

Title of project

**Your name Briain Keddy**

**Your student number n00191932**

Software Project

CA 2 – Develop a PHP shopping cart website

Year 2 2020-21

DL836 BSc (Hons) in Creative Computing

Table of Contents

[1 Introduction 1](#_Toc94698843)

[2 Business Concept 2](#_Toc94698844)

[2.1 Business Idea 2](#_Toc94698845)

[2.2 Business model 2](#_Toc94698846)

[2.3 Market Research 2](#_Toc94698847)

[2.4 Marketing/Advertising 2](#_Toc94698848)

[2.5 Suppliers 2](#_Toc94698849)

[2.6 Competitors 2](#_Toc94698850)

[2.7 Employees 2](#_Toc94698851)

[2.8 Environmental Impact 2](#_Toc94698852)

[3 Requirements 3](#_Toc94698853)

[3.1 Introduction 3](#_Toc94698854)

[3.2 Requirements gathering 3](#_Toc94698855)

[3.2.1 Similar applications 3](#_Toc94698856)

[3.2.2 Interviews 3](#_Toc94698857)

[3.3 Requirements modelling 3](#_Toc94698858)

[3.3.1 Functional requirements 3](#_Toc94698859)

[3.3.2 Non-functional requirements 3](#_Toc94698860)

[3.3.3 Use Case Diagrams 4](#_Toc94698861)

[3.4 Feasibility 5](#_Toc94698862)

[4 Database Design 6](#_Toc94698863)

[4.1 Description 6](#_Toc94698864)

[4.2 Business Reporting Requirements 6](#_Toc94698865)

[4.3 Textual Representation of Data-Set 6](#_Toc94698866)

[4.4 Business Rules 7](#_Toc94698867)

[4.5 Entity Relationship Diagram 7](#_Toc94698868)

[4.6 Tables 8](#_Toc94698869)

[4.7 Database Dictionary 9](#_Toc94698870)

[5 System Design/ Architecture Overview 10](#_Toc94698871)

[5.1 Introduction 10](#_Toc94698872)

[5.2 Model View Controller 10](#_Toc94698873)

[5.3 User Authenticaion 10](#_Toc94698874)

[5.4 Routing 10](#_Toc94698875)

[5.5 Templating 10](#_Toc94698876)

[6 Testing 11](#_Toc94698877)

[6.1 Introduction 11](#_Toc94698878)

[6.2 Functional Testing 11](#_Toc94698879)

[6.2.1 Login/Registration 12](#_Toc94698880)

[6.2.2 Navigation 12](#_Toc94698881)

[6.2.3 Calculation 12](#_Toc94698882)

[6.2.4 CRUD 12](#_Toc94698883)

[6.3 Discussion of Functional Testing Results 13](#_Toc94698884)

[6.4 User Testing 13](#_Toc94698885)

[6.5 Conclusion 13](#_Toc94698886)

[7 Project Management 14](#_Toc94698887)

[7.1 Introduction 14](#_Toc94698888)

[7.2 Project Phases 14](#_Toc94698889)

[7.2.1 Requirements 14](#_Toc94698890)

[7.2.2 Design 14](#_Toc94698891)

[7.2.3 Implementation 14](#_Toc94698892)

[7.2.4 Testing 14](#_Toc94698893)

[7.3 SCRUM Methodology 14](#_Toc94698894)

[7.4 Project Management Tools 15](#_Toc94698895)

[7.4.1 Github Project 15](#_Toc94698896)

[7.4.2 GitHub 15](#_Toc94698897)

[8 Reflection 16](#_Toc94698898)

[8.1 Your views on the project 16](#_Toc94698899)

[8.2 How could the project could be developed further? 16](#_Toc94698900)

[8.3 Assessment of your learning. 16](#_Toc94698901)

[8.4 Completing a large software development project 16](#_Toc94698902)

[8.5 Technical skills 16](#_Toc94698903)

[8.6 Further competencies and skills 16](#_Toc94698904)

# Introduction

Overall aim

Application area

Technologies

PHP, MySQL, Bootstrap, CSS, Vanilla

Tools

IDE, phpMyAdmin, Miro

Project management

GitHub

Business Concept

Requirements

Design

Implementation

Testing

Reflection

# Business Concept

## Business Idea

Online open forum. Users can post their content to the open forum for others to see. Content such as photos, links, opinions, long form text.

## Business model

The business can display advertisements on their forum

## Market Research

Market for Product/Service

Customers - Demographics, Profile

## Marketing/Advertising

## Suppliers

## Competitors

## Employees

## Environmental Impact

# Requirements

## Introduction

The purpose of the requirements phase is to allow for developers to work out what the application should be able to do. It is important to understand what the users would like the application to do rather than the developer deciding what is required.

You can write a bit about your project area. Each paragraph has a blank line between it and the previous paragraph

## Requirements gathering

### Similar applications

Look at and document 2 similar applications. Be sure to include the following for each:

* Screen shots
* Descriptions
* Advantages
* Disadvantages

### Interviews

Conduct interviews with 2 or 3 users to find out what the important features for them for the app are. There may be various issues that arise in multiple interviews. These can be grouped together into a number of themes.

## Requirements modelling

### Functional requirements

Create a numbered list of what the application should be able to do. Start with the most important feature.

### Non-functional requirements

These are requirements which if not met do not stop the application from working, but which mean that the application is not working as well as it should. They are usually based on issues such as:

* Usability
* Performance
* Security

### Use Case Diagrams

Consists of actors and use cases. You should document each individual use case.

Delete the following diagram and insert your diagram. Use draw.io



## Feasibility

This section describes which technologies are planned to be used in the development of the application. It then explains if there are any issues in terms of the technical feasibility of the project, for example, if there are two different types of software which may have compatibility issues.

# Database Design

## Description

A company has a website that sells video games for different consoles. They would need a database for all their games and order places. For each order place, they would need games bought, total price, date of the order, and how long it will take to deliver. The database needs to keep track of all games that are being sold. Customers will have to input their information when registering an account. Customers will also have to input their card details when making a payment for their order.

## Business Reporting Requirements

Substitute in here the information the users of your application will want to be able to view.

1. Organisers need to be able to create, read, update, and delete: festivals, stages, shows, performers, and genres.
2. Users will need to be able to find all festivals ordered by their start date.
3. Users may want to find a festival by a specific start date.
4. Users need to find all festivals using a list of genres.
5. Users need to find the stage for a specific show.
6. Users need to find the shows using a performers name.
7. Performers may need to find the list of festival contacts.
8. Users need to find festivals by location and the location needs to be displayed on a Google Map
9. User may need to find festivals by city
10. Users need to find stages within a festival by the stage’s location
11. Organisers need to display a list of employees that are assigned to a specific festival

## Textual Representation of Data-Set

Substitute in here the tables for your database

**FESTIVAL** (title, description, latitude, longitude, city, start\_date, end\_date, image\_id)

**PERFORMER** (title, description, contact\_email, contact\_phone, image\_id)

**GENRE** (title, description)

**IMAGE** (id, filename)

**SHOW** (date, start\_time, end\_time, performer\_id, stage\_id)

**STAGE** (title, description, location, festival\_id, image\_id)

**GENRE**\_**PERFORMER** (id, genre\_id, performer\_id)

**EMPLOYEE** (name, phone, email)

**FESTIVAL\_EMPLOYEE** (employee\_id, festival\_id, role)

## Business Rules

Substitute in here the business rules for your database

 A **Festival** has many **Stages**.

 A **Stage** belongs to one **Festival**.

 A **Stage** hosts many **Shows**.

 A **Show** is performed on one **Stage**.

 A **Performer** can perform in many **Shows**.

 A **Show** is performed by one **Performer**.

 A **Performer** can have many **Genres**.

 A **Genre** can belong to many **Performers**.

 A **Performer** can have a single **Image**.

 A **Festival** can have a single **Image**.

 A **Stage** can have a single **Image**.

 An **Image** can be associated with a **Performer**, **Festival**, or **Stage**

 A **Festival** can have many **Employees** associated with it

 An **Employee** can be assigned to one **Festival** at a time

## Entity Relationship Diagram

Substitute in here your ERD from draw.io



## Tables

Substitute in here your tables and the relationships between tables from draw.io in the format you used in DBMS with Mohammed.



## Database Dictionary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table | Attribute | Datatype | Range | Required | PK/FK | FK Ref Table |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# System Design/ Architecture Overview

* 1. Introduction

This section will describe the internal functionality of the web framework that you have chosed for the implementation. Add further sections if required by the specification of your web application

* 1. Model View Controller

Explain the follows a model-view-controller design pattern and how it is implemented in your web application.

* 1. User Authenticaion

Explain how user authentication is implemented in the web application framework.

* 1. Routing

Describe the routes that were defined in the web application

* 1. Templating

Describe the templating engine and how it was used to configure/ style the web application.

Add a sequence diagram in this section and other diagrams that illustrate the architecture clearly.

Diagram

Description automatically generated

# Testing

* 1. Introduction

This chapter describes the testing that has been undertaken for the application. This chapter is presented in two sections:

1. Functional Testing
2. User Testing

Functional testing is a type of software testing whereby the system is tested against the functional requirements. The app is tested by looking to see if the actual output for a given input corresponds with the expected output. The tests should be based on the requirements for the app. The results of functional testing can indicate if a piece of software is functional and working, but not if the software is easy to use.

User testing looks to see if a piece of software is easy and intuitive for the user.

* 1. Functional Testing

This section describes the functional tests which were carried out on the app. These functional tests can be categorised as: (whatever is relevant to your app)

Login/Registration

Navigation

Calculation

CRUD

Functional testing generally uses a Black Box Testing technique which means that the internal logic of the system being tested is not of interest to the tester. The tester is only interested in whether the actual output agrees with the expected output.

* + 1. Login/Registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + 1. Navigation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + 1. Calculation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + 1. CRUD

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Discussion of Functional Testing Results

Describe the results from the tests. Address any functionality where unexpected behavior could not be debugged.

* 1. User Testing
  2. Conclusion

Discussion of test results.

# Project Management

## Introduction

This chapter describes how the project was managed. It shows the phases of the project, going from the project idea through the requirements gathering, the specification for the project, the design, implementation and testing phases for the project. It also discusses GitHub as a tool which assist in project management.

## Project Phases

In this section, describe each of the following project phases. Explain any issues which arose for each of the phases.

### Requirements

### Design

### Implementation

### Testing

## SCRUM Methodology

Sprints

## Project Management Tools

### Github Project

Description

Include screen shots

How it worked in practice

### GitHub

Description

How it is used

How it worked in practice

# Reflection

## Your views on the project

Describe how you feel the project went from your perspective.

## How could the project could be developed further?

## Assessment of your learning.

Critically assess your learning. List what skills and competencies you have learned developed in this Continuous Assessment.

List which part of the project would need further development and itemize where you feel you have not satisfactorily completed the continuous assessment.

## Completing a large software development project

Describe what you have learnt from the project, from the point of view of completing a large software development project.

## Technical skills

Describe what you have learnt from the project, from a technical skills viewpoint.

## Further competencies and skills

Describe any extra competencies and skills that would help you with your development in the work place.

# References

Add a list of references that you used to complete the project.

The Department of Technology and Psychology in IADT uses APA 7th referencing style.

Use alphabetical order for your references.

This site gives details about how to cite websites using APA:

https://www.wikihow.com/Cite-a-Website-in-APA

The following is a useful site for creating citations for APA for websites.

<http://www.citationmachine.net/apa/cite-a-website>

You can also use the Referencing tab within Microsoft Word to enter reference information manually. Word then creates an APA style reference.