

**Book Aid**

Giorgia Cavalleri

N00230352

Software Project

Year 2 2024-5

DL836 BSc (Hons) in Creative Computing

Link to resources created as part of the project.

|  |  |
| --- | --- |
| GitHub | <https://github.com/n00230352/software_project.git> |
| Trello | <https://trello.com/b/o26EtsQH/gc-software-project> |
| Figjam | [https://www.figma.com/board](https://www.figma.com/board/RbFlLhkgmsNaQa1fXSv04w/giorgia-cavalleri?t=81osSKZTwCjkpKAp-6) |
| Figma | [https://www.figma.com/design/](https://www.figma.com/design/27t7hUSWLlH0yJbolpy7ec/software-project_giorgia-cavalleri?node-id=0-1&t=81osSKZTwCjkpKAp-1) |
| Screencast | Link to your Screencast |

Table of Contents

[1 Introduction 1](#_Toc190685858)

[1.1 Description 1](#_Toc190685859)

[1.2 Overview 1](#_Toc190685860)

[2 Research 1](#_Toc190685861)

[2.1 Introduction 1](#_Toc190685862)

[2.2 Summary of specific SDG 2](#_Toc190685863)

[3 Requirements 2](#_Toc190685864)

[3.1 Introduction 2](#_Toc190685865)

[3.2 Requirements gathering 3](#_Toc190685866)

[3.2.1 Similar applications 3](#_Toc190685867)

[3.2.2 Interviews and Surveys 3](#_Toc190685868)

[3.2.3 User Personas and User Stories 5](#_Toc190685869)

[3.3 Requirements modelling 5](#_Toc190685870)

[3.3.1 Functional requirements 5](#_Toc190685871)

[3.3.2 Non-functional requirements 5](#_Toc190685872)

[3.3.3 Use Case Diagrams 6](#_Toc190685873)

[3.4 Feasibility 6](#_Toc190685874)

[4 Application Design 7](#_Toc190685875)

[4.1 Design elements 7](#_Toc190685876)

[4.1.1 Layout 7](#_Toc190685877)

[4.1.2 Interaction 7](#_Toc190685878)

[4.1.3 Colour schemes 7](#_Toc190685879)

[4.1.4 Font choices 7](#_Toc190685880)

[4.2 Wireframing 7](#_Toc190685881)

[5 Database Design 8](#_Toc190685882)

[5.1 Description 8](#_Toc190685883)

[5.2 Business Reporting Requirements 8](#_Toc190685884)

[5.3 Textual Representation of Dataset 8](#_Toc190685885)

[5.4 Business Rules 9](#_Toc190685886)

[5.5 Entity Relationship Diagram 10](#_Toc190685887)

[5.6 Tables 10](#_Toc190685888)

[5.7 Database Dictionary 11](#_Toc190685889)

[6 System Design/ Architecture Overview 12](#_Toc190685890)

[6.1 Introduction 12](#_Toc190685891)

[6.2 Model View Controller 12](#_Toc190685892)

[6.3 User Authentication 12](#_Toc190685893)

[6.4 Routing 12](#_Toc190685894)

[6.5 Templating 12](#_Toc190685895)

[7 Testing 13](#_Toc190685896)

[7.1 Introduction 13](#_Toc190685897)

[7.2 Functional Testing 14](#_Toc190685898)

[7.2.1 Login/Registration 14](#_Toc190685899)

[7.2.2 Navigation 14](#_Toc190685900)

[7.2.3 Calculation 14](#_Toc190685901)

[7.2.4 CRUD 15](#_Toc190685902)

[7.2.5 Discussion of Functional Testing Results 15](#_Toc190685903)

[7.3 User Testing 15](#_Toc190685904)

[7.4 Conclusion 15](#_Toc190685905)

[8 Project Management 16](#_Toc190685906)

[8.1 Introduction 16](#_Toc190685907)

[8.2 Project Phases 16](#_Toc190685908)

[8.2.1 Requirements 16](#_Toc190685909)

[8.2.2 Design 16](#_Toc190685910)

[8.2.3 Implementation 16](#_Toc190685911)

[8.2.4 Testing 16](#_Toc190685912)

[8.3 Project Management Tools 17](#_Toc190685913)

[8.3.1 GitHub Project 17](#_Toc190685914)

[8.3.2 GitHub 17](#_Toc190685915)

[9 Reflection 18](#_Toc190685916)

[9.1 Your views on the project 18](#_Toc190685917)

[9.2 How could the project be developed further? 18](#_Toc190685918)

[9.3 Assessment of your learning. 18](#_Toc190685919)

[9.4 Completing a large software development project 18](#_Toc190685920)

[9.5 Technical skills 18](#_Toc190685921)

[9.6 Further competencies and skills 18](#_Toc190685922)

[10 References 19](#_Toc190685923)

[11 Appendices 19](#_Toc190685924)

# Introduction

## Description

In a single paragraph describe what your application will achieve and why.

## Overview

Introduce your SW project here. This is where the entire report is very concisely summarised. You will be covering some/all of the following

* Application area
* Technologies
* PHP, MySQL, Bootstrap, CSS, Vanilla
* Tools, Platforms
* Dev tools: IDE, docker, frameworks
* UX tools: Miro, Figma  
   server-side: phpMyAdmin, sql
* Project management
* GitHub
* Requirements
* Design
* Implementation
* Testing
* Reflection

# Research

## Introduction

The United Nations Sustainable Development Goals (SDGs) are 17 global goals that aim to improve people's lives and protect the planet. They focus on issues such as poverty, hunger, education, health, and the environment. The goals aim to bring countries together to create a more sustainable and fairer world for everyone.

## Summary of specific SDG

SDG 4 aims to have equal access to education and learning opportunities for everyone while also promoting fairness for all.

The UNICEF. (n.d.). *SDG Goal 4: Quality Education*. UNICEF Data. Retrieved from <https://data.unicef.org/sdgs/goal-4-quality-education/> on SDG 4 talks about the importance of quality education for all children, no matter what their background is. It promotes education that includes everyone and helps improve learning results.

The United Nations. (n.d.). Goal 4: Quality Education. Global Goals. Retrieved from <https://www.globalgoals.org/goals/4-quality-education/>article talks about access to education and how it should be equal for everyone; it addresses how to close education gaps for disadvantaged countries. The article aims to get people to be more aware of the educational gaps that are found in the world.

**Case studies:**

The All hands on deck report summaries efforts to improve education quality in underdeveloped areas in India. [(All hands on Deck initiative, 2024)](https://globalschoolsforum.org/sites/default/files/2024-01/all_hands_on_deck_for_sdg_4_csf_case_study_2.pdf)

[case study](https://www.toshibatec.co.uk/assets/Sustainability/CS_SDG4.pdf) – this case study aims to improve education quality by improving health (cleaner water and better energy), which allows more time for school. This project mainly runs in Uganda and Kenya

# Requirements

## Introduction

The purpose of the requirements phase is to allow 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

Mission: Share

A screenshot of a cell phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.

Mission: Share is an app that helps people donate food, clothes, books, and computers and gives people the chance to volunteer. The app allows the user to select the place they wish to donate to and then wait for the charity/non-profit business to accept their donations.

The advantages of this app are that it has an easy and clear layout, so it can be considered user-friendly, while the disadvantages of this app are that the user wouldn’t know how to donate the books and only find out later the process and that they don’t explain what countries they serve and only finding out when searching for it.

* Screen shots
* Descriptions
* Advantages
* Disadvantages

### Interviews and Surveys

I conducted a survey with 10 responses, through which I identified the key features users need for an app that meets their requirements. Through this survey, I identified that over 60% of users find it hard to donate books as they either don’t know where to bring the books or aren’t uncertain about how to do so.

I then asked how often they would be willing to donate books over a certain period, and the results showed that 70% preferred donating monthly, while 30% preferred doing so yearly. I proceeded to ask in which country they would want the donations to go, and the majority expressed a preference for sending books to Africa, which has now helped me identify where to base the app.

The next step was to identify some of the main features of the app. This has led me to have a list of main basic features, which are the following: a donation tracker, which allows the user to see where their donation ends up; a donation history, which allows users to see how many and which donations they have made; the community impact tracker, which shows how the book would improve the community and show some achievements and finally a distribution map so that the user can see where all their books have gone.

Out of all the features, the users consider it somewhat important to know the process of their donation, which links to the donation tracker, but thanks to this, I now know that it’s something to work on in more detail.

I then asked if users would prefer an option to donate money to cover shipping or processing costs. Most users indicated an interest in this option, which is a little controversial as the majority would want to bring it to a place for them to be shipped in a container so the money can then be put towards the shipment form there and then I finally asked if the books should have gone toward a specific community or school and the majority have said yes which now leads me to research for this so that I can include this in the application.

### User Personas and User Stories

A screenshot of a computer

AI-generated content may be incorrect.

As a teacher, I want to find an easy app to use so that my students and I can donate books we no longer need so that children in specific countries can have better access to education.

## Requirements modelling

### Functional requirements

1. Donation tracker
2. Donations to specific Community /Schools
3. Donation History
4. Distribution Map
5. Community Impact Tracker

### 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

A diagram of a person's organization

AI-generated content may be incorrect.

## Feasibility

The technologies that I have decided to use for the development of the application are Laravel for the backend and Bootstrap for the frontend. The application is going to run using xampp which will allow me to host a database where all the user information will be stored

# Application Design

## Design elements

### Layout

Describe the layout of your application. Does this depend on a framework like bootstrap? Is it responsive?

### Interaction

What are the navigation elements, form elements. How does the user interact with the application?

### 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.

## Wireframing

Describe how to navigate from one part of the application to the next by means of wireframing software. This should be tested, summarise the findings here and explain which changes were made based on feedback.



# Database Design

## Description

The book donation platform would need a database to keep track of books, users, and donations. The system would include a **Donation Tracker** to help users follow the status of their donated books from listing to delivery. Donors can choose to send books to **specific communities or schools**, ensuring they go to the right place. A **Donation History** feature would keep a record of all past donations, including details like the date, books donated, and recipient.

(A **Distribution Map** would visually show where books have been sent, making it easy to see the reach of donations. To measure the impact, a **Community Impact Tracker** would display how many books each community or school has received. This system makes donating books simple, organized, and impactful.)

## Business Reporting Requirements

1. Admin can create, read, update, and delete users, donations, recipients and donation statuses
2. Users can track the status of their donation.
3. Users can complete a form to donate, providing details of the book: title, author, condition and select from a list of schools or community
4. Users will receive updates and notifications about their donations.
5. Users may view communities and schools through a Google Map.
6. Users may have access to a donation history page to view all their past donations.
7. Admin will review and approve the donation request.

## Textual Representation of Dataset

**USER** (id, first\_name, last\_name, email, phone\_number, role, image, bio)

**DONATION** (id, school\_community\_id, user\_id)

**GENRE** (id, name, description)

**BOOK** (id, name, condition, donation\_id)

**SCHOOL/COMMUNITY** (id, name, information, location)

## Business Rules

 A **user** can make many **donations.**

 A **donation** belongs to one **user**.

 A **donation** can have many **books.**

 A **book** is from one **user**.

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

 A **book** can have many **genres.**

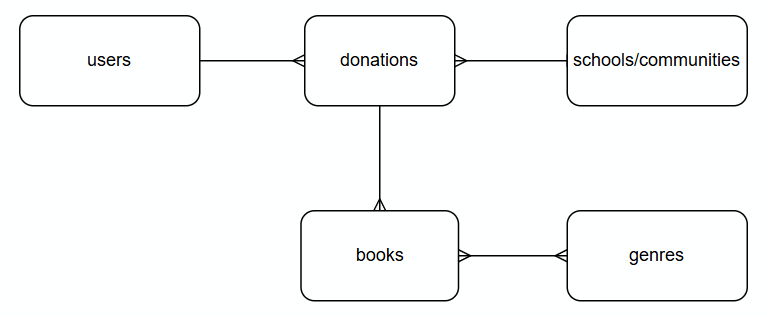
 A **book** can have one **Genres**.

 A **school/community** can have many **donations.**

 A **donation** belongs to one **school/community**.

 An **admin** can review and accept each **donation** before being processed.

## Entity Relationship Diagram



## 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** |
| Festival | title | String | 255 chars | y |  |  |
|  | id | int | 11 chars | y | PK |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table** | **Attribute** | **Datatype** | **Range** | **Required** | **PK/FK** | **FK Ref Table** |
| Festival\_Emplyee | id | int | 11 chars | y | PK |  |
|  | festival\_id | int | 11 chars | y | FK | Festival |
|  |  |  |  |  |  |  |

# 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 or amend headings as neededby the specification of your application

* 1. Model View Controller

Describe the 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, how during the development of the project you deviated from the initial project plan developed in week 1. Comment on your activities for the following project phases. . Explain any issues which arose for each of the phases.

### Requirements

### Design

### Implementation

### Testing

Include your original project plan here



## 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

All hands on Deck initiative. (2024, 01). *All hands on deck for SDG4.* Retrieved from https://globalschoolsforum.org/: https://globalschoolsforum.org/sites/default/files/2024-01/all\_hands\_on\_deck\_for\_sdg\_4\_csf\_case\_study\_2.pdf

UNICEF. (n.d.). *SDG Goal 4: Quality Education*. Retrieved from UNICEF Data: https://data.unicef.org/sdgs/goal-4-quality-education/

United Nations. (n.d.). *Goal 4: Quality Education*. Retrieved from Global Goals: https://www.globalgoals.org/goals/4-quality-education/

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](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.

# Appendices

* Add all original questionnaires and the forms for user-surveys here.
* Add the raw data surveys here.
* Add user-acceptance questionnaires and results here.

­