**DWIT COLLEGE**

**DEERWALK INSTITUTE OF TECHNOLOGY**

****

**TITLE OF PROJECT REPORT [ALL CAPS FONT 16]**

**A MINI PROJECT REPORT**

**Submitted to**

**Department of Computer Science**

**DWIT College**

Submitted by

[Name of the candidate]

[Date]

**DWIT College**

**DEERWALK INSTITUTE OF TECHNOLOGY**

**Supervisor’s Recommendation**

I hereby recommend that this project prepared under my supervision by NAME OF THE STUDENT [ALL CAPITAL BUT NOT BOLD] entitled **“TITLE OF THE PROJ [ALL CAPITAL BOLD]”** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

…………………………………………

[Supervisor Name]

[Designation]

Deerwalk Institute of Technology

DWIT College

**DWIT College**

**DEERWALK INSTITUTE OF TECHNOLOGY**

**Student’s Declaration**

I hereby declare that I am the only author of this work and that no sources other than that listed here have been used in this work.

…………………………………………

[Student Name]

[Date:-]

**DWIT College**

**DEERWALK INSTITUTE OF TECHNOLOGY**

**LETTER OF APPROVAL**

This is to certify that this project prepared by NAME OF THE STUDENT [ALL CAPITAL] entitled **“TITLE OF THE PROJECT [ALL CAPITAL BOLD]”** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| ……………………………………  [Supervisor] | …………………………………………..  [Examiner] |

**ACKNOWLEDGEMENT**

The student can write an acknowledgement in order to thank all the people who directly or indirectly helped to complete the project. In a scientific writing, acknowledgement is an expression of a gratitude for assistance in a work. Receiving credit by way of acknowledgement rather than authorship indicates that the person or organization did not have a direct hand in producing the work in question, but may have contributed funding, suggestions. criticism, or encouragement to the author(s).

[Student Name]

[Roll No.:-]

[Date:- ]

**ABSTRACT**

An abstract is like a movie trailer. It offers a preview, highlights key points, and helps the audience decide whether to view the entire work or not. The parts of an Abstract are as follows:

1. Motivation/problem statement: Why do we care about the problem? What practical, scientific, theoretical or artistic gap is your research filling?
2. Methods/procedure/approach: What did you actually do to get your results? (e.g.. analyzed many literatures, designed and developed a software product)
3. Results/findings/product: As a result of completing the above procedure, what did you learn/invent/create? (e.g., after testing your project, or comparing various algorithms, what did you conclude or find or realize?)
4. Conclusion/implications: What are the larger implications of your findings, especially for the problem/gap identified in step 1? (e.g., where and your project can be used or be useful?)

**Keywords**: *[Choose 4-8 words that best define your work separated by semi-colon]*

**TABLE OF CONTENTS**

**LIST OF FIGURES**

Figure 1: Outline of Document.............................................................................................. 4

Figure 2: Genetic Algorithm.................................................................................................. 8

Figure 3: Class Diagram........................................................................................................ 9

**LIST OF TABLES**

Table 1: Outline of Document............................................................................................... 4

Table 2: Genetic Algorithm................................................................................................... 8

Table 3: Class Diagram.......................................................................................................... 9

**LIST OF ABBREVATIONS**

PHP Hypertext Preprocessor

HTML Hypertext Markup Language

CSS Cascading Style Sheet

**CHAPTER 1: INTRODUCTION**

**1.1. OVERVIEW**

The introduction section has two main purposes: 1) to give an overview of the main points of your report, and 2) to awaken the reader’s interest. It is recommended to rewrite the introduction one last time when the writing is done, to ensure that it connects well with your conclusion. The introduction should include:

* The background for your choice of theme
* A discussion of problem statement
* Objectives of your project
* Scope of your work or project
* A schematic outline of the remainder of your project report

**1.2. BACKGROUND AND MOTIVATION**

The background sets the general tone for your project work. It should make a good impression and convince the reader why the theme is important and your approach relevant. Even so, it should be no longer than necessary.

What is considered a relevant background depends on your field and its traditions. Background information might be historical in nature, or it might refer to previous research or practical considerations. You can also focus on a specific text, thinker or problem.

Academic writing often means having a discussion with yourself (or some imagined opponent). To open your discussion, there are several options available. You may, for example:

* refer to a contemporary event
* outline a specific problem; a case study or an example
* review the relevant research/literature to demonstrate the need for this particular type of project work

If it is common in your discipline to reflect upon your experiences as a practitioner, this is the place to present them. In the remainder of your report, this kind of information should be avoided, particularly if it has not been collected systematically.

**1.3. PROBLEM STATEMENT**

A problem statement' is a short description of the issues that need to be addressed by a problem solving team and should be presented to them (or created by them) before they try to solve a problem. The statement of the problem should briefly address the question: "*What is the problem that the research or project will address?"*  Problem statements often have three elements:

* the problem itself, stated clearly and with enough contextual detail to establish why it is important (i.e., a clear statement that the problem exists; evidence that supports the existence of the problem)
* the method of solving the problem, often stated as a claim
* the purpose, statement of objective

**1.4. OBJECTIVE**

Objective is a claim of one or two sentences in length that outlines the problem addressed by a study. It can be formulated as one or two main statement(s) with (a few) more specific sub-statements or in the form of a hypothesis that will be tested.

**1.5. SCOPE**

One of the first tasks of a researcher is defining the scope of a study, i.e., its area (theme, field) and the amount of information to be included. Narrowing the scope of your project work can be time-consuming. Paradoxically, the more you limit the scope, the more interesting it becomes. This is because a narrower scope lets you clarify the problem and study it at greater depth, whereas very broad research questions only allow a superficial treatment.

**1.6. OUTLINE**

The outline gives an overview of the main points of your report. It clarifies the structure of your report and helps you find the correct focus for your work.

**CHAPTER 2: BACKGROUND RESEARCH**

**2.1. LITERATURE REVIEW**

A literature review usually has an organizational pattern and combines both summary and synthesis. A summary is a recap of the important information of the source, whereas a synthesis is a re-organization, or a reshuffling, of that information. It might give a new interpretation of old material or combine new with old interpretations. Some guidelines for literature review:

* Assess how each source relates to other research within the field. Group sources by theme, topic, or methodology and write the summary of key research.
* Evaluate how the findings of those papers can be relevant for your research
* Critically evaluate research (analyze and synthesize)

**2.2. CURRENT SYSTEM**

Unless your project is a complete new innovation, many similar software products should exist. You can mention about those products and how they are similar to your project work. What parts of those system inspire you, i.e., you decided to include in your project work?

**2.3. THE PROBLEM WITH CURRENT SYSTEM**

What problems did you find in those similar software products that you plan to solve? Try to provide genuine problems that have been proved by some literatures or surveys or studies. Even if you are inventing any problems, provide sufficient and valid reasons to justify them.

**CHAPTER 3: SPECIFICATION AND DESIGN**

**3.1. REQUIREMENT ELICITATION AND ANALYSIS**

The requirements for a system are the descriptions of what the system should do-the services that it provides and the constraints on its operation. A requirement is:

* What a system must do
* A known limitation or constraint on resources or design
* How well the system must do what it does

Requirement elicitation is the process by which you derived or obtained or deduced or collected the requirements, for example, by observation, brainstorming, interview, workshop, prototyping etc. Similarly, requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for your project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

**3.1.1. Functional Requirement**

Functional requirements describe the functionality of the system that can be modeled with use-cases. Functional requirements usually employ the word “*shall*”, e.g., functional requirements for a VoIP can be:

* The user shall add new participant
* The system shall show participants’ count
* The main user shall be able to drop participant
* The user shall be able to summon the operator
* The user shall be able to mute microphone

**3.2.2. Non-Functional Requirement**

Non-functional requirements describe system properties related to:

* system performance, speed (transaction time, screen refresh time, response time), size (project, input, output) , ease of use (training time) , the “*ilities*” (e.g., usability, security, maintainability , availability, reliability, scalability)
* a known limitation or constraint on resources or design
* can include documentation, marketing collateral, product localization, legal compliance restrictions

They typically employ the word “*must*”. e.g., non-functional requirements for a VoIP can be:

* The audio and video quality must be high
* The connection and service must be reliable
* The service must be easy to use
* It must be cheap to use the service
* The service must be available in the local language

**3.3. SYSTEM DESIGN**

System modeling mean representing the system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML). The student represent his/her project using one or multiple UML diagrams (listed below in Table 1) depending the nature of project, an example is given of Use-Case Diagram in section 3.3.2.

|  |  |
| --- | --- |
| **UML Diagram** | **Description** |
| Use Case Diagram | It shows the interactions between a system and its environment |
| Sequence Diagram | It shows interactions between actors and the system and between system components |
| Class Diagram | It shows the object classes in the system and the associations between these classes |
| State Chart Diagram | It shows how the system reacts to internal and external events |
| Activity Diagram | It shows the activities involved in a process or in data processing |

*Table 1: Various Types of UML Diagram*

Apart from these diagrams, the student can also use other methods to represent his/her project:

|  |  |
| --- | --- |
| **Method** | **Description** |
| ER Diagram | An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database |
| Block Diagram | A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks |
| Algorithms Used | It is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. |
| Flowcharts | A flowchart is a type of diagram that represents an algorithm, workflow or process. |

*Table 2: Other Forms Used for Design*

The following guidelines (listed in Table 3) can help students to select UML diagrams that can be applicable/suitable for his/her project work modeling.

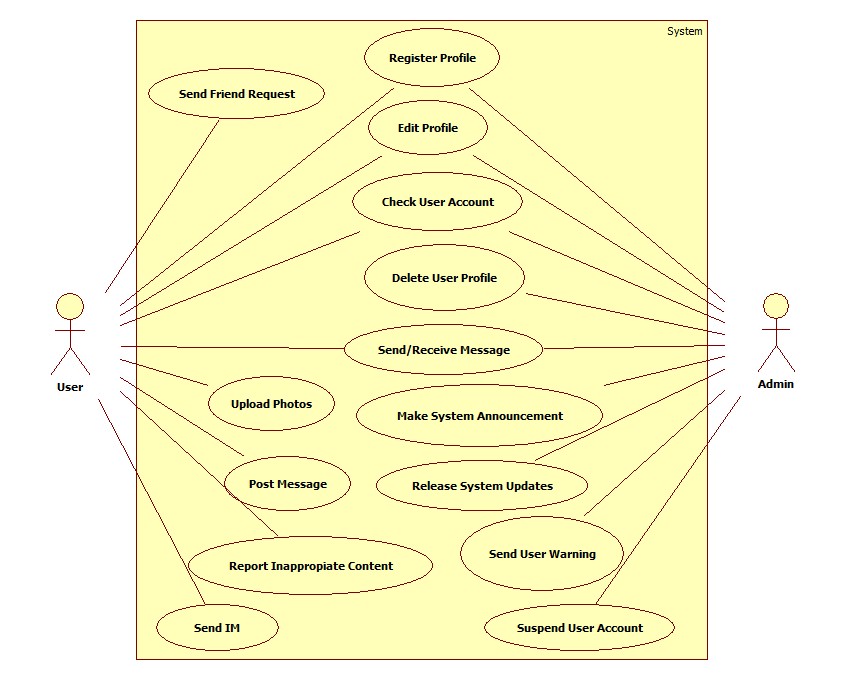
|  |  |
| --- | --- |
| **Perspective** | **Description** |
| External perspective | Model the context or environment of the system, e.g., Use-Case Diagram |
| Interaction perspective | Model the interactions between a system and its environment or between the components of a system, e.g., Sequence diagram |
| Structural perspective | Model the organization of a system or the structure of the data that is processed by the system, e.g.., Class diagram |
| Behavioral perspective | Model the dynamic behavior of the system and how it responds to events. For data, e.g., DFD and for event, e.g., state chart diagram. |

*Table 3: Perspective for Modeling*

**3.3.1. Use Case Diagram**

Use case illustrates a unit of functionality provided by the system. Typically used to communicate the high-level functions of the system and the system's scope (i.e., diagram shows what the system doesn't do). It helps development teams visualize the functional requirements of a system, including the relationship of "actors“. Stakeholders can easily see if needed functionality is present or not present in the system. Generally shows groups of use-cases:

* either all use cases for the complete system, or
* a breakout of a particular group of use cases with related functionality (e.g., all security administration related use cases)

**

*Figure 1: Use-Case Diagram of a Social Networking*

**CHAPTER 4: IMPLEMENTATION AND EVALUATION**

**4.1. TOOL AND TECHNOLOGY**

Write about all the tools and technologies that you used during your project. These tools and technologies can be used for:

* system specification
* system design
* system development and
* system testing

**4.2. IMPLEMENTATION**

The Implementation section is similar to the Specification and Design section in that it describes the system, but it does so at a finer level of detail, down to the code level. This section is about the realization of the concepts and ideas developed earlier. It can also describe any problems that may have arisen during implementation and how you dealt with them.

Do not attempt to describe all the code in the system, and do not include large pieces of code in this section. Instead pick out and describe just the pieces of code which, for example:

* are especially critical to the operation of the system;
* you feel might be of particular interest to the reader for some reason;
* illustrate a non-standard or innovative way of implementing an algorithm, data
* structure, etc..

You should also mention any unforeseen problems you encountered when implementing the

system and how and to what extent you overcame them. Common problems are:

* difficulties involving existing software, because of, e.g., its complexity, lack of documentation;
* lack of suitable supporting software;
* over-ambitious project aims

A seemingly disproportionate amount of project time can be taken up in dealing with such

problems. The Implementation section gives you the opportunity to show where that time has

gone.

**4.3. EVALUATION AND RESULT**

You should describe how you demonstrated that the system works as intended (or not, as the case may be). Include comprehensible summaries of the results of all critical tests that were carried out. You might not have had the time to carry out any full rigorous tests – you may not even got as far as producing a testable system. However, you should try to indicate how confident you are about whatever you have produced, and also suggest what tests would be required to gain further confidence. This is also the place to describe the reasoning behind the tests to evaluate your results, what

tests to execute, what the results show and why to execute these tests. It may also contain a discussion of how you are designing your experiments to verify the hypothesis of a more scientifically oriented project. E.g., describe how you compare the performance of your algorithm to other algorithms to indicate better performance and why this is a sound approach. Then summaries the results of the tests or experiments.

You must also critically evaluate your results in the light of these tests, describing its strengths and weaknesses. Ideas for improving it can be carried over into the Future Work section. Remember: no project is perfect, and even a project that has failed to deliver what was intended can achieve a good pass mark, if it is clear that you have learned from the mistakes and difficulties.

This section also gives you an opportunity to present a critical appraisal of the project as a whole. This could include, for example, whether the methodology you have chosen and the programming language used were appropriate.

**CHAPTER 5: CONCLUSION**

It is important to have a strong conclusion, since this is the last chance you have to make an impression on your reader. The goal of conclusion isn’t to introduce any new ideas, but to sum up everything you’ve written. The components of a good conclusion are:

1. Restate the main idea of your essay, or your thesis statement:
   1. Restate your topic
   2. Restate your thesis statement
2. Summarize (rather synthesize) the three sub-points of your essay:
   1. Include a brief summary of the paper’s main points, but don’t simply repeat things that were in your paper.
3. Leave the reader with an interesting final impression:
   1. Include a provocative insight or quotation from the research or reading you did for your paper.
   2. Propose a course of action, a solution to an issue, or questions for further study.
   3. Point to broader implications.

**CHAPTER 6: LIMITATION**

The limitations of the study are those characteristics of design or methodology that impacted or influenced the application or interpretation of the results of your study. They are the constraints on generalizability and utility of findings that are the result of the ways in which you chose to design the study and/or the method used to establish internal and external validity.

**REFERENCES**

Follow IEEE referencing format, whose referencing details are mentioned below. In the above text, use number reference in a square bracket, for example, [1], or [3, 4].

***Book in print***

[1] B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.

***Chapter in book***

[2] L. Stein, “Random patterns,” in Computers and You, J. S. Brake, Ed. New York: Wiley, 1994, pp. 55-70.

***eBook***

[3] L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 2nd ed. Reading, MA: Addison Wesley, 2003. [E-book] Available: Safari e-book.

***Journal article***

[4] J. U. Duncombe, "Infrared navigation - Part I: An assessment of feasibility," IEEE Trans. Electron. Devices, vol. ED-11, pp. 34-39, Jan. 1959.

***eJournal (from database)***

[5] H. K. Edwards and V. Sridhar, "Analysis of software requirements engineering exercises in a global virtual team setup," Journal of Global Information Management, vol. 13, no. 2, p. 21+, April-June 2005. [Online]. Available: Academic One File, http://find.galegroup.com. [Accessed May 31, 2005].

***eJournal (from internet)***

[6] A. Altun, "Understanding hypertext in the context of reading on the web: Language learners' experience," Current Issues in Education, vol. 6, no. 12, July 2003. [Online]. Available: http://cie.ed.asu.edu/volume6/number12/. [Accessed Dec. 2, 2004].

***Conference paper***

[7] L. Liu and H. Miao, "A specification based approach to testing polymorphic attributes," in Formal Methods and Software Engineering: Proceedings of the 6th International Conference on Formal Engineering Methods, ICFEM 2004, Seattle, WA, USA, November 8-12, 2004, J. Davies, W. Schulte, M. Barnett, Eds. Berlin: Springer, 2004. pp. 306-19.

***Conference proceedings***

[8] T. J. van Weert and R. K. Munro, Eds., Informatics and the Digital Society: Social, ethical and cognitive issues: IFIP TC3/WG3.1&3.2 Open Conference on Social, Ethical and Cognitive Issues of Informatics and ICT, July 22-26, 2002, Dortmund, Germany. Boston: Kluwer Academic, 2003.

***Newspaper article (from database)***

[9] J. Riley, "Call for new look at skilled migrants," The Australian, p. 35, May 31, 2005. [Online]. Available: Factiva, http://global.factiva.com. [Accessed May 31, 2005].

***Technical report***

[10] J. H. Davis and J. R. Cogdell, “Calibration program for the 16-foot antenna,” Elect. Eng. Res. Lab., Univ. Texas, Austin, Tech. Memo. NGL-006-69-3, Nov. 15, 1987.

***Patent***

[11] J. P. Wilkinson, “Nonlinear resonant circuit devices,” U.S. Patent 3 624 125, July 16, 1990.

***Standard***

[12] IEEE Criteria for Class IE Electric Systems, IEEE Standard 308, 1969.

***Thesis/Dissertation***

[13] J. O. Williams, “Narrow-band analyzer,” Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

**APPENDIX I [OPTIONAL]**

This is optional. The students can include anything that gives additional information on the topic explored in the contents of the text.