|epcc|

Software Development Coursework - Part 1

Group 5

April 19, 2024

Contents

1 Introduction

For this coursework we have chosen to create a research tool for prospective undergraduate and postgraduate students. Our user base will therefore mainly be young adults or teenagers searching for a university course. This report will outline the requirements, the design, the planning, the UI design and the potential risks.

2 Requirements

2.1 Requirements capture

We first started with viewing other university research websites in order to have a brief overview of the features they have. Then we presented the result of a questionnaire that aims to gain the potential users' opinions on what a university/courses research tool should introduce.

2.1.1 Other university research tool examples

Knowing how other universities/courses researching websites perform can provide a better understanding about the available or missing features, and highlight any areas for improvement. We have checked many websites including: FindAMasters (FAM) [1], ShanghaiRanking (SHR) [2], TopUniversities (TUS) [3] and TimesHigherEducation (THE) [4]. The features of these websites are recorded in Table 1.

No.	Feature	THE	FAM	SHR	TUS
1	Filter by location	Y	Y	Y	Y
2	Filter by subject	Y	Y	Y	Y
3	Order results by criteria	Y	N	N	N
4	User account	Y	Y	N	Y
5	Results presented in table	Y	N	Y	N
6	Search for particular university	Y	N	Y	Y
7	Filter by course title	N	Y	N	Y
8	Search by keyword	N	Y	N	N
9	Browse courses	N	Y	N	Y
10	Add course/university to favourites	N	Y	N	Y
11	Compare universities	N	N	N	Y

Table 1: The various features found on different university research tools.

Filtering by location and subject are the most common features. This is likely because this is most important to users who are only interested in a particular subject and location. FindAMasters and TopUniversities are more focused on helping users find a particular course at

a university and therefore have more features relating to this, such as a compare university feature, the ability to favourite a course, and the ability to browse through courses. Compared with TimesHigherEducation and the ShanghaiRanking these websites do not use table to display the results, have less data displayed about universities, and do not allow sorting of the data. In general the TimesHigherEducation and ShanghaiRanking websites have more data sorting and filtering functions, whereas FindAMasters and TopUniversities are more visually friendly and have more user friendly functionality.

2.1.2 Questionnaire

In order to create a useful university researching tool, a survey was conducted among 23 students from different education levels. Ideally more students should be asked in order to get more accurate results, however time pressure prevented this.

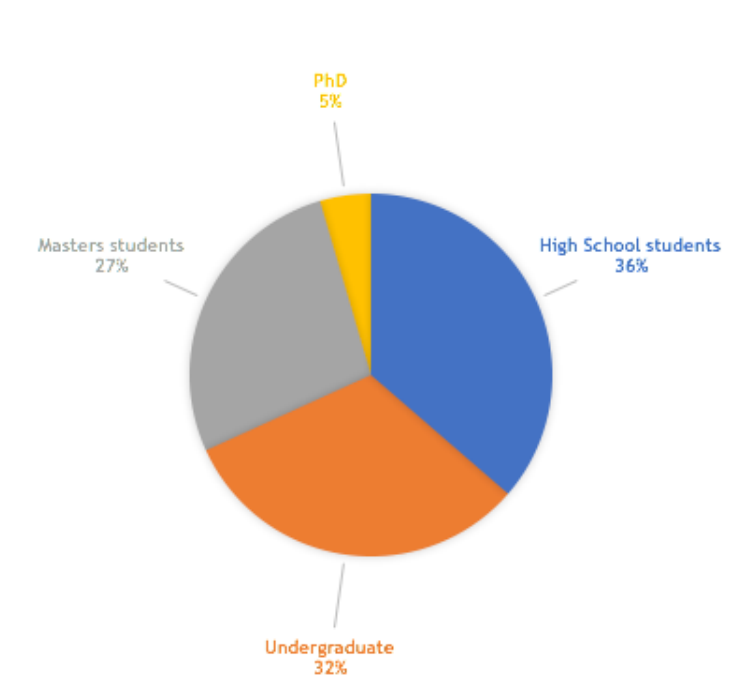


Figure 1: The education level of survey participants.

As seen in Figure 1, 36.36% of them were high school students, 31.82% were undergraduate students, 27.27% were masters students and 4.55% were PhD students. This represents a good sample of our user base, who will likely be high school students looking for their first undergraduate degree, or undergraduates who might be looking into postgraduate study.

The main purpose of the questionnaire is to understand what potential users are looking for when using these kinds of tools. The survey consists of four simple questions to get the par-

ticipants feedback on whether they have used a universities/courses researching tool, and their opinion of the most important criteria to be considered when choosing a particular university.

Participants were asked whether they have used a university research tool before in order to establish their familiarity with such tools. Our results show that 63.64% of the students asked have used some sort of tool for researching universities or courses. However, 36.36% of them have never used any tool for researching. This could be due to many reasons, including the lack of such tools, the potential students might not be aware of the existing tools or for some countries, the ministry of higher education takes control of this procedure.

Another question has been asked to find out the preferable way to use a university/course researching tool this is shown in Figure 2

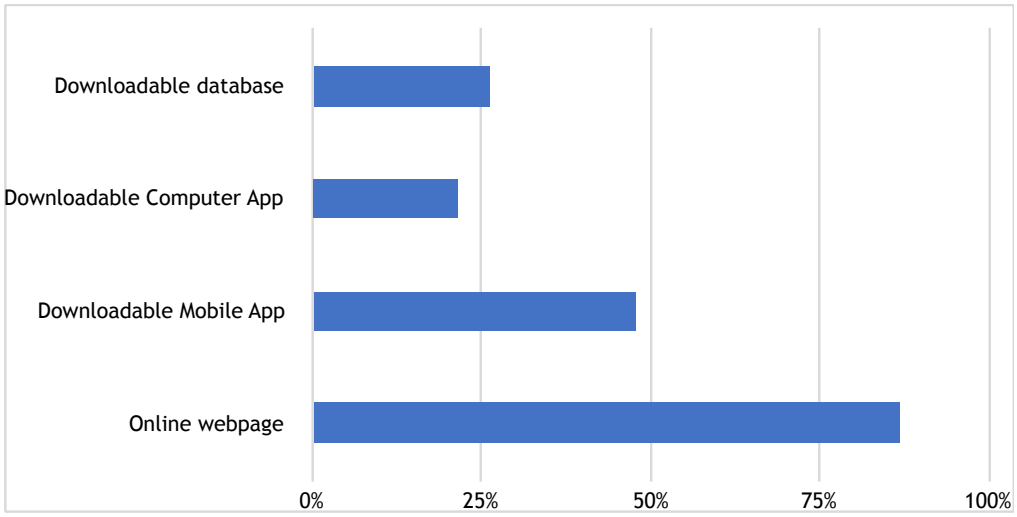


Figure 2: The participants preferences for research tool format.

Most of the prospective students favour to use an online webpage rather than using other methods. The reason for this may be that users are unlikely to use this kind of tool on a daily basis and users would prefer to not have to download and install an application for such irregular use.

The second highest percentage was using a downloadable mobile application. One of the reasons might be that some users would be interested in keeping the information they need accessible to them any time. This brought us to the idea of having an option to the users to create an account in order to save the data they need, their search history and favourites.

In order to develop the universities/courses tool, the questionnaire also sought to understand the most important criteria in the users decision when choosing a particular university. Participants were asked to rank criteria on a scale of 1 to 5 (5 being very important and 1 being not important). This is used to determine which particular criteria should be displayed in the product. The weighted average of the participants preferences is shown in Figure 3.

The majority of those asked selected quality of life in location as being the most important factor. This information is currently not available in the other research tools. Based on this

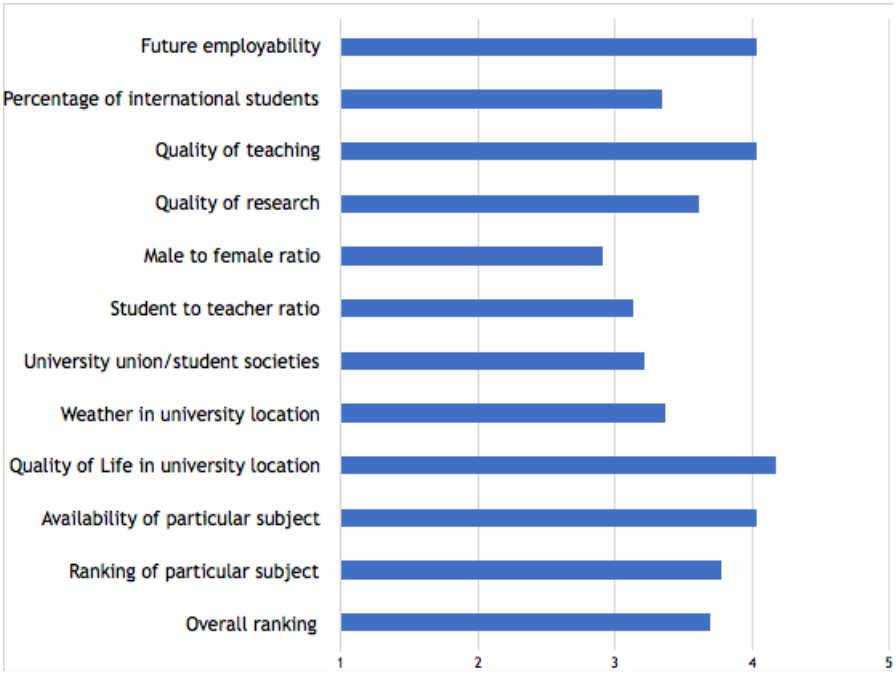


Figure 3: The weighted average of factors survey participants consider important when looking for a prospective university.

criteria relating to quality of life at a particular university should be included in our tool, and could set our tool apart from existing tools. We could also choose to show individual criteria factoring quality of life, such as cost of living and accommodation.

Other criteria were recognised as being less important by those asked. The male to female ratio and the ratio of teachers to students were found to be the least important criteria. This is in contrast with the THE website which includes these criteria in its university data. Based on this, these criteria could be omitted from our product in order to simplify it and not bombard the user with data that they do not consider important.

2.1.3 Interviewing users

If we had more time we would have considered interviewing individuals about their usage of research tools. This could include watching a user use some of the current example of university research tools and asking them direct questions and feedback about their experience. This could be used to gauge any issues and areas for improvement in the current examples and find requirements which may have been missed with other capture methods.

2.2 Functional requirements

Based on the results of the questionnaire and our investigation of current university research tools we have identified the following functional requirements. The priority of these is given by either MUST, SHOULD and COULD. MUST requirements are essential for the correct functioning of the product and will have to implemented first. SHOULD requirements will be implemented after these and are required in order to have a product which functions well and is easy to use. COULD requirements add extra features to the product which are not necessary for its basic usage, but may improve the user experience.

2.2.1 The product will be a website - MUST

Based on the questionnaire results that most people would prefer to use a website based university research tool and that almost all current tools are websites our product will be a website.

2.2.2 The product will use a database to store course/university information - MUST

This requirement is essential to store the wide variety of university and course information. A separate database for course ranking and overall ranking will be used, based on the available data set.

2.2.3 The database will draw data from multiple data sets - MUST

Some users might prefer to access data from different data sources such as the Times Higher Education guide and the Shanghai ranking. This will be able to be done easily in the product without the user having to understand the underlying data structures. Other data relating to quality of life from various sources will also be integrated into the database.

2.2.4 Users can log in and save previous searches and favourites - COULD

Other university research websites such as FindAMasters have the functionality for users to favourite courses and universities in order to store them. Based on this we plan to implement a user log in feature so that users can store past searches and favourites.

2.2.5 Users will be able to mix and match search criteria - MUST

All current examples of university research tools have the ability to search data based on the subject ranking and the location of the university. Without this there is no way for the user to narrow down there results.

2.2.6 Users will be able to filter the results by location and subject - MUST

All current examples of university research tools have the ability to filter the results by location and subject. These criteria are the most important for finding a particular university.

2.2.7 Users will be able customise the search criteria for an advanced search - MUST

This feature allows users to pick certain criteria and weight them according to how important they believe it is. Results will be ordered based on these choices. Based on the fact that users have different preferences when it comes to deciding how to choose a university this allows users to only select criteria that are relevant to them and avoids overloading the user with data.

2.2.8 Users will be able to order universities based on criteria - SHOULD

Ordering the search results by criteria allows users to select which criteria are important to them and then order the results with the highest ranking results first. This is found in nearly all current examples of the university ranking tools.

2.2.9 Users will be able to see what courses are on offer at each university - SHOULD

Based on the questionnaire results users list the availability of courses at a particular university as an important factor in deciding which university to choose. Therefore the ability to see this in the search results should be a requirement.

2.2.10 Users will be able to compare universities - COULD

Comparing universities offers users the ability to have a summary of criteria for two or more universities. This feature is used in a few current examples to give more specific data about particular universities in order to compare them.

2.2.11 Users will have access to the following information about a particular subject at a university - MUST

- Future employability
- Quality of life in university location
- Cost of living
- Cost of accommodation
- Times Higher Education ranking

- Shanghai ranking
- Quality of teaching
- Quality of research
- Courses on offer (filterable by postgraduate/undergraduate)
- Number of students per staff
- Percentage of international students

2.2.12 Users will be able to provide feedback about the tool - COULD

Feedback is important to aid further development of the tool. For the final release this would be useful to improve further versions of the tool. However for the prototype this is not a necessary requirement.

2.3 Non-functional requirements

In this section we shell discuss how our program should work. As with the functional requirements we categorised the non-functional requirements into either MUST, SHOULD, or COULD based on their priority.

2.3.1 Usability-Easy to use and navigate - MUST

When it comes to creating and designing a website, it is essential to take into consideration that users would be able to use the website easily without the need to understand the data structures or the data sets. Therefore, the ability to navigate the website easily is a key point.

2.3.2 Readability - MUST

The font type and size used in the website must be clear and readable so that users could easily scan the page when looking for specific information. The background colour should also be compatible and consistence with the font type, size and colour.

2.3.3 Responsiveness - MUST

It is very common that users use smart phones for searching websites rather than using a PC. Thereby, the website is required to be responsively designed. i.e. Adapting the website to the size of the viewing window. This can be done by changing the layout and format in the CSS (style sheet) without the need to change the data structures. In addition, the site should also be capable of correctly displaying its contents regardless of the user's device.

2.3.4 Multiple languages - COULD

As our website targets worldwide students, multilingual website could assist in attracting users. We decided to start with English as a default language and then depending on our time schedule we can add a feature that enables the users to choose the language they prefer. The chosen languages would be the top 5 most spoken languages in the world.

2.3.5 Performance - MUST

Slow websites can lead to less visitors. Thereby, The response time of the system should be reasonable as the website has to load within three seconds. Periods of stress should be also taken into account.

We could use website's grader tools such as website.grader, Google's PageSpeed Insights and GTMetrix to get some feedback on optimising the website performance in many aspects. This includes the size of the page, page requests and page speed. See section 6.5 for more information.

3 Prototype Design

In this section, we present our technical choices for the prototype. A series of available techniques for web development will be discussed and we will give the reasons for our coding choices.

3.1 Design features

The prototype will be a website which can be accessed through the command line. The prototype will satisfy the fundamental functional requirements below:

- Have databases to store university/course information and present to users
- Allow users to select and assign weights to various criteria
- Allow users to order universities based on the selected criteria

There are two other extended functionalities that we wish to realise in the prototype:

- Compare Universities
- Filter search results based on country and program level

The prototype is not the final version of our design. For the final release it is possible to rewrite the code as we extend the website functionality. The purpose of the prototype development is to demonstrate a simple working version of our design.

3.2 Design components

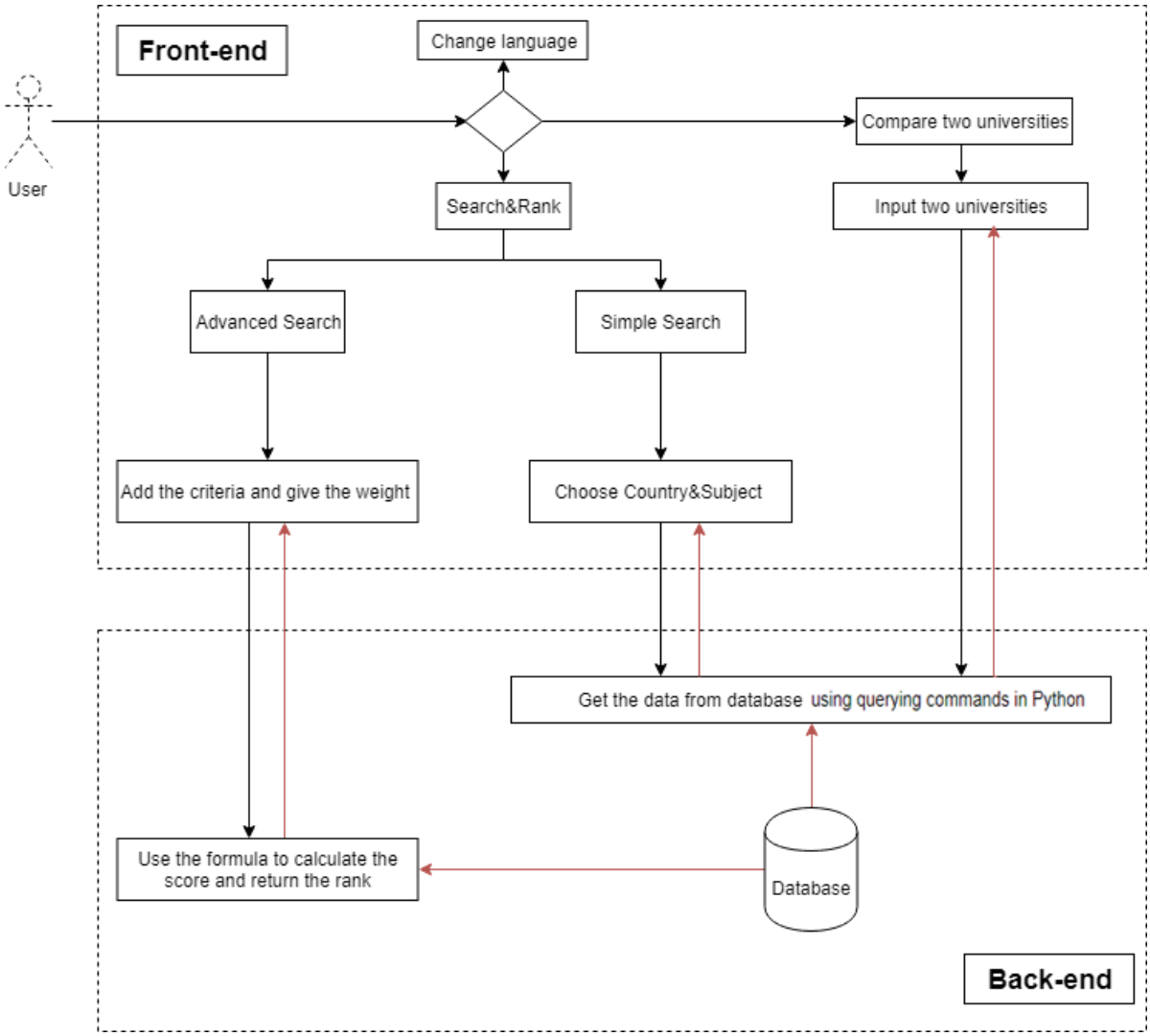


Figure 4: Design components and interconnections for the prototype

3.2.1 Data sources

The data required for the prototype could be largely divide into two parts: one includes statistics about quality of teaching, research and university life, which would be drawn from the current university ranking systems like THE[4] and Shanghai ranking[2], and the other about factors relating to the quality of life in the university cities, which can be found in Teleport [5]. Another part includes university course information and university location, which can be found on the university searching website TopUniversities[3].

Since all our data is from open online sources it is reasonable to acquire data by web scraping.

Python and SQL are both frequently used for data scraping. In this project we will write a Python script for the raw data acquisition as members of the team are more familiar with Python.

3.2.2 Data restructuring

The input of data files will be hard coded into the prototype. The datasets are created by manually inputting data derived from the sources mentioned. This will allow the prototype to read data into one particular format designed exclusively for the prototype and then be ready for the created databases. For the final release, developers are required to create a data querying and sorting module in Python, which is capable of drawing data from designated open sources and automatically producing a input file in the required data format. Since the teams members are coding mostly in Python, data are stored in JSON-like formats and can be loaded at a relatively high speed.

3.2.3 Data Storage

Although datasets for the prototype are not of considerable size, we plan to create several different databases to separate the data for different purposes. For example, statistics for ranking criteria will be stored in one database whereas university location and course information will be stored in another database. The first database will be loaded when webpage calculates the university ranking (following the user-defined combination of criteria) whereas the later database will be loaded when users wish to filter the calculated results. This would allow us to manage the data corresponding to each function with less effort.

Databases of the prototype are created in MongoDB using Navicat. MongoDB is the most frequently used NoSQL database program that allows us to export data in JSON-like files and offers more flexibility and scalability for databases. Navicat is a graphical database management system, which helps us to manage data through the simple GUI without learning any commands to access MongoDB.

3.2.4 Data Processing

The prototype allows users to select university ranking criteria from a series of criteria provided. Users are required to assign different weights (c_{weight}) from 1 to 10 for the criteria they have selected. After users have defined the combination of criteria, the score (C_{score}) corresponding to those criteria will be requested and drawn from databases. Then the overall points (P) of universities will be calculated at the back-end using equation (1), summing over all selected criteria, and the university ranking will be sorted in descending order. Once the results have been computed at the back-end, the website server would return the results to users as shown in Figure 4. All data processing will be handle by the back-end modules coded in Python.

$$P = \Sigma(c_{weight} \times c_{score}) \tag{1}$$

3.2.5 Website Implementation

HTML, CSS and javascript will be used in the front end development. HTML is used to define the standard structure of our webpages while CSS is used to design the layouts for webpages. We also use javascipt to enable interactive actions on webpages. Both front-end and back-end modules are held in Django, the Python-based web framework.

PHP is another popular choice for web application development. Since our team members do not have extensive experiences in PHP, we wish to reduce the usage of PHP as much as possible in order to deliver the qualified prototype with restricted time. It also should be noticed that PHP follows the server side logic, which would offer less flexibility in terms of the UI design. Therefore, we decided not to use PHP in the project.

3.2.6 Interconnections

Interconnection between the front-end and back-end can be established automatically in Django. A block of Python codes will be created to allow the back end to fetch data from databases. We should be able to reduce the overall amount of coding loads while using these two tools. Student licenses are available for both these tools.

3.3 Implementation plan

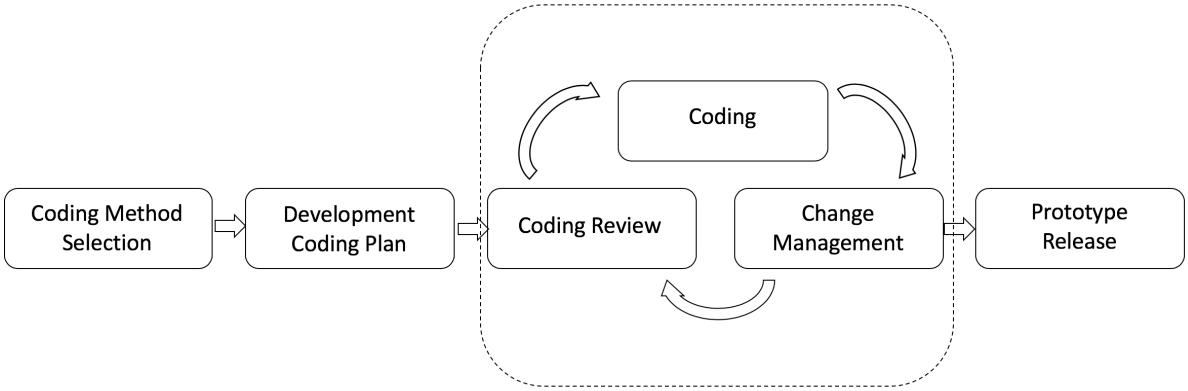


Figure 5: The implementation plan for the prototype

The implementation steps are shown in Figure 5 above. After agreeing on the coding plan, we will develop the prototype in a code-alter-review circle to make sure each block of codes are functioning well.

4 Planning

4.1 Meetings and organisation

Meetings are organised twice a week throughout the project. Before or at the start of a meeting an agenda for the meeting is given. A section of the report or some component of the product design is discussed between all members of the team even though a single member may be responsible for it. Meetings are a chance for others to provide feedback. Communication between team members is done via email, private messaging services (Slack) and in person.

All documents are available to all members of the team. Requirements are risks are stored on the projects shared GitLab page so that they can be accessed and updated throughout the entire project. The report, Gantt chart are produced using ShareLatex and Google Sheets respectively so that all members may edit them and see the changes of others. The prototype will make use of GitHub for version control. This will be linked to the GitLab project.

The schedule and progress is recorded in a Gantt chart which is updated each meeting. Progress are targets are reviewed regularly in order to adjust the schedule to deal with changing circumstances. Risks may be added to the risk assessment throughout the project when they become apparent. Any new risks are assessed at the start of each week.

4.2 Team structure

Our team consists of only four people and therefore roles must be not be too specific. Broadly two of the team are assigned more project/product management and organisational roles, and the other two are assigned more developer roles. For the first part of the assignment each team member has one the following main responsibilities.

- Requirements capture
- Planning and risk management
- Back-end design
- UI design

Meetings are regular such that all team members are involved in all parts of the assignment, providing feedback and suggestions on each others work throughout. This means that all members are informed, accountable and consulted on each area of the project to some extent. This is important as all tasks are related to each other in some way. Furthermore in the second part all members of team will have to work on some part of the development of the prototype and therefore all members should understand the prototype design.

For the development of the prototype all team members will work on the development of the prototype. Those members who were in charge of the back-end and UI design in the first part

will be responsible for making sure that the design is implemented correctly. The other two team members will be responsible for testing the prototype, including usability testing, and ensuring that the development runs to schedule.

For the development of the prototype we propose the four following roles:

- Data capture, cleaning and database creation and formatting (DB)
- Back-end development (BE1)
- Back-end development (BE2)
- Front-end development (FE)

Two back-end developer roles have been made to reflect that there are more tasks relating to this.

4.3 Choice of development model

Choosing and sticking to a development model is important in order to provide structure and organisation to the development process. For this project we have chosen the Design to Schedule model. This is a variation of staged delivery where the code is implemented in stages, with each stage resulting in a working release. In the design to schedule variant the stages are implemented in order of importance up until the deadline. This is ideal for our project as we have a hard deadline. It is also suited to projects where time estimates for tasks may be poor and small projects requiring a low management overhead. This is likely in our case as we are a small team with limited management experience.

4.4 Prototype development tasks

4.4.1 Source requirements

Dependencies: None

Assignment: All

Time effort value: 7 hours

4.4.2 Investigate data sources

Data sources are found as a basis for what can be included into the database.

Dependencies: None

Assignment: All

Time effort value: 2 hours

4.4.3 Design database structure

The database for holding the university data is designed based on the requirements and the available data.

Dependencies: Investigate data sources, source requirements.

Assignment: Main developer

Time effort value: 1 hours

4.4.4 Design search/filter functionality

The main search and filter functionality for the database is designed to meet the requirements.

Dependencies: Design database structure, source requirements.

Assignment: Main developer

Time effort value: 1 hour

4.4.5 Design compare universities functionality

This can be done independently of the database design. This feature is not as high a priority as the database and searching functionality so this should be done after these more important features are designed.

Dependencies: Source requirements.

Assignment: Main developer

Time effort value: 2 hours

4.4.6 Design user page

The user page is designed. This can be done independently of the database design. This feature is not as high a priority as the database and searching functionality so this should be done after these more important features are designed.

Dependencies: Source requirements.

Assignment: Main developer

Time effort value: 2 hours

4.4.7 Design UI

UI can be designed independently of the back-end design so long as the requirements for both are established beforehand.

Dependencies: Source requirements.

Assignment: UI developer

Time effort value: 10 hours

4.4.8 Create database

To create the database the data needs to gathered from the various sources. The data will then be processes and saved in the database.

Dependencies: Design database. Investigate data sources.

Assignment: Database manager

Time effort value: 5 hours

4.4.9 Implement search functionality

The search functionality is the most important feature of the prototype. This must be done as soon as possible, after the database has been created.

Dependencies: Create database.

Assignment: Back-end developer 1

Time effort value: 5 hours

4.4.10 Implement data sorting

Data sorting is not a essential requirement but it is important to implement for a good quality product. The search functionality is more important but these things may be implemented at the same time as we have multiple developers.

Dependencies: Create database.

Assignment: Database manager

Time effort value: 2 hours

4.4.11 Advanced search functionality

The advanced search should be implemented after the the basic search as it uses it as its basis.

Dependencies: Implement search functionality.

Assignment: Back-end developer 1

Time effort value: 5 hours

4.4.12 Implement compare universities functionality

Two or more universities from the database can be compared with each other

Dependencies: Create database.

Assignment: Back-end developer 2

Time effort value: 6 hours

4.4.13 Implement user page

The user page is independent of any of the database functionality. However it should be done after these more important features are fully implemented.

Dependencies: Design user page.

Assignment: Back-end developers

Time effort value: 5 hours

4.4.14 Implement UI

Dependencies: Design UI.

Assignment: UI developer.

Time effort value: 10 hours

4.4.15 Perform usability tests

Dependencies: Create prototype.

Assignment: Product Managers.

Time effort value: 3 hours

4.4.16 Analyse usability tests

Dependencies: Perform usability tests.

Assignment: Product Managers.

Time effort value: 2 hours

4.4.17 Refine prototype

Dependencies: Analyse Usability tests.

Assignment: All.

Time effort value: 4 hours

4.5 Gantt chart

The Gantt chart for the project is shown in the Appendix and can also be found at: `https://tinyurl.com`
It includes the time effort estimate and the actual time effort as well as the schedule for completing each task.

5 Risk Management

The three antagonistic attributes in a software development project are cost, time and quality. Reducing or fixing any one of these will have a negative impact on the other attributes.

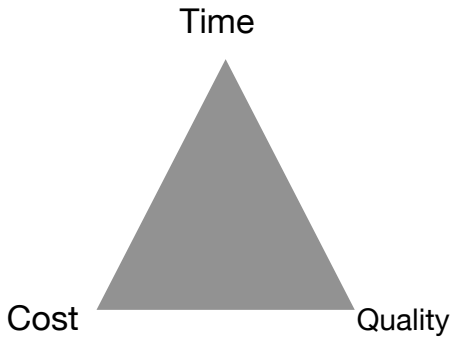


Figure 6: The relationship between the cost, time and quality of a project.

As mentioned in the planning section our project has a fixed deadline and no budget. This eliminates the risks of the prototype not being delivered on time or going over budget. However this increases both the likelihood and potential impact of other risks since there is no flexibility to push back the finish date to accommodate for any issues. As there is no budget for the project there is also no funding to buy extra resources (for example team members or software to increase productivity) if limited time becomes an issue. Therefore quality slippage is likely to result.

Other risks result from the fact that the team is new and relativity inexperienced. This will increase the likelihood of both poor planning and difficulty in the development of the product, as well as team related issues such as lack of communication and cooperation. These risks will impact the overall quality of the product.

5.1 Identified risks

5.1.1 Poor time management/Quality slippage

Time management is important in this project due to the fixed deadline. Poorly managed time can result in parts of the project becoming rushed to meet the deadline. This will reduce the quality. Poor time management is a risk in this project as the team has multiple commitments outside of the project.

5.1.2 Poor planning

Proper planning is essential for good time management of the project. All the necessary tasks in the project must be identified, the time required for a task estimated correctly, and then tasks evenly divided amongst team members in order to ensure that the project runs to schedule and the quality does not slip. Risks must be taken into account during planning process so that extra time may be scheduled to help deal with risks should they occur.

5.1.3 Requirements not correctly identified or prioritised

Identifying and determining the priority of the product requirements decides which design features are implemented and their order of implementation in our design to schedule development model. If this is done incorrectly then the final product will not meet the requirements of the user.

5.1.4 Illness of team members

If team members become ill then there are less people able to work on the project and the time management and quality will suffer. This will impact the productivity of the entire team,

affecting all areas of the project.

5.1.5 Lack of experience

The team has a general lack of experience in planning and task estimation, as well as product design and prototype creation. This means that the scheduling and planning may not be done correctly initially, and be susceptible to change throughout the project. Team members have tasks that they are inexperienced with, such as designing a UI and using unfamiliar tools. This includes using Django, MongoDB and Navicat. Therefore these tasks may take longer than anticipated and the quality may suffer.

5.1.6 Lack of team cooperation

The team must cooperate together so that tasks are completed on time and to ensure that a good level of quality is maintained. As the team is new at working together this may become a problem since roles within the team may take a while to establish.

5.1.7 Lack of team communication

Communication between members of the team is key to ensure that the project is well organised throughout. Lack of communication can lead to tasks being poorly divided between members, resulting in poor management of resources (time and developers).

5.1.8 Difficulty getting data from sources

The prototype requires being able to gather data from a variety of sources and combine them together to form the data set for users to search on. As this data has different formats and sources it may be difficult to find and extract the desired data or combine different data sources. This could affect fulfilling some of the requirements such as searching on various criteria.

5.2 Quantitative risk analysis

A quantitative risk analysis is used to assess the severity of the identified risks. The probability and impact (up to 10) of each risk is determined, and these are multiplied together to give a risk score. Risks are then ordered in terms of this score, with the highest score having the greatest rank. The quantitative risk analysis is shown in Table 2

Based on the risk analysis the risks with the largest score is lack of experience and poor time management. Therefore mitigation strategies should be employed to try and prevent and manage these risks.

No.	Risk description	Probability	Impact	Score	Rank
1	Poor time management/Quality slippage	0.5	8	4	2
2	Poor planning	0.3	8	2.4	3
3	Requirements incorrectly identified or prioritised	0.1	8	0.8	7
4	Illness of team members	0.1	3	0.3	8
5	Lack of experience	0.8	8	6.4	1
6	Lack of team cooperation	0.1	10	1	6
7	Lack of team communication	0.2	8	1.6	5
8	Difficulty getting data from sources	0.3	7	2.1	4

Table 2: The quantitative risk analysis for the project.

5.3 Risk mitigation

5.3.1 Poor time management/Quality slippage

The risk of poor time management can be reduced mainly by proper planning and ensuring each team member follows the schedule. Work should be started and finished on time and team members should keep track of task completion. The team will plan to finish ahead of schedule in case of any unforeseen circumstances. Extra work may have to be done at weekends or evenings in order to to keep to schedule. Tasks can be prioritised in order to fulfil the most important requirements of the product first, so that a working prototype can be delivered in time. Then other less important features can be added if time allows. Any new risks which are likely to affect the project schedule should be documented throughout the project so that they can be mitigated accordingly.

5.3.2 Poor planning

The impact of poor planning can be reduced by reassessing the schedule often and making changes to the schedule or to work assignment when necessary. The plan should be allow some flexibly to accommodate for these changes and be adapted when new risks are identified.

5.3.3 Requirements not correctly identified or prioritised

The requirements capture process is designed so that the correct requirements are identified from the start, by distributing a questionnaire to the user base. The the design features that fulfil the most important requirements are implemented into the prototype first. Usability testing is planned after the UI is developed in order to receive feedback from users so that any changes can be made before the prototype is developed.

5.3.4 Illness of team members

The effects of a team member becoming ill can be mitigated by allowing extra time in the plan in case of such unexpected circumstances. Work between team members should be managed in such a way so that a team member may take over an other's work if necessary. This means that all documents must be available to the whole team and that everyone in the team should be aware of what each team member is working on. For this planning and communication between team members is key.

5.3.5 Lack of experience

The lack of experience of team members means that all tasks are likely to take longer than for a more experienced team. Extra time is scheduled for each task so that team members have time to learn on the job. Team members are encouraged to communicate with each other to share what knowledge they have or acquire during the project.

5.3.6 Lack of team cooperation

Team roles need to be established early on to ensure that tasks are divided equally and that each member understands their job. Regular meetings should be scheduled so that everyone is aware of what others are working on and also so that any issues can be resolved early on. Targets for individuals to complete tasks are made so that team members are likely to keep to the schedule.

5.3.7 Lack of team communication

Good communication is established early on with regular meetings twice a week and daily email/instant messages between members. Subsequent meetings are always scheduled during a meeting in order to maintain regular meetings. Shared pages are set up so that all work relating to the project is accessible to all members of the team.

5.3.8 Difficulty getting data from sources

Sources of data can be identified early on in the project. If this data turns out to not be sufficient then other sources may be found. As the prototype does not have to be fully functioning it will be possible to only use a small sample set of data to demonstrate its usage.

6 UI Design

6.1 Characteristics of a good user interface

6.1.1 Clear Layout

A clear layout is necessary for a website to separate the elements into groups and make the information presented in a structured format. This also provides understanding of the functions with visible hierarchy.

6.1.2 Simple Operations

Some familiar icons should be used on commonly used functions so that users can identify intuitively what would happen when they interact with the website. Meanwhile, several creative elements should be added to attract users' attention on a basis of keeping the familiarity component.

6.1.3 Quick Responses

When users interact with the website they should get a response quickly. For this reason we need to make the interface and the system behind it able to work fast and have good responsiveness. If not, a notification should be presented to user to inform them of what is happening.

6.2 Flow of the website

The user can choose either the Search page or the Compare page.

On the Search page the user can use the Simple Search which would filter the universities by location and programmes. In the advanced search Criteria are chosen and weighted by their importance to the user. This weighting and the data for the criteria are input into the formula which is used to calculate a score for each university. The ranking table will then order the universities according to this score.

For the Compare page, the user can choose two universities, and select at most five different criteria and to compare. The results are presented as a bar chart to the user.

The detailed control flow chart is shown in Figure 7.

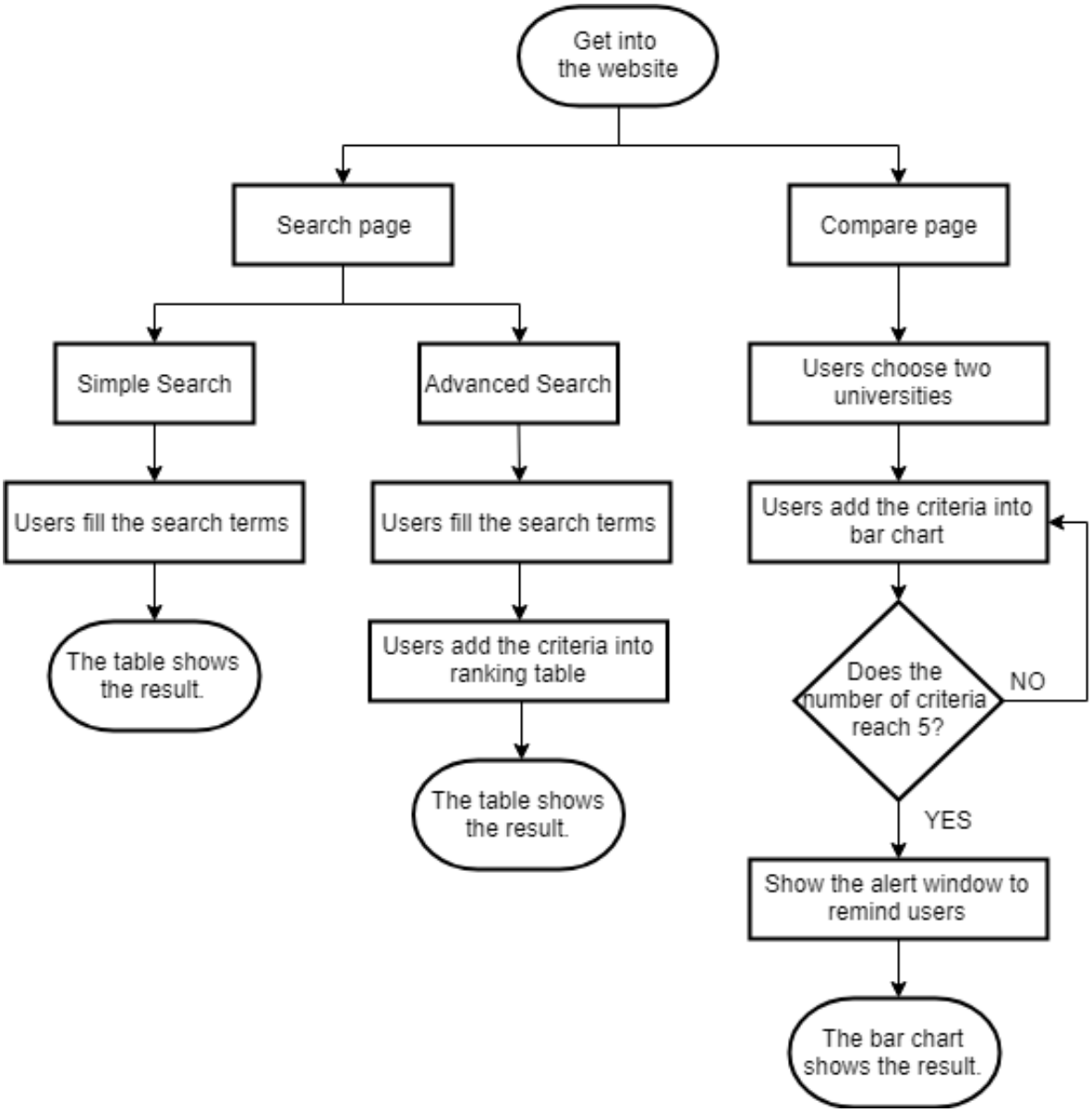


Figure 7: The control flow chart for the user interface

6.3 Prototype of the user interface

In this section, we will post the prototype image, tag the elements in each part of the Web page by the square and number and describe the functionality of it.

6.3.1 the Prototype for Header

Figure 8: The header for the website

Element 1: Language Drop-box
We have designed a language drop-box to support multiple languages so that users can switch the texts in the website into another language. This makes the website available to people from different countries.

Element 2: Log-in Link
When the user clicks on this link, a small log-in form would pop up where users can fill in their username and password. If these are correct the personal information page is loaded.

Element 3: Sign-up Link
When the user clicks on this link, a small sign-up form would pop up. The personal information page is loaded after filling in this form.

Element 4: Search Button
When the user clicks on this button the search page is loaded.

Element 5: Compare Button
When the user clicks on this button, the website will load the Compare page.

6.3.2 the Prototype for Footer

Figure 9: The footer for the website

Element 1: Privacy Policy Link
When the user clicks on this button the website will load the Privacy Policy page.

Element 2: Contact Link
When the user clicks on this link the email application on the user's computer would load with the receiver address is filled with creators' email address automatically.

Element 3: Help Link
When the user clicks on this button the Help page is loaded to give some common Q&As.

Element 4: Copyright Statement
This text lets the public know that our work is protected by copyright law and is not to be copied.

6.3.3 Prototype for Simple Search

For Search page, two kinds of search methods are offered, Simple Search (as default) and Advanced Searched.

Figure 10: The prototype for the Simple Search

Element 1: Container Control
The container control can capture the tab key press and focus on the first element in the container whose tab is clicked. Simple Search is set as default.

Element 2: Degree Radio Button
The user can choose postgraduate or undergraduate degree. This effects the course options column in the table.

Element 3: Country Drop-box
The user can filter universities by the country where they are located to narrow the scope.

Element 4: Subject Drop-box
The user can filter universities by the subject what they provide to narrow the scope.

Element 5: Simple Search Button
When the user clicks on this button the result from the query condition is shown in ranking table below.

Element 6: Ranking Table
The data in the table is received according to the query condition (based on the location and subject) and acquired from the database. The user can use the small triangle button next to the criteria in the table to arrange data in an ascending or descending order.

6.3.4 Prototype image for Advanced Search

Figure 11: The prototype for the Advanced Search

Element 1: Container Control
When the user clicks on the tab of Advanced Search, the container would change so that Simple Search Panel is maintained but the Criteria Panel is added. This panel lets the user choose what aspect they hope to know about the universities.

Element 2: Simple Search Panel
This panel is the same as the one in Simple Search container.

Element 3: Criteria Drop-box

The user can choose one criterion from the drop-box, and then drag the process bar to choose the weight of importance for this criterion.

Element 4: Weight Bar
We put the weight of each criterion from 0 to 10 rather than 0 to 1 to avoid the user confused about whether the weight of all criteria need to add up to 1. In fact the user is free to choose the weight and have no limitation.

Element 5: Add Button
When the user click on this button, the criterion and its weight would be added below. The user is free to add another criterion and weigh.

Element 6: Advanced Search Button
When the user click on this button, the website would produce the result from the query condition and show it in the ranking table below.

Element 7: Ranking Table
The data in the table is received according to the query condition which the user is decided above and acquired from the database. The user can use the small triangle button next to the criteria in the table to arrange data in an ascending order or in a descending order.

6.3.5 Prototype for Compare Page

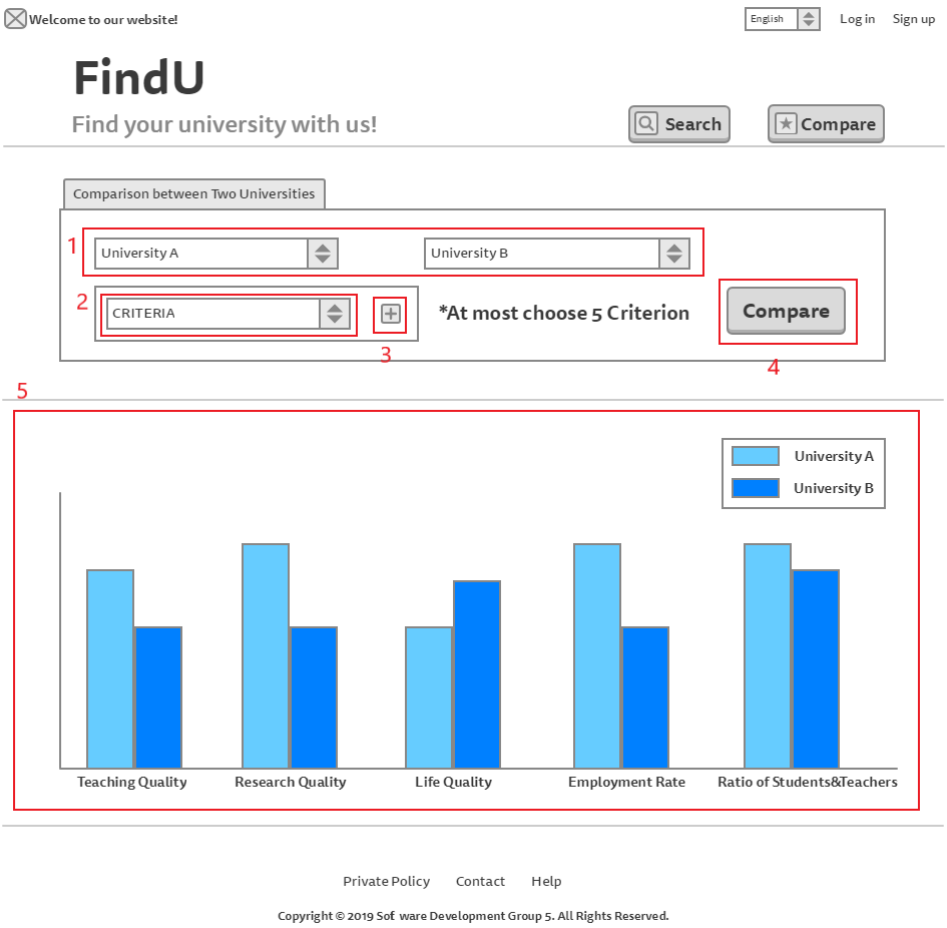


Figure 12: The prototype for the Compare Page

Element 1: University Drop-box
The user can choose two universities from the drop-box or input the first character of the university and the matching would be conducted with the options in the drop-box.

Element 2: Criteria Drop-box
The user can choose one criterion from the drop-box, which would be treated as one element in the x-axis to help to compare universities.

Element 3: Add Button
When the user clicks on this button the criterion would be added into the bar chart below.

Element 4: Compare Button
When the user clicks on this button the website would produce the result from the query condition and show it in the bar chart below.

Element 5: Compare Bar Chart
The data in the table is received according to the query condition (based on the universities and criteria) and acquired from the database. From this chart the user can intuitively reach conclusion about the relative advantages and disadvantages of two universities.

6.4 Design decisions

We have several versions for the User Interface design. With the purpose of creating better user experience and acquiring a user-friendly interface, the initial version has been modified for several times.

6.4.1 Search Box

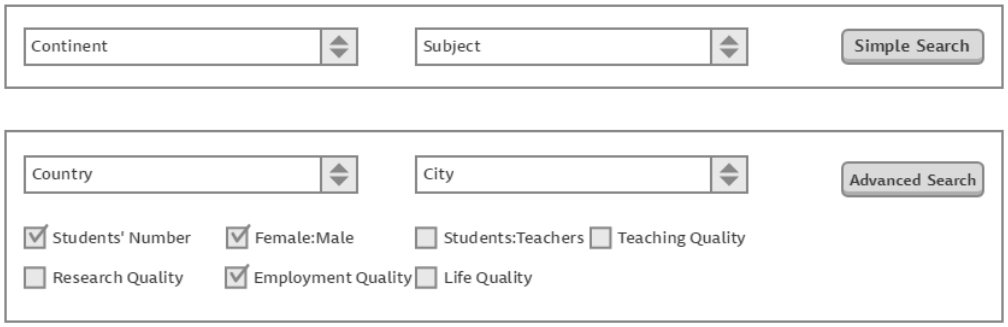


Figure 13: The first version of the Search Box

At the beginning we provided users the search section as is shown in Figure 13. The Simple Search and Advanced Search are in one page. However, this version may lead to some confusion. Users may think that these are two separated functions, with the location and subject not included in the advanced search. However, for our design people can use Simple Search to filter the universities and use Advanced Search to priorities aspects that they care about. Advanced Search is a follow-up and elevated function on the basis of Simple Search.

As a consequence, we set a container control to help users to switch between Simple Search and Advanced Search, which leads to less confusion.

6.4.2 Advanced Search Box

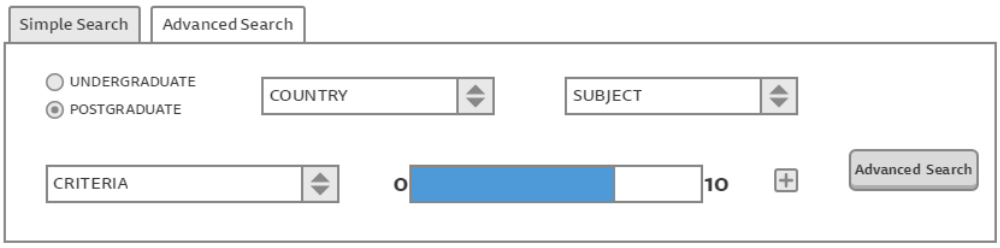


Figure 14: The second version of the Search Box

For the second version of Search Box, we changed the section into a container control. Compared to Simple Search, Advanced Search added three elements: Criteria Drop-box, Weight Bar and Add Button, which would realise one function together. However, as they are arranged separately users may not be conscious of their relationship.

Therefore, we placed these three elements into one panel, which is contained in a rectangular framework to connect them.

6.4.3 Compare Result Table

Figure 15: The first version for compare result table

For the result in Compare section, first we used pie chart to display the comparison outcome (rather than table) in order to show the result more intuitively. However, the pie chart can show the individual strengths and weaknesses of one university, but not easily compare the differences in one criterion between two universities. Therefore, we used a bar chart to show

the result. This kind of graph makes it possible that many universities can easily be compared at once.

6.5 Performance

As we previously discussed in the non-functional requirement section, performance should be taken seriously when it comes to designing a reliable website.

6.5.1 Page weight can be reduced in order to get a better performance

According to Google, most of the recent Google searches are done using smart phones instead of desktop PCs where the data load speed is faster due to the small page size. The page weight (or the page size) includes the files used when creating and designing such as the HTML file and the CSS style sheet, and images, videos, ..etc.

It is obvious that smaller page size introduces a better performance and faster data generation. As the websites' main function is to research text-base data, We can compress and resize the images used in our website and get rid of the unnecessary images and videos. We could also optimise the CSS file by getting ride of the wasted data to add up more space in the style sheets.

6.5.2 Requests can be reduced by reducing the number of page files

The number of page requests affects the performance of the website as the more requests the page gets, the more time it takes to load. Every time a web browser pings the web server of the hosted webpage, the server sends the requested contents (HTTP requests). Thereby, having less number of files on our website assists in reducing the number of HTTP requests required by the users' browsers. We can also view the Network panel of our website using Google Chrome.

As shown in Figure 16, Google Chrome's Network panel is very useful to understand what slowing the page load as it has the ability to illustrate what files a browser is requesting to load and the time it takes for each request.

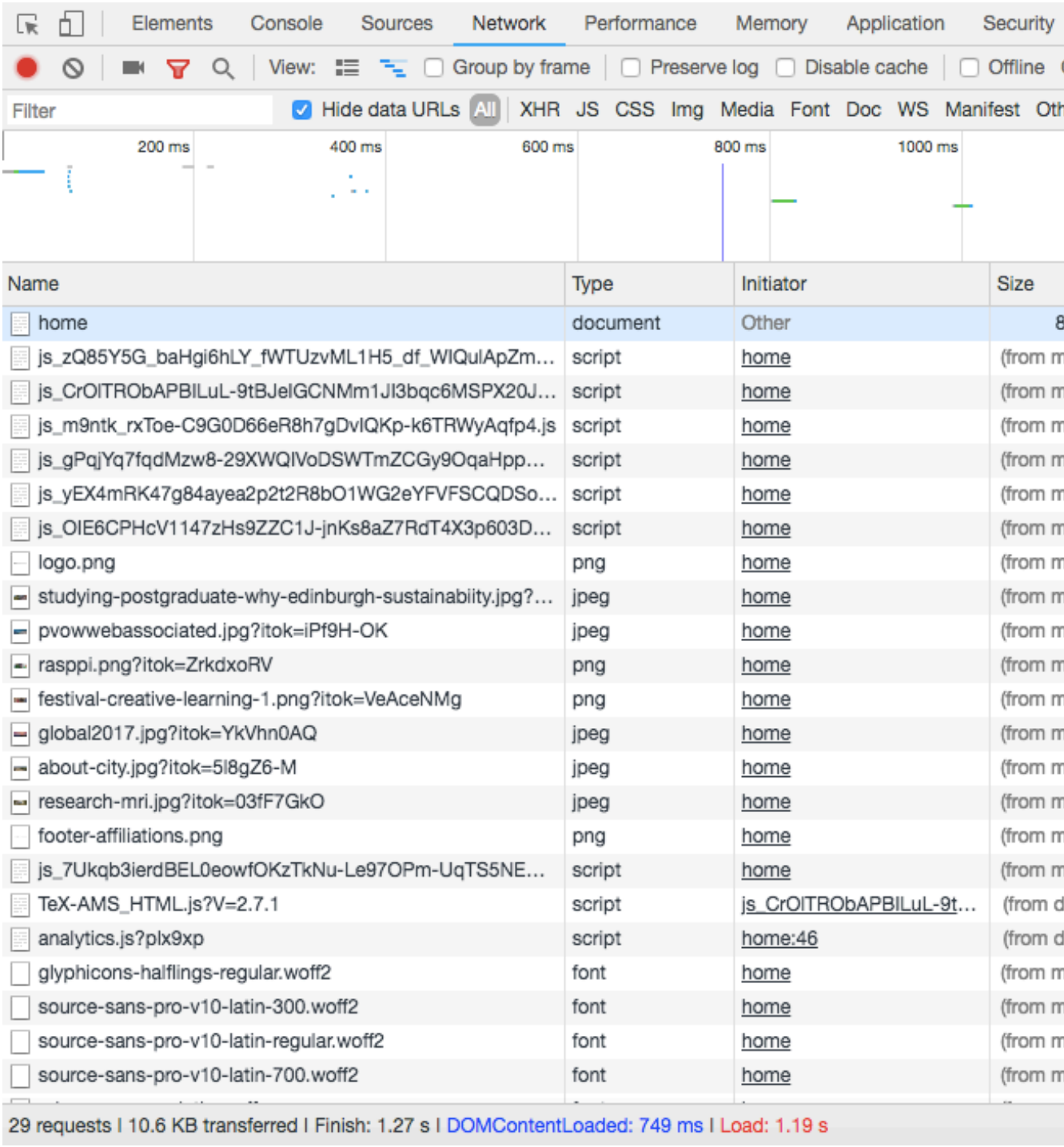
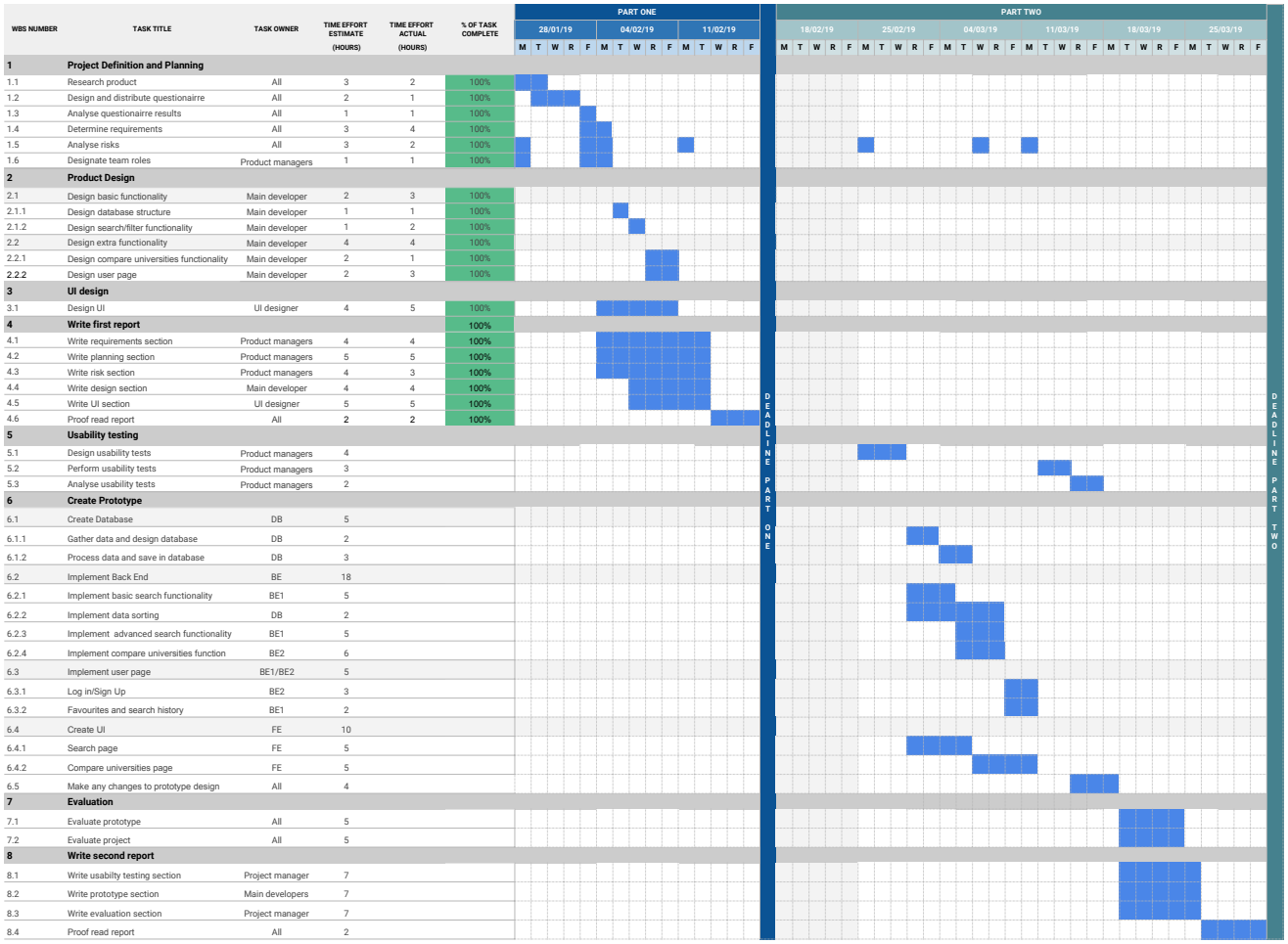


Figure 16: The Network panel of the University of Edinburgh website.

7 Appendix - Gantt chart



WBS NUMBER	TASK TITLE	TASK OWNER	TIME EFFORT ESTIMATE (HOURS)	TIME EFFORT ACTUAL (HOURS)	% OF TASK COMPLETE
1	**Project Definition and Planning**				
1.1	Research product	All	3	2	100%
1.2	Design and distribute questionairre	All	2	1	100%
1.3	Analyse questionairre results	All	1	1	100%
1.4	Determine requirements	All	3	4	100%
1.5	Analyse risks	All	3	2	100%
1.6	Designate team roles	Product managers	1	1	100%
2	**Product Design**				
2.1	Design basic functionality	Main developer	2	3	100%
2.1.1	Design database structure	Main developer	1	1	100%
2.1.2	Design search/filter functionality	Main developer	1	2	100%
2.2	Design extra functionality	Main developer	4	4	100%
2.2.1	Design compare universities functionality	Main developer	2	1	100%
2.2.2	Design user page	Main developer	2	3	100%
3	**UI design**				
3.1	Design UI	UI designer	4	5	100%
4	**Write first report**				**100%**
4.1	Write requirements section	Product managers	4	4	**100%**
4.2	Write planning section	Product managers	5	5	**100%**
4.3	Write risk section	Product managers	4	3	**100%**
4.4	Write design section	Main developer	4	4	**100%**
4.5	Write UI section	UI designer	5	5	**100%**
4.6	Proof read report	All	2	2	**100%**
5	**Usability testing**				
5.1	Design usability tests	Product managers	4		
5.2	Perform usability tests	Product managers	3		
5.3	Analyse usability tests	Product managers	2		
6	**Create Prototype**				
6.1	Create Database	DB	5		
6.1.1	Gather data and design database	DB	2		
6.1.2	Process data and save in database	DB	3		
6.2	Implement Back End	BE	18		
6.2.1	Implement basic search functionality	BE1	5		
6.2.2	Implement data sorting	DB	2		
6.2.3	Implement advanced search functionality	BE1	5		
6.2.4	Implement compare universities function	BE2	6		
6.3	Implement user page	BE1/BE2	5		
6.3.1	Log in/Sign Up	BE2	3		
6.3.2	Favourites and search history	BE1	2		
6.4	Create UI	FE	10		
6.4.1	Search page	FE	5		
6.4.2	Compare universities page	FE	5		
6.5	Make any changes to prototype design	All	4		
7	**Evaluation**				
7.1	Evaluate prototype	All	5		
7.2	Evaluate project	All	5		
8	**Write second report**				
8.1	Write usabilty testing section	Project manager	7		
8.2	Write prototype section	Main developers	7		
8.3	Write evaluation section	Project manager	7		
8.4	Proof read report	All	2		

Figure 17: Caption

References

[1] FindAMasters. https://www.findamasters.com/masters-degrees/computer-science/?1002&Sort=C.

[2] ShanghaiRanking. http://www.shanghairanking.com/ARWU2018.html.

[3] TopUniversities, Jan 2019. https://www.topuniversities.com/universities.

[4] World University Rankings, Oct 2018. https://www.timeshighereducation.com/world-university-rankings/2019/world-ranking#!/page/0/length/25/sort_by/rank/sort_order/asc/cols/stats.

[5] QualityOfLifeInEdinburgh, United Kingdom. https://teleport.org/cities/edinburgh/#summary.