature reviews are often written as arguments. Often, the literature review is where you can establish your research as filling a particular gap or as relevant in a particular way. You have some chance to do this in your introduction, but the literature review section gives a more extended opportunity to establish the scholarly conversation in the way you would like your readers to see it. In doing so, you argue for your place in the conversation, which tends to make the literature review more compelling than a simple reporting of other sources.

7.2 Selecting your Literature

There is a plethora of online databases and content that you use to find and select published material. Your supervisor will guide for where you can look for literature. All you need is to figure out relevant databases related to your field of study and project topic. You can look up scholarly material using academic search engines – such as *Google Scholar* (https://scholar.google.com/) – that direct you to digital and printed resources on your area of interest.

A periodical index is a list of articles classified according to title, subject, and author. It can help you determine which journals you want to locate. Periodicals are excellent sources of information because they offer recent, authoritative discussions of specific subjects. Libraries subscribe to services, such as Springer, LexisNexis, ProQuest, InfoTrac, Gale Virtual Reference (GVRL), and ERIC, that provide access to large databases of journal articles, conference proceedings, newspapers, and other documents.

The Egyptian Knowledge Bank (EKB) (www.ekb.eg). It is the world's largest digital library and knowledge hub granting unlimited resources exclusively for Egyptians. You can simply create an account with your National ID Number and your MTI academic email account. This will give you priceless access to specialized databases of periodicals and books such as IEEE, SAGE, Britannica, Cambridge UP, Clarivate, EBSCO, ELSEVIER, Springer, Lexis Nexis, ProQuest, WIELY, and many others.

CiteSeerx (https://csxstatic.ist.psu.edu/) is an evolving scientific literature digital library and search engine that has focused primarily on the literature in computer and information science. Rather than creating just another digital library, CiteSeerx attempts to provide resources such as algorithms, data, metadata, services, techniques, and software that can be used to promote other digital libraries. CiteSeerx has developed new methods and algorithms to index PostScript and PDF research articles on the Web. Citeseerx provides the following features.

The Association of Computing Machinery (ACM) has developed a Digital Library platform (https://www.acm.org/publications/digital-library). It is the world's most comprehensive database of full-text articles and bibliographic literature covering computing and information technology. This renowned repository includes the complete collection of ACM publications plus an extended bibliographic database of core works in computing from scholarly publishers.

The IEEE Computer Society (https://www.computer.org/csdl) is the premier source for information, inspiration, and collaboration in computer science and engineering. Connecting members worldwide, the Computer Society empowers the people who advance technology by delivering tools for individuals at all stages of their professional careers. The available resources include international conferences, peer-reviewed publications, a robust digital library, globally recognized standards, and continuous learning opportunities.

7.3 Analyzing and Preparing the Literature

The following steps will help you in preparing your review:

1. First, give your selected literature a **brief overview**. Skim through the content and get the essence of what the author is trying to prove or disprove. It would be a good idea to **read the abstract** and first few paragraphs of the introduction in this step. You can also **take notes** during this step.

2. Next, based on what you read, **arrange your material** and think over the headings, subheadings, and divisions you will use for your review. While taking notes you should:
   - Define key terms
   - Look at the statistics
   - Identify key patterns
   - Check emphases, strengths, and weaknesses
   - Check for gaps in the literature
   - Identify relationships between studies
   - Evaluate the methodologies used

3. **Summarize the literature**. You may do this initially in a table or concept map format if you wish to make it easier; however, the final product must be presented in formal paragraph form.

7.4 Writing Your Review

Usually you will need to **synthesize** research rather than just summarizing it. This means drawing connections between sources to create a picture of the scholarly conversation on a topic over time. Many students struggle to synthesize; here are some strategies to help you:

- The point of synthesis is to show your readers how you understand your research, to help them read the rest of your document.

- Synthesis is like hosting a dinner party: imagine all your sources are together in a room, discussing your topic. What are they saying to each other?

- Look at the in-text citations in each paragraph. Are you citing just one source for each paragraph? This usually indicates summary only. When you have multiple sources cited in a paragraph, you are more likely to be synthesizing them (not always, but often).

A literature review uses a basic introduction-body-conclusion structure. Make sure you cite all your references.

7.4.1 Introduction

- Start by identifying the problem statement or your objective (or thesis).
- Explain why this area of study is important.
- A forecast of key topics or texts that will appear in the review
- Give your reasons for selecting the literature you chose as opposed to “other” material that may or may not have been relevant.

7.4.2 Body

- Summarize and synthesize: Give an overview of the main points of each source and combine them into a coherent whole.

- Analyze and interpret: Do not just paraphrase other researchers – add your own interpretations where possible, discussing the significance of findings in relation to the literature as a whole.
- Critically Evaluate: Mention the strengths and weaknesses of your sources.
- Write in well-structured paragraphs: Use transition words and topic sentence to draw connections, comparisons, and contrasts.

7. 4.3 Conclusion

- Summarize the key findings you have taken from the literature and emphasize their significance.
- Explain how your study fills in gaps in the existing reviews or why the repetition is necessary.
- Provides support for an allegation made in the introduction, refute a hypothesis, or simply critique the study to encourage more work in the area.

7.5 Organizing your Review

Literature reviews can take many different organizational patterns depending on what you are trying to accomplish with the review. Here are some examples:

7.5.1 Chronological

The simplest approach is to trace the development of the topic over time (i.e. historically), which helps familiarize the audience with the topic. If you choose this strategy, be careful to avoid simply listing and summarizing sources in order. Try to analyze the patterns, turning points,

and key debates that have shaped the direction of the field. Give your interpretation of how and why certain developments occurred.

7.5.2 Thematic

If you have found some recurring central themes or concepts that you will continue working with throughout your piece, you can organize your literature review into subsections that address different aspects of the topic.

7.5.3 Methodological

If you draw your sources from different disciplines or fields that use a variety of research methods, you can compare the results and conclusions that emerge from different approaches. For example:

- Qualitative versus quantitative research
- Empirical versus theoretical scholarship

7.5.4 Theoretical

The literature review may represent the foundation for your theoretical framework. You can use it to discuss various theories, models, and definitions of key concepts. You can argue for the relevance of a specific theoretical approach or combine various theoretical concepts to create a framework for your research.

7.6 Developing an Annotated Bibliography (AB)

As you are doing your research, create an annotated bibliography (AB). Much of the information used in an annotated bibliography can be

used also in a literature review, so you will be not only partially drafting your literature review as you research, but also developing your sense of the larger conversation going on among scholars and professionals in your topic.

An annotated bibliography is a list of citations (journal articles, books, etc.) where each citation is followed by a brief (about 120-150 words) **evaluative** and **descriptive** paragraph of the article. The purpose of annotating is for the reader to get the "gist" of the article by reading this **one paragraph**.

It is recommended that in your annotation you expose the author's point of view, key findings, and show how their work is relevant (e.g., strengthens, has opposing views, complements, provides a new perspective) to your topic. As you build your AB ask yourself:

- Is the article/source adding information to your topic?
- Is it contradicting or confirming ideas you may have read previously about?
- Is it a new source?
- Is the author an authority in the area (e.g., published a lot in good journals)?

Asking these questions will help you put together a concise annotated bibliography that will later provide key information to put your presentation together.

Indeed, an annotated bibliography will save you **time**, since recalling what the article is about will be easier after reading other papers on your topic, and will **reduce the chances** of word-for-word plagiarism because the summary will be in your words (you will still need an in-text citation though). In general: **summarize**, **assess** and **reflect** on the work

you are reading. By doing this you are engaging in the analysis of the article in meaningful ways – it will pay off later when you put your presentation or article together.

Moreover, an annotated bibliography will help you develop your own thesis and formulate a research hypothesis. A very important part of research is developing a thesis that is debatable, interesting, and current. Writing an annotated bibliography can help you gain a good perspective on what is being said about your topic. By reading and responding to a variety of sources on a topic, you'll start to see what the issues are, what people are arguing about, and you'll then be able to develop your own point of view.

CHAPTER 8

CITATION

By: Dr. Rasha M. Saeed

8.1 Introduction

In semester one of registering for your graduation project, you are required to collect your data from different resources such as the library and the Internet. Therefore, it is essential at this point of your journey to adhere to the rules of academic integrity represented in **citation**.

Acknowledging the work of others is of paramount importance in all areas of life, but especially in academic research, where the currency is not physical objects, but ideas. Use of the work of others without attribution is considered stealing.

You have a moral and a rhetorical obligation to acknowledge your sources. Your moral obligation is to give appropriate credit to the individuals who deserve it – the people who composed the words, created the illustrations, or developed the ideas you are borrowing. Citing sources also serves the rhetorical function of strengthening the credibility of your investigation because it allows you to attribute your findings to relevant subject specialists. If you do not acknowledge your sources, you are accused of **plagiarism**.

- Plagiarism is copying an author's work and passing it off as your own original work or without giving the author proper credit.

- Citation is the way you tell your readers that certain material in your work came from another source. It also gives your readers the information necessary to find that source again, including:
  - information about the author
  - the title of the work
  - the name and location of the publisher
  - date of publication
  - the page numbers of the material you are borrowing

Keep in mind that the citation of sources offers you the opportunity to demonstrate that your research has been **fair** and **thorough**. If your list of sources, for example, omits a major book on your topic or the World Wide Web site of the organization you are investigating, your readers would have cause to **doubt the validity of your findings**. Similarly, if all your information comes from a single source or a single kind of source (e.g., all websites), your readers would likely consider your research to **be biased or incomplete**. In fact, your list of sources is often **a good indicator of the quality of your investigation and the merits of your conclusions and recommendations.**

You should choose a formal method for citing and acknowledging your sources, e.g. **Harvard Style Sheet**. In a formal system of citation, your references will require specific and consistent formatting. A formal system of citation is particularly important if your document or presentation might be used by others to conduct subsequent research. The consistent formatting of citations will assist researchers in locating the sources you used.

8.2 Types of Citations

When you are conducting research, you will use **two types** of citations:

1. **References:** located at the end of the work and display full citations for sources used arranged alphabetically by the surname of the author. Here is an example:

References

Arnold, Thomas and Scheutz, Matthias. (2020). 'HRI ethics and type-token ambiguity: what kind of robotic identity is most responsible?'. *Ethics and Information Technology*, 22 (4): 357–366.

Berger, Arthur Asa. (2016). 'A Discourse on discourse studies'. *Society*, 53:597–602.

Kress, Gunter and Van Leeuwen, Theo. (2012). *Reading Images: The Grammar of Visual Design*. London: Routledge.

Kress, Gunter and Van Leeuwen, Theo. (2020). *Multimodal Discourse: The Modes and Media of Contemporary Communication*. London: Arnold.

Be aware that References are not the same as **Bibliography**. A bibliography includes all the material consulted in writing even if you have not cited them within it. For your graduation project, you are required to include only a list of references.

2. **In-Text** or **Parenthetical Citations** (within the document): used when directly quoting or paraphrasing a source. They are located in the body of the work and contain a fragment of the full citation, such as the example below:

Discourse Analysis (DA) moves the focus on sentences to more complex form of communication; it focuses on "how people use language and the role that this language plays in social and political life and in culture" (Berger 2016:598). According to Berger (2016), integrating semiotics in DA resulted in "a multidisciplinary enterprise that deals with qualitative approaches to communication of all kinds in many disciplines" (598). With the introduction of Critical Discourse Analysis (CDA), analysts focused on revealing hidden connection between language, power and ideology, opening the door to endless research on the taken-for-granted exercise of power relations in different discourse types. As CDA expanded to include mass-mediated texts that comprise different semiotic modes of communication (i.e. linguistic and visual choices), a visual turn in the field took place and brought about a transformation from monomodality to multimodality as advocated by the valuable works of Kress and Van Leeuwen (2012; 2020).

8.3 Style Guides

Style guides, such as APA, Harvard, IEEE, etc., are rules for citation formatting. They include a wide range of rules and guidelines for works in their respective fields, from grammar and language use to the font and size of headings in a work. Generally, style manuals include everything a writer needs to know in order to make their work look and read just like every other work written in that style – the look of the page, the way other authors are referenced in the body of the work, and even the tone of the writing.

In other words, style guides are used as a way of making common elements consistent across documents written by many writers, in many places, and in many circumstances; as a result, readers from any university (or other audience groups) can read a paper written in APA style and know immediately how to navigate the headings of the paper, which details will be listed in the abstract, how quotes will be introduced and marked, where to look for important citation information, and what each citation element represents.

For your graduation project report, you are required to use Harvard Style Sheet. Here are useful links to familiarize you with Harvard Style:

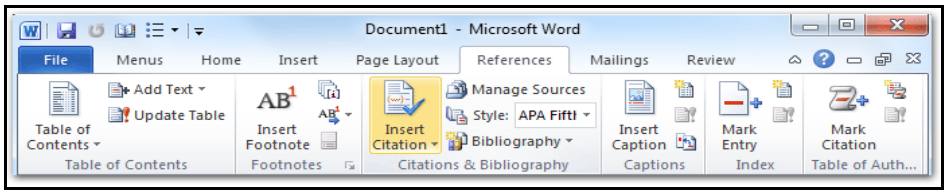
https://library.aru.ac.uk/referencing/files/Harvard_referencing_201920.pdf

https://www.librarydevelopment.group.shef.ac.uk/referencing/harvard.html

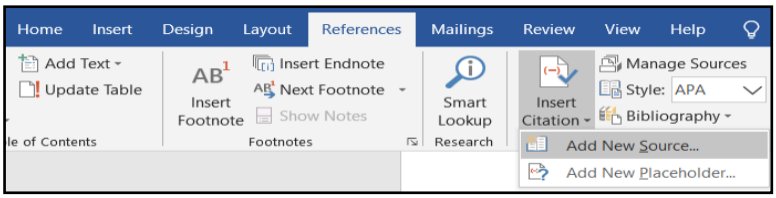
8.3.1 Using the 'Reference Tab' of *MS Word* to Manage Style Sheet

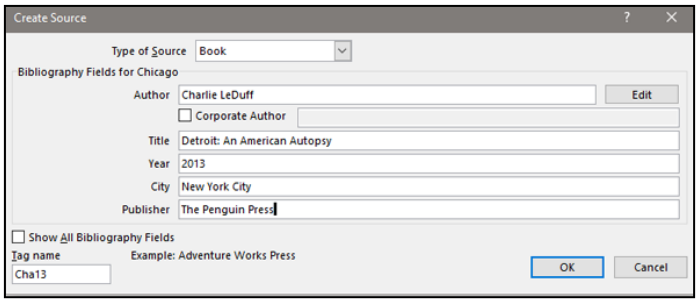
When you write a research paper or your graduation project report, *MS Word* can offer powerful features that you will find useful in compiling your bibliography/references as well as add in-text citation:

1. On the **References** tab, in the **Citations & Bibliography** group, click the arrow next to **Style** and click the style that you want (i.e. Harvard) to use for the citation and source.

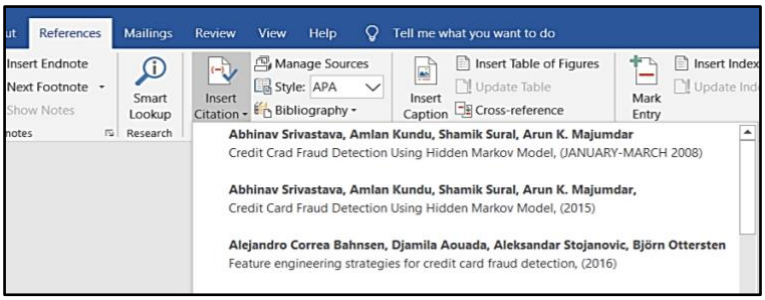


2. Click at the end of the sentence or phrase that you want to cite.
3. On the **Reference** tab, click **Insert Citation**> **Add New Source.**
4. In the **Create Source** dialog box, click the arrow next to **Type of Source**, and select the type of source you want to use (for example, a *book*, an *article* or a *website*), and fill in the required fields.





When you've completed these steps, the citation is added to the list of available citations. The next time you quote this reference, you don't have to type it all out again. Once you've added a source to your list, you can cite it again.



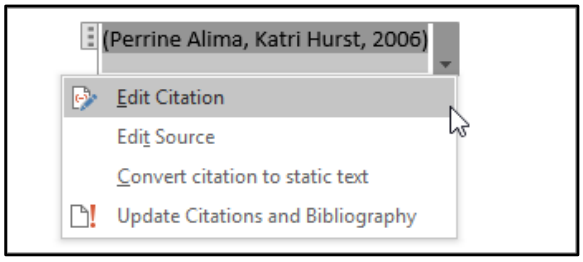
After you've added a source, you may find you need to make changes to it at a later time, following these steps:

1. On the **References** tab, in the **Citations & Bibliography** group, click **Manage Sources**.
2. In the **Source Manager** dialog box, under **Master List** or **Current List**, select the source you want to edit, and then click **Edit**.
3. In the **Edit Source** dialog box, make the changes you want and click **OK**.

8.3.2 Managing In-text Citation Using *MS Word*

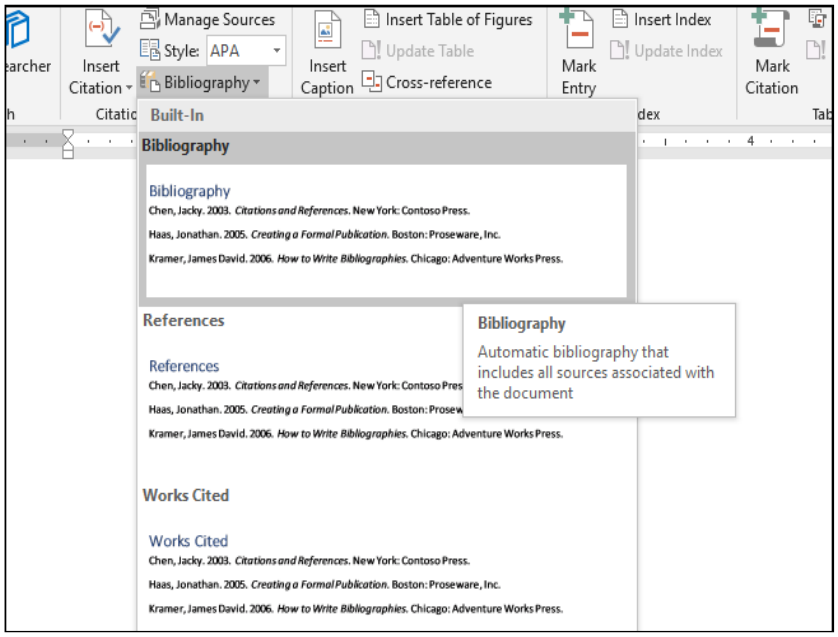
In-text references should immediately follow the title, word, or phrase to which they are directly relevant, rather than appearing at the end of long clauses or sentences. You may as well control what information to use in an in-text citation by either adding or suppressing:

To add details, like page numbers if you're citing a book, select **Citation Options**, and then **Edit Citation**.



8.3.3 Generating & Updating List of References Using *MS Word*

Once you have entered all your citations, you can generate your Reference list at the end of the document. Select **References** from the **Citations & Bibliography** section of the **References** tab. Insert the list of references and make sure you update the list every time you add or make changes to citations.



8.4 Guidelines for Writing In-Text Citation

In-text citation may be presented in the form of quoting, paraphrasing, summarizing and/or secondary referencing.

- **Quoting:** is including a section of a source in your own work using exactly the same words as those used by the original author. If you are directly quoting from a source, then you should include the page number in your citation.

- **Paraphrasing:** is putting someone else's ideas into your own words. It does not mean changing the odd word or rearranging the sentence. When you paraphrase, you should restate the meaning of the original text in your own words. Be sure to cite and reference when you are paraphrasing someone else's work, e.g.:

This this the original quote from Gladwell (2008, p.38)

"Achievement is talent plus preparation. The problem with this view is that the closer psychologists look at the careers of the gifted, the smaller the role innate talent seems to play and the bigger the role preparation seems to play."

Below is an unacceptable paraphrase of the above quote because it follows the original too closely:

Success seems to depend on a combination of talent and preparation. However, when psychologists closely examine the gifted and their careers, they discover that innate talent plays a much smaller role than preparation (Gladwell, 2008, p.38).

The next is an example of an acceptable paraphrase as the meaning of the original has been restated in the author's own words:

As Gladwell (2008, p.38) observes, summarizing studies on the highly successful, we tend to overestimate the role of talent and underestimate that of preparation.

- **Summarizing:** means briefly stating the main ideas or arguments of a complete information source or a substantial portion of an information source. Be sure to cite and reference when you are summarizing someone else's work. A citation for a summary should include the author and date, e.g. (Smith, 2017) or Smith (2017), but there is no need to include a specific page number.

- **Secondary referencing:** means you reference one author who is referring to the work of another and the primary source is not available. **Secondary referencing should be avoided where possible** - if you have only read the later publication you are accepting someone else's opinion and interpretation of the author's original intention. You must make it clear to your reader which author you have read whilst giving

details of the original source by using 'cited in', e.g. (Ecott, 2002, cited in Wilson, 2009) or (Cannon, 1989, quoted in Wilson, 2009, p. 269).

Incorporating in-text citation within your writing may seem tiring task, but following the guidelines below will facilitate your mission:

- When quoting **small amounts** (under two lines), the quotation can be incorporated in the text, without the need to offset it. In this case quotation marks are used to mark the words copied. For example, according to McCurny and Jones (2002) "it is of paramount importance to acknowledge the work of others and to avoid the charge of plagiarism".

- Where it is desired to **quote a chunk around 30 words or more** from the work of another author, the quotation should be introduced by a colon, begun on a new line, and then blocked and indented 1.27cm from the left and right margin. There is no need to use quotation marks. Here is an example:

> If this was an extract from a publication about report-writing that I wanted to include, it would be set off in this way to distinguish it from my own words. When quoting direct from a source, it is essential to provide a full reference, including page number, so that readers can locate the passage if they so desire. (Hunter, 2004, p. 9).

- If the quotation **omits some material** between the source's start and end points, this is indicated by three dots as follows:

> If this were an extract from a book ... it would be set off in this way to distinguish it from my own words. When quoting ... provide a full reference, including page number, so that readers can locate the passage…. (Hunter, 2004, p. 9).

- You may cite the work of others by **referring** to it **without quoting** from it. The next sentence gives an example. According to Adams

(2009) the use of formal methods has little to offer the developer of large-scale systems, however, recent work by Brown and colleagues (Brown, Jones and Green, 2019) suggests that tools are emerging that make the adoption of formal methods in industry a viable prospect.

- Having referred to **co-authors** such as Brown, Jones and Green in full on the first mention of their 2019 publication, they can be referred to on subsequent occasions as Brown et al., (2019).

- If the quotation spans **more than one page**, the form is (Huner, 2004, pp. 9-10).

8.5 Plagiarism Detection Software

Running your document through a plagiarism checker verifies whether your document really is authentically your intellectual property and whether it maintains academic integrity. Naturally, you are allowed to use existing research when writing your graduation project report; this is actually a requirement when writing any scientific paper. However, you have to appropriately identify every single passage you quote.

A plagiarism detector allows you to check if you correctly identified the literature you used. Therefore, if you are not completely certain whether you quoted a text correctly, and if you aspire to verify that you take academic research seriously and that you leave no room for error, you should consider running your document through a plagiarism checker.

To detect and avoid plagiarism in any of your work, then you ought to find the right software. Get yourself something that can match your text with billions of documents present in the web and indicate if there is any

aspect of plagiarism. You will find two options available online: a free software or a fee-based professional checker. These softwares also provide a total percentage of matches, or similarities. You may have already come across the software used by universities – "Turnitin', which provides teachers a percentage of plagiarism when work is scanned. Universities often have an agreed standard of percentage that students need to stick to. Therefore, students can guarantee that the percentage of similarities is below the required number by using a plagiarism checker before the research is delivered.

Printing and submitting a copy of the plagiarism detection report provides as evidence that you are completely honest and that you have taken the necessary precautions to avoid plagiarism. Therefore, **consult your graduation project supervisor about a convenient software to use**.