



African Virtual University

Applied Computer Science: ITI

APPLIED COMPUTER SCIENCE DEGREE PROGRAMME

Dr. Victor Odumuyiwa

Foreword

The African Virtual University (AVU) is proud to participate in increasing access to education in African countries through the production of quality learning materials. We are also proud to contribute to global knowledge as our Open Educational Resources are mostly accessed from outside the African continent.

This module was developed as part of a diploma and degree program in Applied Computer Science, in collaboration with 18 African partner institutions from 16 countries. A total of 156 modules were developed or translated to ensure availability in English, French and Portuguese. These modules have also been made available as open education resources (OER) on oer.avu.org.

On behalf of the African Virtual University and our patron, our partner institutions, the African Development Bank, I invite you to use this module in your institution, for your own education, to share it as widely as possible and to participate actively in the AVU communities of practice of your interest. We are committed to be on the frontline of developing and sharing Open Educational Resources.

The African Virtual University (AVU) is a Pan African Intergovernmental Organization established by charter with the mandate of significantly increasing access to quality higher education and training through the innovative use of information communication technologies. A Charter, establishing the AVU as an Intergovernmental Organization, has been signed so far by nineteen (19) African Governments - Kenya, Senegal, Mauritania, Mali, Cote d'Ivoire, Tanzania, Mozambique, Democratic Republic of Congo, Benin, Ghana, Republic of Guinea, Burkina Faso, Niger, South Sudan, Sudan, The Gambia, Guinea-Bissau, Ethiopia and Cape Verde.

The following institutions participated in the Applied Computer Science Program: (1) Université d'Abomey Calavi in Benin; (2) Université de Ougadougou in Burkina Faso; (3) Université Lumière de Bujumbura in Burundi; (4) Université de Douala in Cameroon; (5) Université de Nouakchott in Mauritania; (6) Université Gaston Berger in Senegal; (7) Université des Sciences, des Techniques et Technologies de Bamako in Mali (8) Ghana Institute of Management and Public Administration; (9) Kwame Nkrumah University of Science and Technology in Ghana; (10) Kenyatta University in Kenya; (11) Egerton University in Kenya; (12) Addis Ababa University in Ethiopia (13) University of Rwanda; (14) University of Dar es Salaam in Tanzania; (15) Université Abdou Moumouni de Niamey in Niger; (16) Université Cheikh Anta Diop in Senegal; (17) Universidade Pedagógica in Mozambique; and (18) The University of the Gambia in The Gambia.

Bakary Diallo

The Rector

African Virtual University

Production Credits

Author

Dr. Victor Odumuyiwa

Peer Reviewer

Rober Oboko

AVU - Academic Coordination

Dr. Marilena Cabral

Overall Coordinator Applied Computer Science Program

Prof Tim Mwololo Waema

Module Coordinator

Instructional Designers

Elizabeth Mbasu

Benta Ochola

Diana Tuel

Media Team

Sidney McGregor

Michal Abigael Koyier

Barry Savala

Mercy Tabi Ojwang

Edwin Kiprono

Josiah Mutsogu

Kelvin Muriithi

Kefa Murimi

Victor Oluoch Otieno

Gerisson Mulongo

Copyright Notice

This document is published under the conditions of the Creative Commons

[http://en.wikipedia.org/wiki/Creative Commons](http://en.wikipedia.org/wiki/Creative_Commons)

Attribution <http://creativecommons.org/licenses/by/2.5/>



Module Template is copyright African Virtual University licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. CC-BY, SA

Supported By



AVU Multinational Project II funded by the African Development Bank.

Table of Contents

Foreword	2
Production Credits	3
Copyright Notice	4
Supported By	4
Course Overview	9
Prerequisites	9
Materials	9
Course Goals	10
Evaluation method	10
Readings and Other Resources.	10
Chapter 1: Introduction	11
1.1 Development Project.	11
1.2 Research Project.	12
Chapter 2: Project Execution	14
2.1 Phase I –Problem Identification, Problem Definition and Literature Review	14
2.1.1 How to choose a topic	14
2.1.2 Problem definition	14
2.1.3 Literature review	15
2.2 Phase II – Analysis, design and implementation	15
2.2.1 Introduction	15
2.2.2 System Development Life Cycle (SDLC)	16
2.2.2.1 Preliminary investigation/analysis	16
2.2.2.2 System analysis	17
2.2.2.3 System design	18
2.2.2.4 System implementation	19
2.3 Phase III – Documentation (Writing-up)	19
2.3.1 Expected content of the documentation	20

2.3.1.1 Preambles	20
2.3.1.2 Chapters	21
2.3.1.3 References	23
Appendix A: Project Outline for a Development Project	24
Appendix B: Project Outline for a Research Project	26
Appendix C: Title Page	29
Appendix D: Declaration	29
Appendix E: Certification.	30
Appendix F: Supervisor's Grading Sheet	30
Appendix G: Examiner's Grading Sheet.	31
Appendix H: Project Proposal Template	32
References.	33

Course Overview

Welcome to Project 2

There is an increasing demand in the industry for graduates with problem solving skills; those with the capability of applying the theories, methods and techniques learnt during their degree programme in solving real life problems. The goal behind applied computer science programme is to produce such graduates who can easily integrate into the industry immediately after college and become indispensable problem solvers. Another goal of the applied computer science programme is to produce entrepreneurs who can create opportunities and bring about innovative solutions. Knowing fully well the demand of the industry, Project 2 has been designed to help students consolidate the knowledge and skills acquired during the course of the degree programme by developing systems (hardware or software) that are applicable to the market. Such systems as a matter of fact should contribute by addressing gaps in the market and should mainly target areas that need solutions unique to the African region. This is not in any way to limit students' horizon; they should think globally but be able also to customize their solutions to the region.

Students will be expected to choose topics from the industry and academia that will target offering a solution to a problem in the African region. Each project title should be approved by the project coordinator and supervised by a member of the teaching staff. The project will run in two semesters. Students may attend seminars on contemporary issues and problems in their project areas. They would be expected to produce all necessary documentation (Proposal, Systems requirements document, Systems Design Document, Test plan and the User Manuals). They would also be expected to give oral presentations of their projects at the end of each semester in order to demonstrate their progress. The deliverables are the developed system and its related documentation.

Prerequisites

Full coverage of Applied Computer Sciences compulsory courses.

Materials

The materials required to complete this course are:

- Computer system
- Internet connection
- Relevant compilers and interpreters

Course Goals

Upon completion of this course, the learner should be able to:

- Design and Implement an information system using the basic concepts learnt throughout the programme
- Demonstrate skills and ability to work with a consultation team to deliver a product as per the clients expectations
- Apply suitable existing technologies to solve a unique African problem.
- Apply technical writing in ensuring correct documentation.
- Present the developed system to a panel for evaluation

Evaluation method

Your final year project shall be evaluated in three phases:

- You will be required to make a presentation of your project proposal to the project coordinator. This evaluation is non-graded but will be used to determine if a topic should be approved or not.
- Continuous assessment will be done by your supervisor through periodic consultations and at the end of the project, a cumulative of the assessment shall be submitted following the template shown in Appendix F. This assessment shall constitute 50% of the total project mark.
- At the end of the session, you will be required to make a presentation before a panel of departmental staff. This presentation shall be graded following the template shown in Appendix G and will constitute 50% of the total mark.

Readings and Other Resources

The readings and other resources in this course are:

Required readings and other resources:

- Dennis A., Wixom B.H. and Roth R.M. (2012), System Analysis and Design, 5th edition, Wiley, USA. Chapters 1,2,3 and 4
- Fowler M. (2004), UML distilled: a brief guide to the standard object modelling language, 3rd edition, Addison-Wesley, Boston, USA.
- Vernon D. (2007), Final year project hand book – Computer engineering students. http://www.vernon.eu/courses/David_Vernon_Project_Manual.pdf (Retrieved on April 13, 2016)
- <http://www.cs.nuim.ie/sites/default/files/FYProjectHandbook-2012-2013-final.pdf>

Chapter 1: Introduction

Your final year project is one of the most important aspects of your degree because it provides you an opportunity to apply the knowledge and skills acquired during the course of your study in solving a problem. It also pushes you to research for and acquire more domain specific knowledge with respect to the problem given. Your final year project allows you to demonstrate your ability to abstract and to combine theories, methods and techniques from different courses. The project should span through the last two semesters of the degree programme.

The complexity of the problem chosen should be of a dimension sufficient enough to prepare a student for the kind of challenges and problems that would be encountered in the workplace after college. Working as a software developer in a company, you will not be asked to explain what programming is or to differentiate between two data structures. You will however be faced with the challenge of designing or implementing a software solution to a given problem. The solution will involve choosing the right data structure to use; it might involve you designing and creating a database or just accessing existing databases; and it might also involve you comparing two different solutions and then coming up with the most effective and efficient.

At this juncture, it is important to highlight the fact that a final year project could either be a research project or a development project. A research project is aimed at solving a research-oriented problem while a development project aims at developing a system that meets users' needs.

1.1 Development Project

A development project is oriented towards creating a system that satisfies a set of requirements. A development project could take different forms such as an application, an information system, a software library, an embedded system, a plugin, an extension, a modification to an existing system, a software protocol etc. Development projects are the most common type of final year projects done by undergraduate students.

This kind of project focusses on applying software engineering principles in developing a quality software solution to a given problem coupled with a thorough evaluation of the developed system. You need to do a proper study of the problem and review relevant literature relating to your problem domain.

Summarily, the methodology for executing a development project consists of the following steps/phases:

- Proper capturing and understanding of the requirements
- Analysis of the problem and the existing system(s)
- Designing a solution to the problem

- Implementation of a system that actualises the designed solution
- Verification of the system to ascertain that it matches the design
- Validation of the system to ascertain that it correctly implemented the solution
- Evaluation of the system to determine how well the solution solves the problem
- Documentation

1.2 Research Project

As mentioned earlier, a research project aims at solving a research-oriented problem. In a research project: you could choose to evaluate the effectiveness of existing solutions to a problem; you might as well choose to compare two or more existing solutions; it might also involve you modifying a solution or coming up with a new solution to a given problem. The focus is on the experimental procedure followed and the thoroughness of the evaluation of the result.

A research project generally starts with the formulation of a hypothesis, followed by the design and implementation of a solution and a series of experiments in order to validate the hypothesis. Intensive literature review is needed in a research project in order to gain a good understanding of the domain. For example, given this research topic “comparative analysis of Naïve Bayes and Support Vector Machine for sentiment analysis of news comments”, your hypothesis could be “Support Vector Machine should outperform Naïve Bayes classifier in terms of speed and accuracy in analysing expressed sentiments in news comments”. For the above topic, a student will have to first of all study the literature to understand the problem domain which is sentiment analysis of news comments. This will be followed by a study on machine learning in order to understand what classification tasks are and the techniques involved. Not only that, a thorough literature review about the two classification techniques mentioned in the topic would also be conducted. The next thing will be to implement the two classification techniques and pre-process news comments that will be used in the experimentation. Once the software and the data are ready, a series of experiments should be carried out and the result of the experimentation analysed (generally statistically). The result of the analysis should be discussed and the hypothesis should be revisited based on the result for confirmation or refutation.

Generally the methodology for experimental research in computer science consists of the following steps:

- Formulation of hypothesis
- Identification and selection of performance measurement criteria
- Design and implementation of a solution to the problem
- Design and conduction of experiments and collection of results
- Analysing resulting data from experiments
- Interpretation and explanation of results
- Confirmation or refutation of the hypothesis based on the results obtained.

- Documentation

It is important to note that your first goal is to solve the problem. The next objective is to implement the solution. This should be followed by experimentation in order to evaluate how well your solution performs. The solution is the main output of your work and the implementation is the tool you will be using for experimentation. Before your experimentation, you should ensure that your software is well tested so as to ascertain that it correctly implements your solution. The experimentation only serves the purpose of measuring the effectiveness and efficiency of your solution. It is however important to note that in a research oriented project, you can choose to adopt an existing implementation of a solution and tweak it to reflect your own solution to the problem and to suit the experiments to be carried out. On the other hand, in a development oriented project, the emphasis is more on the development/implementation of the proposed system with a higher scope of software development work.

Proper time management is a factor in successfully carrying out a research project. You must ensure you allocate time for each of the stages of the project. If you successfully developed a solution without implementing it, your project is far from being completed. If you implement without experimentation, your solution's efficiency and effectiveness will not be validated.

Chapter 2: Project Execution

There are three main stages involved in executing a final year project. The first stage involves identifying the problem to solve and the concepts related to the problem. It involves proper studying of relevant literature. The second stage is the implementation and testing stage which entails designing a solution to the problem and implementing the solution in form of a system (mainly software). During this stage, proper testing of the software should be done and if it is a research-oriented project, experiments should be carried out during this stage. The third stage is the documentation stage which entails reporting the process followed in solving the problem as well as the solution and result gotten.

2.1 Phase I –Problem Identification, Problem Definition and Literature Review

2.1.1 How to choose a topic

Generally, a student is allowed to pick a topic of his/her choice but the topic should be moderated by the advisor/supervisor. In some cases your supervisor will choose a topic for you. The choice of topic should be affected by the following: the curriculum for the program showing the subject areas a student should have been exposed to during the course of the degree programme; the programming tools the student is already familiar with; the time duration expected for the project to be completed; and the industry expectation of a degree holder in Computer Science. The complexity of the project should be determined by those four factors.

Once you have been assigned a supervisor, you should identify two to three different problems you are interested in solving and then present them to your supervisor who should guide you in choosing one out of the three. Once a problem to solve has been identified, it should be formulated into a topic and a project proposal should be written following the template provided in APPENDIX H and submitted to the departmental project coordinator.

2.1.2 Problem definition

A problem identified does not automatically imply that the problem is well defined. After identifying a problem, you should go ahead with the help of your supervisor to define the problem. Problem definition entails asking the right questions about the problem, identifying the sub-problems in the problem. A well-defined problem will lead to a well-designed solution. You need to spend enough time to think through the problem so as to ensure that it is well defined. When you are defining the problem, you should also set the boundaries and give a scope for your work.

2.1.3 Literature review

After defining your problem, you cannot just start coding immediately. Remember that this is a project and not just a programming assignment. Once a topic has been formulated and the problem defined, the next thing is to do an extensive study of the literature to fully understand the theories and concepts associated with the chosen topic. Literature review entails exploring existing solutions to the problem you are about to solve in case there are any. It is a very important aspect because it reveals your mastery of the theoretical and conceptual framework upon which your work is based. Every concept mentioned in your topic should be effectively explored during the literature review. The following sources of information could be consulted for your literature review:

- Books around the subject area
- Periodicals such as articles in academic journals and conference proceedings
- Web resources such as web pages, wikis and online libraries.

You should be careful in using web resources because there is no guarantee of quality control of their content. While books and academic articles go through various review processes before they are published, unfortunately this cannot be said of online resources like web pages and wikis.

It is important to know that your literature review might propel you to modify your topic or redefine your problem. This is because of increase in your domain knowledge as you search through the literature. The more knowledgeable you are in a domain, the more informed you are in formulating a problem and defining such in the domain.

You should endeavour to keep a log of materials you consult and their sources. Summarizing the content of the materials will be of help during the documentation stage. Your summary should be a synthesis of the material. Avoid lifting of sentences or paragraphs from the materials. You should rather express your understanding of the materials in your own words. You should not just swallow everything you see in the literature, your ability to critique the contents expressed by other authors will go a long way to show your depth of understanding and mastery of the domain.

2.2 Phase II – Analysis, design and implementation

2.2.1 Introduction

This is the second phase of the project. This is a critical phase in the execution of your project. It is referred to as the donkey-work phase. Once you are comfortable to a level with the domain knowledge, your next step is to analyse the system you intend to create.

For a development project, during this phase, you will first analyse the existing systems (may be a manual system) and document requirements for the new system you are about to create. Once thorough system analysis has been done, you will go ahead to design the system you

intend to create and after which you implement the design by writing codes for the business logic and creating a database for the data storage layer. System verification, validation and evaluation are also carried out during this phase.

For a research-oriented project, this phase covers the design and implementation of a solution to the problem given. All other steps involved in a research-oriented project such as the design and conduction of experiments, collection and analysis of resulting data from experiments, interpretation and explanation of results and confirmation or refutation of the hypothesis based on the results obtained are all carried out during this phase.

2.2.2 System Development Life Cycle (SDLC)

Developing a system can be likened to the process of building a house or a car. Taking the building of a house for example, the owner will first describe his/her vision or imagination of the house to the architect. The architect will now come up with a sketch to represent the description made by the owner. S/he tries to ascertain if the description captures the requirement of the owner. There could be a series of interactions between the owner and the architect and several iterations of the sketch until the owner agrees that his/her vision has been properly captured. The architect then goes ahead to do a proper design in the form of a plan for the house which will be implemented by the building engineer. During the building process, there are probabilities that the owner can come up with some changes and new decisions as per the features and look of the house. All these changes are also considered and integrated as the engineer builds the house. The same process is followed in a software development project (Dennis, Wixom & Roth, 2012).

In order to effectively execute a development project, you must understand the different phases of the system development life cycle. System development life cycle describes a process for planning, designing, developing, testing and deploying an information system. It is composed of five phases which are: preliminary investigation, system analysis, system design, system implementation and system maintenance. In the context of a final year degree project, we shall limit ourselves to the first four phases.

2.2.2.1 Preliminary investigation/analysis

This phase is needed to study and observe the work environment and understand the objectives and goal of an organization. It involves studying the business process of an organization in order to understand what her needs are and the reasons for the intended automation of a business process. During this phase, interviews are carried out, ethnographic observations are performed and questionnaires could also be administered. This phase is needed to understand the problem at hand, determine if there is already a solution, propose alternative solutions, produce feasibility report and submit a preliminary plan with recommendations. In the case of a final year degree project, you might be given a topic which requires that you choose a case study; if this is the case, you would have to follow the process above with regard to the organization chosen as a case study in order to properly identify the

requirements. If on the other hand, you do not have a case study, you might have to critically think out the requirements or consult your colleagues and your supervisor by describing the kind of system you intend to build to them and then solicit their help (as prospective users) in telling you the kind of expectations they would have with regard to the functionalities and features of the system. You can then analyse their responses and combine these with knowledge garnered from the literature to come up with clearly stated requirements for the system.

2.2.2.2 System analysis

This phase entails analysing and defining the requirements. During this phase, data collection and analysis are carried out to effectively identify the problems with the existing systems and to outline the features expected of the new system. During this phase, you are expected to come up with use cases for the new system. This entails identifying the various categories of prospective users of the system. For each category, you must define the functionalities that the system must offer for the users in that category in order for them to effectively carry out their work using the system. System analysis can be done either by using the structured approach or the object-oriented approach.

The structured approach

In the structured approach, use case diagrams are used to present the different use cases for a system. The next step is to perform process modelling. This entails modelling the business process for which the information system is to be created. You will need to create a context diagram for your system. You will also need to come up with data flow diagrams to show the different processes and entities that make up the system and how data flows among them. Another task to be performed under system analysis is the development of the conceptual data model for the proposed information system. If you intend to use a relational database management system, this entails creating the entity-relationship (ER) model for your data layer.

The object-oriented approach

In the object-oriented approach, UML diagrams are used for analysis and design. According to the creators of UML, any object-oriented approach to developing a system must be use case driven, architecture centric, and iterative and incremental. Use case driven implies that functional requirements and behaviour of the system are defined using use case diagrams. Architecture centric implies that the architecture of the system should determine and stimulate the specification, construction and documentation of the system. Three architectural view of a system are needed: functional view, static view and dynamic view. While the functional view captures the external behaviour of the system based on the perspectives of the user, the static view on the other hand focusses on describing the structure of the system in terms of attributes, methods, classes, relationships and messages. The third view, which is the dynamic view, describes the internal behaviour of the system by showing the interaction between the

components of the system; it describes the communication that happens internally within the system and state changes within an object. System development being incremental and iterative means that the whole development process involves creating and testing several iterations of the system with each iteration bringing the system closer and closer to the final product which should meet all the user requirements and needs (Dennis, Wixom & Roth, 2012).

UML 2.0 defines fourteen different diagramming techniques that could be used to model a system. These diagrams can be broken into two major groups: structure diagrams for representing data and static relationships in a system and behaviour diagrams for representing the dynamic relationships among objects in a system. Unlike the structured approach where you can classify each diagram as being used either in the analysis or design phase, most UML diagrams can be used in both the analysis and design phases except the use case diagram which is only applicable during the analysis phase. It is therefore up to you to decide on how you will use these diagrams in your project. However, I will make some propositions which I think should be able to help you in executing your project.

It is advisable that you include an activity diagram in your analysis phase in order to illustrate the flow of activities in a use case. You should also use a less developed class diagram for creating a conceptual model of the system. During the design phase, you should create well detailed class diagrams. You can as well include object diagrams. A sequence diagram will be of interest in the design phase for modelling the behaviour of objects within a use case. Based on the nature of the system you are developing, you can as well include a communication diagram and behaviour state machine diagram in your design.

2.2.2.3 System design

After a thorough analysis has been carried out and the requirements have been well defined, the system should be designed. This phase “decides how the system will operate in terms of the hardware, software and network infrastructure that will be in place; the user interface, forms, and reports that will be used; and the specific programs, databases and files that will be needed” (Dennis, Wixom & Roth, 2012). The design phase determines what the system will look like and how it will operate. During this phase, you are to develop flow charts and algorithms for the processes in the system. After this, you should design the user interface (your knowledge of human computer interaction should come into play here). The user interface should be designed with the goal of creating a rich user experience. Design prototypes should be created and iterated over using prospective users until a prototype is agreed upon by the users as the best interface for ensuring rich user experience. If you are following a structured approach, your entity-relationship diagram (ERD) should be converted into a relational model (which is the logical data model) during the design phase. For an object-oriented approach, well detailed class diagrams, object diagrams, sequence diagram communication diagram and behaviour state machine diagram should be used in your design stage depending on the complexity of the system.

2.2.2.4 System implementation

This is the longest phase of the SDLC. It entails the construction of the system based on the design created in the previous phase. It entails creating the database on a chosen database management system, creating the user interfaces with the look and feel, programming the business logic of the application and integrating all the components of the information system together. This phase also entails proper testing of the application. Software testing is done to verify and validate the application. While verification ensures that the system has been correctly built by ensuring that the system is the correct implementation of the designed solution, validation on the other hand ensures that you “have built the correct system” by testing to ensure that the system actually solves the original problem for which it was designed. There are three types of test that should be of interest during the implementation phase: unit tests, integration tests and system tests.

Unit tests are carried out on individual unit (i.e. a method or a programme module that performs a specific function) of the system in order to ascertain that the unit performs its function as defined in the specification. There are two approaches that could be followed in carrying out unit tests: black-box testing and the white-box testing. Black-box testing is done by creating several test cases based on specifications that covers the input space for the different units of the system. The black-box testing is the most widely used approach in unit testing. White box testing is more of code review.

Integration tests are done to ascertain that the different program modules are working together without error. While unit tests certify individual units or modules of an application, integration tests, on the other hand, certify whether or not the different modules are working together correctly. The four approaches to performing integration tests include: user interface testing, use scenario testing, data flow testing, and system interface testing.

System tests are performed to assess how well the system meets the system requirements defined in early stages of the SDLC. System testing involves doing requirement testing, usability testing, security testing, performance testing and documentation testing.

2.3 Phase III – Documentation (Writing-up)

What most people will end up seeing as regarding your project is the documentation. What is kept in the library for future references is the documentation. What your examiners will grade, to a large extent, is the documentation. Documentation is a very important stage of a final year project execution. Your work is as good as your communication of it. If you did a good work and made a poor documentation, the work could be poorly graded. On the other hand if you did a poor work and made a good documentation, the work could be graded well above average. It therefore means that the project documentation is as important as the project design and implementation. In fact, you should allocate about one-third of the project period to documentation.

Documentation entails developing a story line around the work done. It is about narrating the problem you solved, why you solved it, how you solved it, and the results you got. As you have learnt that system analysis and design precede implementation so it is in documentation; you should not just jump into writing, you need to plan and structure your documentation. Structuring entails creating an outline you are going to follow in writing-up. A sample outline for a development project is shown in Appendix A while that of a research-oriented project is shown in Appendix B. These sample outlines are just a form of guide and should be adapted depending on the nature of the project.

Plagiarism is a very serious offence. Copying and pasting from websites or books and articles is not allowed. Your report must be original and void of plagiarism. Remember that you are to provide synthesis of consulted literature and not doing a direct copying of content of the literature. However, if you are so pleased with the way an idea is expressed in the literature and you feel that rewriting such in your own words will not make the same impact as the original, then you can do a direct quotation of such content and reference the author.

2.3.1 Expected content of the documentation

2.3.1.1 Preambles

In the documentation, you need to include some preambles such as:

Title page: A sample format is provided in Appendix C.

Declaration: You should declare on this page that the project was carried out by you and that the content originated from you, meaning that you did not plagiarise. You must sign this page. The page should be formatted as shown in Appendix D.

Certification: This page is to be signed by the supervisor/advisor indicating that the project was carried out by the student. A sample format is provided in Appendix E.S

Dedication: This page is optional. You might choose to dedicate your work to someone or an organization or a nation or anything you like.

Acknowledgement: This page allows you to show your gratitude to those who assisted you in one way or the other. You might choose to acknowledge your parents, siblings, friends, professors, supervisors etc.

Table of content: This should contain the list of chapters and top-level sections with the corresponding page numbers. It is advised that you generate this automatically.

List of figures: All the figures in the documentation and the pages where they appear should be listed on this page.

List of tables: All the tables in the documentation and the pages where they appear should be listed on this page.

Abstract: This page should describe in not more than 300 words, the What, Why and How of the project. You should summarise the whole project by telling us about the problem being solved, the motivation and the reason for solving the problem, the technique you used in solving the problem, and the results you got.

2.3.1.2 Chapters

Chapter One - Introduction

This chapter should provide a general overview of the project. It should introduce the project by stating the problem to be solved and giving a justification or motivation for solving the problem. In this chapter, you should also tell us the aim of the project and highlight the objectives to be achieved in the project. It is also very important to define the scope of work in this chapter. Tell us the boundaries you are setting for your work. For a research project, your hypothesis should also be presented in chapter one.

Chapter Two – Literature Review

This chapter centres on the background study. You should treat all the concepts associated with your topic. You are also to present the “state-of-the-art” of product and systems related to your work. In this chapter, you should highlight the performance measurement criteria for assessing systems in the domain. If there are no standard criteria, you might have to come up with some criteria. The objective is to compare the existing systems based on the criteria identified. Remember that this chapter should be a synthesis and not a direct copy from the literatures. You should also carefully structure this chapter into subsections by creating a flow that drives the story home. You are not obliged to use the words “Literature Review” as the title of this chapter. It is advised that you formulate a title in your own words that reflects the content of the chapter.

Chapter Three – System Analysis and Design (for a development project)

This is the chapter that documents most of the work done in phase II of the project execution. You should give a brief introduction of this chapter before you start developing the subsections. You should explain the software development methodology you used in the developmental stage of the project. All the outputs of the system analysis activities should be presented. Functional and non-functional requirements of the system should be highlighted. Proper analysis of the requirements should be discussed. Use cases developed should be presented and all the stages of the process modelling should be properly documented in this chapter. For an object-oriented approach, all UML diagrams used such as class diagrams, sequence diagrams, object diagrams etc. should be documented in this chapter. All the various designs for the system such as flowcharts, algorithms, user interface design and logical data model design should be presented and clearly explained.

Chapter Three –System Analysis and Design (for a research-oriented project)

In a research-oriented project, your chapter three should present the steps you followed in carrying out the research. You should start by a brief introduction followed by a section on problem modelling which should show how you broke down the problem into smaller problems representing tasks to be performed. Inputs and outputs of each sub-problem should be highlighted and the solution you designed for the problem should be discussed. You will also need to show how you analysed and designed your software. You might use different structured systems analysis and design or UML diagrams to explain this.

Chapter Four – Implementation

This chapter should focus on how you implemented your design. You need to discuss your choice of tools, programming language, libraries and platforms used and the reason for the choice. You might run through the developed system components explaining how they work by showing screen shots of relevant interfaces and indicating how the system satisfies the user requirements e.g. by showing how each task is performed on the system (this could also be included as an Appendix). The way the system was tested should be reported. Discuss how you selected your test cases and test data and the rationale behind your selection method. Provide a summary of test results showing that your software passed the test. Using the metrics you presented in chapter two, you should evaluate the performance of your system and compare it with the existing systems.

Chapter Five – Evaluation (this chapter only applies to a research-oriented project)

In this chapter, you should highlight the metrics you used to evaluate your software. The experimental procedure for each metric and measurement method should also be described. This should be followed by a description of the inputs (dataset) for each experiment. Summary of your results should be presented as well as the statistical or other techniques you used in deducing your results. A comparison of your results with other systems could be reported where possible. You should discuss your findings and fall back to your hypothesis for confirmation or refutation.

Chapter Five – Conclusion and Recommendation (This would be chapter six for a research-oriented project)

This chapter provides a closure for the whole project documentation. It summarizes the project by providing concluding remarks, highlighting encountered problems during the project execution and providing some recommendations that could be of help to anyone that chooses to improve on the project work.

2.3.1.3 References

Nobody is an island of knowledge. You need to stand on the shoulders of others in order to see far. This section should contain a list of materials you consulted and cited in the documentation. Web references should be included as well. You should use the APA referencing style.

Appendix A: Project Outline for a Development Project

Title page

Certification

Dedication

Acknowledgement

Table of Content

List of figures

List of tables

Abstract

Chapter one – Introduction

- 1.1 Overview
- 1.2 Problem Definition
- 1.3 Justification
- 1.4 Aim and Objectives
- 1.5 Scope of Study
- 1.6 Methodology
- 1.7 Arrangement of Dissertation

Chapter Two – Literature review

All concepts in the title should be looked into and related works should be reviewed

Performance measurement criteria should be developed

Existing systems should be compared based on criteria developed

Chapter Three – System Analysis and Design (use this if you followed a structured analysis and design approach)

- 3.1 Introduction
- 3.2 Software Development Methodology
- 3.3 System Analysis
 - 3.3.1 Requirement Analysis
 - 3.3.2 Use Case Analysis
 - 3.3.3 Process Modelling

- 3.3.3.1 Context diagram
- 3.3.3.2 Data Flow Diagram Fragments
- 3.3.3.3 Level 0 Data Flow Diagram
- 3.3.3.4 Level 1 Data Flow Diagram
- 3.3.4 Data Modelling (ER Model)
- 3.4 System Design
 - 3.4.1 Architecture Design
 - 3.4.2 User Interface Design
 - 3.4.2.1 Use scenario development
 - 3.4.2.2 Interface structure design
 - 3.4.2.3 Design prototypes
 - 3.4.3 Data Storage Design (Conversion of ERD to Physical Data Model)
- 3.5 Conclusion

Chapter Three – System Analysis and Design (use this if you followed an object-oriented analysis and design)

- 3.1 Introduction
- 3.2 Software Development Methodology
- 3.3 System Analysis
 - 3.3.1 Requirement Analysis
 - 3.3.2 Use Case Analysis
 - 3.3.3 Activity Diagram
 - 3.3.4 Class Diagram
- 3.4 System Design
 - 3.4.1 Sequence Diagram
 - 3.4.2 Detailed Class Diagrams
 - 3.4.3 Communication Diagram
 - 3.4.4 Behaviour State Diagram
 - 3.4.5 User Interface Design
 - 3.4.5.1 Use Scenario Development

3.4.5.2 Interface Structure Design

3.4.5.3 Design Prototypes

3.5 Conclusion

Chapter Four – Implementation

4.1 Introduction

4.2 System Requirements

4.3 Choice of Implementation tools

4.4 Software Components (Run through the components of the software displaying user interfaces)

4.5 Testing

4.7 Evaluation (Comparison with existing systems)

4.8 Conclusion

Chapter Five – Conclusion and Recommendation

5.1 Conclusion

5.2 Problems Encountered

5.3 Recommendation

References

Web References

Appendix

Appendix B: Project Outline for a Research Project

Title page

Certification

Dedication

Acknowledgement

Table of Content

List of figures

List of tables

Abstract

Chapter one – Introduction

- 1.1 Overview
- 1.2 Problem Definition
- 1.3 Research Objectives
- 1.4 Justification
- 1.5 Scope of Study
- 1.6 Research Methodology
- 1.7 Thesis Arrangement

Chapter Two – Literature review

All concepts in the title should be looked into and related works should be reviewed

Performance measurement criteria should be developed

Existing systems should be compared based on criteria developed

Chapter Three – Analysis and Design

- 3.1 Introduction
- 3.2 Requirements Analysis
- 3.3 Analysis
- 3.4 Design
- 3.5 Conclusion

Chapter Four – Implementation

- 4.1 Introduction
- 4.2 System Requirements
- 4.3 Choice of Implementation Tools
- 4.4 Software Components (Run through the components of the software displaying user interfaces)

4.5 Testing

4.5.1 Verification

4.5.2 Validation

4.6 Conclusion

Chapter Five – Evaluation

5.1 Introduction

5.2 Evaluation metrics

5.3 Experimental setup

5.4 The Experiments

5.5 Results

5.6 Discussion of Findings

5.7 Conclusion

Chapter Six – Conclusion

6.1 Summary

6.2 Problems Encountered

6.3 Contribution to Knowledge (if any)

6.4 Future Work

References

Web References

Appendix

Appendix C: Title Page

PROJECT TITLE
By
Your full name
Submitted to
Department of Computer Science
African Virtual University
In partial fulfilment of the requirements for the award of Bachelor Degree in Computer Science.
Supervisor: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Month, Year

Appendix D: Declaration

DECLARATION

I hereby declare that this project, which I now submit for assessment on the program of study for the award of Bachelor Degree in Computer Science, is an original work carried out by me and has not been taken from the work of others except in cases where such work has been properly cited and referenced.

Signature.....

Date.....

Appendix E: Certification

CERTIFICATION

This is to certify that this project work submitted to the Department of Computer Science, African Virtual University is an original work carried out by [the student's name] under my direct supervision.

Name of Supervisor: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Signature.....

Date.....

Appendix F: Supervisor's Grading Sheet

SUPERVISOR'S GRADING SHEET			
Student's Name			
Project Title			
		Total Awardable Mark	Mark's Awarded
	Abstract	5	
	Introduction	5	
	Literature review (Synthesis)	10	
	Methodology	10	
	Analysis	15	
	Design	15	
	Application (Software)	20	
	Evaluation	10	
	Conclusion	5	
	References	5	
	Total	100	

Supervisor's Comment	
Supervisor's Name	
Supervisor's Signature	
Date	

Appendix G: Examiner's Grading Sheet

EXAMINER'S GRADING SHEET			
Student's Name			
Project Title			
		Total Awardable Mark	Mark's Awarded
Documentation (60%)	Abstract	5	
	Introduction	5	
	Literature review (Synthesis)	5	
	Methodology	5	
	Analysis	10	
	Design	10	
	Application (Software)	10	
	Evaluation	5	
	Conclusion	5	
	References	5	

Viva (40%)	Presentation	15	
	Depth of understanding	15	
	Question and Answers	10	
	Total	100	
Examiner's Comment			
Examiner's Name			
Examiner's Signature			
Date			

Appendix H: Project Proposal Template

FINAL YEAR PROJECT PROPOSAL	
Student's Name	
Student's Number	
Supervisor's Name	
Title	State your project topic and let it reflect the problem you are trying to solve
Project description	In not more than 300 words, describe the proposed project and explain how relevant it is to computer science.
Approach	Describe the approach you will follow in executing your project and provide an overview of the main activities. You should provide a plan indicating work schedules, expected output at each stage and performance indicators for progress evaluation. You can use a Gantt chart or any other relevant chart to provide detailed timetable of work.
References	Provide some relevant references preferably from books or academic publications (conference/journal papers).
Second reader's opinion	This should be left blank.

References

- Dennis A., Wixom B.H. and Roth R.M. (2012), System Analysis and Design, 5th edition, Wiley, USA. Chapters 1, 2, 3 and 4.
- Fowler M. (2004), UML distilled: a brief guide to the standard object modelling language, 3rd edition, Addison-Wesley, Boston, USA.
- Vernon D. (2007), Final year project hand book – Computer engineering students. http://www.vernon.eu/courses/David_Vernon_Project_Manual.pdf (Retrieved on April 13, 2016).
- <http://www.cs.nuim.ie/sites/default/files/FYProjectHandbook-2012-2013-final.pdf> (Retrieved on April 13, 2016).

The African Virtual University Headquarters

Cape Office Park

Ring Road Kilimani

PO Box 25405-00603

Nairobi, Kenya

Tel: +254 20 25283333

contact@avu.org

oyer@avu.org

The African Virtual University Regional Office
in Dakar

Université Virtuelle Africaine

Bureau Régional de l'Afrique de l'Ouest

Sicap Liberté VI Extension

Villa No.8 VDN

B.P. 50609 Dakar, Sénégal

Tel: +221 338670324

bureauregional@avu.org