

Software Project

**Your name**

**Adam Gallagher**

**Your student number**

**N00211418**

Software Project

Online shop for bidding on second hand guitars

Year 2 2022-23

DL836 BSc (Hons) in Creative Computing

Link to resources created as part of the project.

|  |  |
| --- | --- |
| GitHub | https://github.com/y2-SW-project/swproject23-AdamGallagher27 |
| Video | Link to your video file (MS Stream, YouTube) |

Table of Contents

[1 Introduction 1](#_Toc96009471)

[2 Business Concept 2](#_Toc96009472)

[2.1 Business Idea 2](#_Toc96009473)

[2.2 Business model 2](#_Toc96009474)

[2.3 Market Research 2](#_Toc96009475)

[2.4 Marketing/Advertising 2](#_Toc96009476)

[2.5 Suppliers 2](#_Toc96009477)

[2.6 Competitors 2](#_Toc96009478)

[2.7 Employees 2](#_Toc96009479)

[2.8 Environmental Impact 2](#_Toc96009480)

[3 Requirements 3](#_Toc96009481)

[3.1 Introduction 3](#_Toc96009482)

[3.2 Requirements gathering 3](#_Toc96009483)

[3.2.1 Similar applications 3](#_Toc96009484)

[3.2.2 Interviews 3](#_Toc96009485)

[3.3 Requirements modelling 3](#_Toc96009486)

[3.3.1 Functional requirements 3](#_Toc96009487)

[3.3.2 Non-functional requirements 3](#_Toc96009488)

[3.3.3 Use Case Diagrams 4](#_Toc96009489)

[3.4 Feasibility 4](#_Toc96009490)

[4 Web application Design 6](#_Toc96009491)

[4.1 Layout 6](#_Toc96009492)

[4.2 Interaction 6](#_Toc96009493)

[4.3 Colour schemes 6](#_Toc96009494)

[4.4 Font choices 6](#_Toc96009495)

[4.5 Wireframes 6](#_Toc96009496)

[5 Database Design 7](#_Toc96009497)

[5.1 Description 7](#_Toc96009498)

[5.2 Business Reporting Requirements 7](#_Toc96009499)

[5.3 Textual Representation of Data-Set 7](#_Toc96009500)

[5.4 Business Rules 8](#_Toc96009501)

[5.5 Entity Relationship Diagram 8](#_Toc96009502)

[5.6 Tables 9](#_Toc96009503)

[5.7 Database Dictionary 10](#_Toc96009504)

[6 System Design/ Architecture Overview 11](#_Toc96009505)

[6.1 Introduction 11](#_Toc96009506)

[6.2 Model View Controller 11](#_Toc96009507)

[6.3 User Authenticaion 11](#_Toc96009508)

[6.4 Routing 11](#_Toc96009509)

[6.5 Templating 11](#_Toc96009510)

[7 Testing 12](#_Toc96009511)

[7.1 Introduction 12](#_Toc96009512)

[7.2 Functional Testing 12](#_Toc96009513)

[7.2.1 Login/Registration 13](#_Toc96009514)

[7.2.2 Navigation 13](#_Toc96009515)

[7.2.3 Calculation 13](#_Toc96009516)

[7.2.4 CRUD 13](#_Toc96009517)

[7.2.5 Discussion of Functional Testing Results 14](#_Toc96009518)

[7.3 User Testing 14](#_Toc96009519)

[7.4 Conclusion 14](#_Toc96009520)

[8 Project Management 15](#_Toc96009521)

[8.1 Introduction 15](#_Toc96009522)

[8.2 Project Phases 15](#_Toc96009523)

[8.2.1 Requirements 15](#_Toc96009524)

[8.2.2 Design 15](#_Toc96009525)

[8.2.3 Implementation 15](#_Toc96009526)

[8.2.4 Testing 15](#_Toc96009527)

[8.3 SCRUM Methodology 15](#_Toc96009528)

[8.4 Project Management Tools 16](#_Toc96009529)

[8.4.1 Github Project 16](#_Toc96009530)

[8.4.2 GitHub 16](#_Toc96009531)

[9 Reflection 17](#_Toc96009532)

[9.1 Your views on the project 17](#_Toc96009533)

[9.2 How could the project could be developed further? 17](#_Toc96009534)

[9.3 Assessment of your learning. 17](#_Toc96009535)

[9.4 Completing a large software development project 17](#_Toc96009536)

[9.5 Technical skills 17](#_Toc96009537)

[9.6 Further competencies and skills 17](#_Toc96009538)

[10 References 18](#_Toc96009539)

# 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

My business idea is an online shop that allows users to sell their guitars or start bids for them. This will give an alternative and sustainable option for buying guitars instead of relying on buying guitars new from larger companies like Epiphone or fender.

## Business model

The business model is loosely based off how Depop works. Depop allows users to sell their clothes online and they take a commission off every transaction. My website would work in the same way.

## Market Research

The musical instrument business was worth USD 14.20 billion in 2022 and is expected to grow 7.4% from 2023 to 2030. There is a gap in the market for users to find and sell sustainably sourced guitars.

The most obvious demographic for guitars is musicians but there is also a market for music shops that don’t have their own online store. These two demographics make up the majority of the users for this platform. It also goes without saying not all customers are musicians or shop owners, some may just be people buying an instrument as a gift.

## Marketing/Advertising

I would conduct market research to see how I could connect with musicians online. Another strategy I considered is getting a music influencer or Youtuber to endorse my product.

## Suppliers

There is no need for suppliers for this business as the users are their own suppliers as they are the ones selling and buying their own guitars.

## Competitors

The major competitors in this space are the traditional instrument distributors like fender or Epiphone. They have a major market share but most of them don’t have an option to purchase second-hand guitars, their goal is to sell their new products.

## Employees

As I mentioned in the supplier’s section where I said the customers would handle supplying their own guitars.

## Environmental Impact

While guitars are not the worst contributor to climate change, they are far from perfect. Some of the most endangered tree species are used for creating guitars for example Honduran rose wood or Pau Brazil. This store hopes to reduce the environmental impact.

From manufacturing new guitars.

# Requirements

## Introduction

(Not sure what to write here)

## Requirements gathering

For my requirement gathering I did competitor analysis on fender and gear for music.

I also did two interviews with two musicians I know. They were the ones who recommended I look at fender and gear for music.

### Similar applications

The first competitor I looked at was a more traditional guitar shop. Fender is the world’s largest seller of guitars. The main advantage of buying from Fender is you can trust the quality of their instruments are superb and you can guarantee what they advertise is accurate to the product. A major negative feature of fender is the products are sustainably sourced and there is no option to sell or buy second hand instruments.

A group of guitars

Description automatically generated with medium confidence

Figure 1: fender shop page

Their website design is very sleek and minimalistic. It is very easy to navigate and understand every detail about the product you are buying see figure 1.

A picture containing diagram

Description automatically generated

Figure 2: single product page fender

You can see again in figure 2 fenders minimalistic design. Figure two also highlights two of the issues with Fender. You can also see that the finger board is made from ebony wood which is an endangered wood. Despite the environmental impact and the lack of a second hand shop feature the fender website is very well designed website that is intuitive and aesthetic.

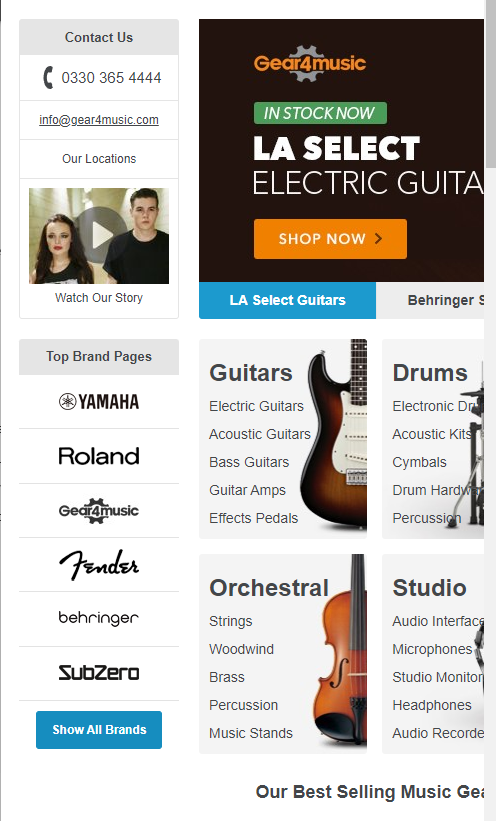


Figure nonresponsive web design

The second competitor I analysed was gear for music. Gear for music is one of the biggest online vendors of instruments in Europe. They have a huge variety of stock, and their website is very easy to use. Despite the websites size and practical functionality, the website is not designed very well and feels very clustered at parts see figure 4. The major draw back with gear for music is that their website is not responsive. When I scaled the website to a mobile size the components were overlapping and made it hard to navigate see figure 3.

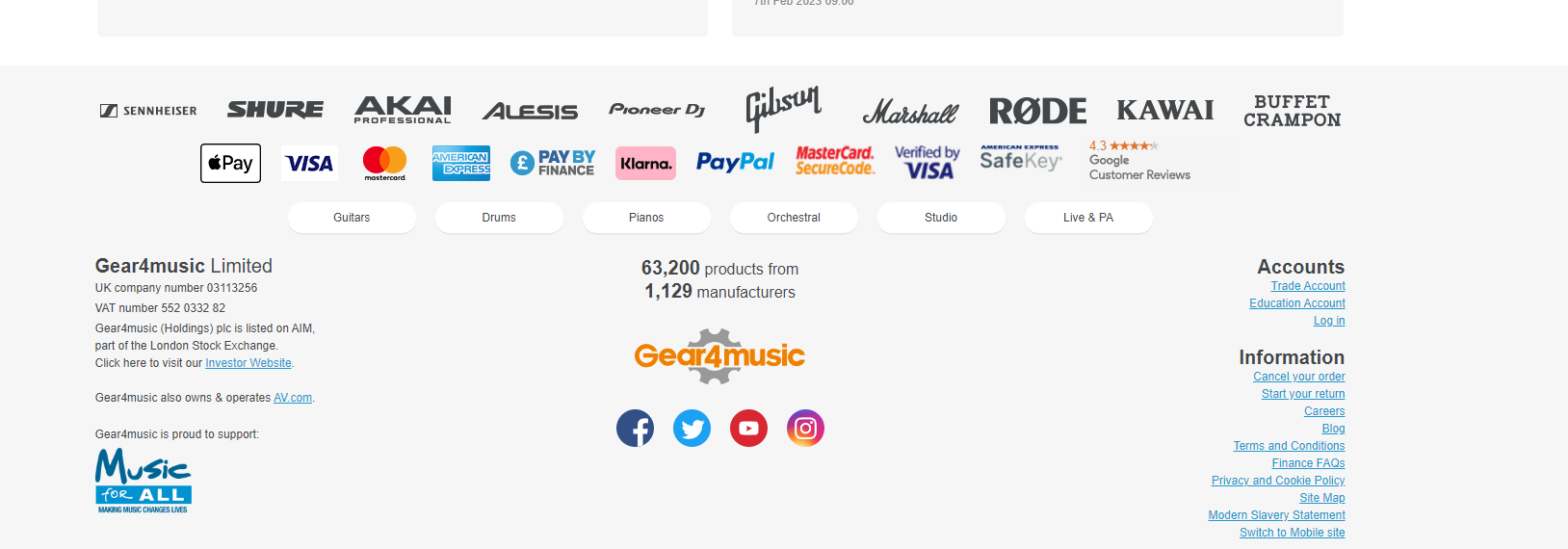


Figure 4 over cluttered footer

The best part of this design was in the product page. While the aesthetic was not perfect it was very functional and easy to make a purchase and see the specifications of each product see figure 5.

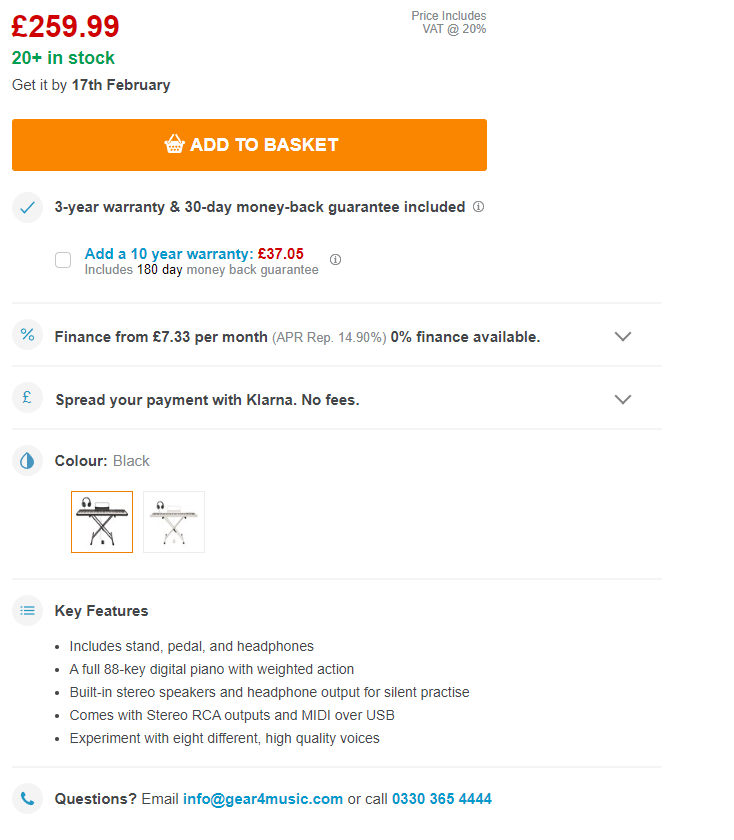


Figure 5 gear for music product purchase

The main problem with the aesthetic in my opinion is the colour palette. The red of the price doesn’t pop enough and the green that symbolises the amount left in stock clashes with the other colours on the page.

### Interviews

Graphical user interface, text, application, email

Description automatically generated

Figure 6: interview questions

I conducted two interviews with two musicians I knew. I prepared five questions for the interviews see figure three.

Question 1:

For the first questions the first user said they buy their instruments in their local music shop when the other users cited the fender website or gear for music for all their instruments.

Question 2:

The first user said they prefer buying in shop because they can test the instrument before buying it. The second user likes the convenience of shopping online and there generally very quick at delivering whatever she ordered.

Question 3:

Both users had the same answer that they wouldn’t mind bidding, but they would still like an option to buy the product outright. The first user reiterated that knowing the quality or sound of the instrument would help them make a purchase.

Question 4:

The first user said they want it to be easy to use and give many options for different types of instruments. The second user likes when the website gives detailed explanations of the specifications of each instrument.

Question 5:

Both users had the same answer again. They both disliked that you are never certain as to what you are buying, and quality may vary from different online vendors.

## Requirements modelling

### Functional requirements

1. Users can buy guitars.
2. Set up a shop and sell guitars.
3. Make bids or buy outright.
4. Admin uses can do crud on all post’s comments.
5. Comment on posts.
6. Wishlist / favourite guitars.

### Non-functional requirements

The three primary non-functional issues are Usability, Performance and Security.

I want my website to be as user friendly as possible I plan to do user tester to try remedy any issues in this section. I will eager load all my data and optimize my code to improve performance. Security is a very important issue when it comes to ecommerce websites. I would use a trusted API for handling purchases like snipcart.

### Use Case Diagrams

Diagram, engineering drawing

Description automatically generated

## Feasibility

Laravel 8.1

Livewire 2.6

Bootstrap 5.3

Php 8.1

JavaScript ES13

PhpMyAdmin 4.9

# Web application Design

## Layout

For my layout I will be using bootstrap and livewire to have my design be responsive and have some of my components be reactive for example the like button and the comments on each post. I aim to have a minimalistic design that is heavily inspired by the depop and fender websites.

## Interaction

The main feature of my website will be the ability to make posts (putting on instruments for sale) and the ability to make bids on them. For the user to make a post they would have to fill out a form which would gather information about the guitar they are selling. Examples of data would be the make, the condition, the colour. I want the bidding to be reactive. The product will list what the top bid is currently, and the user can raise that by putting in their own bid. This will update the database / current highest bid without a page refresh using livewire.

## Colour schemes

Describe the colour palette that you will use consistently across the web application

## Font choices

Specify the fonts that you will use for different types of text. Include samples for paragraph text, headings and bold and italicised text.

## Wireframes

Describe how to navigate from one page to the next by adding a diagram of the different screens and what the main functionality is.



# 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 the performer's 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. Users 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 Dataset

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 chosen 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 Authentication

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 assists 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

Include a Gantt chart



## SCRUM Methodology (optional)

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 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 workplace.

# 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.