

# Martial Manager: A Comprehensive Martial Arts Management System

# Course

# Module

# Supervisor

# Fullname

# Pnumber

Abstract

**Table of Contents**

1. ***Introduction*** 3
   1. …..
   2. …..
   3. ….
   4. …..
   5. …..
2. ***Project planning*** 10
   1. …..
   2. …..
   3. ….
3. ***Research*** 12
   1. ……..
   2. …….
      1. ……..
      2. ………
   3. ………
4. ***Requirements*** 15
   1. ………
      1. ………
      2. ……..
   2. …….
5. ***Design*** 17
   1. ……………..
      1. ………
      2. ……
   2. …………….
      1. ………..
   3. ……………...
      1. ……..
6. ***Implementation*** 21
   1. …………….
   2. …………..
   3. …………
      1. …..
      2. …..
      3. …..
      4. …..
      5. …..
   4. …………………
   5. ………………….
      1. ……
      2. ……
7. ***Testing*** 36
8. ***Critical analysis*** 37
   1. …………….
   2. ……………
   3. …………..
      1. …….
      2. …….
   4. ……………..
   5. ……………….
   6. …………….
9. ***Conclusion*** 42
10. ***Appendices*** 43

***THESE ARE INDICATIVE YOUR LIST OF APPENDECIES MAY DIFFER so you may or may not have multiple iteration of certain artifacts. All depends on your project. But please include all iteration of your documentation from the first deliverable up until the final to demonstrate any changes that you have made as you when along the process .***

*Appendix 1 – Project Contract* 44

*Appendix 2 – Project Plan* 49

*Appendix 3 – Project Plan Final Iteration* 55

*Appendix 4 – Literature Review* 68

*Appendix 5 – Functional Specification* 80

*Appendix 6 - Functional Specification Final Iteration* 92

*Appendix 7 – Design Documentation* 102

*7a - Information architecture and website flow* 103

*7 b - System architecture* 105

*7 c - Software architecture* 107

*7 d - Website design and wireframes* 108

*Appendix 8 – Initial Database Design* 115

*Appendix 9 - Database Design Final Iteration* 120

*Appendix 10 – Initial Prototype* 126

*Appendix 11 – Progress Reports* 131

*Appendix 12 – Test Plan* 152

*12 a - User Testing* 165

*Appendix 13 – Screenshots of the working system* 169

*Appendix 14 – Software development strategy* 183

*Appendix 15 – User Guide* 186

*Appendix 16 – Code base screenshots* 188

**Chapter 1- INTRODUCTION**

* 1. – Project Background

This section should outline the context and motivation behind the project. It explains the purpose of the platform, the target audience, and the gap in existing solutions that the system aims to fill.

* 1. – System overview

This part provides a high-level description of the system and its core functionalities. It should detail the key features, while also highlighting security measures and access controls. The overview must reflect how the system fulfils the initial contractual requirements.

* 1. – Starting forms

This section describes the initial documentation used to define the project scope, including the ethical review and considerations taken to ensure ethical compliance. It should include details on the contractual requirements, key deliverables, risk analyses, and scheduling tools that were established at the project’s outset. These forms serve as the foundation for the system’s development and guide the overall project direction.

**Chapter 2- PROJECT PLANNING**

#### ****2.1 Software Development Lifecycle and Chosen Methodology****

This section outlines the overall development process adopted for the project. It describes the chosen methodology—whether agile, waterfall, or another approach—and explains how it guided the project through iterative cycles of planning, development, and evaluation.

#### ****2.2 Time Management****

This part details the strategies used to allocate and track time during the project. It covers the planning of a project timeline using tools such as Gantt charts, Github/ version control and any other tools to help track the establishment of key milestones, and the ongoing monitoring of progress against deadlines. The discussion should include how adjustments were made in response to delays or unexpected challenges, ensuring that the project remained on track.

#### ****2.3 Project Scope****

This section defines the boundaries and objectives of the project. It explains the core functionalities and any advanced features that were intended for the final system, delineating what was within scope and what was considered out of scope.

**Chapter 3 – RESEARCH**

This chapter should provide an overview of the literature relevant to the project, synthesizing key theories, methodologies, and findings that have informed the project’s approach. It should summarize the main points from the literature review, discuss how existing research relates to the project objectives, and identify any gaps that the project aims to address. Detailed references and additional background material should be included in an appendix for further context.

* 1. Subtitle
  2. Subtitle
     1. – subsection heading
     2. – subsection heading
  3. Subtitle

**Chapter 4- REQUIREMENTS**

The **Requirements** chapter should provide a clear outline of the system’s functional and non-functional requirements. It should begin by listing the **core functionalities** essential to the system, referencing any contractual requirements or initial specifications. Additionally, any **advanced or additional features** implemented beyond the core scope should be highlighted.

Each **core use case** should be described in detail, explaining its purpose and how it was implemented. Screenshots should be included to support explanations and demonstrate key functionalities. Any **changes** made to the original requirements during development should be discussed, including:

* What **modifications** were made from the initial requirements to the final system.
* **Why** these changes were necessary (e.g., feasibility, user feedback, technical constraints).
* **How** these changes improved or impacted the system.

4.1 Subtitle

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx(see fig 1.) xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

***Figure 1:*** *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

**Chapter 5 - DESIGN**

This section of the report should provide an overview of the system's architecture, including initial design considerations and how they evolved throughout the project. It should cover key components such as **System Architecture**, explaining the overall structure (e.g., MVC framework) and the relationship between front-end and back-end elements. It should also reflect upon the UML diagrams(e.g. class diagram or how object-oriented principles were applied.) The **Database** subsection should discuss the Entity-Relationship Diagram (ERD), table structures, and changes made during development. Lastly, the **User Interface** section should outline initial UI designs, tools used for prototyping, and any modifications based on usability needs. The report should emphasize how iterative improvements were made to the design of both front and back end of your system.

If the system includes dynamic features (e.g., reducing page redirects, using JavaScript for interactivity), these **design choices** should be explained, along with how they contribute to usability and performance. The chapter should provide a **holistic view** of how the requirements evolved and were met in the final product.

#### CHAPTER 6 - IMPLEMENTATION

#### This chapter should detail the development process, highlighting key functions and complex components with supporting code snippets and reference screenshots of the working system, and explain how challenging issues were resolved during system development. It should form the bulk of your report.

#### CHAPTER 7 – TESTING

#### This chapter should outline the testing methodologies used—such as user testing and black-box testing—detailing test cases for each use case and referencing the full test plan in the appendix.

#### CHAPTER 8 – CRITICAL ANALYSIS

#### This chapter should offer a holistic evaluation of the project, discussing what aspects worked well, what challenges were encountered, and suggesting areas for future improvement and measure project success and if scope was met.

#### CHAPTER 9 – CONCLUSIONS

#### This chapter should summarize the overall outcomes of the project, reflecting on the objectives achieved, the lessons learned throughout development, and potential directions for future work.

#### Appendices

*Appendix 1 –* Project Contract

*Appendix 2 –* Project Plan

*Appendix 3 –* Project Plan Final Iteration

*Appendix 4 –* Literature Review

*Appendix 5 –* Functional Specification

*Appendix 6 -* Functional Specification Final Iteration

*Appendix 7 –* Design Documentation

7 a - Information architecture and website flow

7 b - System architecture

7 c - Software architecture

7 d - Website design and wireframes *Appendix 8 –* Initial Database Design

*Appendix 9 -* Database Design Final Iteration *Appendix 10 –* Initial Prototype

*Appendix 11 –* Periodic Progress Reports

*Appendix 12 –* Test Plan

12 a - User Testing

*Appendix 13 –* Screenshots of the working system *Appendix 14* – Software development strategy *Appendix 15 –* User Guide

Appendix 16 – Code base screenshots

Appendix 1 - **Project Contract**

**INSERT PROJECT CONTRACT**

Appendix 2 - **Project plan**

*This is an extension of the gannt chart/ project plan you have created on your project contract.*

*It should include:*

* *an introduction*
* *a section on your chosen software development lifecycle methodology*
* *a task and dependencies table(which can be taken from your project contract)*
* *optionally a gannt chart too if you have made one/ screenshots of any applicable project management software you may be using e.g. github, Trello, miro, jira etc.*

*Optionally you can include some or all of the following sections:*

* *a project justification*
* *scope*
* *stakeholders*
* *list of deliverables*
* *project success factors*

***Do some research into project plan documents to get some inspiration.***

Appendix 3 **Project plan** **Final iteration**

**INSERT PROJECT PLAN FINAL ITERATION(updated gannt chart)**

Appendix 4 **Literature review**

* *min 1800 and max 2500 words excluding reference list*
* *needs to be Harvard referenced please see the following for guidance -* [Harvard - DMU Referencing Guide - LibGuides at De Montfort University](https://library.dmu.ac.uk/refguide/harvard)

Appendix 5 – **Functional Spec**

***research into functional spec documents to see what sorts of things to include, but ideally you should include:***

* *Introduction - a brief overview of system and what its aims are.*
* *brief overview of the technical requirements of the system and touch on business need of the project*
* *System actors description*
* *Use case summary – i.e list of of all use case names (split into core and non core)*
* *Use case descriptions*
* *Use case UML diagram*

**Use case notation example (you will need one for each use case description):**

|  |  |  |
| --- | --- | --- |
| ***Use Case Name:*** | *Register as User* | |
| ***Primary Actor:*** | *User* | |
| ***Business Goal:*** | *To register yourself as a user on the system* | |
| ***Precondition*** | *Must be a student and own a vehicle* | |
| ***Success Condition*** | *A student is now registered on the system* | |
| ***Main Path*** |  | |
| 1, Nonregistered user clicks ‘Register’ on the Index page | | |
| 2. System will display Register page. | | |
| 3. User enters information into the fields and presses Sign Up. | | |
| 4. System redirects the newly Registered user to the Log in page. | | |
| ***Variant Paths*** |  | **Result** |
| Step 3. User enters invalid data | | System will display validations and request information to be re-entered. |
| Step 4. Database/ server fault | | System will display error message to user and refresh redirecting them to start registration again |

Appendix 6 – **Design Docs**

**You could use some or all of these, this will depend on your own individual system and the technologies and approach you are using.**

Some examples of documentation to include:

* Information architecture and website flow diagram
* System architecture/ Software architecture
* Wireframes
* Data flow diagram
* Class diagram
* Sequence diagram
* Database design
  + Overview
  + Table types
  + Entity relationship diagram

**And so on…**