JOMO KENYATTA UNIVERSITY
OF
AGRICULTURE AND TECHNOLOGY

IT DEPARTMENT

FINAL YEAR RESEARCH PROJECT STRUCTURE

Formatting

- Apart from the title page (see title page below) the rest of the document should , should be in **Times New Roman, font twelve (12)**
- The whole document should be **1.5 lines** spaced
- Use the APA format for referencing
- Each chapter begins on a new page
- All the diagrams/figures must be properly captioned(use APA format for captioning)

1. STRUCTURE OF RESEARCH PROPOSAL

A) Title page to include:

i. Title

- A concise statement of the main topic
- Should capture succinctly the research area and the problem domain.
- Should be a reflection of the contents of the document.
- Fully explanatory when standing alone.
- Should not contain redundancies such as 'a study of…..or 'an investigation of……
- Abbreviations should not appear in the title.
- **bold, font 16 and centered**

ii. Author's name and affiliation

- Avoid use of words like 'By'….. 'from"…..
- Preferred order of names- start with 1st, middle followed by last name.
- Full names should be used, initials should be avoided.
- Also include the name of the supervisor

Affiliation should be well illustrated i.e. '*A proposal/ research project submitted to the Department of Information Technology ………in the School of…………. in partial fulfillment of the requirement for the award of the degree of …….. Jomo Kenyatta University of Agriculture and Technology.*'
-affiliation statement should be font 14, italics and centered

The year should follow at the bottom of the caption.

Note:
For proposals and final report (spiral bound) the cover page should include the title, author and affiliation (all on one page) and centered.

B) Declaration – Should include both the candidate's and the supervisor's declaration and duly signed.

This proposal/research project is my original work and has not been presented for a degree in any other University

………………….	…………………
Signature	Date

This proposal/research project has been submitted for examination with my approval as University Supervisor

………………	………………
Signature	Date

Note: Paginate using roman numbers starting with the declaration page.

C) Abstract- This is a brief statement of the problem and the research area, proposed solution, methodology and the expected outcome of the project. Should not exceed one page.

D) Table of contents

- The rubric should be in title case
- The chapter titles should be in caps and bold.
- The subheadings should follow each chapter title and should be in title case.
- Subheading of rows should be – Chapters & Pages indicated once at the top of each column e.g.,

CHAPTER 1	PAGE
1.1 Introduction ……………………………1	
1.2 Statement of the Problem……………..2	

Reference
Appendices

E) List of Tables

F) List of Figures

G) Acronyms

H) Definition of terms

- Define terms in the text that are not common.

CHAPTER 1
INTRODUCTION

This chapter should include the following;

1.1 Background –Should show understanding and genesis of the problem within your client operations.

- Talk about the global perspective followed by the local scenario
  - A checklist would assist in coining the details

1.2 Project Overview - Should introduce the research area formally

-Talk about the global perspective followed by the local scenario.
-Briefly introduce key computational principles behind the research area.
- A checklist would assist in coining the details

1.3 Statement of the problem

- Must indicate exactly what the problem is.
- Indicate why and how it is a problem. Give information to support this e.g. by use of statistics or evidence. This should be derived from background information to illustrate connectivity.
- The problem statement should address the current realities and should be abreast of the technology trends. The problem should not be just automation or transition from paper-based to digital system.

-It should show the magnitude and effects of the problem
-Must indicate exactly what the research problem is.
-This should be tied to your title/research area.

1.4 Proposed Solution

-State what will be developed. This research seeks to …… not the prototype.
-The research component should be clearly captured within the solution
-Outline key operations expected from the system to be developed.
-The proposed system should not merely transit from a manual to a digital system.
-Students need to compare recent models-globally and regionally to come up with the best solution without giving an old technology/going backwards.

1.5 Objectives

-One general objective which should be in line with the title.
-Three to four specific objectives related to the general objective
-Should be numbered
-Each specific objective should capture a key deliverable of the project; a research objective, design objective, implementation and testing.
-These should be the student's objectives (project-level) not the client's. It should be measurable within the 8 months of the projects
- Should not be questions in the questionnaire.
-They should be SMART (Specific, Measurable, Achievable, Reliable and Testable)
-They should be simple statements-not a paragraph

1.6 Research Questions

- They should be in line with the specific objectives and equal in number.
- Have to be numbered (1, 2, 3…..) and should be questions and not statements.
- Gives us a breakdown of the areas that would be covered.
- They should be broad areas that will bring value
- They should not have short answers that can be answered in a few words
- Research questions should be limited to research areas-NOT the client's problem or proposed prototype

1.7 Justification

- Should illustrate why the researcher is conducting the research and whom it shall benefit.
- Should indicate why the proposed solution will solve the client's problem.
- What is it contributing to that research area/Problem area
- Should outlay the relevance of the research at that point in time

1.8 Proposed Research and System Methodologies

- Briefly describe the proposed System implementation methodology-An outline of the methodology without discussions
- The tools and techniques used in the methodologies should be contextualized to the project/ system to be developed.
- Justify your choice of method
- Covers the life cycle of the research

1.9 Scope

- This is a kind of a disclaimer. It should cite the focus of the study geographical area or target group/ population.
- Acknowledge the potential limitations of your study, such as constraints in data availability, methodology, or resources.
- State what you will confine yourself to as far as the project is concerned

Note:

- Paragraphing should be consistent. Either leave space or indent between paragraphs.
- Spacing and indenting should not be used together.
- One sentence paragraphs are unacceptable.
- A paragraph should have a minimum of five sentences.

Table of contents should be followed by:

- List of figures/ tables- Should be labeled as per the chapters in which they are found e.g. the first figure in chapter one should be labeled as Figure 1.1

CHAPTER 2

LITERATURE REVIEW

This chapter should include;

2.1 Introduction- Tells us what to expect to be covered.

- Briefly introduce the research topic and its relevance to the application area.
- Give a brief background of the increasing application of the technology (research topic) within the specific application area setting. For instance the use of chatbots in healthcare settings.
- State the purpose of the literature review and briefly outline the objectives

2.2 Theoretical review

- Start by identifying and defining the key concepts or variables relevant to your research area. For instance in machine learning (ML) elucidate on concepts such as data, features, labels, algorithms, training etc.
- Identify key theoretical divisions defining your area of research. This could be different schools of thought around the subject and principles behind them. For instance if your research is on chatbots in heathcare discuss the different types of chatbots commonly used in healthcare, such as rule-based chatbots, retrieval-based chatbots, and generative chatbots.
- Highlight on the advantages and limitations of each kind of division/school of thought. For instance the pros and cons of the different types of chatbots.

.

2.3 Case Study Review

- Explore the various applications of the research topic to the application domain. For instance in our example the application of chatbots in triage, diagnosis, counseling etc.
- Identify key implementations relevant to your problem. Briefly explain their key successes and possible misses.

2.4 Integration and Architecture

- Explore the various options for integrating / implementing the research problem with the application. For instance how can chatbots be integrated with existing healthcare systems? How can cryptography be implemented in a student management system?
- Address the issue of possible design architectures or frameworks

2.5 Summary

- A brief summary of key findings from the review.

2.6 Research gaps

-Research gaps should be explained and how this research seeks to resolve them.

REFERENCES

Project research should have references. Minimum of 10 peer review references. Students are encouraged to adhere to the age of the references not be more than 5 years except in a few exceptions

APPENDICES: Project Resources, Budget, Project Schedule

2. STRUCTURE OF PROJECT REPORT

The Final Report will include the above two chapters plus

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

This chapter should indicate;

3.1 Introduction:
Introduction to the chapter detailing what the chapter covers.

3.2 Describe the Systems Development Methodology that is used in the research. This is just to introduce it before the actual implementation begins.

3.3 Feasibility Study:
A statement on the project feasibility should be presented.
-Should capture detailed reports on the economic, technical, operational and any other feasibility aspect critical to the application successful implementation.

3.4 Requirements elicitation:
Data Collection:
-Describe the data collection tool, its preparation and how it was administered and attach it as an appendix. Use tools like Interviews, Observation, questionnaires, etc. This tool should be approved by the Supervisor before it is administered.
-Describe the sampling techniques used and how the sample size was determined
-The data collected must be relevant to the problem and based on the objectives of the research.
-It should be useful in deducing system requirements

3.5 Data Analysis:
Analyze the data collected using Statistical tools (Excel, SPSS) and represent the findings using any analytical tool (pie charts, bar graphs, line graphs, etc)

3.6 System Specification

- A detailed specification of system requirements including functional and non-functional requirements

3.7 Requirements Analysis and Modeling

- Analyze the gathered requirements to identify dependencies, conflicts, and potential solutions.
- Structure the requirements to conform to software functionalities and components

- Tools such as low-level DFDs , use case diagrams and conceptual and analysis class diagrams can be used to visualize and clarify requirements

3.8 Logical Design

- A logical representation that captures the essential structure, behavior, and functionality of the system.
- Should include:
- **3.8.1 System Architecture**: Define the high-level structure and organization of the software system. Determine the major components, modules, and their relationships. Consider architectural patterns such as client-server, layered, or microservices architecture. Artifacts such as class diagram and component diagram can be developed to capture the architectural details.
- **3.8.2 Control Flow and Process Design:** Define the control flow and sequencing of activities within the system. Specify the processes, tasks, and their relationships. Identify the key processes and explain them clearly. Use flow charts/activity, sequence, state chart diagrams as well as detailed DFDs. Pseudo code can also be used to describe some tasks algorithmically.
- **3.8.3 Design for non functional requirements:** Define security strategies employed and how they have been integrated into the system. Also define error and exception handling strategies. Define any other mechanism employed to enhance the efficiency, effectiveness and appeal of the system.

3.9 Physical Design

- It involves making decisions about hardware and software platforms, databases, networking, deployment environments, and other physical aspects of the system.
-Should include:
- 3.9.1 Database Design:
- Design the physical structure and organization of the database system. Specify the database schema, tables, fields, and relationships based on the logical data model. Determine the appropriate database management system (DBMS) and storage mechanisms. Consider factors such as data integrity, security, indexing, and query optimization.
- **3.9.2 User Interface Design**: Design the interfaces between system components, modules, and external systems.
-Design input and output forms
-Create wireframes using tools such as Adobe XD, Marvel and InVision studio.

CHAPTER 4

SYSTEM IMPLEMENTATION AND TESTING, CONCLUSIONS AND RECOMMENDATIONS

4.1 **Introduction**-A brief on the chapter

4.2 **Environment and Tools** – Explain the implementation environment in terms of tools used for programming the backend, frontend and middle layers.

4.3 **System Code Generation**-Explain clearly the process of code generation. Identify key processes; describe their implementation showing code snippets used.

4.4 **Testing**-Define clear testing strategies by developing a testing suite.

- Should describe all tests subjected to your system and the results
- Show snapshots/screenshots of results of various tests and at various stages of the software.

4.5 User **Guide**

- The software system should have a user guide /help implementation

- The documentation should include instructions on setting up and using the software.

4.6 **Conclusions** - Did you solve your client's problem and to what extent?

-What are the most significant accomplishments?

-what are the limitations?

Indicate the challenges encountered in the study that may have limited the study.This may include: Skills, Money, tools, etc. time is NOT a limitation

4.7 **Recommendations**- Should be derived from the conclusions

-What improvements can be made in future?

REFERENCES

Research project should include references

APPENDICES

Instruments

Letters of introduction

Interview Transcripts and Questionnaires

Budget

Project Schedule

Notes on Testing

Testing

Objectives of testing

- Build quality in the system developed
- Demonstrate the working capabilities of your system
- Assess progress and suitability of the system

What is testing

Testing entails subjecting the system developed by the students to set of valid inputs to validate the outputs of the system against pre meditated set of outputs.

-Identify, design then implement the testing.

-Clearly state the Validation (e.g.Usability, Blackbox, Acceptance) and verification tests (e.g.Smoke testing, Stress test, White box) conducted.

TESTING SCOPES THAT CAN BE ADOPTED

Gui Testing: Attempt to cover all the functionality of the system and fully exercise the GUI itself. A GUI has many operations that need to be tested. The objective at this point will be test sequencing of events and relevance in terms of colors and appearance of the GUI interface

Usability testing is a technique used in user-centered interaction design to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system.

The students should be able to test the system with their friends and peers and report on how he peers felt about the system. A form or documentary evidence should accompany the report to show this.

Performance testing is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

The system for the students should actually be subjected in stressful conditions such as multiple entries of data ,quick succession of activities to determine its performance level. This should then be reported back to the group for assessment by the supervisors.

White box testing: tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. The student should submit the flow charts so that set of test cases are subjected to the flow and response of the system determined from the flow chart. This would then help compare the internal performance against expected general performance of the system.

Load testing: Load testing is a generic term covering Performance Testing and Stress Testing. A process of testing the behavior of the Software by applying maximum load in terms of Software accessing and manipulating large input data. It can be done at both normal and peak load conditions. This type of testing identifies the maximum capacity of Software and its behavior at peak time.

Regression Testing: Similar in scope to a functional test, a regression test allows a consistent, repeatable validation of each new release of a product or Web site. Such testing ensures reported product defects have been corrected for each new release and that no new quality problems were introduced in the maintenance process. Though regression testing can be performed manually an automated test suite is often used to reduce the time and resources needed to perform the required testing.

Whenever a change in a software application is made it is quite possible that other areas within the application have been affected by this change. To verify that a fixed bug hasn't resulted in another functionality or business rule violation is Regression testing. The intent of Regression testing is to ensure that a change, such as a bug fix did not result in another fault being uncovered in the application.

Stress testing: Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements to determine the load under which it fails and how. A graceful degradation under load leading to non-catastrophic failure is the desired result. Often Stress Testing is performed using the same process as Performance Testing but employing a very high level of simulated load.

This testing type includes the testing of Software behavior under abnormal conditions. Taking away the resources, applying load beyond the actual load limit is Stress testing.

Smoke testing: A quick-and-dirty test that the major functions of a piece of software work without bothering with finer details. Originated in the hardware testing practice of turning on a new piece of hardware for the first time and considering it a success if it does not catch on fire.

Unit Testing: Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components of a product to ensure their correct behavior prior to system integration.

This type of testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is separate from the test data of the quality assurance team.

Acceptance Testing: Testing to verify a product meets customer specified requirements. A customer usually does this type of testing on a product that is developed externally.

This is arguably the most importance type of testing as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the clients requirements. The QA team will have a set of pre written scenarios and Test Cases that will be used to test the application.

Black Box Testing: Testing without knowledge of the internal workings of the item being tested. Tests are usually functional.

Functional Testing: Validating an application or Web site conforms to its specifications and correctly performs all its required functions. This entails a series of tests which perform a feature by feature validation of behavior, using a wide range of normal and erroneous input data. This can involve testing of the product's user interface, APIs, database management, security, installation, networking, etc Functional testing can be performed on an automated or manual basis using black box or white box methodologies.

This is a type of black box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional

Testing of the software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

Integration Testing: The testing of combined parts of an application to determine if they function correctly together is Integration testing.

Testing in which modules are combined and tested as a group. Modules are typically code modules, individual applications, client and server applications on a network, etc. Integration Testing follows unit testing and precedes system testing.

There are two methods of doing Integration Testing Bottom-up Integration testing and **Top down Integration testing**.

Bottom-up integration: this testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

Top-Down integration: in this testing, the highest-level modules are tested first and progressively lower-level modules are tested after that.

Security Testing: Security testing involves the testing of Software in order to identify any flaws ad gaps from security and vulnerability point of view.

Portability/Compatibility Testing: Testing to ensure compatibility of an application or Web site with different browsers, OSs, and hardware platforms. Compatibility testing can be performed manually or can be driven by an automated functional or regression test suite.

Portability testing includes the testing of Software with intend that it should be re-useable and can be moved from another Software as well.

Conformance Testing: Verifying implementation conformance to industry standards. Producing tests for the behavior of an implementation to be sure it provides the portability, interoperability, and/or compatibility a standard defines.

Identification of Test case generation

A test case is a set of conditions or variables under which the student determines if the application is working. It includes Test case ID, Condition to be tested, Expected results and actual results. Include a column that shows if the test passed or failed.

The students should generate a minimum of 20 test cases to enable the examiners assess the full functionality of the system. The test cases should cover each unit or module of the system. The assumption in this scenario is that the students will have five modules and so 4 test cases will be applied to each module to determine its suitability.

A table should be generated to indicate the set of inputs and the valid outputs to be produced from the system. These inputs will help form test cases and outputs as assessment criteria for suitability of the system.

Software testing tools in the market:

-Selenum

-HP Quick Test Professional

-IBM Rational Functional Tester

-Silk Test

-TestComplete

-Testing Anywhere

-WinRunner

-LoadRunner

-Visual Studio Test Professional

-WATIR