



FunTime – a web portal assisting the teenage population to find the right extra-curricular activities

Sara Disney

C2101407

MSc Computing and IT Management

Supervisor: Daniela Tsaneva

School of Computer Science and Informatics

Cardiff University

08/09/2021

Acknowledgments

I would like to firstly thank my dissertation supervisor, Daniela Tsaneva, for her continuous support and direction within this project. Particularly the first month of the project discussing the logic with me and providing plentiful resources to help my independent study.

I would also like to thank my family who have been down the phone to speak to throughout the year. Deciding to undergo a master's degree and quitting a fulltime job to pursue it was a big decision which was made a lot easier with the support they offered.

Contents

Abstract.....	5
Introduction	5
Aims and Objectives.....	7
Background research	8
Problem.....	8
Web Portal	9
Recommendation engines	9
Collaborative filtering	9
Content-based filtering	10
Other recommendation options	10
Examples of recommender systems	11
Extracurricular activities	11
Well-known recommender systems	12
Overview	14
Approach.....	15
Requirements and design	15
User Survey	15
Personas.....	22
Requirements.....	24
UML Use Case Diagram	24
Non-Functional Requirements	24
Use Cases and State Transition Networks	25
Prototypes / Wireframes	34
System architecture	38
Algorithm design	39
Cosine similarity	39
Euclidean distance	40
Database design.....	40
Data sources.....	43
Decision on variables used.....	43
Implementation	44
UML Activity Diagram	45
Algorithm	47
System development	49
Flask Framework	49

Database	50
Forms	50
User requirements	51
Inputting logic to Flask	52
Map integration	55
Social Media Sharing	57
Frontend styling	57
Results.....	58
Testing.....	58
Analysis	67
Conclusion.....	68
Future work.....	70
Reflections	71
References	74
Appendix	76

Abstract

Psychological studies have repeatedly demonstrated the benefits that extracurricular activities offer the teenage population. However, there are currently a variety of problems that exist within this topic, which this project aims to resolve. Many teenagers may need support deciding on extracurricular activities through not enough choice. Equally, in some areas, with such a variety of extracurricular activities that exist, there can be an overwhelming sense of not knowing what to do.

FunTime is a web portal with the aim to help teenagers discover the extra-curricular activities that are most suited to them, including options from sports, arts, and skill-based activities. Users of the portal will be provided with suggestions of suitable organisations in their local area and the relevant information that accompanies this.

The recommender is built using a hybrid approach of a basic database query search, followed by principles of content-based filtering: specifically Euclidean distance. This enables the users' needs and wants to be fulfilled and provide accurate and insightful recommendations that users may not have considered before.

Introduction

Growing up, after quitting the one extracurricular activity I did at 13, I had so much spare time but did not know what to do with it. I did not know what clubs existed near me, let alone any that would take newcomers. Only beginning university, I joined new clubs and societies, wishing I had much earlier. I am now a strong advocate for being active and understand the challenges that finding clubs or organisations have. My anecdotal story was the beginning of the motivation to choose this project, further research into the topic demonstrated there is a real importance for teenagers to partake in extracurricular activities.

Currently, there is no readily available database that consists of all activities alongside a short description to understand what it is. In addition, there is no database with details of these activities, or where a person may take part in them. Some websites have tried to put together a few organisations that exist in their area, equally if a person went onto Facebook

groups, they might find organisations promoting themselves, but there is no fully comprehensive site or database where a user is able to discover what exists.

Since there is nothing that states what extracurricular activities exists, there definitely is not a tool that suggests the best activities to people. That is what this project aims to resolve. Firstly, it requires building a database that consists of a range of activities that could grow over time as more activities and organisations are inputted. Then using the data, provide users with the best options for them.

To solve this problem, data science methods and tools are going to be used. The reader will not require a deep understanding of data science, but the project will go on to discuss different recommendation systems and similarity measures prevalent in the field of data science.

The project considers how to build a system that will suggest appropriate extracurricular activities to teenagers. To do so, it aims to understand the target audience alongside parents as a secondary actor to use the system. I build a prototype to demonstrate at a small scale how this problem can be solved to the local area of Cardiff, Wales using a sample of extracurricular activities and associated organisations that exist.

The project begins by outlining the aims and objectives, then discusses background material. The background material examines the need for this project surrounding extracurricular activities, recommendation engines, well-known examples of websites using recommendation systems and examples of the closest tool to extracurricular activity suggestions: quizzes. After discussing this material, I discuss the approach and design. This includes use cases, wireframes, algorithm design and database design. The implementation discusses making the design into a reality, the challenges that became present from this and how I worked to resolve them. The final section tests the implementation according to the aims, objectives, and use cases from the beginning of the project. This is before evaluating the overall project and reflecting on the lessons learned. I also discuss how this project can be developed further in the future.

Aims and Objectives

The overarching aim of this project is to produce a prototype web portal that returns appropriate suggestions to teenagers regarding extracurricular activities that are best suited to them. This is creating a recommendation prototype for a sector that has been untouched by both academia and businesses.

Within this, there are several mini objectives that the project aims to address.

1. An exploration of appropriate methods to produce recommendations is required. This will be the logical backend that supports the portal.
2. Understand user requirements in the design phase. This will shape the portal for both the frontend and backend.
3. A varied activity database is required. This is to be designed to be inclusive to all members of society who may use the portal if it continued past prototype stage.
4. The algorithm created should cover aspects that define what features form an activity, and what users look for in activities. This is to ensure the portal produces valid suggestions.
 - a. Related to points 2 and 3, a range of feature options suited to different user circumstances should be available for users to choose from.
5. Build a basic user interface. This should be fully functional and provide the user with information in a clear manner.
6. A limited number of results should be produced. This is to balance giving users options but not presenting them with the phenomenon of overchoice. Overchoice could lead to difficulty in making a choice of partaking in a particular activity which would make the suggestion redundant.

Background research

To understand the problem area further and appropriate solutions, I undertook background research. To begin with, I aimed to identify the need for assistive technology regarding teenagers and extra-curricular activities. Then proceeded to research into recommendation engines plus evaluating examples of recommender systems that are widely used.

Problem

Academic research has repeatedly demonstrated the importance of extra-curricular activities particularly within the older children age bracket. Extracurricular activities offer a range of benefits for teenagers identified as: increased confidence supporting social interaction, aspirations to go onto higher education, more soft skills, and a sense of wellbeing and belonging (Donnelly *et al.* 2019). Donnelly *et al.* (2019) go on to explore how mental health and wellbeing is supported through these activities offering an important space to relax and have fun. Psychological studies have shown that disruptive behaviours and substance use is negatively correlated with participating in extracurricular activities (Driessens 2015). Similarly, Fredricks and Eccles (2006) found that teenagers who were involved in school extracurricular activities were less likely to drop out of their school or become arrested as young adults. And Mahoney (2000) identifies that confidence and trust in others is created during a time of life where intimacy and faith are challenged. Clearly there is a strong importance for extracurricular activities within teenagers' lives; having a web portal that could support this may produce benefits being the step towards influencing teenagers to join and maintain continued attendance.

Understanding the importance of extracurricular activities has identified the context to this project, however, the need of a system which can assist teenagers to find the right one requires further exploration. Kupisk (2017) claims that "finding extracurricular activities that are a good fit for your teen can take time". She continues to state that teens sometimes require help to envision what they could do and what exists. Terzian *et al.* (2009) explored the reasons behind teenagers' lack of involvement in out-of-school programs and identified lack of awareness about what exists as a key reason. Many participants in their study did not know about the activities local to them, however, expressed interest in wanted to participate in something. Therefore, there is a gap for a product to assist teenagers to find what exists and inspire them into trying such activities.

Donnelly et al. (2019) from the social mobility commission defines extracurricular activities in a broad sense to encompass a full picture of types, this includes: “sporting (e.g., football, boxing, cricket), musical (e.g., learning an instrument, membership of an orchestra), artistic (e.g., drama, dance, drawing), and social (e.g., youth groups, scouts/girl guides) activities” (p. 6).

Web Portal

A web portal is a website that is designed to enable users to get information, bringing information together from various sources (Hasabe *et al.* 2015).

Recommendation engines

Assistive technology has the underlying engine of recommendation systems. Recommender systems are a more personalised approach rather than a simple search engine. Ricci et al. (2011) differentiate between personalised and non-personalised recommendation systems: something like a top ten popularity list counts as the latter and is typically not addressed by researchers for recommender systems. The idea is personalised systems “prioritise items likely to be of interest to the user” (Burke *et al.* 2011, p. 1). Different types of recommender systems are available to use, identified in Burke’s (2002) paper. The two most common recommender engines will be explored here in detail with mention of other options.

Collaborative filtering

One of the most well-known recommender engines is collaborative filtering, here recommendations are based off information provided from other users (Burke 2007). In practice, this could be user recommendations being based upon what other users enjoy and rate highly. The idea here is that if users like or want items that are like each other then there is strong chance that they will have the same utility for future items and so subsequent recommendations will be based off their counterparts’ other likes (Schafer *et al.* 2017). To exemplify this, if user X and user Y both watch and rate positively the same movie, user X will then be recommended with other movies that user Y has watched and vice versa.

Collaborative filtering can be split into user-based and item-based sections. User-user method is a memory based collaborative approach that identifies users who are similar to one another through use of the nearest neighbours’ algorithm, then items are suggested that are popular with these users of similar characteristics. The recommendation engine

relies on information such as likes and dislikes to compute a similarity index between users then proceed to recommend appropriate items. In comparison item-based collaborative filtering looks for similar items rather than similar users. The theory is that users will like items similar to what they previously have liked.

However, there are problems associated with collaborative filtering. These include new items requiring an additional knowledge source since they will not have any reactions from current users (Schafer *et al.* 2007). A hybrid approach may be useful to incorporate this “cold-start” problem. Equally, Schafer *et al.* (2007) note how malicious users could bias the system with “sybil” profiles – this would mean the algorithm is working from data, which is not valid, thus creating poor recommendations.

Content-based filtering

Another famous recommendation engine is content-based filtering. Recommendations from this approach is generated from the features associated with particular items and the ratings a user has given them. It is user-specific based upon the content of items, and the interests of the users are discovered based on their history on the site. What makes this different to item-based collaborative filtering is that it looks for similar items to what the user has looked for before, rather than taking into consideration ratings and preferences.

Thus content-based recommendation system provides recommendations to a user based on description or features and identifies similarity according to these attributes. There are different approaches to measure similarity distance including Cosine similarity, Euclidean distance, and Manhattan distance (Subramanian 2020).

A similarity measure is comparing how alike two data objects are where there is a distance with dimensions that represent features of the objects (Polamuri 2015). The smaller the distance, the greater degree of similarity. Similarity is usually measured in the range of 0 to 1 where 1 means $X=Y$, and 0 means $X \neq Y$ (Polamuri 2015). Similarity measures are calculated from a vector that corresponds to an item’s features, and a vector that translates to the user’s preferred features.

Other recommendation options

Although collaborative and content-based are the most common recommendation techniques, Burke (2007) also notes upon demographic and knowledge-based techniques.

Demographic provides recommendations based on a demographic profile of a user, and knowledge-based produces suggestions based on inferences on user's needs and preferences.

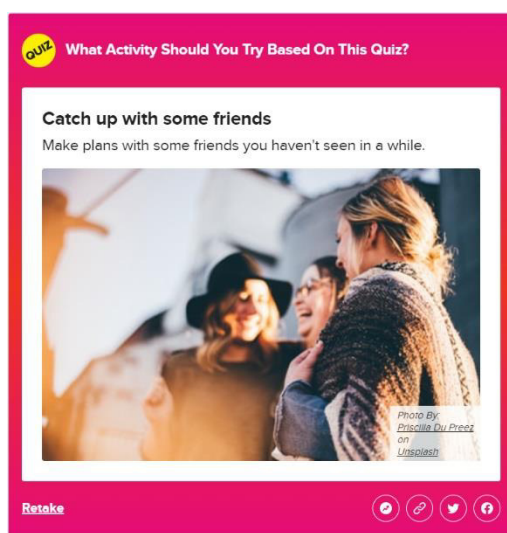
More complex recommender systems introduce the concept of hybrid recommendations which combine at minimum two of the recommendation techniques noted. Hybrid recommendations can be implemented in seven different ways: weighted, switching, mixed, feature combination, feature augmentation, cascade, and meta-level (Burke 2007).

Discussion surrounding algorithm decisions are noted upon in the algorithm design section to understand which is most suited to the requirements.

Examples of recommender systems

Extracurricular activities

Searching for a recommender for extracurricular activities has proven challenging – suggesting that there is currently no such system that exists. Using search criteria such as: “recommend me an extracurricular activity”, “recommend activity”, “what activity should I do”. The closest similarity for a recommender fitting with this topic would be some online quizzes. Although a quiz is not the same as what this project will be producing, it will be useful to analyse to understand how they are suggesting activities to people. Alongside analysing such quizzes, analysis of alternative recommender systems will also be useful for background knowledge to this project.



Buzzfeed's “What Activity Should You Try Based on This Quiz?” using a range of questions with mid-sized images attached. These questions include picking an outfit, an animal, a drink, a tv show, and other simple questions. The end of the quiz produces a title of the activity, a small description with large image underneath. There is also the ability to share this result with friends on Facebook, Twitter, as a link or to pin to your personal Buzzfeed profile as a sort of tagging system.

Quiz: Which fitness activity suits you?



Result: Spinning or cycling

Share Your Result



Spinning is an indoor cycling cardio that typically takes place in a gym, but since the pandemic began lots of people are enjoying spinning at home. Spinning is low impact fast cycling that's usually done to music, and you don't need rhythm or coordination to do it, but as it is a cardio workout, it aims to increase your heart rate during the session. Contact your GP if you aren't sure whether you should do a spinning class.

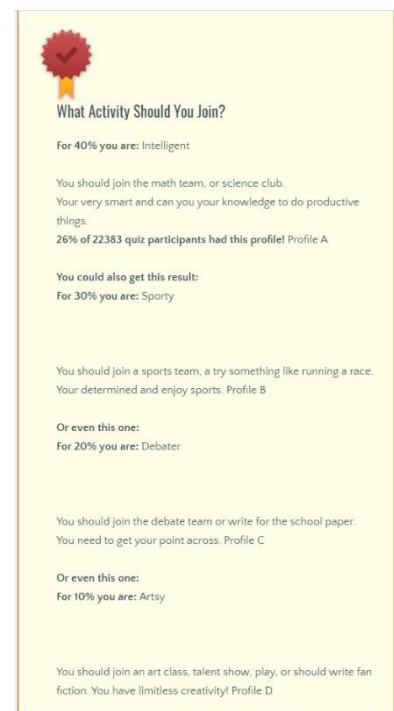
The British Heart Foundation offers a similar resource of a quiz to suggest which fitness activity suits you. It provides seven questions asking the user to choose location of their activity (quiet studio, busy studio, outdoors, home), willingness to pay, aims, etc. The result of a person's answers is produced with a large image, title, description of what the activity is and the ability to share your result on to social media.

The last quiz-based system I wish to review is from allthetests.com who ask a series of questions based on your personality and then produce options of what activity you should join.

One problem with this quiz is that the answers provided to the questions were limiting meaning that a user had to find a best fit to one of their answers rather than something that was closely aligned with themselves.

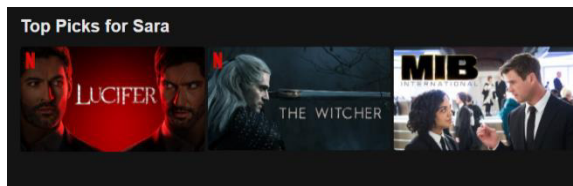
Well-known recommender systems

Recommender systems have been applied largely in the ecommerce sector alongside streaming websites. The household names who use recommender systems include Netflix, Amazon, Tinder and YouTube (Severt 2018).



Streaming services:

Netflix –



Type of recommender system used and why:

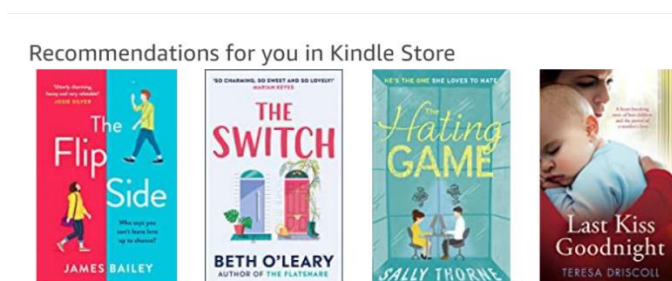
Netflix is famous for its recommendation engine. Netflix offers a basic ranking

recommendation system where it promotes the “Top 10 in the UK Today”. This is their simplest tool based off user data. Netflix also uses a hybrid approach to show users movies which they might not have chosen originally. It looks at the content, but not just genres within that. Using the millions of data points from users across the world, it understands who watches what, when and how many times. When a user clicks on a title, Netflix show a percentage match. Netflix’s approach is very sophisticated, through working with explicit data where users state that they like something in conjunction with implicit data where a user is watching the same genre repetitively as an example.

UI Design: Netflix offers a “Top Picks for [name]” section in their platform, whilst also promoting the movies they expect a particular user to like throughout. The items are presented well visually with a cover image that displays what the film or series is. When a title is opened, a short description and details of the title is provided with the option for users to thumbs up or down the movie, consequently supporting the recommender engine.

E-commerce:

Amazon -



Amazon offers a range of products such as prime video, kindle store and the original e-commerce items service. Using the kindle store as an example here for recommending books, the recommendation system

provides options based upon what a user’s previous purchases were. This is likely a hybrid recommendation engine mixing collaborative filtering with content filtering.

Amazon's kindle store uses a similar design to Netflix whereby each item is displayed as an in-line image and clicking on it to reveal further information.

Overview

Currently there is no tool available that recommends extracurricular activities based upon an individual's preferences. The quizzes that are available are limited in both content and logic with none of them offering what this project aims to provide. On the note of the recommendation systems, the most commonly used ones have a high number of current users which enables complex engines to be used through hybridity.

Approach

To complete this project within the timeframe, the agile methodology was adopted. This methodology was adopted with use of Kanban via the Trello app. The benefits of Kanban were immediately noticed with its 'pull approach': when a task is finished, a new item is immediately pulled to begin working on (Ahmad et al. 2016). Other benefits of using Kanban included better visibility of work, and the ability to return on ideas.

Agile was a natural decision choice in comparison to waterfall development. With agile, tasks could be broken up and tested throughout. Furthermore, with extracurricular activities being a topic that has not been explored in this way: decisions on what should be included in the database, what inputs were necessary, and what logic to use were all potential threats to a waterfall, step-by-step approach. Therefore, through having the flexibility that agile offers enabled changes to take place throughout the project instead of bottlenecks or realising errors at the end.

Requirements and design

User Survey

One element in the design phase that I viewed critical was understanding users' thoughts. As in any product management role, the importance is to gain insight into what users want rather than projecting what I think a user wants. Therefore, I conducted a short online survey aimed at parents and legal guardians of teenagers to understand their reality and build my prototype around their feedback. Due to the time it took to gain ethical approval for this user survey I conducted this much later in the project than I anticipated in my initial plan. However, the benefits of agile meant that I could work on the implementation, and after conducting the user survey, return to the implementation and make any edits that I deemed appropriate. The aim of the questions was to focus on what features of activities are considered important when decision-making, what information people want about activities, and then a focus on the system design.

Please find the user survey questions in the appendix.

Some notes on this survey's methodology before addressing the results. This survey was conducted online and was put into a Facebook group of a relatively affluent area. This may lead to the results not being representative of the overall population. However, for this

study, this does not cause much of an issue as the aim is to simply get ideas from a small sample of the population to understand their point of view. Any responses from members of the target population builds a clearer picture to create and edit a system from, rather than working wholly from my own ideas.

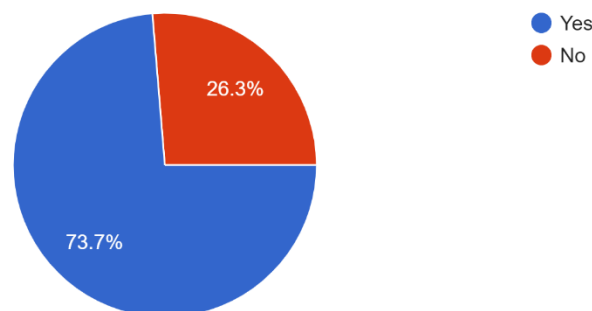
Question 1 and 2:

The first two questions were to ensure participants had read the participation information sheet, consented to the research and are in the target population identified. These were to ensure ethical approval was met but also an attempt to produce valid responses. All users answered that 'Yes' to both questions – which were mandatory to continue.

Question 3:

Did/does your teen take part in extracurricular activities which they have been part of for a minimum of one year?

19 responses



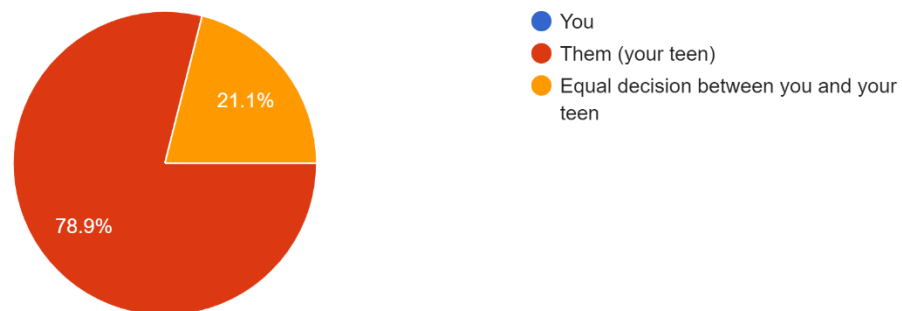
The aim of this question was to find out the proportion of teenagers who currently take part in an extracurricular activity for a regular period of time. Over 70% of participants claimed that their teen either has done or currently partakes in an extracurricular activity that they have been part of for a minimum duration of one year. Although this is a high proportion of respondents, this question raises three points – the first, is the activity that the teen does the most suited to them? Perhaps a different activity would be better aligned with what they like. The second, would they be interested in taking part in another activity? Just because they do one activity, does not mean they would not have time for another. And thirdly, regarding the respondents who answered 'No' should not be dismissed as there are some teenagers who have not found the right activity for them. This demonstrates that

there is a need for this system, and connotes extracurricular activities are an important part of many teenagers' lives.

Question 4:

Who is more likely to choose the extra-curricular activities for your teen?

19 responses

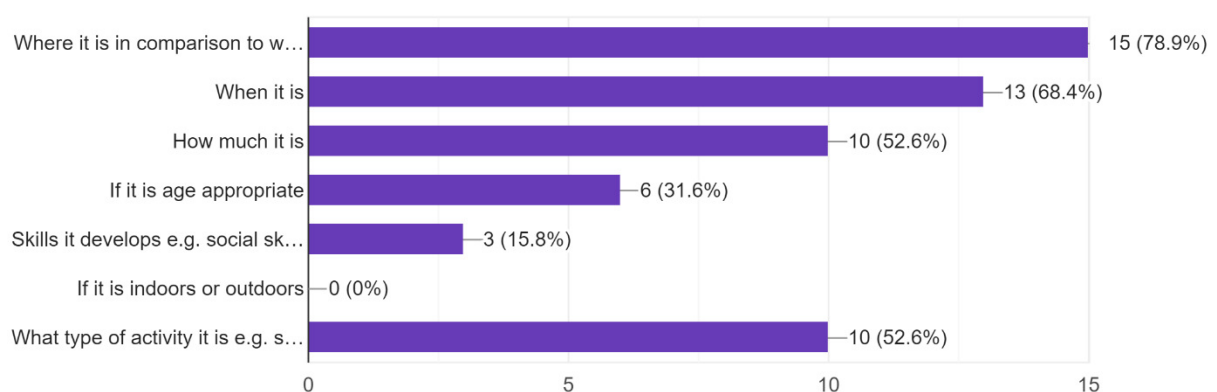


This question aimed to understand who the primary user of the web portal would be. This would impact how the site would be presented. This resulted in concluding that teenagers would become the primary user of the site, and parents would become the secondary user. No parent stated they would choose an activity for their teenager, but some did state they would have a part to play in the decision-making process – so these may want to use the portal.

Question 5:

Select the top 3 features you consider most important when choosing an extracurricular activity

19 responses

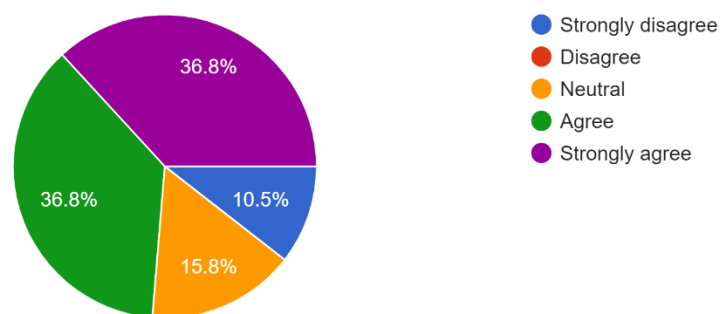


Question five's purpose was to understand the features of recommender system and to break them down according to importance. As expected, location was the mode response followed by when it is, how much it is and the type of activity it is. These results have supported the decisions made in the logic of the algorithm and have put more weight on the decision to break the algorithm down into "must haves" and "wants" (which will be addressed in more depth further in this paper).

Question 6:

To what extent do you agree with this statement: 'Distance from home impacts decisions regarding which extracurricular activities my children can do.'

19 responses

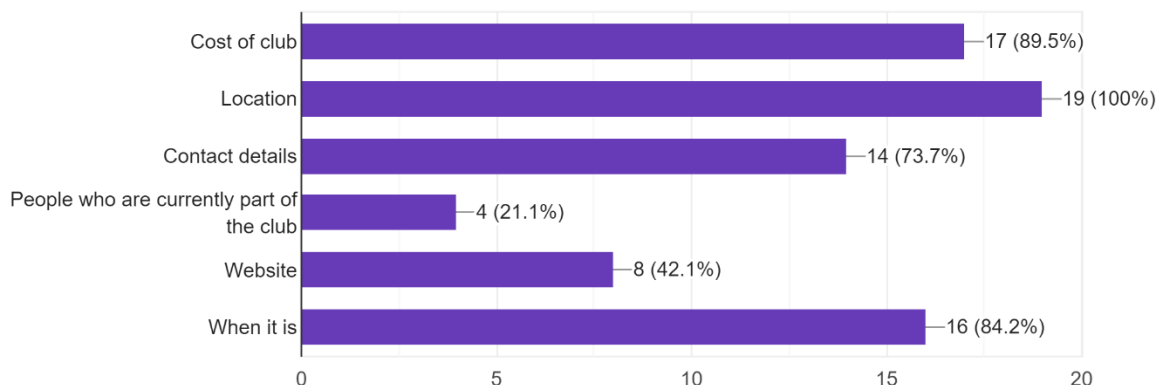


This question was to focus on the location/distance aspect in greater detail to understand if it was as important as I hypothesised. Although there are some outliers who do not view distance to impact extracurricular activity choice, for many this is important with over 70% either agreeing or strongly agreeing to the statement. This has supported the decision to put a strong focus on location of activities on the portal.

Question 7:

What information would you like to know about an activity before your teen joins?

19 responses

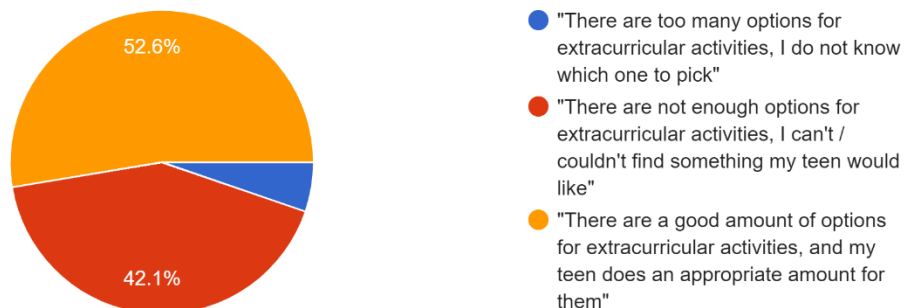


The information that is stored in the database for each activity and then displayed on the activity webpage was also useful knowledge to gain from users. If I could find out what they wanted to know about extracurricular activities, then I could gather the appropriate information into the database, so the portal is functional for users. Again, location came out highly important with every participant wanting to know the location of an activity before joining. Closely following was the cost, time, and contact details. Interestingly, the people who are already in the club was not considered important, so I did not add this as a column in the activity table in my database.

Question 8:

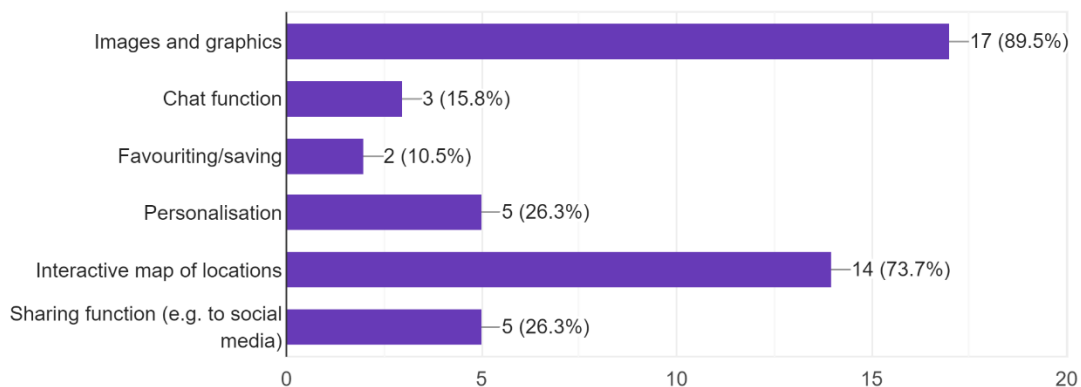
Which statement do you agree with more?

19 responses



Are there any features of websites that you particular like? [Please tick all that apply].

19 responses

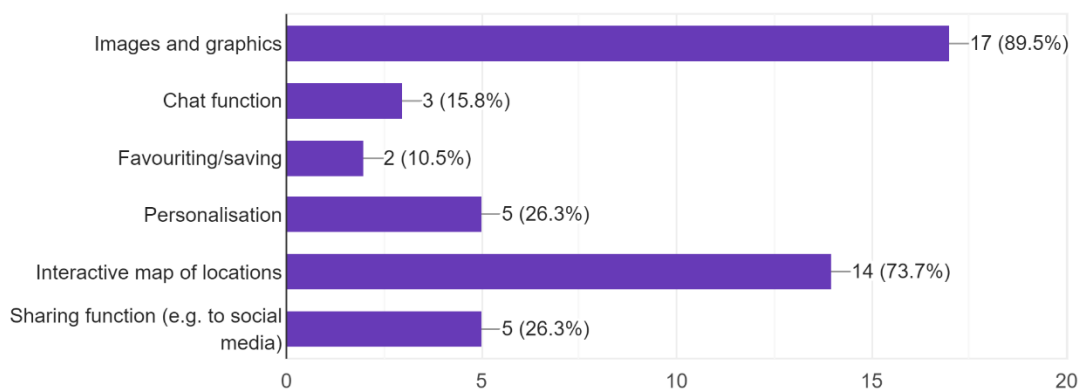


The results of question 8 were the most unexpected. Although a large proportion of respondents said that their teen either do or did take part in an extracurricular activity for at least a year in question 3, a surprising percentage stated that there are not enough options for extracurricular activities. This suggests that finding the “right” activity is a challenge, and some support is desirable to find activities that their teenager would enjoy.

Question 9:

Are there any features of websites that you particular like? [Please tick all that apply].

19 responses



I wanted to discover some insights into the use of the system, particularly the UI and extra features that users may like, additional to the basic logic of the activity suggestion part.

Images and graphics were the modal answer, shortly followed by an interactive map.

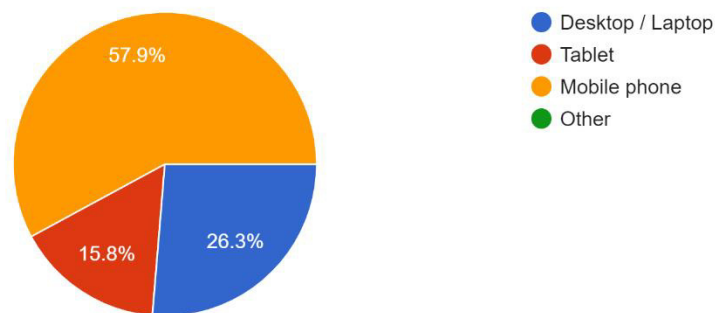
Although I had a use case for users to be able to locate activities, this response has led me

to include an interactive map integration to display activity locations. Chat function and the ability to favourite was not very popular so I have decided to not spend time creating the logic for these functions since it would not add much to the user experience and meant more time can be applied to more desirable functionality.

Question 10:

What type of device do you primarily visit websites on?

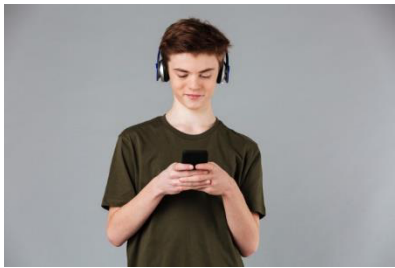
19 responses



The final question of the user survey addressed device use. Due to this survey concluding after the initial implementation, I had not created the web portal mobile-first, however, this result demonstrated the necessity of making a system that is responsive to mobile devices.

Personas

Harry Parke



Primary user

Occupation: School pupil

Age: 14

“I want to try something new after school with my mates.”

Harry is a school pupil who is bored of watching tv afterschool but does not know what to do instead. He wants to find something to do that he can suggest to his friends or where he could make new friends when doing the activity.

He wants to find something to do that is sports-related and wants to be suggested with appropriate activities suited to his age. He will use the platform initially on a one-off basis to find an activity to do but would become a more regular user if there was a social element included or if there were updates from the club he joins.

Being a teenager, he will use the platform on a range of devices including his mobile phone, his laptop and web browser on his Xbox gaming console.

Goals:

- Be recommended activities that are related to his interest in sports.
- Find contact details for clubs he is interested in joining.
- See his recommended activities to look at later.

Karen Hill



Primary user

Occupation: Bookkeeper and parent

Age: 46

“I want to find activities my teenage kids will be interested

in.”

Karen is a working mother with two teenage kids. Her partner and her both work and do not get home until 6pm so want to find activities that their kids can do straight after school. Karen wants to find activities that her children will enjoy and stick to, but also thinks it's important to develop their social skills and keep active.

Karen does not want to feel out of touch with her children so wants appropriate recommendations based off what her children seem to be interested in currently alongside her own thoughts.

She will mainly use the web portal on her desktop, however, may use her mobile phone sometimes too. Her use will be irregular, however, if a developed social feature or updates from the extra-curricular activities were provided her use would become more regular.

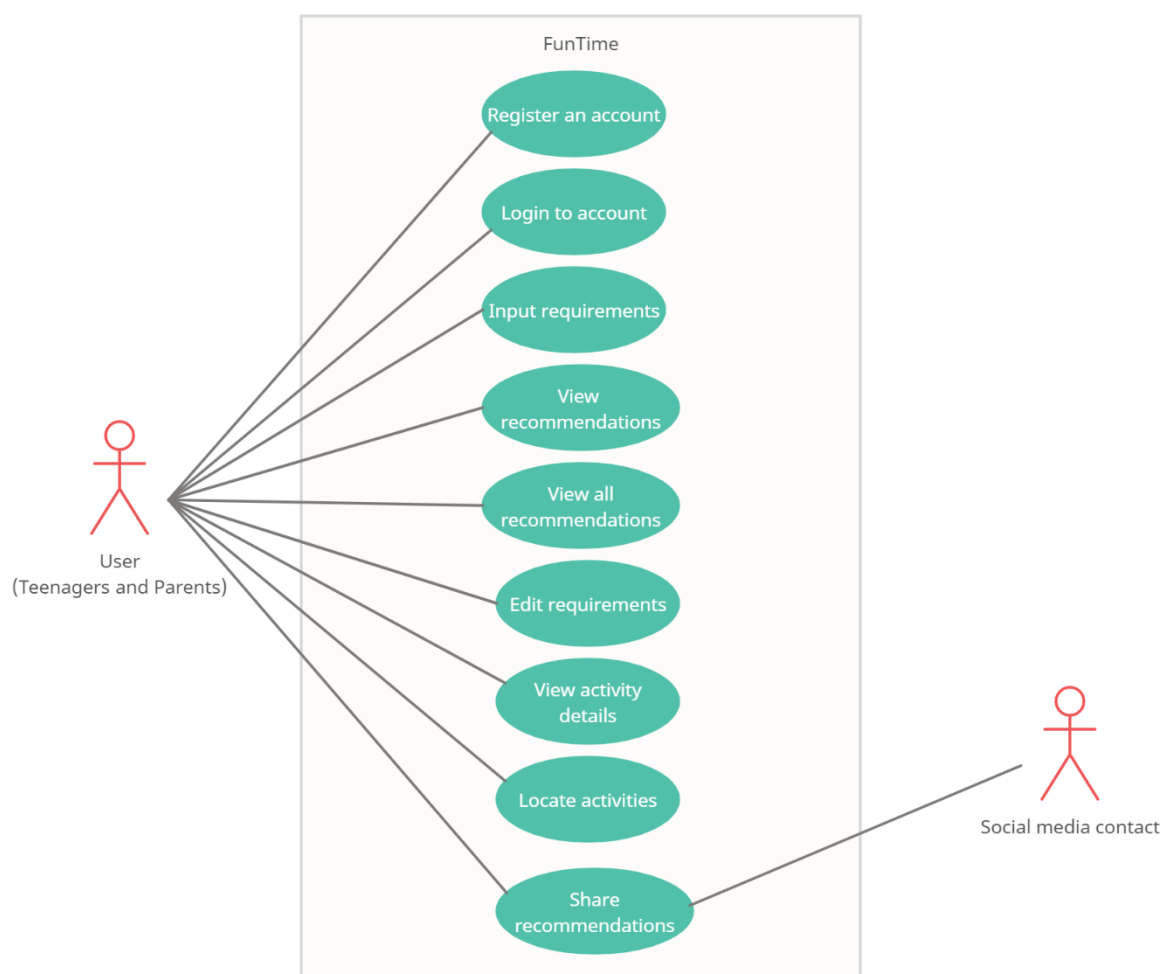
Goals:

- Receive extracurricular activities recommendations that her teenagers may be interested in.
- Be recommended with fun and safe extracurricular activities for her children.
- Contact providers of the extracurricular activities to find out more information.

Requirements

For this project to be successful, a range of requirements are necessary for the user to experience a functional and effective portal. Although there is some debate regarding where registration and logging in should be use cases, they have been included here because it is part of the initial development, therefore have been counted as functional requirements.

UML Use Case Diagram



Non-Functional Requirements

- The web portal should be secure, with user data secure and protected against malicious attacks.
 - o User passwords should be hashed in the database, rather than stored as plaintext

- Escape values to rule out cross site scripting (XSS)
- Be protected against database injection attacks with input validation
- The web portal should have suitable functions to meet user needs to be suggested with extracurricular activities
 - The system should be functionally correct with results being the closest match to a users' preferences
 - The system should be functionally appropriate through presenting the user with only the necessary steps to get suggestions.
- The performance of the application should be between 90-100% within Lighthouse standards
 - The capacity of the database maximum limit should meet requirements with ability to expand with increased data.
 - Processing times should be quick, with page load times less than 3 seconds.
- The portal should be user-friendly to both parents and teenagers who will be varied in their computer literacy.
 - Create an aesthetically pleasing interface by maintaining a house style on each page with set colours, navigation, and fonts.
 - Navigation should be obvious and accessible, with the ability to return to the homepage in one click.
 - Users are presented with error messages to protect them against making errors
 - Users to be provided with appropriate informative text to understand how to use the system.

Use Cases and State Transition Networks

Use Case No: 1	Use Case Name: Register an account	Rating: Must Have
Purpose: For users to register their details onto the web portal to create a new account		
Main Actor: Teenagers	Secondary Actor: Parents	
Pre-conditions: N/A		
Trigger: Users click on the register button		

Description

Main flow:

1. Users enter the registration page
2. Users input their details into the boxes
3. Users click the register button

Alternative flows:

1. User inputs invalid information
2. User clicks the register button
3. User is returned to registration page with feedback

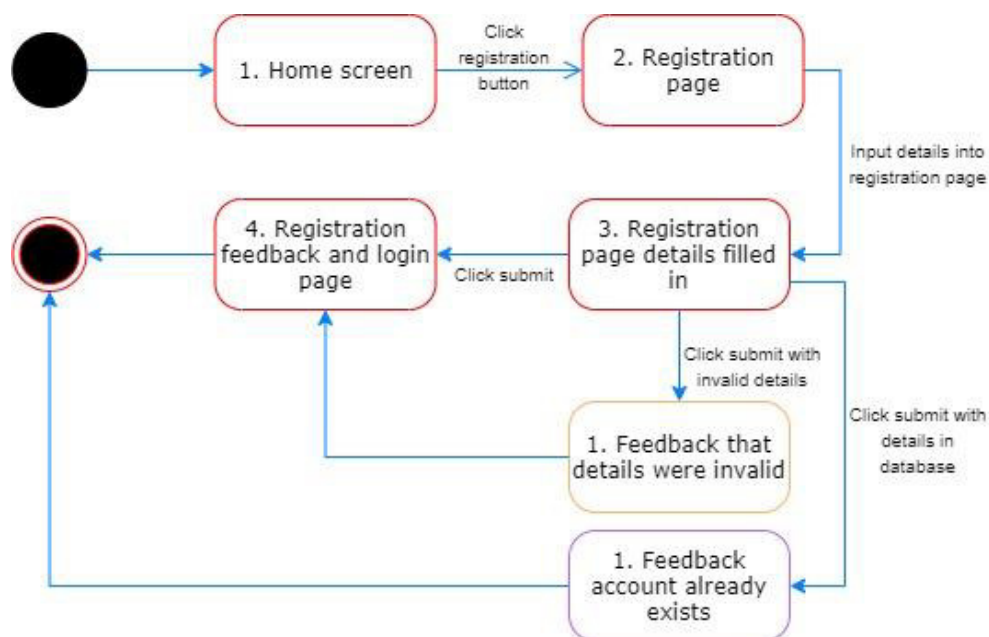
Exceptions:

1. User is already registered

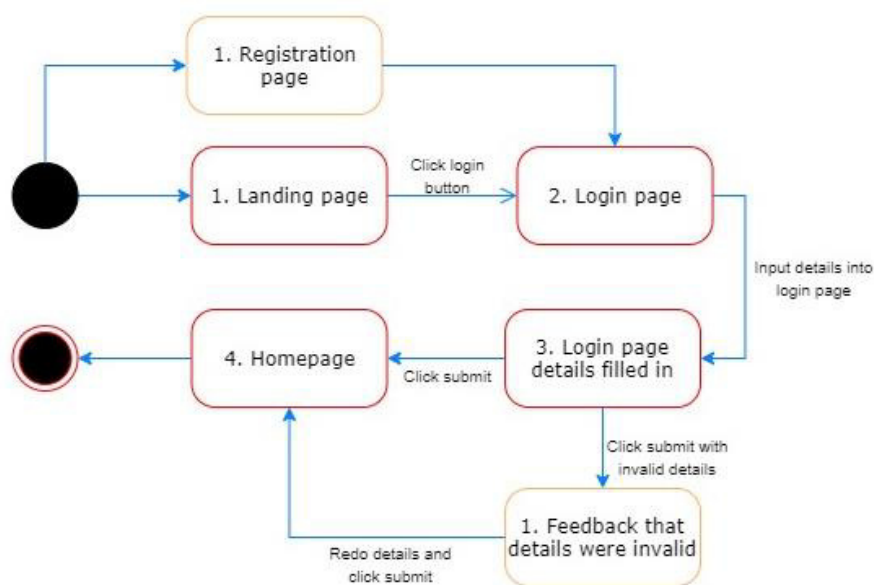
Extensions: N/A

Related Use Cases: Login to account

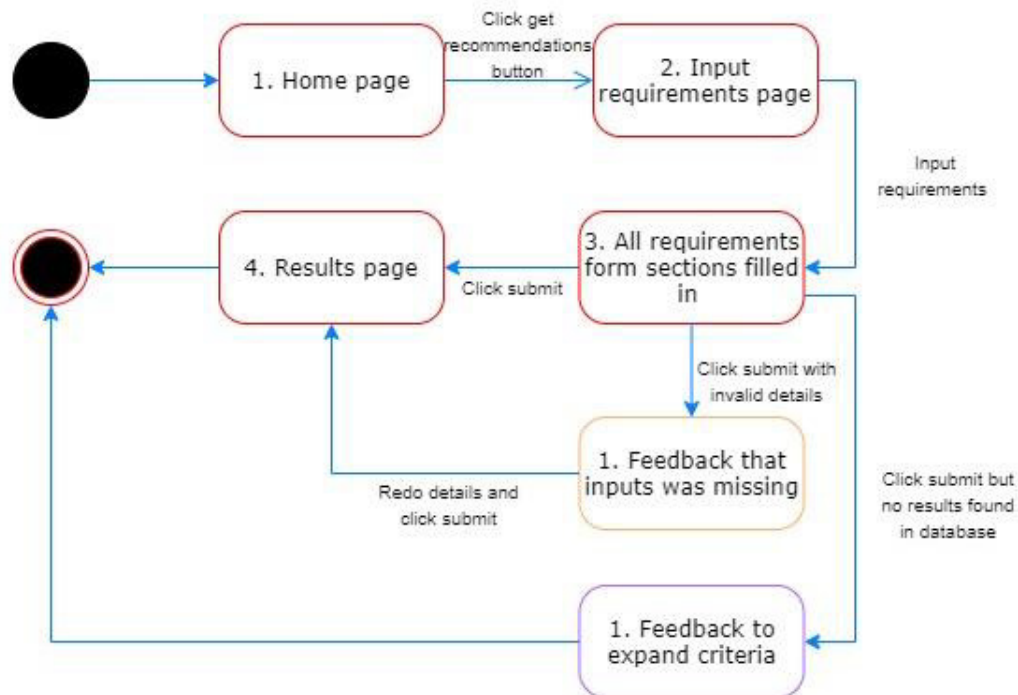
Post-conditions: Users details are saved in database, and they will be redirected to the login page.



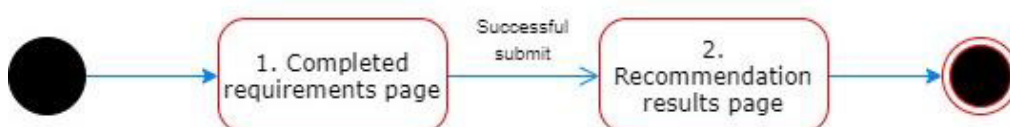
Use Case No: 2	Use Case Name: Login to account	Rating: Must Have
Purpose: For users to login using their details they have registered with		
Main Actor: Teenagers	Secondary Actor: Parents	
Pre-conditions: Users have registered their details		
Trigger: Users click on the login button		
<div>Description</div> <div><div>1. Users enter the login page</div><div>2. Users input their details into the boxes</div><div>3. Users click the login button</div></div> <div>Alternative flows:</div> <div><div>1. User inputs invalid details into the boxes</div><div>2. Users click login button</div><div>3. User is returned to login page with feedback</div></div>		
Extensions: N/A		
Related Use Cases: Register an account		
Post-conditions: Users are redirected to the homepage.		



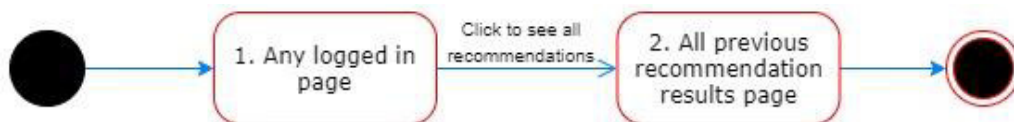
Use Case No: 3	Use Case Name: Input requirements	Rating: Must Have
Purpose: For users input their activity requirements and desirables		
Main Actor: Teenagers	Secondary Actor: Parents	
Pre-conditions: Users have logged in		
Trigger: Users click either the ‘start your search button’ or the ‘input’ option from the navigation.		
<div>Description</div> <div>Main flow:</div> <div><div>1. Users enter the input requirements page</div><div>2. Users answer all the questions</div><div>3. Users click to complete the task</div></div> <div>Alternative flows:</div> <div><div>1. Users do not answer all the questions</div><div>2. Users click to complete the task</div><div>3. Users are returned to input page with feedback</div></div> <div>Exceptions:</div> <div><div>1. No results are found in database within the parameters specified</div></div>		
Extensions: Users can either scroll down the page or click the next button to be taken to the next step without scrolling.		
Related Use Cases: View recommendations		
Post-conditions: Users are redirected to a results page		



Use Case No: 4	Use Case Name: View recommendations	Rating: Must Have
Purpose: For users to view their recommended activities		
Main Actor: Teenagers	Secondary Actor: Parents	
Pre-conditions: Users have logged in		
Trigger: Users have completed their requirements input		
Description Main flow: <div><div>1.</div>User is redirected to the recommendation page</div> <div><div>2.</div>Users are provided details about their top recommended activities</div>		
Extensions: Users can re-do their requirements if they wish		
Related Use Cases: Input requirements		
Post-conditions: Users results are stored in the database.		

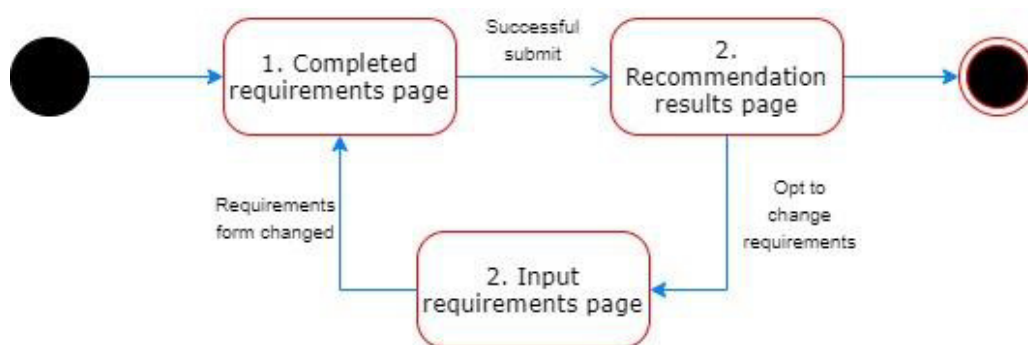


Use Case No: 5	Use Case Name: View all recommendations	Rating: Should Have
Purpose: For users to review all their recommended activities in one location		
Main Actor: Teenagers	Secondary Actor: Parents	
Pre-conditions: Users have completed their requirements		
Trigger: Users have got a minimum of one result		
<div>Description</div> <div>Main flow:</div> <div><div>1. User clicks link to see all recommendations</div><div>2. All recommendations are presented on one page</div></div>		
Extensions: Page will be blank if the user has not completed the input requirements form at least once.		
Related Use Cases: View recommendations		
Post-conditions: N/A		

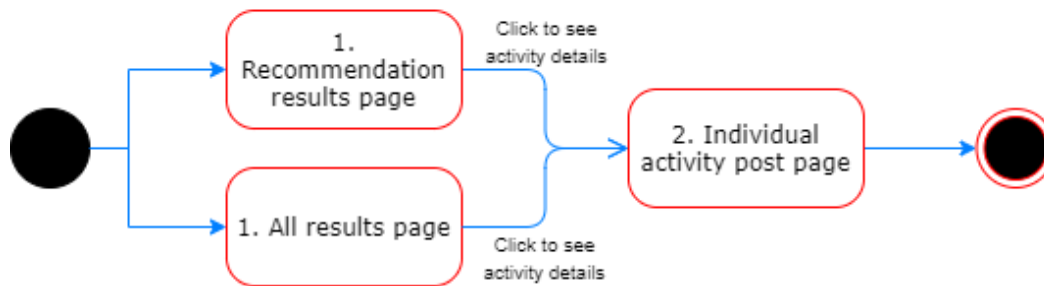


Use Case No: 6	Use Case Name: Edit requirements	Rating: Should Have
Purpose: For users to change their requirements		
Main Actor: Teenagers	Secondary Actor: Parents	
Pre-conditions: Users have already inputted some requirements		
Trigger: Users decide they want to change their answers		

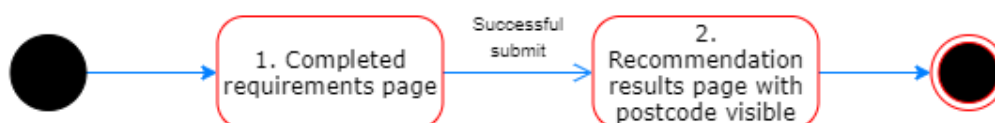
Description Main flow: <ol style="list-style-type: none"> 1. Users can go back on their input page 2. Users can edit their answers 3. Users are revealed with an updated result
Extensions: N/A
Related Use Cases: Input requirements
Post-conditions: The latest result will be displayed. Old and new results will be stored in all recommendations page.



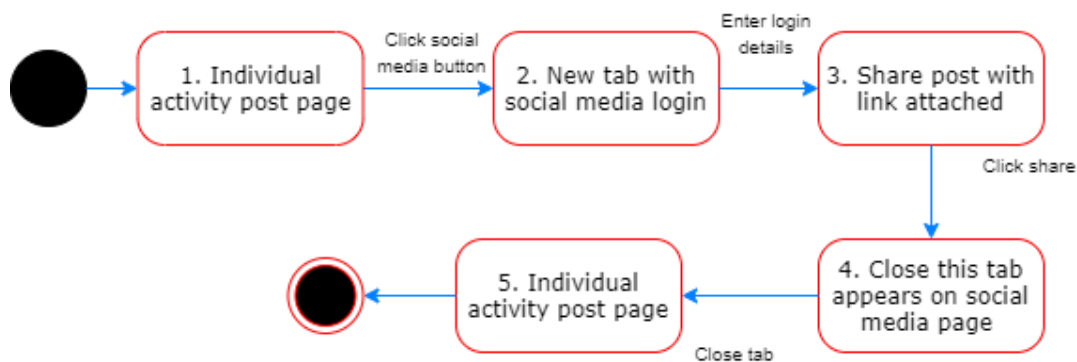
Use Case No: 7	Use Case Name: View activity details	Rating: Must Have
Purpose: For users to view detailed information about a particular activity organisation that they are recommended		
Main Actor: Teenagers		Secondary Actor: Parents
Pre-conditions: User has got at least one recommended activity		
Trigger: User clicks on button to view activity details		
Description <ol style="list-style-type: none"> 1. User is on either recommendation results or all results page 2. User clicks on button to view activity details 3. User is taken to a detailed page 		
Extensions: N/A		
Related Use Cases: View recommendations, view all recommendations		
Post-conditions:		



Use Case No: 8	Use Case Name: Locate activities	Rating: Should Have
Purpose: For users to visualise where their recommended activities are		
Main Actor: Teenagers		Secondary Actor: Parents
Pre-conditions: Users have completed their requirements		
Trigger: Users are on a results page		
Description <ol style="list-style-type: none"> 1. Users are on a results page 2. Users can see the postcode of the activity 		
Extensions: Users to see these locations on an integrated map		
Related Use Cases: View recommendations		
Post-conditions: Users can enter individual activity page where full address will be stated		

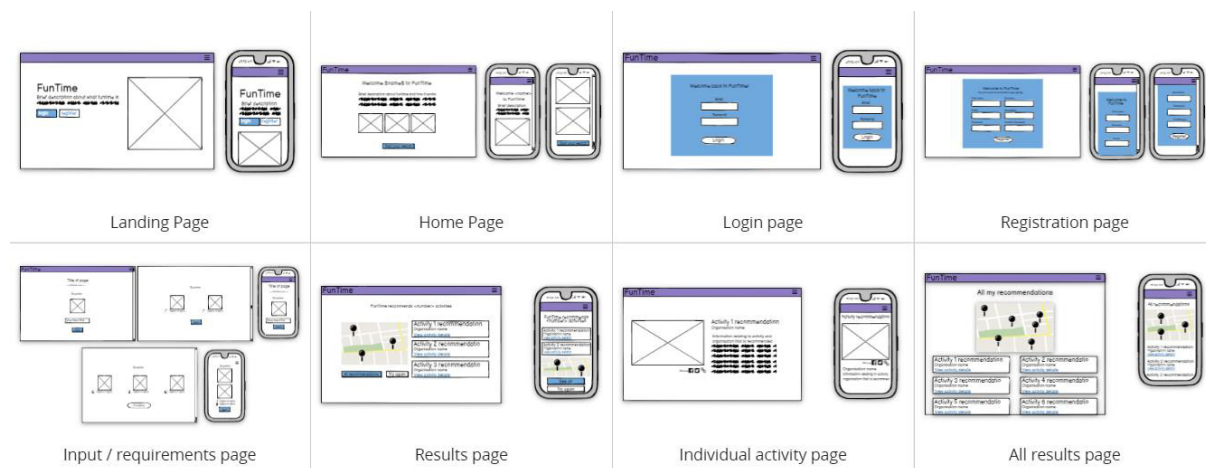


Use Case No: 9	Use Case Name: Share recommendations	Rating: Could Have
Purpose: For users to be able to send their recommendations to other people, so that, they can make use of the recommendation		
Main Actor: Teenagers	Secondary Actor: Parents, Social Media Contact	
Pre-conditions: User has got at least one recommended activity		
Trigger: User clicks on button to view activity details		
Description <ul style="list-style-type: none">1. User is on individual activity post page2. User clicks on share button3. User has options to share on Facebook, Twitter, or email		
Extensions:		
Related Use Cases: View recommendations		
Post-conditions: User is returned to results page once completed, user has some sort of feedback that share has worked.		



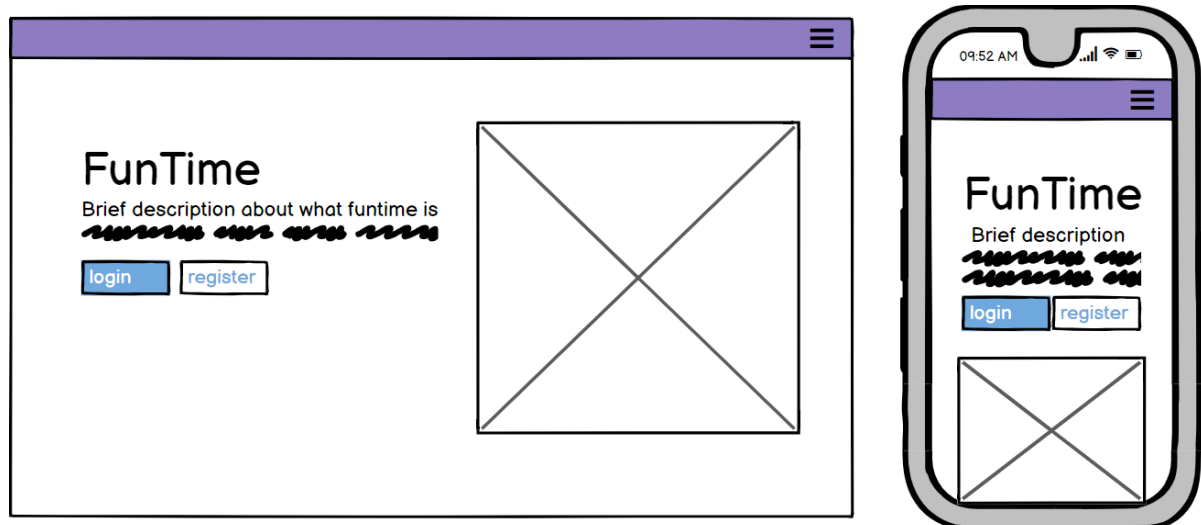
Prototypes / Wireframes

For the user interface, I used the web application 'Balsamiq' to design basic wireframes of each page. Initially, these designs were only for desktop devices, however, following the results of the user survey (question 10)– with the benefit of working in agile – I returned to the designs to create mobile equivalents. The wireframes consist of a 'landing' page, 'home' page, 'login' page, 'registration' page, 'user input' page, 'results' page, 'individual activity' page, and 'all results' page.



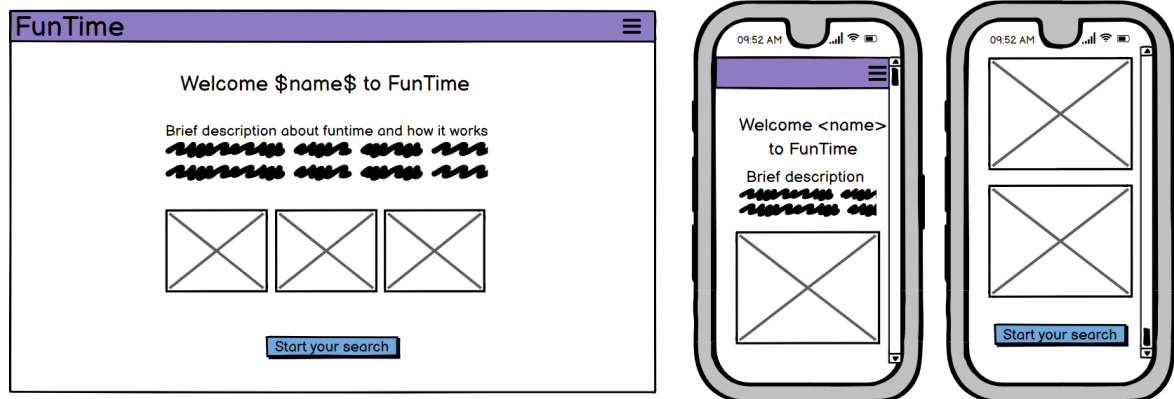
1. Landing Page

This is the page that users will be directed to when they first enter the website. This is to create a difference for users to make it very clear when they are logged in versus when they are not. I also wanted a clear direction to register and login, so that, users understand they need to login to use the tools that the portal offers.



2. Home Page

This is the main home for all users once they are logged in. It provides some information about the portal. Although it does not have much utility, it is a page necessary for users to return to as a base.



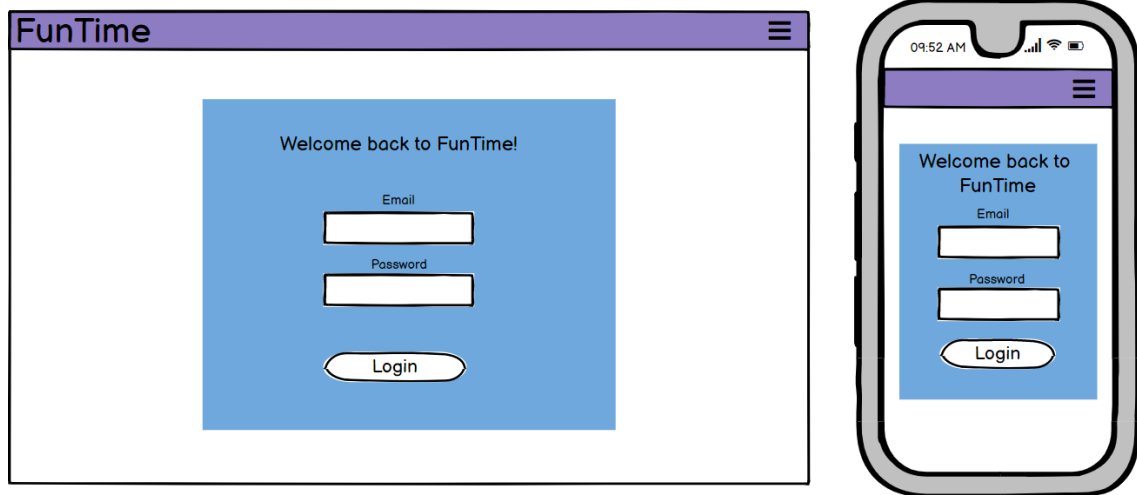
3. Registration Page

This is a basic form page where the user will create their account.



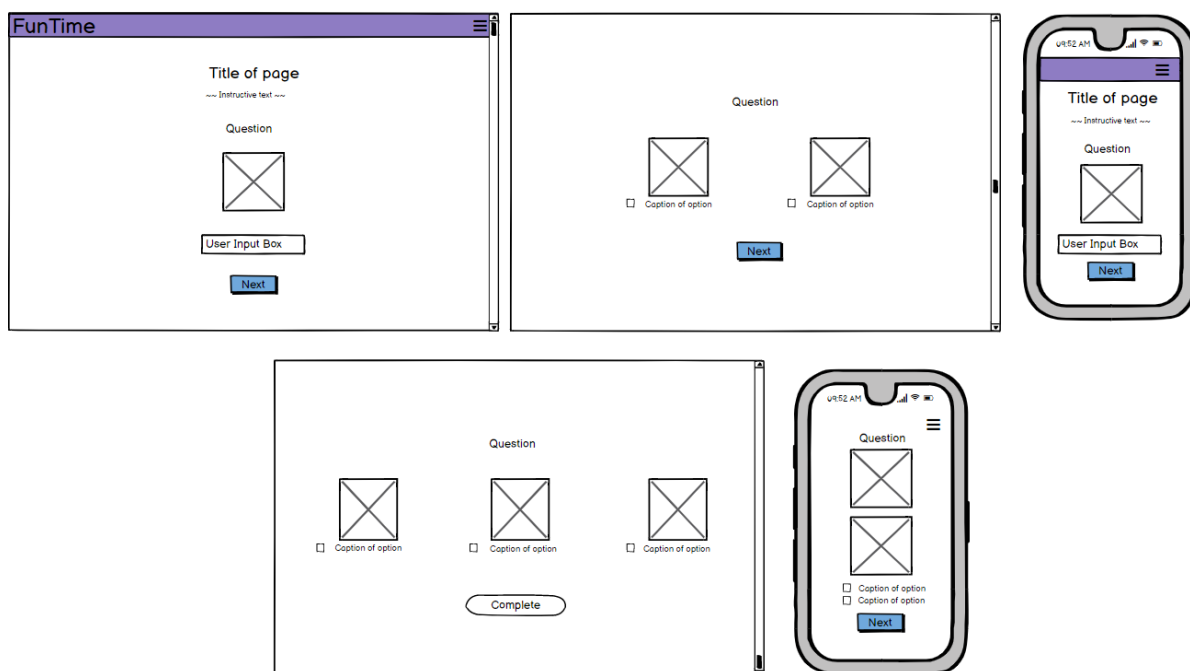
4. Login Page

This uses a similar design to the registration page. Although I asked for a username in the registration page, I will only ask for the users email and password to login.



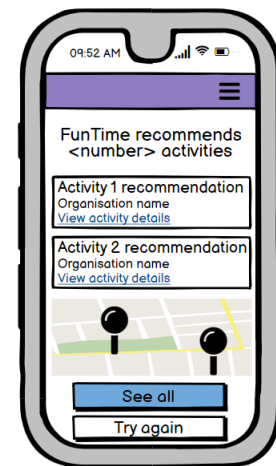
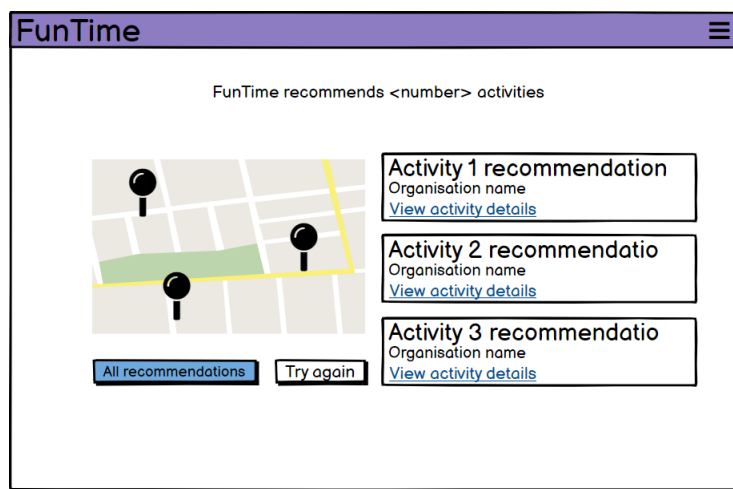
5. Input page

Here the users will be filling in details that are required to make the recommendation system work. This is their user preferences. This works by being one long page because, when using CSS, a Flask Form over multiple pages is extremely challenging without saving the session. The user will be asked a series of questions and can either scroll or click the next button between each answer. There are images that represent the question or answers. Some answers will be input boxes, whilst some will be radio buttons.



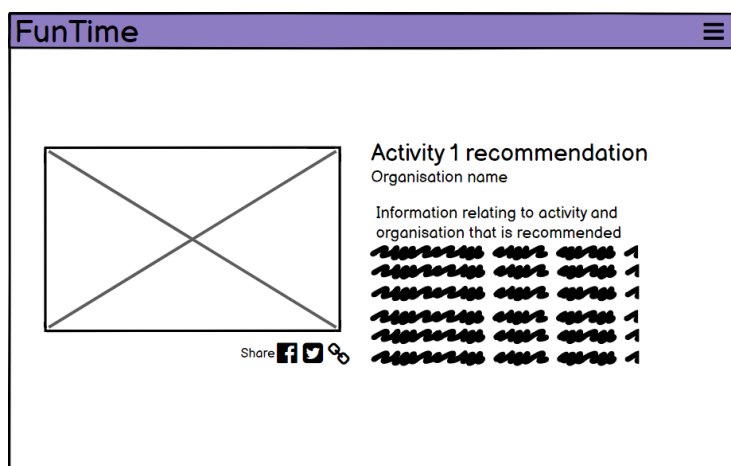
6. Results page

The results page will show an interactive map that has markers for each of the activity organisation locations that have been recommended as a result of the users most recent input. The activity name, organisation name and a link to view more details will be listed on the right column, whilst the map will be on the left. This information will be dynamically displayed according to the user results, and the number of results will change according to what the algorithm and database produce.



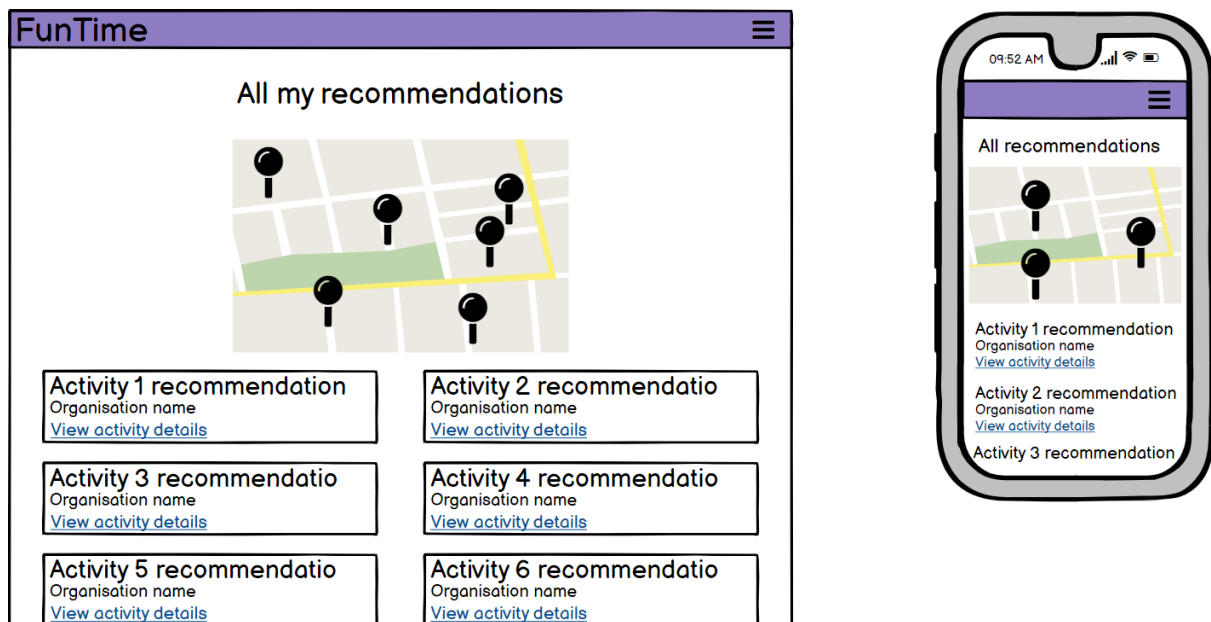
7. Individual activity page

The single activity page will present the most information about a particular activity that was recommended to the user. It will be in depth and can be accessed by the 'View activity details' link on both the results and all results pages. Instead of a map, an image of the activity will be on the left side of the page along with the buttons to share to different social media platforms.



8. All results page

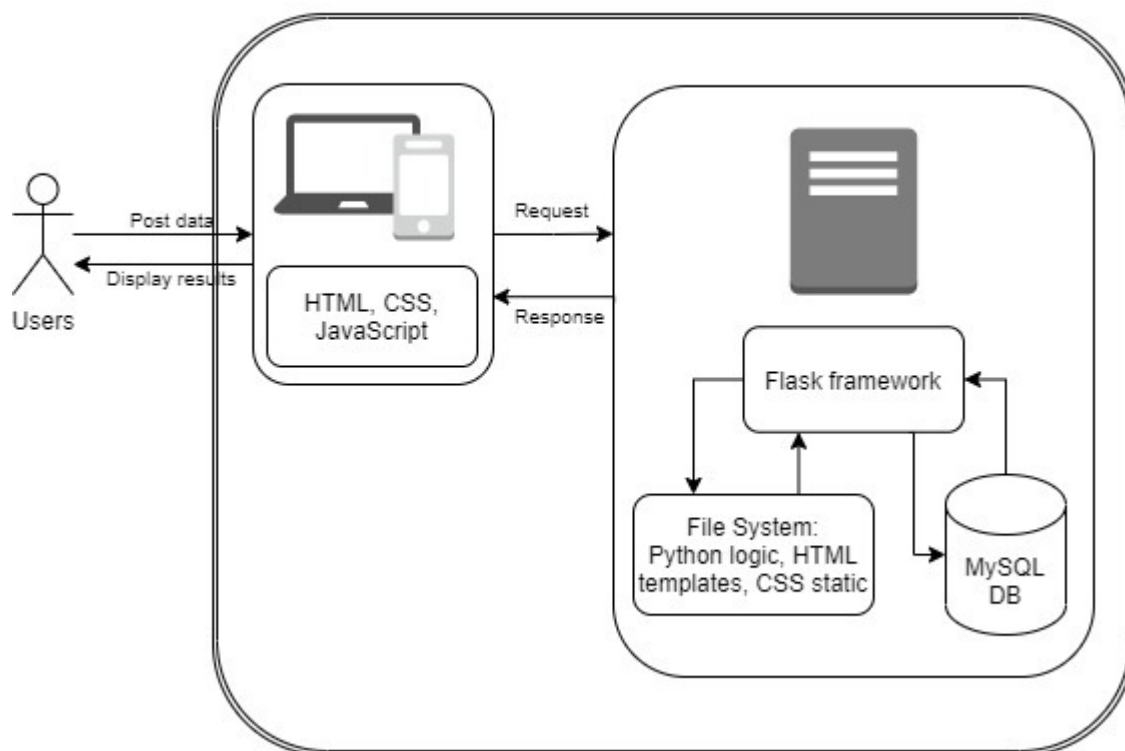
The 'all results' page is an extension of the results page, however, shows the user every result that they have ever gathered from the portal. Similar to the results page it will show a map with pointers to all of the recommendation results produced for that user.



System architecture

The diagram below demonstrates the system architecture of the web portal. This consists of a basic HTML, CSS and JavaScript frontend, and the server-side consisting of a Flask framework which holds the file system including Python logic, HTML templates and CSS static pages. The server side also consists of a MySQL database which is called from the flask framework.

FunTime portal system architecture



Algorithm design

A key deciding factor in the algorithm design was the cold-start problem that exists within collaborative filtering and wanting to create a portal that does not require other users' data. These two factors were pivotal in deciding to use a content-based filtering system. Once the decision had been made on this, exploration on the best similarity measure was necessary.

Cosine similarity

After reviewing background material on different recommendation engine models, one that struck out was cosine similarity. This was the initial idea for the recommendation system to be based on. Cosine similarity works on the premise that if a user likes one thing, it will recommend other items that have similar features. A basic model that was practiced used descriptions of each extracurricular activity. A user would input an activity that they currently like and it would result in a similar activity that they should try. This is an example of content-based filtering as it explores what the user has done before rather than the user rating a selection of different activities. Although this is useful for many recommendation systems, for example: books, movies, and retail, there were obvious problems with this algorithm being the backend logic of this project. To expand, questions arose such as: what if someone does not like any activity in the first place? What if someone is interested in

something completely new as they are just being shoe-horned into an activity that is just like one they already do? And the idea of this project is to recommend the “right” extracurricular activity for them. Just because a user currently likes one activity, does not mean it is the “right” one, but this algorithm is restrictive to what a user already knows. Therefore, this algorithm design was scrapped for a more appropriate option.

Euclidean distance

Researching Euclidean distance, often it is associated to collaborative filtering, however, my project is using this similarity measure for a content-based approach. The Euclidean distance is the straight-line distance between two points, where a point is a vector. The actual distances are not too much of interest, but the nearest neighbour to a particular point is more relevant (Shimodaira 2015). The premise behind this algorithm is to create a user vector and then compare the similarity through seeing which are the nearest neighbours to the activity vectors. The ability to create vectors based upon different features for each activity and a vector for the user based upon features of each activity proved a viable and successful option. Rather than retrieving results that are similar to what a user likes currently, the user is able to state their preferences based upon features and the result closest match (nearest neighbour) is recommended. This enables the “right” activity to be recommended, allowing new activities perhaps they were unaware of to be presented to the user rather than being restrictive to the user in a sort of “filter bubble” (Pariser 2011).

Database design

For this project, I decided to use an SQL database as part of the backend architecture. There are many benefits to using a database over a CSV file, and with the requirements set in this project a SQL database proved the most suitable option. Along with being easy to integrate with my Flask application, it enabled an aesthetically pleasing UI rather than displaying a pandas data frame that would have been the result of the alternative approach.

Four main tables exist. These are User, Activity, User_requirements and User_results.

1. User

The User table is a basic table to store information about the user primarily used for registering and logging in. Users require to be authenticated to enter the portal and see recommendations.

```
class User(UserMixin,db.Model):
    id=db.Column(db.Integer,primary_key=True)
    firstname=db.Column(db.String(35),unique=False, nullable=False)
    lastname=db.Column(db.String(35),unique=False, nullable=False)
    username=db.Column(db.String(15),unique=True, nullable=False)
    email=db.Column(db.String(120),unique=True,nullable=False)
    password_hash=db.Column(db.String(128))
    password=db.Column(db.String(60),nullable=False)
```

2. Activity

The activity table is used to store all the data for each activity organisation that exists. For each activity a row exists for different variables that it offers. For example, if an organisation offers services for multiple ages, a new row exists for each age variable.

```
class Activity(db.Model):
    id=db.Column(db.Integer, primary_key=True)
    name=db.Column(db.String(40), nullable=False)
    description=db.Column(db.Text, nullable=False)
    org_name=db.Column(db.Text, nullable=False)
    Type=db.Column(db.String(15), nullable=False)
    time=db.Column(db.String(20), nullable=False)
    age=db.Column(db.Integer, nullable=False)
    sex=db.Column(db.String(6), nullable=False)
    InOrOut=db.Column(db.String(10), nullable=False)
    social=db.Column(db.String(15), nullable=False)
    postcode=db.Column(db.String(8), nullable=False)
    longitude=db.Column(db.String(10))
    latitude=db.Column(db.String(10))
    budget=db.Column(db.String(15), nullable=False)
    address=db.Column(db.Text, nullable=False)
    city=db.Column(db.String(40))
    contact_name=db.Column(db.Text)
    contact_details=db.Column(db.Text)
    URL=db.Column(db.Text)
    image=db.Column(db.String(40))
```

3. User requirements

The user requirements table is inserted information from the authenticated user as a result of their submitted form. In order to retrieve all information such as longitude and latitude, the user inputs their postcode and I use geopy in python to create a variable for longitude and latitude which is included in the variable for the database session.

Furthermore, so that I know who the requirements are about, the User id is another column. Although this could have been done relationally, I found it easier to add as a variable using the 'current_user' variable in Flask.

```
class User_requirements(db.Model):
    id=db.Column(db.Integer, primary_key=True)
    postcode=db.Column(db.String(8), nullable=False)
    sex=db.Column(db.String(6), nullable=False)
    InOrOut=db.Column(db.String(10), nullable=False)
    age=db.Column(db.Integer, nullable=False)
    budget=db.Column(db.String(15), nullable=False)
    distance=db.Column(db.Integer, nullable=False)
    Type=db.Column(db.String(15), nullable=False)
    time=db.Column(db.String(20), nullable=False)
    social=db.Column(db.String(15), nullable=False)
    longitude=db.Column(db.Float)
    latitude=db.Column(db.Float)
    user_id=db.Column(db.Integer)
```

4. User results

Once the algorithm has compared information from the activity table to the user requirements, the activities which have been identified as the closest match are inputted into the user results table. This has the same schema as the activity table as it is a simple replication but with a few additional columns. The additional columns include a 'new_id', 'user_id' and 'number_of_records'. The 'new_id' was a late addition as a result of a bug meaning that once an activity was inserted into the table it could not be inserted again due to its primary key status. Therefore, I created a 'new_id' which increments appropriately allowed for the same record to enter the table as many times as it is produced as a result. The 'user_id' column links the result to the user who did it, and the 'number_of_records' column means that I can translate the number of results from a user input into the number of activities that get displayed on the webpage.

```
class user_results(db.Model):
    new_id =db.Column(db.Integer, primary_key = True)
    id=db.Column(db.Integer)
    name=db.Column(db.String(40), nullable=False)
    description=db.Column(db.Text, nullable=False)
    org_name=db.Column(db.Text, nullable=False)
    Type=db.Column(db.String(15), nullable=False)
    time=db.Column(db.String(20), nullable=False)
```

```
age=db.Column(db.Integer, nullable=False)
sex=db.Column(db.String(6), nullable=False)
InOrOut=db.Column(db.String(10), nullable=False)
social=db.Column(db.String(15), nullable=False)
postcode=db.Column(db.String(8), nullable=False)
longitude=db.Column(db.String(10))
latitude=db.Column(db.String(10))
budget=db.Column(db.String(15), nullable=False)
address=db.Column(db.Text, nullable=False)
city=db.Column(db.String(40))
contact_name=db.Column(db.Text)
contact_details=db.Column(db.Text)
URL=db.Column(db.Text)
image=db.Column(db.String(40))
user_id=db.Column(db.Integer)
number_of_records = db.Column(db.Integer)
```

Data sources

One challenge that this project faced was the lack of data source to build a recommendation engine from. Many other topics that have been used in recommender systems have files that can be downloaded and used for the benefit of their work, such as movie recommendation engines. However, there was no such database to extract from. Therefore, a lot of time was spent to build a database that included the information necessary to produce a working prototype.

Beginning with a list of potential activities was the first step, then using Oxford dictionary definitions to add an activity description, and then finding the organisations that do the activity with information surrounding it took extensive time. To keep this in the remit of the project, only a selection of activities was collected and a minimum of one – maximum of four organisations were attached to each activity. If this project was to be continued, the ability for organisations to input their details into the portal could be added to expand the database.

Decision on variables used

As no database was present for this project to base ideas from, one task was to decide how to describe all activities. Initially I tried to work out what variable titles would be appropriate. Then, within these variables decided what the options within them would be. This decision was a result of the user survey, background research and my own thoughts. I

did not want to overwhelm the user with too many questions or options, but equally needed a solid set of requirements so that their best matches would be resulted from the algorithm, to meet the overall project aim.

I settled on:

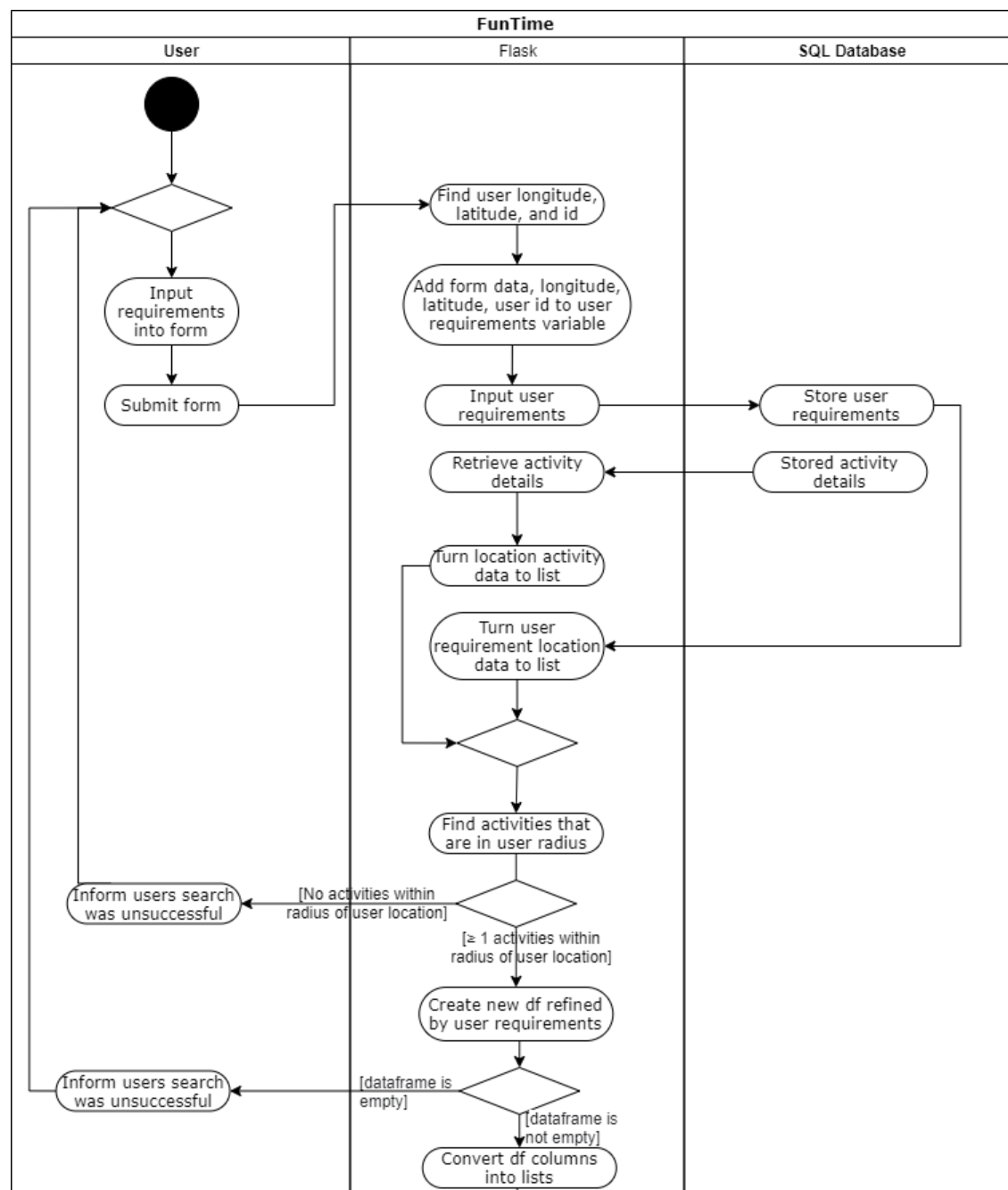
```
class RequirementsForm(FlaskForm):
    postcode = StringField('postcode', validators=[DataRequired()])
    distance = IntegerField('travel', validators=[DataRequired(), NumberRange(min=1, max=20)])
    age = IntegerField('age', validators=[DataRequired(), NumberRange(min=13, max=18)])
    sex = RadioField('sex', choices=[("Male"), ("Female")], validators=[DataRequired()])
    budget = RadioField('budget', choices=[("Low", 'Cheap'), ('Mid', 'Average'), ('High', 'Not an issue')], validators=[DataRequired()])
    Type = RadioField('type', choices=[("Sport"), ('Arts / Culture'), ('Skills')], validators=[DataRequired()])
    time = RadioField('When', choices=[("Afterschool"), ("Weekend")], validators=[DataRequired()])
    social = RadioField('group or independent', choices=[("Group"), ('Independent')], validators=[DataRequired()])
    InOrOut = RadioField('indoors or outdoors', choices=[("Indoors"), ("Outdoors")], validators=[DataRequired()])
    submit = SubmitField('Complete')
```

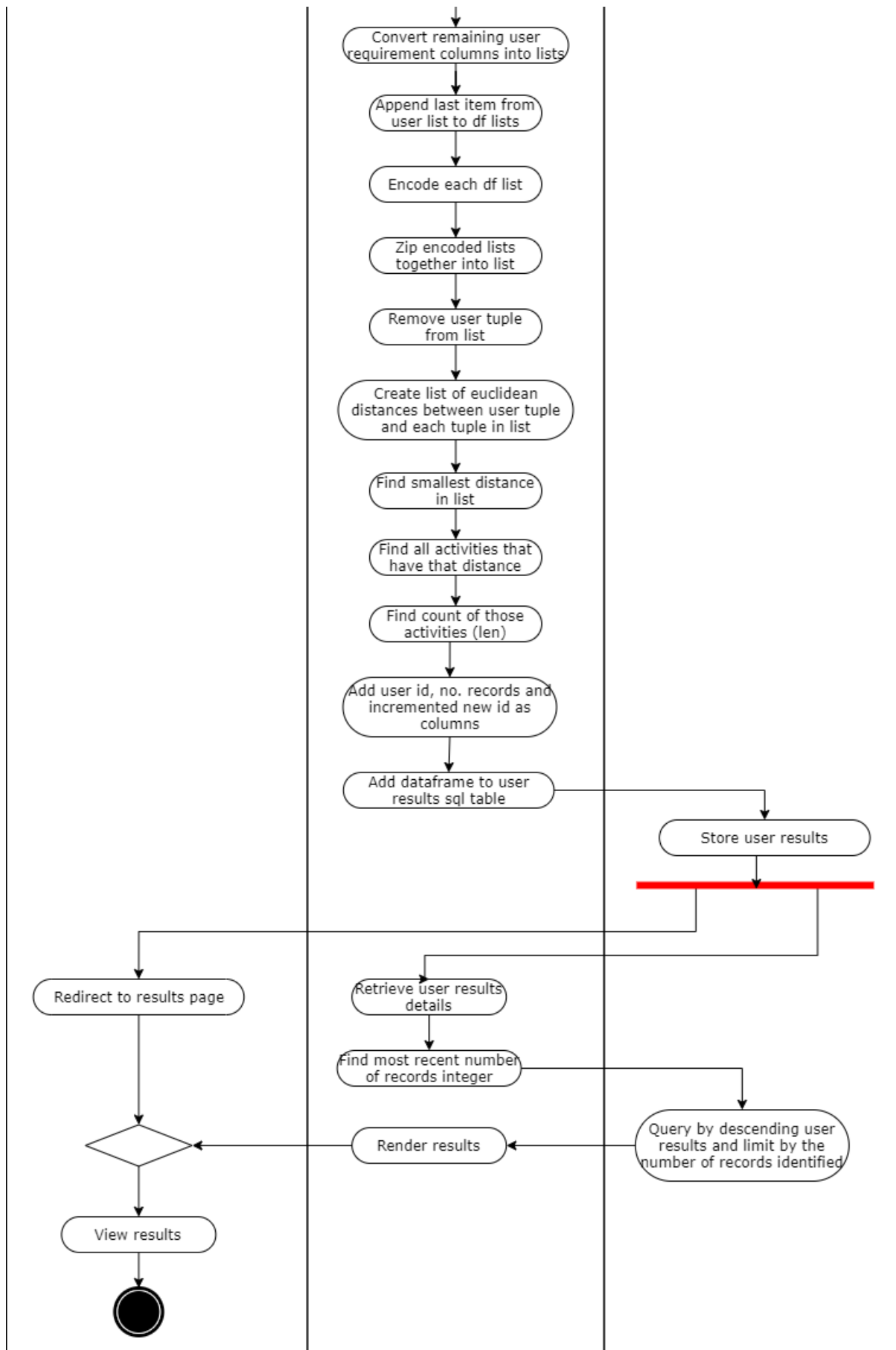
Implementation

To create this prototype, as already noted, I used an agile working method which allowed changes and edits throughout. The agile approach was done in groupings to make the implementation clear: I began with practising the algorithm, then flask framework, smaller use cases, implementing the algorithm, integrating an interactive map, and then styling with CSS.

Below is an activity diagram to demonstrate the process of the application logic.

UML Activity Diagram





Algorithm

I began by focusing on the algorithm that would produce suggestions. I practiced this in Jupyter Notebook because I find that a useful tool to write and rewrite successful code. As mentioned already, the algorithm design was between two similarity measures which was narrowed down to use Euclidean distance.

An integral part of the code was the pre-processing, as Euclidean distance requires data to be real values, encoding was necessary. There are many options to encode data, however, a simple solution was scikit-learn 'LabelEncoder' that enables the transformation of non-numerical labels to numerical labels. Once the data was encoded, I needed a method to put all of the variables from the encoded data together, so that each activity had its own vector. The solution to this was zipping the variables. To exemplify this: if the encoded data was as follows:

```
Activities = [Badminton, Art, Hockey, Swimming]
```

```
Type = [1, 2, 1, 1]
```

```
Time = [1, 1, 2, 2]
```

```
InOrOut = [1, 1, 2, 1]
```

This would be zipped together as a list in a variable:

```
features = list(zip(Type, Time, InOrOut))
```

To produce:

```
[(1, 1, 1), (2, 1, 1), (1, 2, 2), (1, 2, 1)]
```

Once the data was encoded, it was a matter of finding the Euclidean distance from what a user's data would be. I began by creating user dummy data with pre-set values simply to get the Euclidean distance to work. Euclidean distance can be calculated using from 'scipy.spatial import distance' and using the distance.euclidean code. Continuing with the above example, this worked like this:

```
from scipy.spatial import distance
```

```
user = (1, 2, 2)
```

```
ecd_list = []
```

```
for x in features:
```

```
    ecd = distance.euclidean(x, user)
```

```
    ecd_list.append(ecd)
```

This produces a list of distances from the user variable to the vectors within the features list, with the smallest number being the closest distance.

In this example, the results are:

```
[1.4142135623730951, 1.7320508075688772, 0.0, 1.0]
```

The next step of the practice logic was to retrieve the smallest number from the list and identify which activity that corresponded to. In this example, the smallest number was 0.0 and was in index position 2 meaning that the activity closest to the user requirements was Hockey.

After practising on basic variables, I incorporated pandas into the logic to read a CSV with a greater volume of dummy data to ensure the logic worked, which it did.

After creating a basic model that the recommender system would be based off, I moved over to begin on the start of the algorithm which focused on a simple search query. The idea of this was that some things people need to have, and I could not attach those attributes in an equivalent way to the recommender logic. For example, the location of activities is critical whereas if it is indoors or outdoors is simply a preference. Therefore, I narrowed the database down by the logic of a search – if the data matches the user requirements, the dataframe will include that row, otherwise it will not.

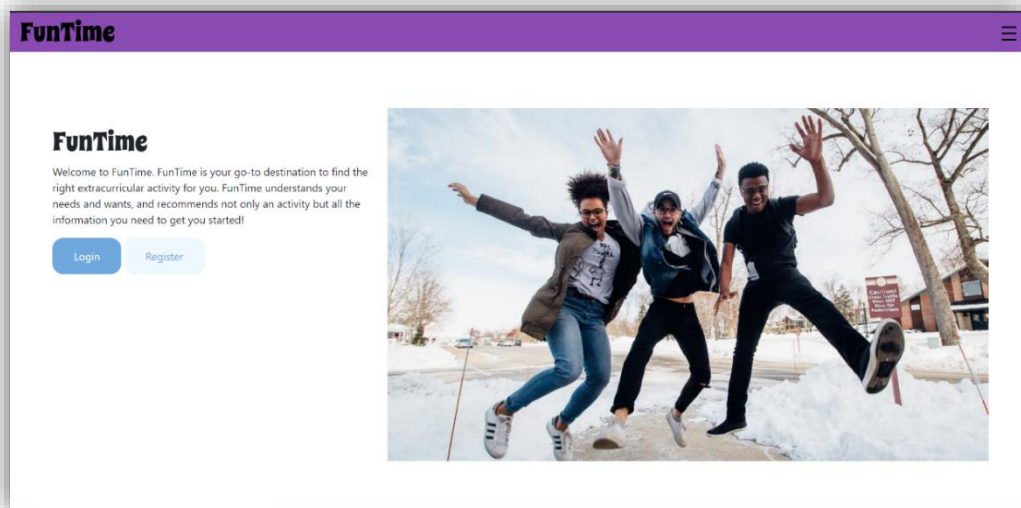
The most difficult step was the distance measure for locations. I found a function online (Stackoverflow 2011) called haversine which calculates the circle distance between two points on earth using longitude and latitude. Then with a list to iterate over in a for loop, checked if the distance between the user longitude and latitudes to the longitude and latitudes in the list were within the radius specified. Making this a new column into the dataframe, this enabled a new search to be conducted where we know which activities are in the user radius, and which are not.

Once the location distance was created, it was simply deciding on which variables should be included in the search and which should be included in the recommender. It was decided that age, sex, distance, and budget were all necessary in the search section.

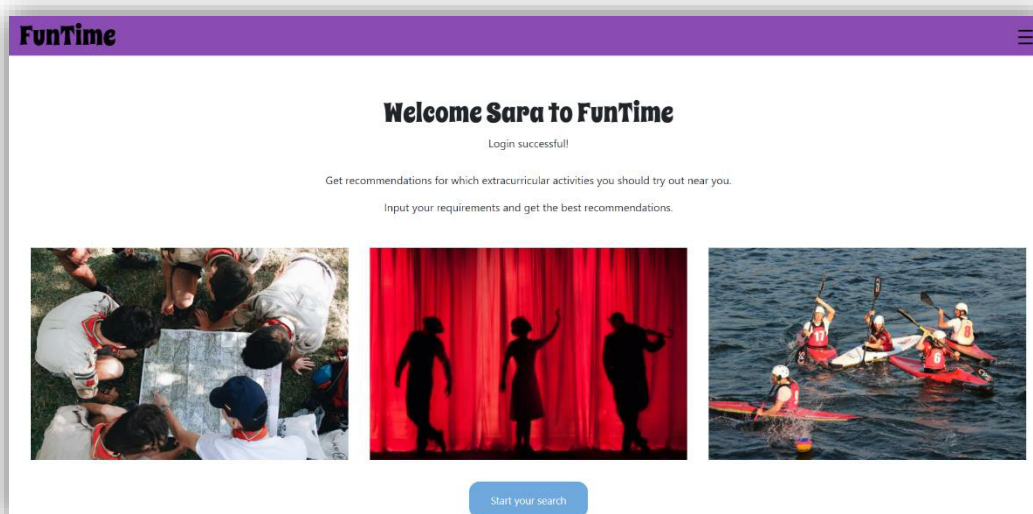
System development

Flask Framework

After settling with the basic template that the algorithm would follow to produce suitable recommendations, I moved over to work on the Flask framework. I set up the framework creating a routes file, an API file named `__init__`, static file, a layout HTML page, and a home HTML page to get going. Once the basic framework was set up anything else could go on top easily.



Landing page



Home Page

Database

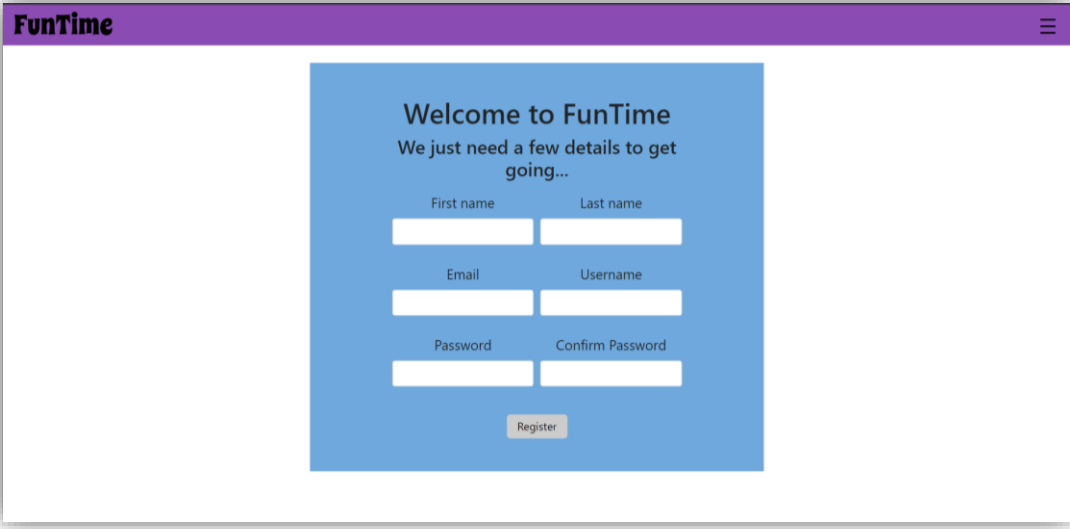
A good starting point was setting up the initial two user stories whereby people are able to register and login to an account, these were easy wins as the code is not difficult. This went into the 'models' python file and I ensured that user's password was hashed to deal with the non-functional security requirement. These tables were then created in SQL using python on the terminal 'from portal import db', 'db.create_all()'.

Simultaneously to setting up the Flask framework, I went about finding the data to be input into the framework. This laborious task took a couple of weeks persistent work to find all of the activities and decide upon the variables it would be measured by. I kept the records in a CSV file that could be edited freely.

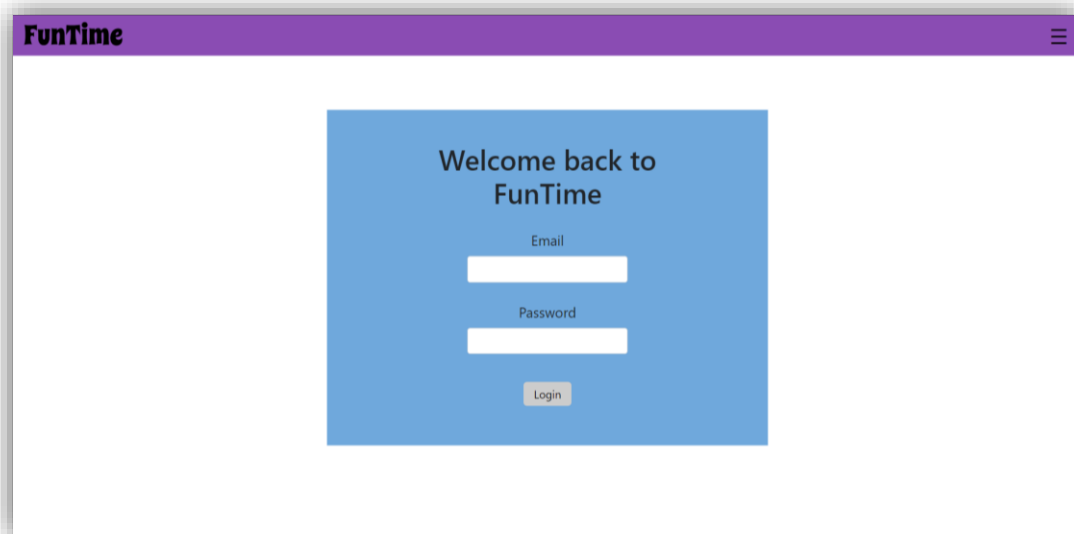
Once the variables were set upon, I could create the database integration with Flask. Using the python file 'models', the activity table for the database was created. This table went under a few iterations whilst defining what requirements users would choose from to base their suggestions off.

Forms

Parallel to the models created, appropriate forms were required that would enable user input. With the user model, an appropriate register and login form was created ensuring that all of the variables in the model was accounted for.

The image shows a web browser window with a purple header bar containing the text "FunTime" on the left and a hamburger menu icon on the right. The main content area is white and features a blue rectangular registration form. The form has the heading "Welcome to FunTime" followed by the subtext "We just need a few details to get going...". Below this, there are four input fields arranged in two columns: "First name" and "Last name" in the top row, "Email" and "Username" in the second row, and "Password" and "Confirm Password" in the third row. At the bottom of the form is a light orange button labeled "Register".

Register page



Login page

User requirements


After setting up a good proportion of the backend logic, the next step was to work out how user requirements would be gathered, since up to this point only dummy data was used. I used a form just as the user would to create an account, and then created a user requirements table in models. The user would be asked to choose an answer from each question. I created either written answers through using StringField or IntegerField and used RadioField to create options with predefined answers. Once I had a method and implementation concluded to get the user requirements from the frontend, the next step was inputting all of the practice logic into the flask framework and piece it all together.

FunTime

What type of activity are you looking for?

To start we want to know some of the basics. These are your 'musts'.


Please input your postcode




Next

Input requirements page

Do you prefer activities that are largely indoors or outdoors?



☐ Indoors



☐ Outdoors

Complete

Input requirement page continued

Inputting logic to Flask

This process ran into several problems which I had to find solutions for. These problems were:

1. How to retrieve the data from the SQL database to use pandas logic.
2. How to make user requirements encode the same as the activity encoded data.
3. How to show only the current users results.
4. How to show only the current users results associated to their most recent search

5. How to display results on a HTML page in a more user-friendly way than a dataframe which pandas offers.

1. To address the first problem, I found an easy solution that created a variable named 'engine' which linked to the SQL account using sqlalchemy, and then using pandas selected the appropriate table from the database. I did this with both the activity table and user requirements table. This created a pandas dataframe for each of the tables to follow the logic that I had already made in Jupyter Notebook.

```
sql_data = pd.read_sql("SELECT * FROM activity", engine)
```

2. For the second point, I realised that the user requirements row needed to be considered as another row in the activity table. This meant that the data would encode the same way as the rest of the data. This was done by turning the user requirements dataframe to lists and getting the last item from the list and putting it in a variable. Then appending that to the end of the list created of the equivalent variable from the activity dataframe (named 'within_parameters'). Once this was part of the list, it could be encoded as part of the same list. E.g.,

```
Type = within_parameters['Type'].to_list()
user_type = user_req['Type'].to_list()[-1]
Type.append(user_type)
type_enc = le.fit_transform(Type)
```

3. To show only the current users results, I used a tool that is offered in the Flask framework to identify 'current_user'. In the dataframe, before pushing it up into a table in SQL named 'user_results', I created a new column where all of the results from the algorithm that was just processed would have the user's id next to it.

```
user = current_user.id
user_id = []
for eachitem in first_tuple_elements:
    user_id.append(user)
```

```
results['user_id'] = user_id
```

Therefore, when displaying the results, the SQL table that stored all of the results was queried by the 'current_user.id'

```
user_id = current_user.id  
results = user_results.query.filter(user_results.user_id == user_id)
```

However, this only solved the problem to query by a user, meaning that if a user had completed multiple requirements forms every result they have ever received would be displayed. Therefore, this made sound logic for the 'all results' page, but not to display the results of their most recent search.

4. This led to the fourth issue, I could find all of the users results, but not the results of their most recent search. To overcome this issue, using a similar idea to the third point, I created another new column in the dataframe before pushing into SQL. Since the number of results could be different each time, I found this number using the following code:

```
number_of_records = (len(first_tuple_elements))
```

(Where 'first_tuple_elements' is the id of every activity with the smallest distance to the user requirements.)

This number is then appended to each line of the items that are going in the results in a similar way to how the user id was.

```
records_no = []  
for eachitem in first_tuple_elements:  
    records_no.append(number_of_records)  
  
results['number_of_records'] = records_no
```

Then to display the results, a more refined query was enabled that would find the current users most recent recommendation results.

```
activities = user_result_sql['number_of_records'].to_list()  
number = activities[len(activities)-1]
```

```
number = int(number)
results = user_results.query.order_by(desc(user_results.new_id)).filter(user_results.user_id
== user_id).limit(number)
```

5. The fifth issue identified was once the results were gathered, how this would actually render on the HTML templates. The variables were parsed in the render_template() parentheses.

```
return render_template('results.html', title='Results', results=results, center=center, number=number)
```

Then on the HTML page, I used a for loop to display each of the results information.

```
{% for result in results %}
<div class="results_box">
  <h3>{{ result.name }}</h3>

  <p>{{ result.org_name }}</p>

  <p><a href="{{ url_for('post', user_results_new_id=result.new_id) }}"> Visit activity page</a></p>

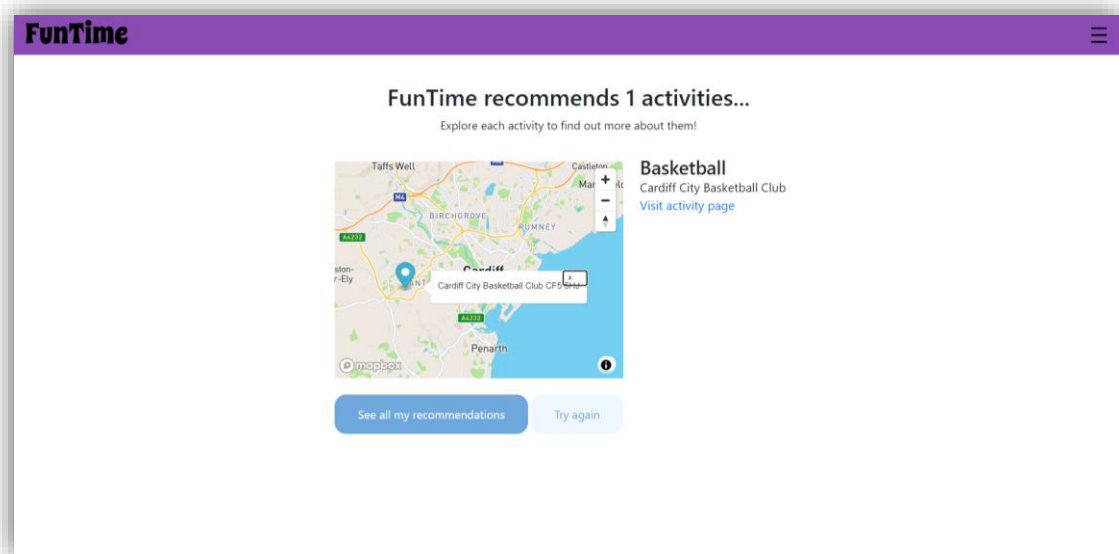
</div>
```

Once this logic was implemented, the base of the web portal was complete. The algorithm works and users are able to input what they need and want and are resulted in suitable extracurricular recommendations.

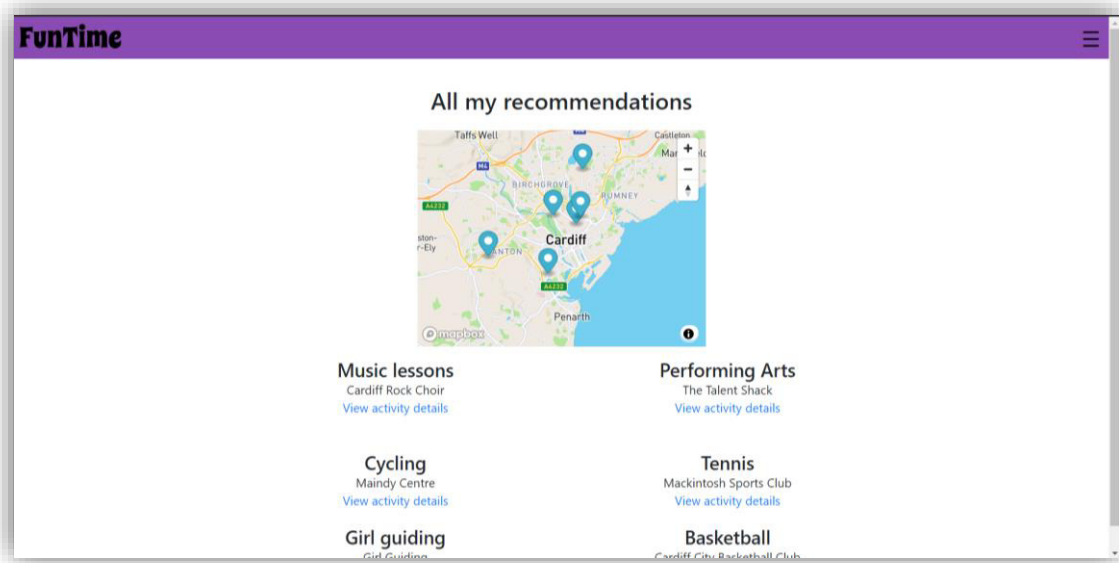
Map integration

Following the user survey that shaped the last iteration of the use cases, an interactive map was identified to be a desirable addition to the web portal. I researched Google maps and Mapbox as possibilities and decided to integrate the Mapbox API into my project. This decision was based off of the stylish UI of Mapbox plus the easy-to-follow documentation that Mapbox offers. Mapbox provides a base map which can be customised. I added

navigation control in the top right of the map and set the centre of the map to the centre of Cardiff (this was because the data used in this prototype revolves around Cardiff, however, could be edited according to the user's postcode or any other longitudinal / latitudinal location). Then I used a for loop within the JavaScript of the Mapbox code to set markers to each of the result locations, with the organisation name and postcode included in the popup associated to the marker. The Mapbox integration is on the 'all results' and 'results' html pages.



Results page



All results page

Social Media Sharing

To complete the use cases, I required a method to share on social media. Flask has its own social media API with a range of sites. I decided to use this and choose the icons for Facebook, Twitter, and Google plus. Because the results pages are queried using the logged in current user id, these pages could not be shared as it would not work for other people who are either not logged in or have a different user id. Therefore, it made sense to put the sharing icons only on the single activity page as this is not refined by any user id logic. Equally this page does not require a user to be logged in to view, meaning that users can share the page to anyone to access.

Frontend styling

Once most of the use cases had been covered from a logic and rendering perspective, nearing the end of the implementation I turned to styling the frontend using CSS. Following the wireframes created, I styled each page for the desktop view first. Then as a result of the user survey I returned to these pages to focus on styling for mobile responsiveness. The styling did not deviate much from the wireframes. I used some bootstrap integration to help style the forms in a greater UI than how Flask renders the input. I also used Google fonts to change font of the title 'FunTime', but other than this the rest of the styling was done by myself in CSS.

Results

Testing

To test the prototype, I decided to complete unit testing of different isolated code. I focused on the main requirements identified in the design phase of the project.

Registration:	
Test 1	
Description	Test that a user is able to register an account
Expected results	A user enters the registration page, inputs their details, and is redirected to the login page with a notification of a successful registration.
Actual results	A user enters the registration page, inputs their details, and is redirected to the login page with a notification of a successful registration.
Pass / Fail	PASS
Test 2	
Description	Test that a user needs the same password to verify the account
Expected results	User clicks the register button, and an error message appears.
Actual results	User clicks the register button, and an error message appears.
Pass / Fail	PASS
Test 3	
Description	User requires unique username and email
Expected results	User clicks the register button, and an error message appears.

Actual results	User clicks the register button, and an error message appears.
Pass / Fail	PASS

Login:	
Test 1	
Description	Test that a user is able to login to their account
Expected results	A user enters the login page, inputs their details, and is successfully logged in with a notification of the login success.
Actual results	A user enters the login page, inputs their details, and is successfully logged in with a notification of the login success.
Pass / Fail	PASS
Test 2	
Description	Test that a user must be registered to login
Expected results	User clicks the login button and is provided with an error message.
Actual results	User clicks the login button and is provided with an error message.
Pass / Fail	PASS
Test 3	
Description	Test that a user can logout of their account
Expected results	User clicks on the logout button and is logged out of their account and redirected to the landing page.
Actual results	User clicks on the logout button and is logged out of their account and redirected to the landing page.

Pass / Fail	PASS
-------------	------

Input requirements:	
Test 1	
Description	Test that a user is able to input their requirements.
Expected results	A user enters the input page, puts in valid data into every question, clicks submit and is redirected to a results page.
Actual results	A user enters the input page, puts in valid data into every question, clicks submit and is redirected to a results page.
Pass / Fail	PASS
Test 2	
Description	Test that a user must be in a valid radius of an activity in the database.
Expected results	User inputs a postcode and radius that does not match any items in the database, clicks the submit button, and is resulted in an error message with prompt to expand search.
Actual results	User inputs a postcode and radius that does not match any items in the database, clicks the submit button, and is resulted in an error message with prompt to expand search.
Pass / Fail	PASS
Test 3	
Description	Test that a user must answer all questions to get a recommendation

Expected results	User does not answer all questions, clicks the submit button, and is directed to the first question which they have not answered.
Actual results	User does not answer all questions, clicks the submit button, and is directed to the first question which they have not answered.
Pass / Fail	PASS
Test 4	
Description	Test that a user must input data in the 'must haves' section that matches at least one item in the database.
Expected results	User inputs data that does not match any items in the database, user is not redirected to results, and it presented with a message to provide feedback why it failed.
Actual results	User is not redirected to the results; however, no message is flashed to provide feedback to explain why the search failed.
Pass / Fail	FAIL

View recommendations:	
Test 1	
Description	Test that a user gets results appropriate to their search.
Expected results	A user has input valid requirements and is displayed a minimum of one recommendation.

Actual results	A user has input valid requirements and is displayed a minimum of one recommendation.
Pass / Fail	PASS
Test 2	
Description	Test that the map pointer matches recommendations
Expected results	User clicks on the map pointer which is at the longitude and latitude of the activity; activity organisation name and postcode is displayed.
Actual results	User clicks on the map pointer which is at the longitude and latitude of the activity; activity organisation name and postcode is displayed.
Pass / Fail	PASS

View all recommendations:	
Test 1	
Description	Test that a user can see every recommendation they have ever had.
Expected results	A user enters the 'all results' page and a list of all of their previous recommendations is visible.
Actual results	A user enters the 'all results' page and a list of all of their previous recommendations is visible.
Pass / Fail	PASS
Test 2	
Description	Test that the map pointer matches all recommendations

Expected results	User clicks on the map pointer which is at the longitude and latitude of any activity; activity organisation name and postcode is displayed.
Actual results	User clicks on the map pointer which is at the longitude and latitude of any activity; activity organisation name and postcode is displayed.
Pass / Fail	PASS

Edit requirements:	
Test 1	
Description	Test that a user can return to their requirements page and put in new data.
Expected results	A user has got a recommendation, clicks the try again button and is returned to the input requirements page.
Actual results	A user has got a recommendation, clicks the try again button and is returned to the input requirements page.
Pass / Fail	PASS

View activity details:	
Test 1	
Description	Test that a user can see details about a specific activity on its own page
Expected results	A user clicks to view activity details; they are taken to a new page which has detailed information about the activity.

Actual results	A user clicks to view activity details; they are taken to a new page which has detailed information about the activity.
Pass / Fail	PASS

Share recommendations:	
Test 1	
Description	Test that a user can share their recommendations.
Expected results	A user clicks on a share link and are redirected to the social media site with the specific page included as a link.
Actual results	A user clicks on a share link and are redirected to the social media site with the specific page included as a link.
Pass / Fail	PASS

Non-Functional requirements:

Security:	
The web portal should be secure, with user data secure and protected against malicious attacks.	OVERALL RESULT: PASS
User passwords should be hashed in the database, rather than stored as plaintext.	PASS
Escape values to rule out cross site scripting (XSS)	PASS
Be protected against database injection attacks with input validation.	PASS
Functional Suitability:	

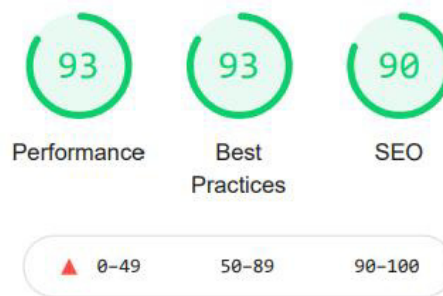
The web portal should have suitable functions to meet user needs to be suggested with extracurricular activities.	OVERALL RESULT: PASS
The system should be functionally correct with results being the closest match to a users' preferences.	PASS
The system should be functionally appropriate through presenting the user with only the necessary steps to get suggestions.	PASS
Performance efficiency:	
The performance of the application should be between 90=100% within Lighthouse standards*	OVERALL RESULT: PASS
The capacity of the database maximum limit should meet requirements with ability to expand with increased data.	PASS
Processing times should be quick, with page load times less than 3 seconds.	PASS
Usability:	
The portal should be user-friendly to both parents and teenagers who will be varied in their computer literacy.	OVERALL RESULT: PASS
Create an aesthetically pleasing interface by maintaining a house style on each page with set colours, navigation, and fonts.	PASS
Navigation should be obvious and accessible, with the ability to return to the homepage in one click.	PASS

Users are presented with error messages to protect them against making errors	PASS
Users to be provided with appropriate informative text to understand how to use the system.	PASS

Interface testing:

Mobile responsiveness: To test the UI (visual appearance) of each page when it is in a mobile view.	
Landing page	PASS
Home page	PASS
Registration page	FAIL
Login page	PASS
Input requirements page	PASS
Results page	PASS
All results page	PASS
Individual activity page	FAIL

*One key indicator of the non-functional requirements was to ensure that the Lighthouse scores for certain metrics were above 90%, in the green. To go into further depth on this point, I tested the landing page against a Lighthouse report. I had to remove accessibility from the report due to an error resulting. The error came from the button text being white and not contrasting from the white background of the overall page. Therefore, I removed that metric from the report as it did not display any scores. However, testing against performance, best practices and search engine optimisation produced high results – all over 90%. This test was done on my local environment; however, the result should not differ on the deployed site.



Overall, the implementation was a success with the overall result of functional and non-functional requirements being a pass. All of the 'must have' use case requirements were successfully completed in a timely manner which enabled time to be spent on the 'should' and 'could have' requirements.

Analysis

This project has created a suitable solution to suggest appropriate extracurricular activities to teenagers. The robustness of the solution can be witnessed through the varied conditions it may be used. Through creating an opportunity of choice with an 'input requirements' page, the solution behaves reasonably to find the closest match to what a user wants – it may not be the exact match but will be the best fit.

One challenge that this project has overcome in its use of algorithm is the cold-start problem. When I researched the different options, I was faced with this problem being mentioned repetitively in collaborative filtering. This project, however, uses the Euclidean distance tool (which is often used in collaborative filtering techniques) to find similarity based off of user preferences and applies to the content of each activity. This manages to overcome the cold start problem, creating a successful solution where I am able to gather the necessary data from the user clearly. A user's preferences do not rely on a large proportion of data to already exist, it is solely based on their own preferences. And once many users have input their data, other methods of recommendation systems could be developed into the application.

The original specification of this project was to create a system that suggested extracurricular activities to teenagers. My initial understanding of the project was not to include organisations but simply suggest the actual activities – after discussions with my

supervisor this was clarified to be a working prototype where location was a key aspect specific to actual organisations that run the activity. The original specification did not consider the scope of the system, therefore after discussions with my supervisor we decided that building a prototype based in the Cardiff area would be successful within the timeframe and would not waste time building an overly extensive database.

The solution is innovative due to both its application and unusual two-step logic. Often recommender systems are used for ecommerce and streaming services. These are areas where there is much data already present of either the item being sold or the user buying. However, this project considers a different type of use case application to the logic that has already been built up in this field.

The two-step logic was derived from Burke *et al.*'s (2007) work discussing Hybrid recommender systems, particularly the meta-level type whereby one recommendation technique is applied, which is then the input used by the next technique. In this case, rather than using two types of recommendation systems such as collaborative followed by content-based, I have used a basic search followed by content-based filtering. The reason behind this was because in the use case of extracurricular activities, some features people require – for example, it would not be beneficial to recommend an activity that matches someone's requirements perfectly who lives in Cardiff, when the activity is based in Bath. Similarly, it would not be a working recommender system if a 13-year-old was recommended an activity that was for 15–17-year-olds only.

Although the solution may be useful initially, the product built does not have any features that will prompt users to return. Once a user has found all of their closest matches according to different requirements, the product does not offer anything else. Nonetheless, the product was not designed for such a use case since the main aim was to create a system which suggests suitable extracurricular activities.

Conclusion

To summarise, this project has proven successful in completing the desired aims and objectives stated at the beginning. Users are able to receive appropriate suggestions that are best suited to their interests.

Alongside the success of the recommendation system, another achievement has been creating a more well-rounded experience for the user with a registration, login, personalisation, and interactive map features. Being my second ever web application created, I was happy to secure my foundation knowledge with features such as registration and logging in, whilst developing my skillset using APIs.

Current recommender systems depend on a user's information regarding what they already like in an item-to-item approach, however, this approach would not be successful for this use case. I did not want the algorithm to be influenced by what the user may think they like e.g., they may think they like Netball, having never tried Tennis. Because these items are so different, it is likely that user would never get recommended Tennis if we found out they like Netball. So, I went about an alternative route to break down activities according to their features. These features build a vector that describes an activity, and users are asked about their preferences regarding these features instead of at an activity level.

The logic built in this project produces activities that are at the closest distance from the similarity measure which could result in a number from 1 to any number upwards. The reason behind this was because wanting to find the best match is the primary goal, but because the vector information is not weighted to any item, many activities may be the result of one user input. This is positive because it gives users choice but not an overwhelming amount. Problems with this, however, arise when only one recommendation is produced. If only one recommendation is produced users are not given much choice. But the flip side of this is that users are able to enter in new requirements by completing the form again, so they are not stuck with just one forever. Another issue with this may be the consistency, a future edit could be that the code works to display the top five recommendations only, or top ten if that activity dataset was fuller. This could be put in order of similarity, creating a more consistent user experience across the board.

The practical application of the system is widespread; this system could be introduced to schools as part of their 'Life skills' classes or as an activity to do in morning tutor group sessions. Extracurricular activities have been proven time and time again to have tangible benefits in adolescent development ranging from physical exercise to building fundamental social skills. If a system like this was rolled out to the target population, the difficulty of having to research activities is taken away and a simple form is all they need to complete.

Future work

This work is a basic prototype to demonstrate how the logic of recommender systems can be applied into circumstances that have not been explored yet. Due to this nature of unexplored territory both academically and in a business context, there was the cold-start issue for both the activity data within the database and the user data which an algorithm could be based upon. Therefore, there is much future work that could be done to sit on top of this project.

Although there are plenty of opportunities to develop this application, I aim to note upon a few ideas here.

1. Develop the database by expanding to the rest of the UK. As noted, this prototype has a limited number of Cardiff based organisation so this has great room for additional work to expand the locations and activities included.
2. Allow users to pick multiple options from each section rather than being limited to one answer from each question.
3. The method that user requirements are gathered could be done more conversationally. By introducing a chat bot, user requirements could be discovered through an inquisitive question and answer method. This would help improve validity of the algorithm to ensure meaning is understood, whilst also improving the overall user experience.
4. The logic in the algorithm could be developed to produce a fully hybrid solution. Once there are many users on the system, there may be enough data to explore the opportunity to build a collaborative filtering system alongside the current content-based system.
5. Make another user persona from an activity organisation leader, this person would login as an activity organisation rather than a parent or teenager and could post from their perspective. This would help to retain users on the site and build more of a social element to the system.

Reflections

To reflect on my research, learning, and development in this time, I will address items that were a success along with exploring shortcomings, with possible opportunities to improve in the future.

The main success for this project is the skills learnt and developed throughout this project. To begin the project, I had not studied data science nor had any experience with the Pandas module. Equally I had not learnt to code before October last year, so to develop these skills has been a huge success for me personally, but also to complete the project to this standard. Understanding the value of particular concepts such as content and collaborative filtering was extremely useful to begin this project – plus background research into machine learning concepts. These underpinned my approach and meant I did not feel overwhelmed or as if I was going into a project completely blind. Spending the time researching these items meant that I could learn and apply skills in this area to the project. The effect of learning these concepts led to a successful implementation that meets the project aims and objectives. If I was to do this project again, I would spend even greater time on background research as I cannot stress the importance of gaining a full understanding before starting the project. Although I took a good amount of time to research, throughout the algorithm design phase, I did find myself having to return to do more research.

Another successful aspect of this project was the project management tools I used to ensure a timely build and submission. For the overall project I used a Gantt chart to keep track of high-level tasks according to the weeks available to conduct the project. This was a beneficial decision because I felt calm throughout the project being able to meet milestones and know that I was working to the submission deadline at a steady pace. Within the implementation I used Trello Kanban board to keep track of the issues I needed to complete in order to complete the functional and non-functional requirements identified in the design phase. I found that working on the recommender logic and being able to iterate over it throughout the project a good decision since this was the main item of the project, and also led to bugs being caught and resolved before completion. Working in an agile methodology was positive because as van Solingen (2020, p. 1.4) states ‘agility suits situations in which there is a lot of uncertainty’; this project had much uncertainty about the direction of the algorithm logic and also due to a delay in ethical approval, had to return to sections as a

result of the user survey. This is the first personal project that I have completed under the agile methodology; after using it I have experienced the benefits it offers to pick up tasks and return to make edits fluidly. In the future, these ways of working will be continued in further projects.

The last success I wish to note upon here is the contribution that this project has to the field of knowledge regarding recommendation systems. Pinela (2017) identifies that one clear challenge of a content-based recommendation system is its tendency to over-specialise. In other words, the system will recommend items similar to what has already been consumed or liked by the user, bringing back the notion of the “filter bubble” mentioned during the design phase. This algorithm design overcomes this problem in two ways. Firstly, a user can repeat their input of requirements multiple times and so the user can play about with different options rather than being siloed into one option forever. Secondly, through using features of activities, rather than a user stating what activities they either already like or think they like, a user may not be recommended with their “perfect” recommendation. By this I mean, they may be recommended with an activity that does not fit all of their “wants” and can explore suggestions that they may have not considered if a “perfect” match had occurred.

Although I felt there were successes within this project, I have identified clear room for improvement and ideas if I were to do the project again.

One key aspect I feel would have been better if I was to do this project again would be to develop it mobile-first. Because mobile design was an afterthought, I came across problems in making some of the pages responsive. For example, the registration page did not work as I wanted it to, and the form order also skewed.

I think that the logic of the recommender system would have been developed more smoothly if I had completed the activity diagram wholly at first, rather than going straight into the code. If I had created the activity diagram to understand all of the steps I needed to do before touching code, I think the development would have run much smoother. Instead, I began coding much too early, then would hit problems and try to work out what I needed to do next.

Finally, although this project was my second-ever web application that I have developed, if I were to complete this again, I would try to produce a better front-end with a framework such as React. The reason I did not do this here was because of time constraints and wanted to focus on completing the project with a working backend that would produce recommendations to users. However, with the project being aimed at teenagers I think this would be a great addition to the overall system.

Overall, this project has been successful to create a working prototype of a system that produces suitable extracurricular suggestions for teenagers. This project shows the ability of manipulating the logic of recommendation systems that is so commonly used for movies, books, and ecommerce and apply it to a whole new area. I am highly satisfied with the outcome of the project – it has heightened my personal growth in both managing a project and learning new concepts and skills. These skills will carry through with me into my technical career and give me the confidence that I am competent in the technical field.

References

Allthetests.com. 2021. What Activity Should You Join. Available at:

<https://www.allthetests.com/personality-tests/hobbies/quiz30/1370304656/what-activity-should-you-join>

Amazon. 2021. Available at: <https://www.amazon.co.uk>

British Heart Foundation. 2021. Which Fitness Activity Suits You? Available at:

<https://www.bhf.org.uk/information-support/heart-matters-magazine/activity/which-fitness-activity-suits-you>

Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Model User-Adap Inter* 12, pp. 331-370. <https://doi.org/10.1023/A:1021240730564>

Burke, R., et al. 2007. Hybrid Web Recommender Systems. *The Adaptive Web*, LNCS 4321, pp. 377-408.

Burke, R., et al. 2011. Recommender Systems: An Overview.

Buzzfeed. 2021. What Activity Should You Try Based on This Quiz? Available at:

https://www.buzzfeed.com/elzbeth_a/answer-some-questions-and-we'll-give-you-something-dz05mfs4g9

Donnelly, M. et al. 2019. An Unequal Playing Field: Extra-Curricular Activities, Soft Skills and Social Mobility. Social Mobility Commission.

Driessens, C. M. 2015. Extracurricular activity participation moderates impact of family and school factors on adolescents' disruptive behavioural problems. *BMC Public Health*, 15(1110). DOI: 10.1186/s12889-015-2464-0

Fredricks, J., and Eccles, J. 2006. Extracurricular Involvement and Adolescent Adjustment: Impact of Duration, Number of Activities, and Breadth of Participation. *Applied Developmental Science*, 10(3), pp. 132-146.

Hasabe, D., et al. 2015. Web Portal for Providing Various Services. *International Journal of Innovative Research in Advanced Engineering*, 3(2).

- Kupisk, D. 2017. Acing Afterschool: Making extracurricular activities work for your teen. Available at: <https://parenthetical.wisc.edu/2017/01/23/acing-afterschool-making-extracurricular-activities-work-for-your-teen/> Accessed: 20th July 2021.
- Mahoney, J. 2000. School Extracurricular Activity Participation as a Moderator in the Development of Antisocial Patterns. *Child Development*, 71(2), p. 502.
- Netflix. 2021. Available at: <https://www.netflix.com>
- Pariser, E. 2011. *The Filter Bubble: What the Internet Is Hiding From you*. London: Viking/Penguin Press.
- Pinela, C. 2017. *Content-Based Recommender Systems*. Available at: <https://medium.com/@cfpinela/content-based-recommender-systems-a68c2aee2235>
- Polamuri, S. 2015. Five most popular similarity measures implementation in Python. Available at: <https://dataaspirant.com/five-most-popular-similarity-measures-implementation-in-python/>
- Ricci, F., et al. 2011. Recommender Systems Handbook. 1. DOI 10.1007/978-0-387-85820-3_1
- Schafer, J. B., et al. 2007 Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds) *The Adaptive Web*. Lecture Notes in Computer Science, 4321. https://doi.org/10.1007/978-3-540-72079-9_9
- Shimodaira, H. 2015. Similarity and recommender systems. *Learning and Data Note 2. Informatics 2B*.
- Stackoverflow. 2011. *Haversine Formula in Python*. Available at: <https://stackoverflow.com/questions/4913349/haversine-formula-in-python-bearing-and-distance-between-two-gps-points>
- Subramanian, D. 2020. *Building a Content-Based Book Recommendation Engine*. Available at: <https://towardsdatascience.com/building-a-content-based-book-recommendation-engine-9fd4d57a4da>
- Terzian, M. et al. 2009. Why teens are not involved in out-of-school time programs: the youth perspective.

Van Soligen, R. 2020. Agile. Dpunkt.

Images included on the website are provided by Unsplash. Unsplash is a source of free images to be used by anyone. Available at: <https://unsplash.com/>

Appendix

Hosted website: <http://funtime-funtime.apps.cs.cf.ac.uk/>

N.B. This website is hosted on OpenShift and so requires a working VPN connection with OpenShift or to be connected to eduroam internet.

Extracurricular activity survey

FunTime – a web portal assisting the teenage population to find the right extra-curricular activities.

Information Sheet version 1

You are being invited to take part in a research project. Before you decide whether or not to take part, it is important for you to understand why the research is being undertaken and what it will involve. Please take time to read the following information carefully and discuss it with others, if you wish.

Thank you for reading this.

1. What is the purpose of this research project?

This is a student project which aims to create an assistive web portal for teenagers to use, providing them recommendations of extra-curricular activities that are nearby and suited to their interests.

2. Why have I been invited to take part?

You have been invited because you are or have been parent to a teenager.

3. Do I have to take part?

No, your participation in this research project is entirely voluntary and it is up to you to decide whether or not to take part. If you decide to take part, the following information will discuss the research project with you and ask you to sign a consent form. If you decide not to take part, you do not have to explain your reasons and it will not affect your legal rights. If you are a Cardiff University student, involvement in this research project will have no effect on your education or progression through a degree course.

You are free to withdraw your consent to participate in the research project at any time, without giving a reason, even after signing the consent form.

4. What will taking part involve?

You will be asked to complete a short online survey once. You can do this in your own time, but it will need to be completed in one sitting.

5. Will I be paid for taking part?

No.

6. What are the possible benefits of taking part?

There will be no direct advantages or benefits to you from taking part, but your contribution will help us understand what requirements parents may have for a tool of this nature.

7. What are the possible risks of taking part?

There should not be any risks of taking part, however, all information is anonymous to mitigate any potential risks.

8. Will my taking part in this research project be kept confidential?

All information collected from (or about) you during the research project will be kept confidential and any personal information you provide will be managed in accordance with data protection legislation. Please see 'What will happen to my Personal Data?' (below) for further information.

9. What will happen to my Personal Data?

This research will not collect Personal data such as an individual's name, address, email address or date of birth.

Cardiff University is the Data Controller and is committed to respecting and protecting your personal data in accordance with your expectations and Data Protection legislation.

Further information about Data Protection, including:

- your rights
- the legal basis under which Cardiff University processes your personal data for research
- Cardiff University's Data Protection Policy
- how to contact the Cardiff University Data Protection Officer
- how to contact the Information Commissioner's Office

may be found at <https://www.cardiff.ac.uk/public-information/policies-and-procedures/data-protection>

10. What happens to the data at the end of the research project?

After the end of the research project, the data will no longer be stored.

11. What will happen to the results of the research project?

It is the intention to publish the results of this research project within an MSc project. Participants will not be identified in any report, publication, or presentation.

12. What if there is a problem?

If you wish to complain or have grounds for concerns about any aspect of the manner in which you have been approached or treated during the course of this research, please contact Sara Disney or Daniela Tsaneva. If your complaint is not managed to your satisfaction, please contact Shancang Li lis117@cardiff.ac.uk.

If you are harmed by taking part in this research project, there are no special compensation arrangements. If you are harmed due to someone's negligence, you may have grounds for legal action, but you may have to pay for it.

13. Who is organising and funding this research project?

The research is organised by Sara Disney within the school of Computer Science and Informatics in Cardiff University. The research is currently not funded.

14. Who has reviewed this research project?

This research project has been reviewed and given a favourable opinion by the COMSC research ethics committee.

15. Further information and contact details

Should you have any questions relating to this research project, you may contact us during normal working hours:

Sara Disney, +44881604851, disneys1@cardiff.ac.uk

Thank you for considering to take part in this research project. If you decide to participate, you will be given a copy of the Participant Information Sheet and a signed consent form to keep for your records.

12/08/21

***Required**

1. I confirm that I have read the information sheet dated 12/08/21 version 1 for the above research project. *

Mark only one oval.

☐ Yes

☐ No

2. I confirm that I have understood the information sheet dated 12/08/21 version 1 for the above research project and that I have had the opportunity to ask questions and that these have been answered satisfactorily. *

Mark only one oval.

☐ Yes

☐ No

3. I understand that my participation is voluntary and I am free to withdraw at any time without giving a reason and without any adverse consequences (e.g. to medical care or legal rights, if relevant). I understand that if I withdraw, information about me that has already been obtained may be kept by Cardiff University. *

Mark only one oval.

☐ Yes

☐ No

4. I understand that data collected during the research project may be looked at by individuals from Cardiff University or from regulatory authorities, where it is relevant to my taking part in the research project. I give permission for these individuals to have access to my data. *

Mark only one oval.

☐ Yes

☐ No

5. I understand who will have access to personal information provided, how the data will be stored and what will happen to the data at the end of the research project. *

Mark only one oval.

☐ Yes

☐ No

6. I understand that after the research project, anonymised data may be made publicly available via a data repository and may be used for purpose not related to this research project. I understand that it will not be possible to identify me from this data that is seen and used by other researchers, for ethically approved research projects, on the understanding that confidentiality will be maintained. *

Mark only one oval.

☐ Yes

☐ No

7. I understand that anonymised excerpts and/or verbatim quotes from my questionnaire may be used as part of the research publication. *

Mark only one oval.

☐ Yes

☐ No

8. I understand how the findings and results of the research project will be written up and published. *

Mark only one oval.

☐ Yes

☐ No

9. I agree to take part in this research project. *

Mark only one oval.

☐ Yes

☐ No

Survey

10. Please confirm that you have read the participation information sheet and signed the appropriate consent form (by answering the questions in part 1). *

Mark only one oval.

☐ Yes

☐ No

11. Are you a current parent/guardian or previous parent/guardian to a teenager (aged 13-18)? If no, please end the survey. *

Mark only one oval.

☐ Yes

☐ No

- 2 Did/does your teen take part in extracurricular activities which they have been part of for a minimum of one year?

Mark only one oval.

☐ Yes

☐ No

13. Who is more likely to choose the extra-curricular activities for your teen?

Mark only one oval.

- ☐ You
- ☐ Them (your teen)
- ☐ Equal decision between you and your teen

14. Select the top 3 features you consider most important when choosing an extracurricular activity

Tick all that apply.

- ☐ Where it is in comparison to where you live
- ☐ When it is
- ☐ How much it is
- ☐ If it is age appropriate
- ☐ Skills it develops e.g. social skills
- ☐ If it is indoors or outdoors
- ☐ What type of activity it is e.g. sport, learning-focused, etc.

- 5 To what extent do you agree with this statement: 'Distance from home impacts decisions regarding which extracurricular activities my children can do.'

Mark only one oval.

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

16. What information would you like to know about an activity before your teen joins?

Tick all that apply.

- ☐ Cost of club
- ☐ Location
- ☐ Contact details
- ☐ People who are currently part of the club
- ☐ Website
- ☐ When it is

17. Which statement do you agree with more?

Mark only one oval.

- ☐ "There are too many options for extracurricular activities, I do not know which one to pick"
- ☐ "There are not enough options for extracurricular activities, I can't / couldn't find something my teen would like"
- ☐ "There are a good amount of options for extracurricular activities, and my teen does an appropriate amount for them"

- 8 Are there any features of websites that you particular like? [Please tick all that apply].

Tick all that apply.

- ☐ Images and graphics
- ☐ Chat function
- ☐ Favouriting/saving
- ☐ Personalisation
- ☐ Interactive map of locations
- ☐ Sharing function (e.g. to social media)

19. What type of device do you primarily visit websites on?

Mark only one oval.

☐ Desktop / Laptop

☐ Tablet

☐ Mobile phone

☐ Other