

<PROJECT TITLE>

**BIT304 FINAL YEAR PROJECT I**

<< Student ID >> << Student Name >>

Submitted to the

FACULTY OF COMPUTING AND DIGITAL TECHNOLOGY

(SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY)

In partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

HELP UNIVERSITY

JUNE 2020

###### **Abstract**

A brief description of the project in 200 to 300 words to enable readers to gain a quick and accurate impression of your project. In particular, you should explain why this project was initiated e.g., weaknesses in existing system and how you manage to solve the problems. You should then explain the results of your implementation and findings, indicating the significance and benefits to the users of the system.

To increase an efficiency of a product, nowadays many web development companies are using different project management systems. A company may run a number of projects at a time, and requires input from a number of individuals, or teams for a multi-level development plan, whereby a good project management system is needed. Project management systems represent a rapidly growing technology in IT industry. As the number of users, who utilize project management applications continues to grow, web based project management systems enter a critical role in a multitude of companies. Thus, a proper project management system plays a distinctive part in ensuring reliable, robust and high quality web applications for customers. Developing a web based project management system and showing how, in turns, it helps users to handle projects. These processes in everyday´s working life, is the scope of the thesis. The reliability and robustness of a web based project management system has also been set as the structure of the current thesis. Finally, a web based project management system has been developed, which highly meets the standards and requirements set by the company. The web based project management system uses an already integrated TRAC application that has improved to suite companies needs.

# **Declaration**

I hereby declare that the report presented here as part of the requirement of BIT304 is original and no parts of this report had been plagiarised from any other resources unless those indicated with proper referencing. This report will be the property of HELP University and cannot be distributed in any form without the written consent of HELP University.

Student Names:

Date:**Acknowledgement**

This project will not be possible without the help of a number of people. We would like to express our deepest gratitude to <name> who has devoted much of his/her time to ………

# **Table of Contents**

ABSTRACT ii

**DECLARATION iii**

**ACKNOWLEDGEMENT iv**

**TABLE OF CONTENTS v**

**LIST OF TABLE viii**

**LIST OF FIGURE ix**

**LIST OF SYMBOLS AND ABBREVIATION x**

Chapter 1: Proposal 1

1.1 Introduction 1

1.2 Company Background 1

1.3 Issues with Current System 1

1.4 Benefits and Constraints of Proposed System 1

1.5 Project Description 1

1.6 Project Aims and Objectives 1

1.7 Project Scope 1

1.8 Software and Hardware Requirements 1

1.9 Development Methodology 1

CHAPTER 2: PROJECT MANAGEMENT PLAN 2

2.1 Introduction 2

2.2 Work Breakdown Structure 2

2.3 Work Items List 2

2.3 Risks Management 2

**2.5 Gantt Chart 3**

Chapter 3: REQUIREMENTS ANALYSIS 4

3.1 Introduction 4

3.2 Requirements Gathering 4

**3.3 Functional Requirements**

3.3.1 Use Case Diagram 4

3.3.2 High-Level Use Cases 4

**3.4 Non-Functional Requirements**

3.4.1 Technical Requirements 4

3.4.2 Usability Requirements 5

**3.4.3 Reliability Requirements 5**

**3.4.4 Security Requirements 5**

Chapter 4: ITERATIONS 6

**4.1 Introduction 6**

**4.2 Iteration 1 6**

4.2.1 Introduction 6

4.2.2 Purpose 6

**4.2.3 Context 6**

**4.2.4 Schedule of Iteration Workflows 6**

4.2.5 Iteration Schedule Breakdown 7

4.2.6 Resource Summary 7

4.2.7 Evaluation Criteria 7

4.2.8 Analysis and Design Artifacts 8

4.2.9 Implementation and Testing 8

4.2.10 Iteration Review and Evaluation 10

**4.3 Iteration 2 11**

**4.4 Iteration 3 11**

**CHAPTER 5: CONCLUSION 12**

BIBLIOGRAPHY 13

Appendix A: GANTT CHARTS 14

Appendix B: PROGRESS REPORTS 15

**APPENDIX C: MEETING REPORTS 16**

Appendix D: MINUTES OF MEETINGS 17

**APPENDIX E: PROJECT EVALUATION 18**

Appendix F: Formatting of Report 19

D.1 Binding 19

D.2 Length of Report 19

D.3 Typing 19

D.4 Spacing 19

D.5 Margins 19

D.6 Subdivisions 19

D.7 Tables and Diagrams 19

# **List of Tables**

Table 1.1: Summary of Web Applications using React Native. 13

Table 1.2: Number of Developers using Web Builders 18

Table 2.1:

Table 2.2:

**List of Figures**

Figure 1.1: Web Applications using React Native. 13

Figure 1.2: Examples of Web Builders 18

Figure 2.1:

Figure 2.2:

# **List of Symbols and Abbreviations**

JS JavaScript

HTML HyperText Markup Language

# **CHAPTER 1: PROPOSAL**

## **1.1 Introduction**

Summarise the project in a few sentences.

## **1.2 Company Background**

Provide some background information about the company, e.g. nature of the business, organisation structure, history, etc.

## **1.3 Issues with Current System**

Outline the problems or weaknesses in the current system; it could be a manual system or existing system.

## **1.4 Benefits and Constraints of Proposed System**

Explain how your proposed system should be able to overcome the weaknesses defined in Section 1.3. Describe the risks/constraints that you have identified for the system.

## **1.5 Project Description**

Provide a detailed description of the project you are proposing, outlining all the important functionalities, future users/stakeholders of system, and how they should access the system. Include an architecture diagram (network diagram) to illustrate the overall system.

## 

## **1.6 Project Aims and Objectives**

List all the objectives you wish to achieve for this project. They should be accurate and precise as you will be reviewing them individually in the concluding chapter.

## **1.7 Project Scope**

List the constraints and scope of the project. Remember not to overestimate by trying to implement too many modules if you are not very familiar with the methodology or languages used to build the software. At the same time, do not narrow the scope too steeply to the point that you’ll end up finishing the project 4 months earlier.

## 

## **1.8 Software and Hardware Requirements**

List all software and hardware required for this project.

## **1.9 Development Methodology**

Indicate the type of methodology that you will be using e.g., CASE, RAD, spiral, prototyping, etc. Explain the major phases involved and why you chose the particular methodology.

# 

# **CHAPTER 2: PROJECT MANAGEMENT PLAN**

## **2.1 Introduction**

## State the purpose of this plan.  The purpose of the plan should be to describe how the project is going to be managed. This plan should also identify the key project milestones and document project team roles and responsibilities.

## **2.2 Work Breakdown Structure**

A work breakdown structure (WBS) is a list of all the individual tasks that are required to complete the project. It is essential in planning and executing the project because it is the foundation for developing the project schedule, for identifying milestones in the schedule. Refer to Table. 2.1 for a sample WBS template – you will need to complete the table with phases and activities that are appropriate to your project. List all phases and activities for BIT304 and BIT305.

**Table 2.1: Work Breakdown Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phases and Activities** | **Start Date** | **End Date** | **Duration in Days** | **Number of Resources (hours)** |
| 1. Define requirements    1. Define user requirements    2. Define system requirements    3. Define risks    4. Develop project plan |  |  |  |  |
| 1. Iteration1   2.1 Analysis  2.2 Design  2.3 Implementation  2.4 Testing |  |  |  |  |
| 3. Iteration 2 |  |  |  |  |
| 4. Iteration 3 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**2.3 Risks Management**

Project risk management is the art and science of identifying, analysing and responding to risks throughout the life of a project. A risk management plan documents the procedures for managing risks throughout the project. Use the template provided in Table 2.2.

**Table 2.2 Risks Management**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Potential Impact on Project Success**  **L/M/H** | **Likelihood of Occurrence**  **L/M/H** | **Mitigation Plan** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### **2.4 Gantt Chart**

Briefly describe the structure and major timelines of your Gantt Chart, the project start and end date, as well as the tools used to produce the Gantt Chart

Produce a planned Gantt Chart to show activities that have to be completed and when.

* What the various activities are
* When each activity begins and ends
* How long each activity is scheduled to last
* Where activities overlap with other activities, and by how much
* The start and end date of the whole project

The Gantt Chart should be placed in the Appendix A.

# **CHAPTER 3: REQUIREMENT ANALYSIS**

## **3.1 Introduction**

Provide a short introduction on what you wish to achieve in this stage.

## 

## **3.2 Requirements Summary**

Summarise the requirements for this system, identifying in detail the users, main functions/modules, the input and expected output of the system.

### 

### **3.3 Functional Requirements**

### **3.3.1 Use Case Diagram**

Draw the use case diagram with CASE tools e.g., Rational Rose, StarUML, etc.

**Use footnotes to explain your level of contribution in producing the use case diagram.**

### **3.3.2 High-Level Use Cases**

Produce a high-level use case for every use case defined in the Use Case Diagram (Section 3.3.1).

|  |  |
| --- | --- |
| **Use Case 1[[1]](#footnote-1)** |  |
| Goal in Context |  |
| Primary Actors  Secondary Actors |  |
| Description |  |

|  |  |
| --- | --- |
| **Use Case 2**[[2]](#footnote-2) |  |
| Goal in Context |  |
| Primary Actors  Secondary Actors |  |
| Description |  |

**Include all the high-level use cases for this project. However, use footnotes to highlight the high level use cases that you were responsible for developing.**

### **3.4 Non-Functional Requirements**

### **3.4.1 Technical Requirements**

Describe the operational characteristics related to the environment, hardware and software of the organization. E.g. the client components of a new system might be required to operate on mobile devices or desktop PCs running the Windows OS and using Internet Explorer. The server components might have to be written in Java and communicate with one another using a component interaction standard e.g. CORBA (common Object Request broker Architecture) or SOAP (Simple Object Access protocol).

### **3.4.2 Usability Requirements**

Describe the operational characteristics related to users e.g. the user interface, related work procedures, online help, and documentation. E.g. a web-based interface might be required to follow organisation-wide graphic design, guidelines e.g. menu placement and format, colour schemes, use of the organisation’s logo, etc.

### **3.4.3 Reliability Requirements**

### Describe the dependability of a system – how often a system exhibits behaviours e.g. service outages, and incorrect processing and how it detects and recovers from these problems.

### **3.4.4 Security Requirements**

### Describe which users can perform what system functions under what conditions. E.g. access to certain system functions might be limited to managers at a certain level or employees of a specific department. Security requirements can also apply to areas such as network communications and data storage. E.g. an organisation might require encryption of all data transmitted over the internet and control of all database server access through use of a username and password.

### **CHAPTER 4: ITERATIONS**

## **4.1 Introduction**

How many iteration cycles will you need to complete the project (include BIT304 and BIT305)? Explain why you have assigned the various use cases in each of the iteration cycle.

**Table 4.1. Iteration Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Iteration** | **Use Cases** | **Proposed Start Date** | **Proposed End Date** |
| 1 | 1 and 2 |  |  |
| 2 | etc |  |  |
| 3 | etc |  |  |
| etc | etc |  |  |

**4.2 Iteration 1**

**4.2.1 Introduction**

State the basic goals for this iteration along with a summary of the start and end dates for this iteration. Update the GANTT chart for this iteration.

**4.2.2 Purpose**

The completion of an iteration comprises some defined portions of a phase. The completion of a phase is a milestone. Describe the contribution that this iteration provides toward the completion of the milestone.

**4.2.3 Context**

Provide a brief description of how this iteration plan fits into the overall project plan. You may refer to the project plan, development case, or any other relevant artefacts. List the particular use cases that will be addressed in this iteration.

**4.2.4 Schedule of Iteration Workflows**

A summary of the activities in each workflow (by specific use case) participating in this iteration is given here. Any risks or events that could cause slippage in the dates should be given here. Provide a breakdown tabular form, listing *only the workflows relevant to this iteration*.

Table 4.2. Iteration Workflow Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Workflow** | **Start Date** | **End Date** | **Duration (days)** |
| **Use Case 1: Registration Portal** |  |  |  |
| Analysis |  |  |  |
| Design |  |  |  |
| Implementation |  |  |  |
| Testing |  |  |  |
| **Use Case 2:** |  |  |  |

**4.2.5 Iteration Schedule Breakdown**

Provide a list of tasks for each workflow given in Table 4.2. A format for this list is given in Table 4.3.

Table 4.3. Iteration Plan Task Breakdown by Workflow

| **Task Name** | **Start** | **Finish** | **Assigned To** |
| --- | --- | --- | --- |
| **Use Case 1 (Registration Portal)** |  |  |  |
| **1.1 Analysis** |  |  |  |
| 1.1.1 <etc> | <date> | <date> | <name> |
| **1.2 Design** |  |  |  |
| * 1. 2 <etc> | <date> | <date> | <name> |
| **1.3 Implementation** |  |  |  |
| 1.1.3 <etc> | <date> | <date> | <name> |
| **1.4 Testing** |  |  |  |
| 1.1.4 <etc> | <date> | <date> | <name> |

**4.2.6 Resource Summary**

Provide a list of resources needed to complete this iteration. Resources include people, equipment, facilities, software, etc. These resources should be confined to those people or things that are limited in nature, or are unavailable at the time of writing of this iteration plan. For example, if a server is to be installed during this iteration, you would not necessarily include it as a resource in the next iteration, unless there is some problem with its availability or performance. This section is intended to be a management tool, not an inventory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Resources | Usage | Quantity | Cost |
| Hardware | | | |  |
| 1 | Computer (Windows 10) | Development of web application | 2 computers | RM0 (Existing computers) |
| Software | | | |  |
| 2 | Meteor js | Development of front end and back end applications | 1 platform | RM0 ( Open source) |
| Human Resources | | | |  |
| 3 | Expert in Meteor js from the industry | Conduct training to introduce this system to team members | 1 expert | RM200 |

**4.2.7 Evaluation Criteria**

Describe the criteria by which you will evaluate the success of this iteration e.g. what are the anticipated results, deliverables; iteration to be completed in the specified time frame; familiarisation with software, hardware; skills attained, etc.

**4.2.8 Analysis and Design Artefacts**

Provide a description/models of the artefacts that are the deliverables affected by this iteration. Include only those that are completed in this iteration, if you are "managing by deliverables." Again, *use only the parts that are relevant and correspond to the Workflow Breakdown. Only provide the Analysis artefacts if there have been changes made to the artefacts defined previously (for subsequent iterations).*

If your proposed system is an object-oriented system, include the analysis class diagram to represent the static structure of your problem domain. If your proposed system is not an OO system, include the data flow diagrams (upto Level 2) in this section.

**4.2.9 Implementation and Testing**

**Implementation**

Outline the implementation strategy you used. What tools or languages did you use? Include also problems encountered.

Pick up the screen shots of important processes - explain how they work.

## **Testing**

Do not tell us what black box testing, function testing, integration testing are. These can be found in the textbook. Instead list all the testing you have done with each module, giving attention to the following:-

1. how exactly you tested the system (the type of data used)
2. the kind of errors you found in the code.
3. did the system solve the problem that it was set out to solve?

**Unit Testing**

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules.

| **Unit Test Plan** | |
| --- | --- |
| Module Name/ID: \_\_\_\_\_\_\_\_\_ | Program Name/ID: \_\_\_\_\_\_\_\_\_\_\_ |
| **1. Module Overview** | |
| *Briefly define the purpose of this module. This may require only a single phrase: i.e.: calculates overtime pay amount, calculates equipment depreciation, performs date edit validation, or determines sick pay eligibility, etc.* | |
|  | |
| **1.1 Inputs to Module** | |
| *[Provide a brief description of the inputs to the module under test.]* | |
|  | |
| **1.2 Outputs from Module** | |
| *[Provide a brief description of the outputs from the module under test.]* | |
|  | |
| **1.3 Logic Flow Diagram/Segment of Code** | |
| *[Provide logic flow diagram if additional clarity is required.]* | |
|  | |
| **2. Test Data** | |
| *(Provide a listing of test cases to be exercised to verify processing logic.)* | |
|  | |
| **2.1 Positive Test Cases** | |
| *[Representative data samples should provide a spectrum of valid field and processing values including "Syntactic" permutations that relate to any data or record format issues. Each test case should be numbered, indicate the nature of the test to be performed and the expected proper outcome.]* | |
|  | |
| **2.2 Negative Test Cases** | |
| *[The invalid data selection contains all of the negative test conditions associated with the module. These include numeric values outside thresholds, invalid Characters, invalid or missing header/trailer record, and invalid data structures (missing required elements, unknown elements, etc.)* | |
|  | |
| **3. Interface Modules** | |
| *[Identify the modules that interface with this module indicating the nature of the interface: outputs data to, receives input data from, internal program interface, external program interface, etc. Identify sequencing required for subsequent string tests or sub-component integration tests.]* | |
|  | |
| **4. Test Tools** | |
| *[Identify any tools employed to conduct unit testing. Specify any stubs or utility programs developed or used to invoke tests. Identify names and locations of these aids for future regression testing. If data supplied from unit test of coupled module, specify module relationship.]* | |
|  | |

**System Testing**

Report the findings for each use case in the following format:-

| **Use Case Tested:** | | use case name | | | |
| --- | --- | --- | --- | --- | --- |
| **Test Description:** | | Provide a concise description of the procedure used by this test script. | | | |
| **Pre-conditions** | | Detail any of pre-conditions for execution of this test script. | | | |
| **Post-conditions** | | Detail the post-conditions of executing this test script. | | | |
| **Notes:** | |  | | | |
| **Result (Pass/Fail/Warning/Incomplete)** | |  | | | |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | P | F |
|  |  | |  |  |  |
|  |  | |  |  |  |
|  |  | |  |  |  |

Discussion of test results must be supported with evidence of test cases and test scripts.

Conclude whether you have successfully implemented the proposed system.

**4.2.10 Iteration Review and Evaluation**

How well did the estimated schedule correspond with the actual implementation times? Based on work items and work item completion times, draw up a 'project burndown' chart which shows the actual progress of the project. On the same chart draw up a similar graph based on estimated work item completion times. Discuss any differences. Were there any tasks that took a significant amount of time that did not appear in the work item list? Discuss these tasks in terms of project scheduling and budget.

**4.3 Iteration Plan 2**

etc etc

**4.4 Iteration Plan 3**

etc etc

**Chapter 5: Conclusion**

Discuss all the challenges, limitations, and highlight all the issues that you have encountered. How did you solve the problems?

**Bibliography**

**Use the APA referencing style adopted by the School of Information and Communication Technology.** The references should be arranged in alphabetical order.

List all the references that you used for the project

## **APPENDIX A: GANTT CHARTS**

Present your Baseline and final Tracking Gantt Charts here. Use an appropriate software tool to produce your Gantt Charts.

# Identify the responsibilities of each team member on the Gantt Charts. **APPENDIX B: MONTHLY PROGRESS REPORTS**

# Compile all the monthly progress reports here. You should have three monthly progress reports for the month of January, February, and March. This component is used to record all the continuous progress throughout the whole semester for BIT304. You should document this appropriately to show your continuous effort towards your project development.

Note: Refer to Monthly Progress Report Template

# **APPENDIX C: MEETING REPORTS**

# Compile all the weekly meeting reports here. You should have at least 9 meeting reports with your supervisor. Make sure you take the initiative to make appointments with your supervisor. You should meet your supervisor at least 9 times throughout the semester to show your project’s progress and prototype development. This will be taken into account in the evaluation of prototype development in both presentation and final report.

Note: Refer to Supervisor Meeting Report Template

# **APPENDIX D: MINUTES OF MEETINGS**

Note: The minutes of meetings will be used to determine the work allocated and done by each student. Provide sufficient detail to describe the complexity and equality of work distributed.

Note: Refer to Minutes of Meetings Template

**APPENDIX E: PROJECT EVALUATION**

Note: Refer to Project Evaluation Template

**APPENDIX F: FORMATTING OF REPORT**

**Note: DO NOT include this section into your actual report.**

### **D.1 Binding**

The report must be bound in clear plastic covers. You must use comb binding for your final report.

### **D.2 Length of Report**

There is no limit on the number of words, but generally it should not exceed 70 pages and should be more than 55 pages, excluding the appendices. Do not print any source codes but include all program codes in the CD.

### **D.3 Typing**

The report should be prepared using an electronic word processor, e.g. Microsoft Word. The printed text must be clear and legible. The acceptable font type is Times New Roman and the font size for general text is 12 points.

Analysis and design models should be drawn using a CASE tool, e.g., Rational Rose, StarUML, etc. Hand-drawn diagrams will not be accepted.

### **D.4 Spacing**

Use 1.5 spacing throughout the text. However, you can use single-spacing for tables, notes, footnotes, table labels and figure labels. Font size for tables, notes, footnotes, table figures and figure labels is 10 points.

### **D.5 Margins**

The margins are as follows:-

Top: 3 cm

Right: 3.5 cm

Left: 3 cm

Bottom: 3 cm

Typing should not be below the bottom margin. All tables and figures must conform to the margin requirements.

### **D.6 Subdivisions**

The internal organisation should be used consistently throughout the report. The text should be divided into chapters and sub-chapters eg., 1.1, 1.2, etc. The maximum subdivision is up to level-3 (eg: 1.1.1) any other section heading above this should be underline and bold

### **D.7 Tables and Figures**

Tables should be centered on the page. Each table must have a reference number and a caption, e.g., **Table 1.1 Comparison between Apple and Microsoft**, **ABOVE** the table itself. Try to place the tables near to the text which refer to it. The next table in the chapter should be referenced as Table 1.2 and so forth. Tables in Chapter 1 should have a prefix of 1, tables in Chapter 2 should have a prefix of 2, etc.

Figures should also have a reference number and a caption, e.g., **Figure 1.1 Context Diagram**, **BELOW** the figure itself. Try not to let the figure exceed one page. Try to place the figures near to the text which refer to it. The next figure in the chapter should be referenced as Figure 1.2, etc. Figures in Chapter 1 should have a prefix of 1, figures in Chapter 2 should have a prefix of 2, etc

1. [↑](#footnote-ref-1)
2. **Fong is responsible for producing this high level use case.** [↑](#footnote-ref-2)