Guideline for

Project Documentation
Limit - 6000words (+/-500)

a) Title Page (First Page)

The full title of the project
Full Name & ID in brackets
Daffodil International University

b) Abstract (Second Page)

This section should consist of a synopsis of the project (150-200 words) stating the nature and scope of the work undertaken, and a high level summary of the outcomes.

c) Content Page (Third Page)

This section should show the page numbers of chapters, sections and sub-sections, a list of figures and tables, and a list of appendices.

d) Acknowledgement (Fourth Page)

This section is optional, but you may wish to pay tribute to particular people who have given you special assistance or support.

CHAPTERS

1. Introduction

1.1 The system developed
1.2 Justification for the method or framework used
1.3 The solution that emerged
1.4 The main aims and objectives of the project
1.5 A short overview of the remaining chapters

2. Feasibility Study

2.1 Economic Feasibility(Cost-benefit analysis of the system)
2.2 Technical Feasibility(Technical resources are there or required)
2.3 Operational Feasibility (Usability advantages for users within the system)

3. Analysis

3.1 Requirements

3.1.1 List of functional requirements
3.1.2 List of system-wide non-functional requirements

The functional requirements should be prioritized using the MoSCoW prioritization technique.
A requirements catalogue should be included in an appendix.

3.2 Use Cases

3.2.1 One Use Case model of the system
Use Case descriptions should be included in an appendix.

3.3 Architecture

3.3.1 System Architecture
Interfaces with other systems (human or automated)
An overview of the technical architectures to be used for development and Implementation (DBMS & Web Servers, Client end, Internet Locally)

3.3.2 Initial Class Diagram
It will only show classes and the relationships between them
Some classes may be missing
Some relationships may be unclear

4. Design

4.1 Structural Model

4.1.1 Detailed Class Diagram (Classes, relationships, methods & attributes)

The above must be modelled using UML notation

Detailed class definitions should be included in an appendix.

4.2 Behavioural Model

4.2.1 Sequence Diagrams Or Collaboration Diagrams

Describes the behavior of the actors and classes in YOUR system

The above must be modelled using UML notation

5. Implementation

5.1 Choice of programming language
5.2 System cutover from the development architecture to the implementation architecture
5.3 Data migration from the development architecture and/or existing systems to the implementation architecture
5.4 Training

User manual will be added in the appendix section showing the usability of the system

6. Other Project Matters

6.1 Project Management **(if possible)**
6.2 Risk Management **(if possible)**
6.3 Configuration Management **(if possible)**
6.4 Testing
6.4.1 Unit Testing
6.4.2 Integration Testing

Test scripts & Test results need to be shown in this section.
SAMPLE TEST SCRIPTS are attached herewith.

7. Conclusion

7.1 Strengths and weakness of your system(Whether or not you achieved your aims and objectives)
7.2 Problems Identified & Resolved(What problems occurred and how YOU overcame them)
7.3 Further Improvements (Things that YOU may do differently in any further projects that YOU undertake and YOUR reasons for doing so)

8. References

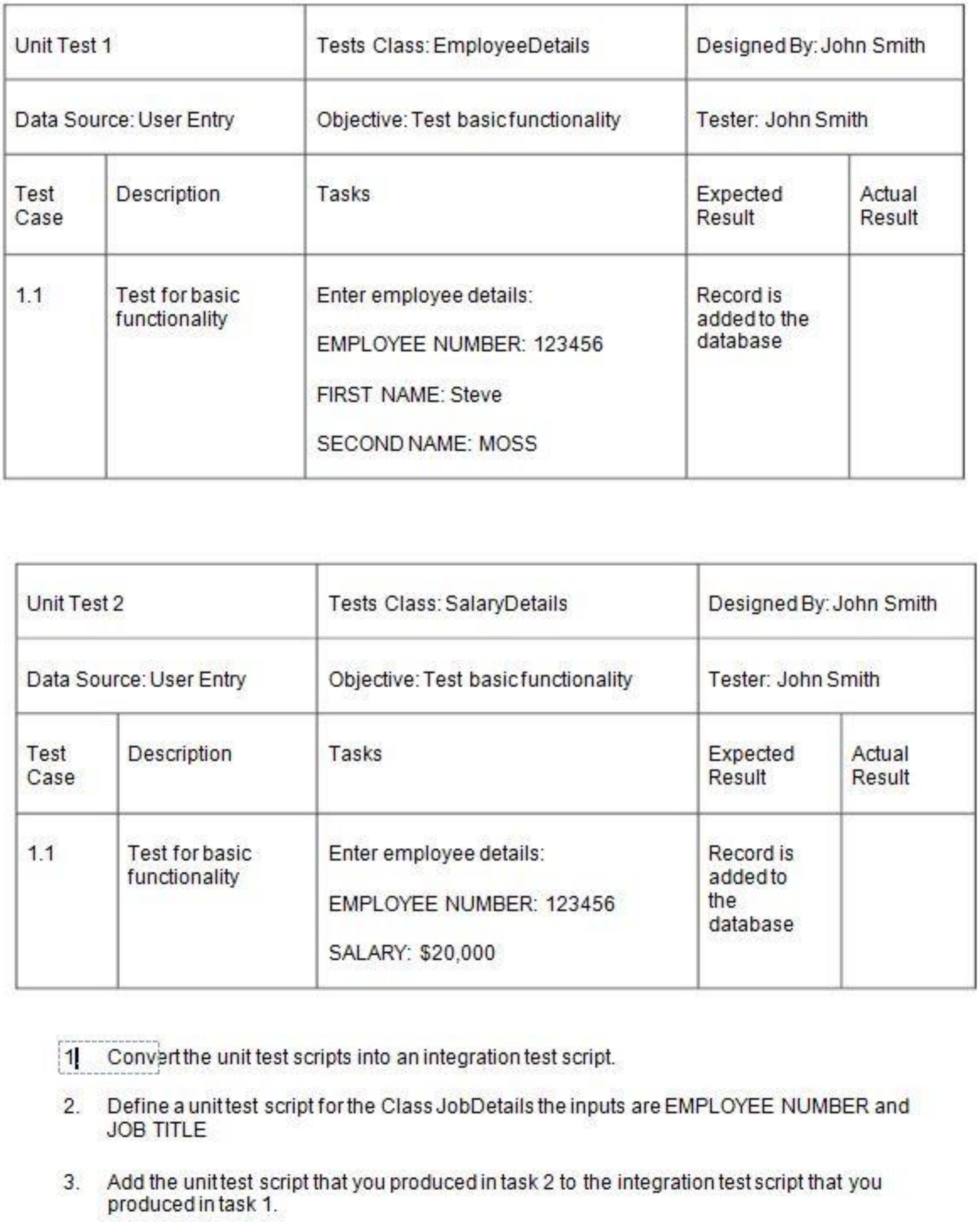
Outline of references

All references you cite within the body of your report should be fully referenced here using the Harvard Style.
No reference should appear here unless it has been cited in the main body of the report.

9. Appendices

9.1 Requirements Catalogue
9.2 Use Case Descriptions
9.3 Detailed Class Definitions
9.4 Test Scripts
9.5 User Guide
9.6 System Code

SAMPLE TEST SCRIPTS



Unit Test 1		Tests Class: EmployeeDetails	Designed By: John Smith	
Data Source: User Entry		Objective: Test basic functionality	Tester: John Smith	
Test Case	Description	Tasks	Expected Result	Actual Result
1.1	Test for basic functionality	Enter employee details: EMPLOYEE NUMBER: 123456 FIRST NAME: Steve SECOND NAME: MOSS	Record is added to the database	

Unit Test 2		Tests Class: SalaryDetails	Designed By: John Smith	
Data Source: User Entry		Objective: Test basic functionality	Tester: John Smith	
Test Case	Description	Tasks	Expected Result	Actual Result
1.1	Test for basic functionality	Enter employee details: EMPLOYEE NUMBER: 123456 SALARY: $20,000	Record is added to the database	

1. Convert the unit test scripts into an integration test script.

2. Define a unit test script for the Class JobDetails the inputs are EMPLOYEE NUMBER and JOB TITLE

3. Add the unit test script that you produced in task 2 to the integration test script that you produced in task 1.