

Software Project

**India Heath**

**N00213565**

Software Project

Develop an accountability platform

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-indihth |
| 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

An accountability platform that people can use for starting/maintaining habits. Users get paired with a partner and then agree on the frequency they are in contact and what their goals are. This platform would start by targeting people who are learning a language and want to develop a strong daily habit of doing something for their language. An option would be to require a user to upload an image of them doing their task by a certain time each day and for that to be sent to their partner. This would meet the need for people to have outside accountability.

## Business model

The base account would be free, but a premium option would offer more in-depth options such as different partners for different habits, eg. One partner for weekly workouts and another for daily language learning goals. The platform would also be monetized with ads. The option for partnerships with other services such a Babble for language learning is possible, companies related to popular hobbies.

## Market Research

**Market for Product/Service**

The initial target market will be people in the online language learning community who are mostly studying independently. This market of people already has a strong community with more and more joining each day. They will use the service because it offers a better thought out and more user-friendly way of staying accountable, which is something many people already practice. A starting focus on language learners will eventually make way for a wider variety of activities the platform could be used for, e.g. exercise, learning an instrument, school work, business building and much more.

**Customers - Demographics, Profile**

The average customer would be between 18 and 35, this being the range in the current community. Either students or young working people who are establishing their careers. A large portion also have interests in science and technology, many involved in development themselves. This leads to a large portion being quite tech savvy. Another section is the slightly older, working professional group who are interested in learning a language but, unlike the young people, don’t have as much time per day to commit. This group are looking for easy and clear methods to get started with and are willing to pay for it.

## Marketing/Advertising

Initial advertising would be word of mouth around the established discord and YouTube communities. There are also possibilities of asking prominent YouTube creators (Days of French n’ Swedish channel for example) to promote the platform since it’d be non-profit and for the benefit of the community.

## Suppliers

The platform would be web hosting and a domain. There’s no physical product to be produced.

## Competitors

Many competitors have the same idea of offering a mostly free service as part of a non-profit want to help further education and self-improvement.

1. Studygang.app (female only option, focus on specific exams)
2. Studytogether.com (stats, leaderboards, competitions, very nice looking)
3. FocusMate (not everyone is comfortable with video, only 3 free sessions per week but reasonably priced subscription)
4. Various reddit subs for finding partners (potentially harder to find serious people)
5. Discord servers

## Employees

At least two developers would be needed with experience in both front and back-end, and a UX designer.

## Environmental Impact

If employees choose to remote work, the environmental impact would be even lower than a standard tech start-up. Electricity consumption would be the major factor with at least three employees working on computers throughout the day and then servers running to host the website. To reduce carbon emissions the website could use green hosting.

Using an online calculator(https://www.websitecarbon.com/website/studytogether-com/) on an equivalent competitors website, the carbon emissions came to 206.32kg of CO2 per year.

# 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

Here I will be analyzing two similar websites, gathering the pros and cons from each and then taking this information to users. I will be interviewing and surveying the target users to gauge their interest in the proposed features. After this user data has been collected, I will then proceed to refine the proposed features and alter where needed to fit the target users needs.

## Requirements gathering

### Similar applications

#### Studytogether.com

This platform focuses mainly on traditional study, offering goal checklists (see *figure 2)*, group or solo study rooms as well as a community aspect of leaderboards and events. The dashboard (*see figure 1*) immediately gives the user options to join a study room, make their own or study solo.

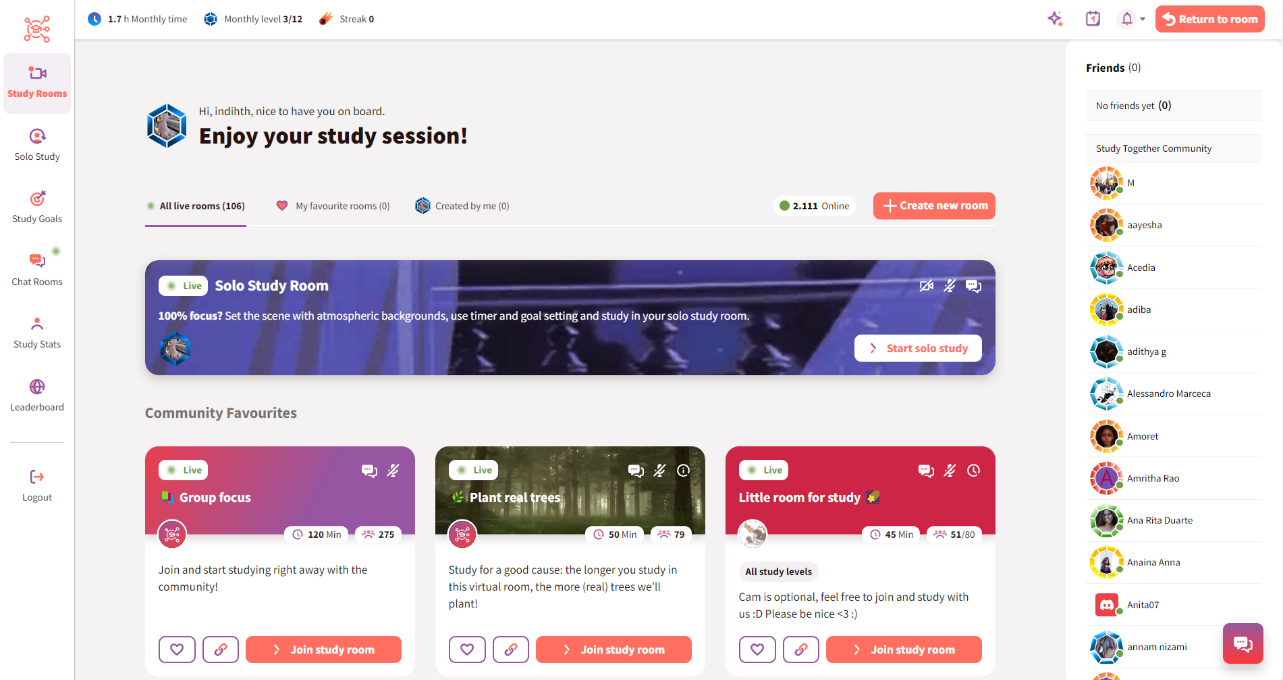


Figure 1 - Dashboard

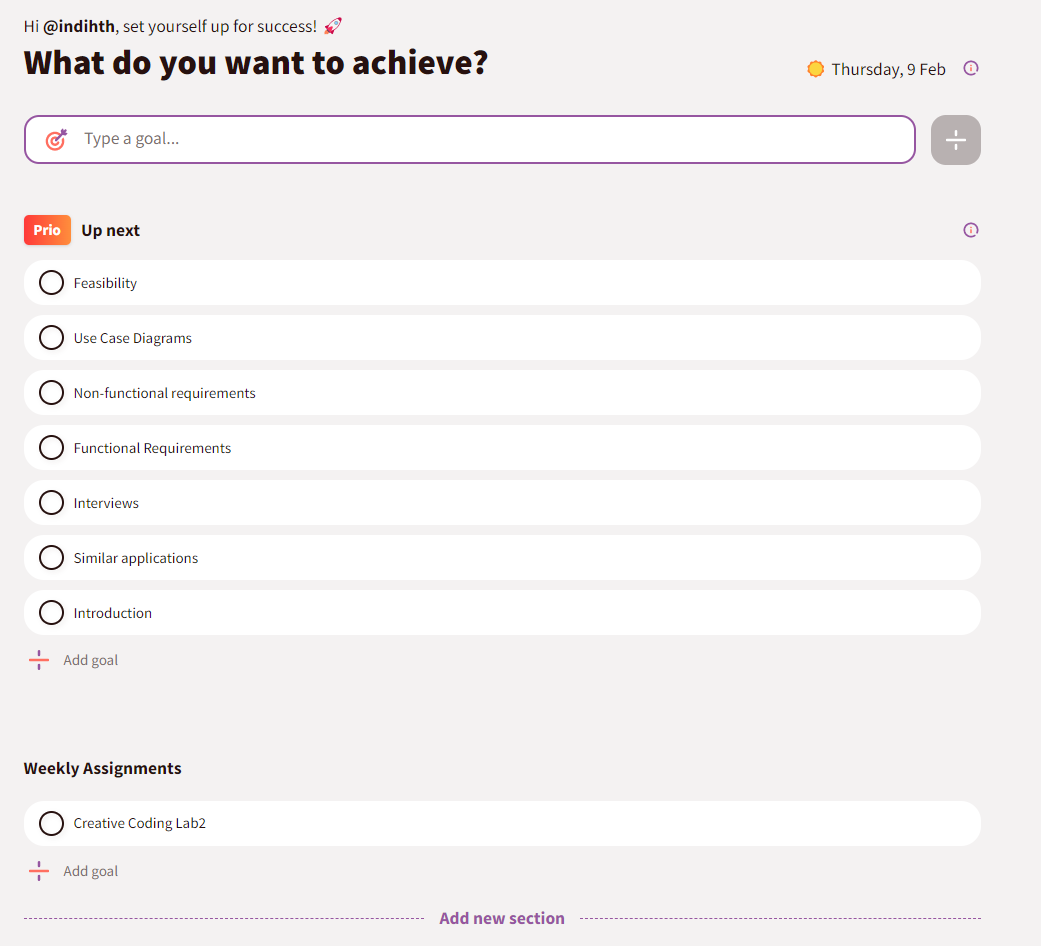


Figure 2- Goal lists

There are a few community engagement areas such as the challenges (*see figure 3)* and the leaderboard (*see figure 4)*. Each month the platform runs challenges to encourage users to study more or create new habits. The leaderboard ranks users monthly and can also be set to ‘Friends Only’, which is a particularly nice option and gives the opportunity to create a more personal study group.

Graphical user interface

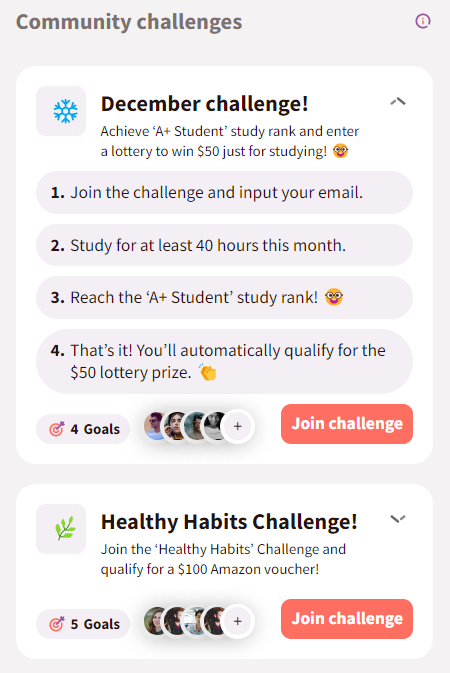
Description automatically generated

Figure 3 - Community Challenges

Figure 4 - Leaderboard

The platform has a more in-depth statistics view page (*see figure 4*) which gives users a chart which can be filtered by day, week, month or all time study hours. There is also badges and stones users can collect by logging more study hours along with a weekly streak tracker. The streak settings can be changed to require less or more time spent studying each day.

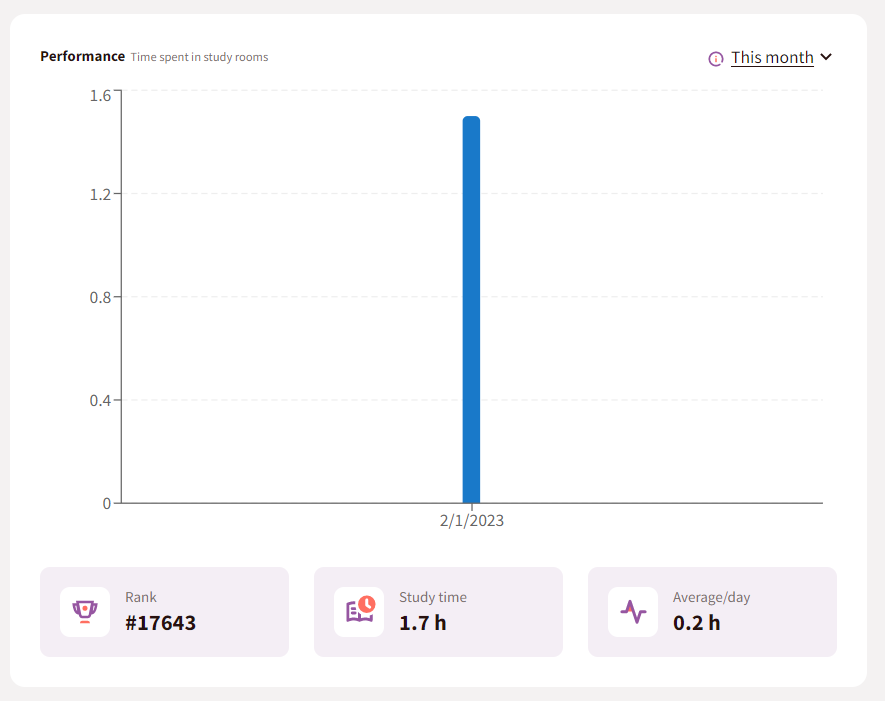


Figure 5 - Statistics

Graphical user interface, application

Description automatically generated

Figure 6 - Statistics 2

There’s a nice mini statistics section at the top left of the screen (*see figure 7*) where users can quickly see their total study hours of that month, their level and their study day streak. These small elements are great for motivation and show the user their progress in a quick snapshot, encouraging them to continue.

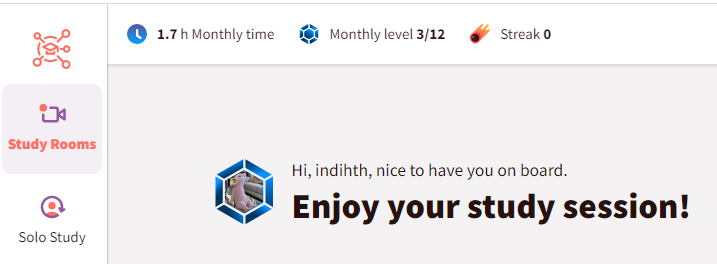


Figure 7 - Mini Stats

An interesting aspect of the platform is that in many places there are little info popups (*see figure 9*) and sections explaining the theory (*see figure 8*) behind parts of the platform.

Text

Description automatically generatedText

Description automatically generated

Figure 8 - Explaining goal setting

Figure 9 - Info popup

Advantages:

* Easy goal setting with the option to create multiple lists of goals. For example, goals for each study subject, for the week or the month.
* Info points throughout the platform so the users understand why the included features are beneficial.
* The study and streak stats are great for users to view their progress. The streak is a way to gamify the process but also rewards consistency, regardless of the total amount. Badges and other earned rewards are also good for keeping engagement up.
* Community engagement with leaderboards, either global or friends only, can be motivating for users and gives goals to work towards, i.e., getting higher on the leaderboard.

Disadvantages:

* There’s no option to input hours studied outside of the platform or for other activities, which for traditional study is fine but not for the purposes of this project.
* There is no way to filter the study rooms currently online. A user may only want to view rooms where a webcam is mandatory or vice versa. Currently, the only option is to scroll and read each description.
* The leaderboard is just one huge pool of users. There are no league options to work up to. This could be discouraging for some users who will never want or be able to study 160hrs per month. A league system would motivate users to push just a little more within their current level. For example, a user in a level 3 league who studies 15hrs per month may work to be at the top of that league by increasing to 17hrs. Small and sustainable improvements are much more helpful than sudden, lofty goals. Being ranked #12255 isn’t going to be motivating for most users. Duolingo’s league system is quite good as it encourages just a little more than the user is currently doing.

#### StickK

This platform uses financial sakes to keep people on track with their goals. Setting up a commitment requires users to decide if they want to put money on the goals or not, and who will get that money if they fail. The options vary but it’s recommended people choose an ‘anti-charity’ to be the recipient if they fail. Anti-charities being something that the person strongly disagrees with, such the NRA for people who are against gun violence. This is meant to further motivate people to stick to their goals, otherwise their money goes to an organization they dislike. The platform also gives the option for a referee and the frequency of check-ins. The referee is sort of like an accountability partner as they must verify the user has completed their daily/weekly goals. Users can submit journal updates with photos to help keep them on track (*see figure 10*).

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 10

There is very little community on this platform because the motivation is solely financial so community is less important but it would be nice to see options to search for and connect with users of similar goals.

Graphical user interface, text, application

Description automatically generated

Figure 11

Advantages:

* If used correctly, extremely motivating as negative consequences are involved.
* Easy to setup, the process of making a new commitment was straight-forward.
* Clear ways to check-in/report daily activities were done.

Disadvantages:

* The platform doesn’t have many options once a commitment is made but for its model that isn’t necessary.
* The design is quite boring and slightly dated looking.

### Interviews

**Issue 1# - Ease of logging**

I interviewed someone with experience in food tracking, using an app. The main takeaways were that it took some time to get used to but the app made it easier by saving the most recent entries for each meal logging. For example, she ate the same thing for breakfast every day so that app would have that food be her first option when inputting breakfast. This made the whole process much easier, removing barriers and pain points. When entering new foods you just have to select the portion size or amount. There’s also recipes pre-filled for easy adding of common meals.

**Issue 2# - Predictions**

Another helpful feature she found was how the app would give her a prediction at the end of each day. For example, if she had eaten 1300 calories that day, the app would say “If you keep to this diet for 5 weeks you will have lost x amount of weight.” This feature could be used in relation to building skills, e.g if someone logged, they’d read for 20 minutes/x number of pages, the site could tell them “If you read this much every day, in x weeks you’ll have finished x book/read for x number of hours.”

From this interview I had a few ideas on how to make it easy for users to add their frequent activities to their logs and give them a prediction based on their current level of activity.

**Issue 3# - Tracking is counterproductive for some**

From the questionnaire I shared with a language learning community I found that some people didn’t find tracking their time helpful in the long run because they became too focused on getting their numbers up for the sake of numbers, not for their learning. To counter this, I would like to give the option for people to simply track their habits rather than their time. Each user should be given the option to use checkboxes or more in-depth time tracking. A simple checkbox would help users with the streak portion of tracking and not distract them with numbers.

*Questionnaire link: https://forms.office.com/e/fQu2FGBUSQ*

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.

1. Allow creation of multiple daily tasks that can be checked off each day
2. Input time spent on each goal
3. Allow premium users to create multiple categories of goals (language, fitness etc.) and have different partners for each goal
4. View statistics for activities logged
5. View partners activity log
6. Comment/respond to partner's log

### Non-functional requirements

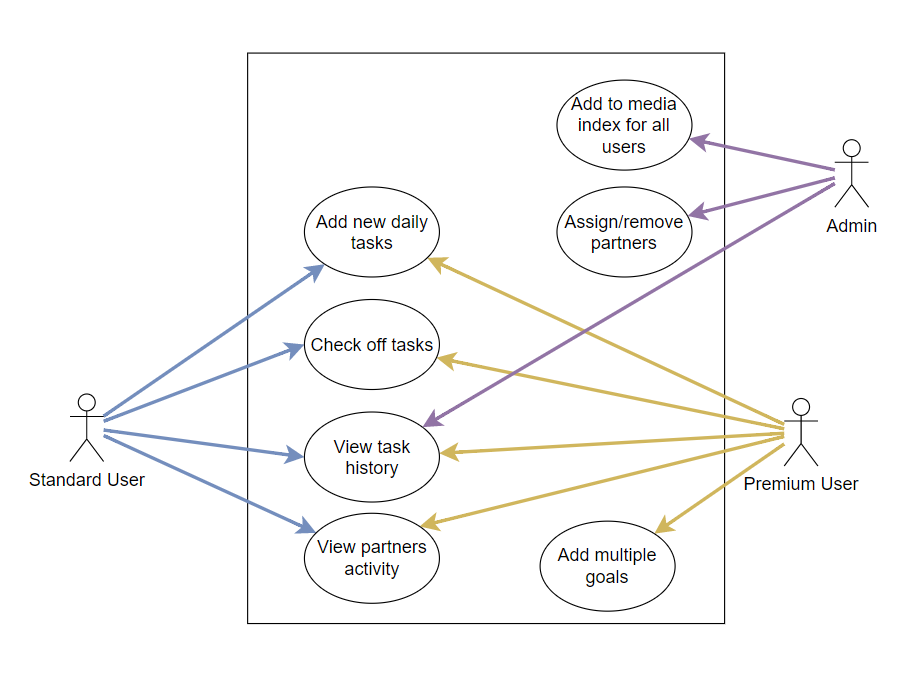
These are requirements which if not met do not stop the application from working, but which mean that the application is not working as well as it should. They are usually based on issues such as:

* Usability
* Performance
* Security

1. Notify user when their partner logs an activity
2. Track the users streak (consecutive days of study)
3. Ability to choose just checkboxes or time tracking on goals
4. Auto suggest the users more recent task when logging something
5. Let user pick logged media from a general index available to all on site
6. Give users a prediction of level in x weeks each time they log something

### Use Case Diagrams

This use case diagram illustrates what functions each user type can perform. The standard user is able to add new tasks, tick them off and view their and their partners’ history. The premium user can do all of this and add multiple goals. The admin can add new media (tv shows, movies etc.) to the index for users to select from, assign and remove partners and view any user history.



## 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.

The technologies to be used are the PHP framework Laravel, along with Breeze for authentication for the back-end, the Bootstrap framework for CSS and SCSS for front-end and SQL with phpMyAdmin for the database. I don’t foresee any complications with these technologies as I have used them together in the past with no compatibility issues.

# Web application Design

## Layout

Describe the layout of your web 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.

## 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.