

**Software Engineering and Testing. BSC Year 2, 2020/2021**

**(Assignment 3 - 20%)**

**Assessment 3: Design and Draft Implementation**

**Submitted by: Names, Student numbers**

**Submission date**

**Declaration**

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of an Ordinary Degree in Computing in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated.

Author: Habiba Nour Dated: 18/MARCH/2024

Author: Steven Kelly Dated: 18/MARCH/2024

Author: Rochelle Mullen Dated: 18/MARCH/2024

Author: Piotr Momat Dated: 18/MARCH/2024

# Abstract:

The below document will outline the methodology, structure, and development plan of the pet service website we are building. Using PHP, CSS, HTML, and MySQL, we aimed to build a system that provides a user-friendly website that will allow them to book our services or allow businesses to offer the services. The website contains a booking system, a login/sign-up system, a review page, and a contact us page. This document will break down the website’s structure, ERD design, UML design, user interfaces and use cases.

# Project Definitions

**2.1** The purpose of this document is to give an overall outlook and design structure of our website. Below you will see the design and implementation of the website including UML design, ERD design, database implementation and a general outline of the functionality of the website.

**2.2** The project we are designing is a front-end website for a local pet walking, pet sitting and daycare business. Users will be able to log in, book the service they like or leave reviews. Business operators will be able to manage times and bookings. The administrator will have full privileges of the entire website.

**2.3** The website has a few elements to get it to work, but our main priorities are ease of use and simplicity for the front end. We believe this is important as the user needs to have a clear understanding of the services provided and the layout of the site. Also, for ease of use, we have databases connected to the system to manage user and other site data. We are using MySQL to write the database, and PHP, CSS, and HTML are handling the scripting and front end of the website.

# 2.4 The software system has 2 main components, front-end and back-end. The front end is a simple design. It has three squares on the homepage which will bring the user to the area they would like (those being pet sitting, daycare and walking). When the user picks the service, they would like they will be brought to a booking page of that service. From the business end, they will be able to view their bookings and edit their available slots which site users with an account are able to see.

# Document Revision

**Rev**. *2.0* **Date:** *18/MARCH/2024*

# Methodology

**4.1** System models – UML (Habiba)

**4.2 Use of, and necessity of OOAD (Piotr)**

Object-Oriented Analysis and Design (OOAD) is crucial in software engineering for developing complex systems by breaking them down into small pieces. It enables the understanding of system requirements and defines objects, facilitating robust, scalable, and maintainable software.

**4.3** What is a class diagram? (Rochelle)

A class diagram is a visual representation of structure of the system you are creating showing the classes as well as the attributes and methods of these classes. It also shows the relationships between classes. Class diagrams can also be used to help develop ERD’s needed for the database that the systems will need. Class diagrams can be used in many of the stages of developing software such as helping you find out what are the main components of your software and understand the requirements needed for the system.

**4.4** Static Versus Dynamic Case Diagrams? (Rochelle)

Static case diagrams are only show which entities interact with each. They don’t show what occurs when this interaction happens. Dynamic diagrams do show what happens when they interact.

**4.5 What is an ERD? (Piotr)**

An Entity-Relationship Diagram (ERD) is a tool in database design, that shows the structure of data and how different entities relate to each other. Typically, we would use symbols such as rectangles to represent entities, diamonds to represent relationships, and lines to connect entities and relationships.

**4.6** Purpose of using classes? (Habiba)

**4.7** Volatile versus Persistent storage – Object Instances / Database? (Steven)

**4.8** User Interface template chosen and how it can aid in executing the functional specification of the project. (Steven)

1. **Requirements (Habiba)**

4.1 Use Cases

4.2 Use Case Specifications

(Specifically – how use case specifications have been used as a means to develop the classes***/attributes/methods*** and database ***tables***)

1. **Case Diagrams (Rochelle)**

**Class Diagram** – Show all relationships, multiplicities, associations, generalisations (inheritance), aggregations (compositions) - See lecture 4.

Paragraph to explain ALL design decisions.

**Entity Relationship Diagram** – Show all relationships, multiplicities,

**Conceptual Diagram**:A diagram of a company

Description automatically generated

**Physical ERD of DATABASE**:

A diagram of a computer

Description automatically generated

Paragraph to explain ALL design decisions.

# Conclusions

**7.1** We have progressed a lot with our project. Successfully developed the database structure and class diagrams which gave us a clear system architecture. The front-end interface has most of the website functionalities, the same as the back end.

**7.2** Minor adjustments were made to the original proposal where we optimised data management like DB tables and new variables.

**7.3** Overall, the project has not changed a lot in architecture. However, we have improved a lot in the development of the website by sticking to our original goals. We're dedicated to adding the rest of the features and testing everything properly to make sure our pet service website works great.