Software Development Coursework - Part 2

Group 5

April 19, 2024

Contents

1 Usability Testing **3**
- 1.1 Test cohort . . . 3
- 1.2 Testing procedure . . . 3
- 1.3 Analysis plan . . . 4
- 1.4 Test cases . . . 4

2 Usability analysis **5**
- 2.1 Test description . . . 5
- 2.2 Deviations . . . 5
- 2.3 Questionnaire results . . . 5
- 2.4 Prototype changes . . . 9

3 Prototype description **11**
- 3.1 Framework Structure . . . 12
- 3.2 Code Implementation . . . 13

4 Prototype Evaluation **16**
- 4.1 Design alternation/replacement . . . 16
- 4.2 Requirements met . . . 17
- 4.3 Future work . . . 17

5 Project Evaluation **19**
- 5.1 Schedule evaluation . . . 19
- 5.2 Risk evaluation . . . 20
- 5.3 Product development model evaluation . . . 20

6 Appendix **20**
- 6.1 Data sources . . . 20
- 6.2 Test cases . . . 21
- 6.3 Usability questionnaire . . . 21
- 6.4 Gantt chart . . . 22
- 6.5 Source code . . . 24

1 Usability Testing

1.1 Test cohort

The test cohort for the usability testing should reflect the target user group. For our website this should mostly be high school or university students who want to search for a university to study at for either an undergraduate or postgraduate degree. These will mostly be young people from the ages of 16-23, with the exception of a few mature students. The genders of the test cohort should be balanced to ensure that our website is tested by all genders, as students of all genders will use the website. Ideally around 50 people should be tested in order to sample a wide range of people. This should also include people with different accessibility requirements in order to ensure that these people are catered to.

1.2 Testing procedure

We plan to run usability tests where we ask the user to perform a selection of task scenarios under observation. These scenarios are selected so that all features of the website are used. Users are observed in order to directly see how they use the website, how long it takes them to perform the scenarios and if there are any issues they encounter. They are then given a survey to fill in about their experience. If given more time we would then plan to analyse the results and then improve the website according to the feedback. The tests would then be performed again, repeating this cycle until the quality is acceptable. However due to time constraints, we plan to only test, analyse the results and improve the product once.

This section contains the usability tests scenarios. We are going to analyse the expected and actual results of running these tests in section 2.

Scenario 1: Compare Page

You are a student who wish to apply for an undergraduate degree at University of Cambridge and University of Oxford. Select these two universities in the comparison table and try to identify which one is the best choice when you consider the quality of teaching as your primary concern.

Scenario 2: Basic Search

Ask the user to perform a Basic Search for universities in the UK which are good for computer science.

Scenario 3: Advanced Search 1

A new user attempts to use the advanced search by firstly selecting the university location and preferred subject, scaling his/her preferable criteria(s) and then press the Filter button to proceed. The actual results should be as expected. For example, if a user selects Location: United Kingdom, Subject: Computer Science, scales quality of life to good and then presses filter.

Scenario 4: Advanced Search 2

Ask the user to perform an advanced search for universities in the UK. They are not interested in any particular course, but would like to select a university where the cost of living is not too high, with a strong research track record, and a good international outlook.

1.3 Analysis plan

The results from the usability test will be both quantitative and qualitative. The quantitative results consist of data collected during the observation of the usability test, such as how long it took the user to perform tasks, and if they made any mistakes, as well as data from the survey that users will be asked to fill in after the usability test.

To analyse the observation data we plan to take the average values of the time taken to perform the tasks. These are needed to understand how a variety of users may find using the website. For example if one user struggles to use the website then it is likely that given a large number of people others will too.

The questionnaire questions are given in the Appendix. The results of these will be used to analyse the test cohort as well as how they perceived the website. Questions are asked about the gender, age and education level of the participants to determine the demographic of the average cohort participant. Users are asked how they useful they found the website on a whole, as well as how useful or difficult to understand they found particular features of the website. Those which are lowly rated will become targets for improvement.

Analysing the qualitative data is less straightforward. Comments and suggestions made by the users we will noted down during the testing. There is also the option to leave suggestions in the questionnaire. We plan to look into all comments made by users, however we will give priority to comments that are made by multiple users as this will reflect something that is impacting the usability for most users.

1.4 Test cases

We have run some test cases before proceeding with the usability tests. The purpose of this is to provide users a stable platform. Thus, focusing on the users' experience regarding the

level of ease when using the website. All performed test cases have passed except for running the website on Internet Explorer where the page was displayed but lost some of its design functionalities. A copy of the test cases is in the appendix.

2 Usability analysis

2.1 Test description

There were two types of usability tests: the first one was to let the participants follow the provided scenarios and record the time they spent. The second was to let them choose their preferable options and features to use. In the second method, we focused on writing down the tasks they performed as well as recording the time spent. Section 1.2 provides some possible testing scenarios. After the test, the participants are asked to fill in a survey to get their feedback in order to improve our services and evaluate tool's effectiveness from the users' prospective. The main features of the website are Simple search, Advanced search and Compare two universities. Participants were asked in person to perform a usability test on these three features. Results are shown in section 2.3.

The conducted survey consists of nine simple questions. In the first three questions, participants were asked about their gender, age, level of education, in order to correctly analyse the survey results. The questionnaire then sought to get participants' feedback on their experience of using the website. Questions 4,5 and 6 were about asking participants for their opinions of the website. The last two questions were asked to understand what features were difficult for them to deal with. A link to the survey can be found in the appendix.

2.2 Deviations

The proposed plan was to perform some usability tests on large number of students with different ages, levels of education and from different countries. However, as the project has a fixed deadline, there were only 13 participants and most of them are currently postgraduates. We also ended up with many more female participants than male. Our current website is in English, which is the international language of business. The website was supposed to be multilingual but due to the time limitation, we did not apply this feature whereas the main purpose is to have an understandable and stable web page.

2.3 Questionnaire results

The questionnaire was conducted among 13 students, 15.38% of them are male and 84.62% are female. Figure1 shows the age ranges of those who had taken the survey, where most of

them (with a percentage of 69.23%) are in the range of 18-24 years old and 30.77% are in the range of 25-34 years old. The typical time spent on filling in the survey was 4 minutes.

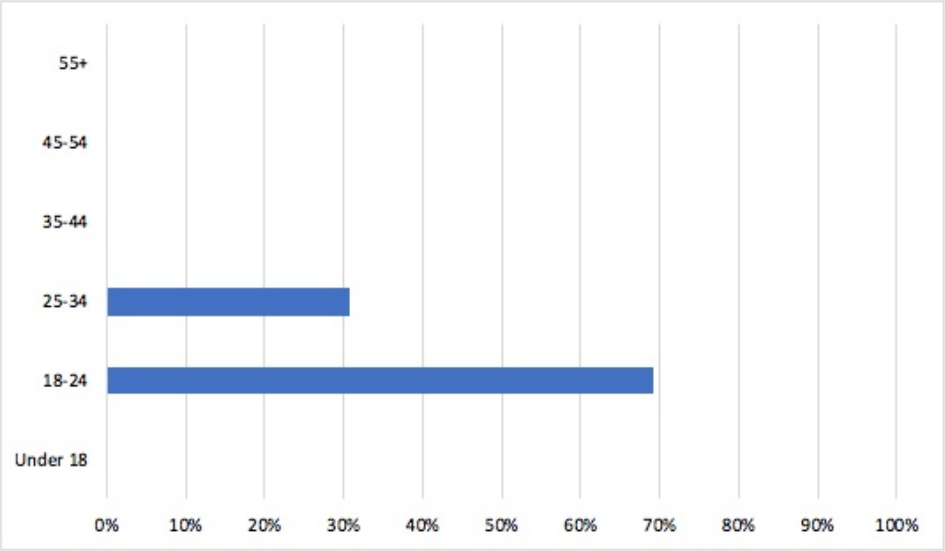


Figure 1: Participants' age ranges

As part of the usability testing, it is essential to get the participants' level of education. Figure2 illustrates the percentage of the students' education levels, where it is clear that most of them are postgraduate students with a percentage of 53.85%. The second highest percentage (30.77%) were high school students, 7.69% were undergraduate and 7.69% were PhD students. As mentioned in section 1.1 our test cohort should mostly be high school students. However given our limited time and resources to perform the tests having 30.77% high school students is fair. We should therefore have a reasonable reflection of how our target audience would view our website.

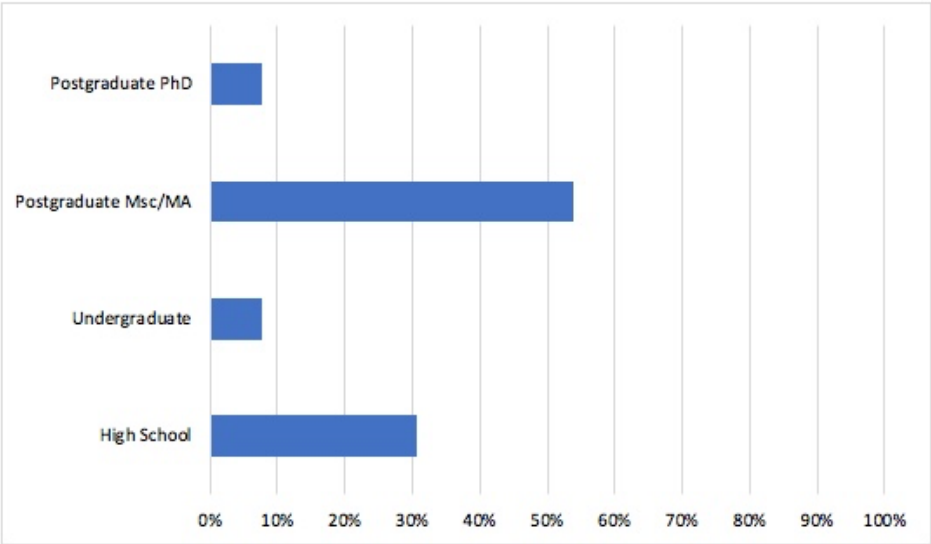


Figure 2: Participants' levels of education

As we aim to get the users' feedback after using the website, we have asked them to select words to describe the prototype. As can be seen in Figure3, 100% of the users have chosen to describe the website as "Useful", which shows that using the website has given them some useful information about the available universities and courses. Around 46 % of users also think the website is unique which means that its features have managed to differentiate it from other current university research tools. However, there were some users who think that the website is ineffective and unreliable, which could be due to some issues with it. These will be addressed in section 2.4. None of the users asked thought that the website was impractical, which should mean that the overall design of the website is usable.

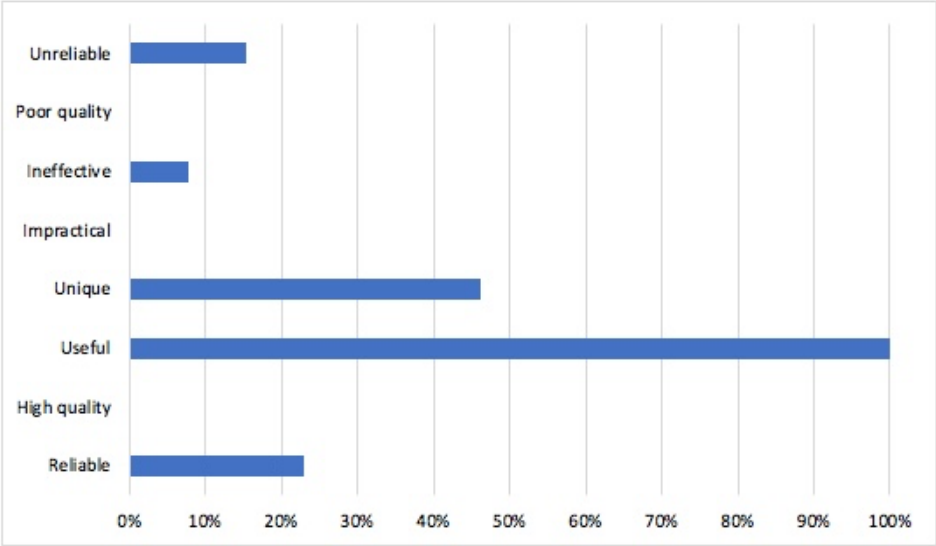


Figure 3: Participants' feedback on the website

In the questionnaire we also sought to understand the most useful feature, thus, Participants were asked to rank Simple search, Advanced search and Compare two universities features on a scale of 1 to 5 (5 being very useful, 1 being not useful) based on their experience of using the website. The weighted average of the the participants measurement is shown in Figure4.

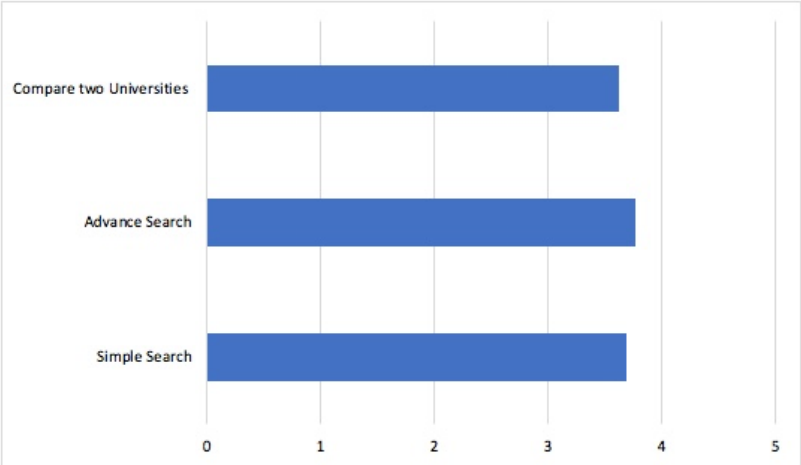


Figure 4: The weighted average of features participants consider useful when using the website

The weighted average of ranking these three features were very close, "Simple Search" is 3.69/5, "Advanced Search" is 3.77/5 and "Compare" is 3.62/5. This means that the "Advanced Search" was the most popular feature for the users. The results of this mean that the

participants' thought that all the features of the website were useful, and based on this no single feature needs to be significantly improved.

In order to improve the websites' presentation and usability, another question was asked to get the users' impressions on what features were difficult to understand or use. The feedback can be found in the next section (2.3.1).

2.3.1 Usability testing results

As mentioned in section 2.1 We had two types of usability tests. When users performed the tests according to the provided scenarios, which are mainly divided into three, -each related to one feature-, the typical time spent on each was as follows:

- 1:11 minutes on "Simple search" feature.
- 2:49 minutes on "Advanced search" feature.
- 1:00 minute on "Compare two universities" feature.

The average time users spent when performing the usability tests based on their preferable options and features was 3:41 minutes.

Based on the results of the usability testing we have identified the following issues with the prototype.

- Results are not maintained when the page is changed
- The font colour for the table is too light
- The weights on the sliders are not quantified
- The results appear too far down the page
- The compare page is hard to find

The first of these was found by all users during the test and is due to a bug in the code. In the final version of the prototype we expect this to be fixed. Other issues are due to the readability and layout of the website, which make it confusing for users.

2.4 Prototype changes

Based on the issues found during the usability testing we have made the following changes to the prototype. These are show in Figure 5.

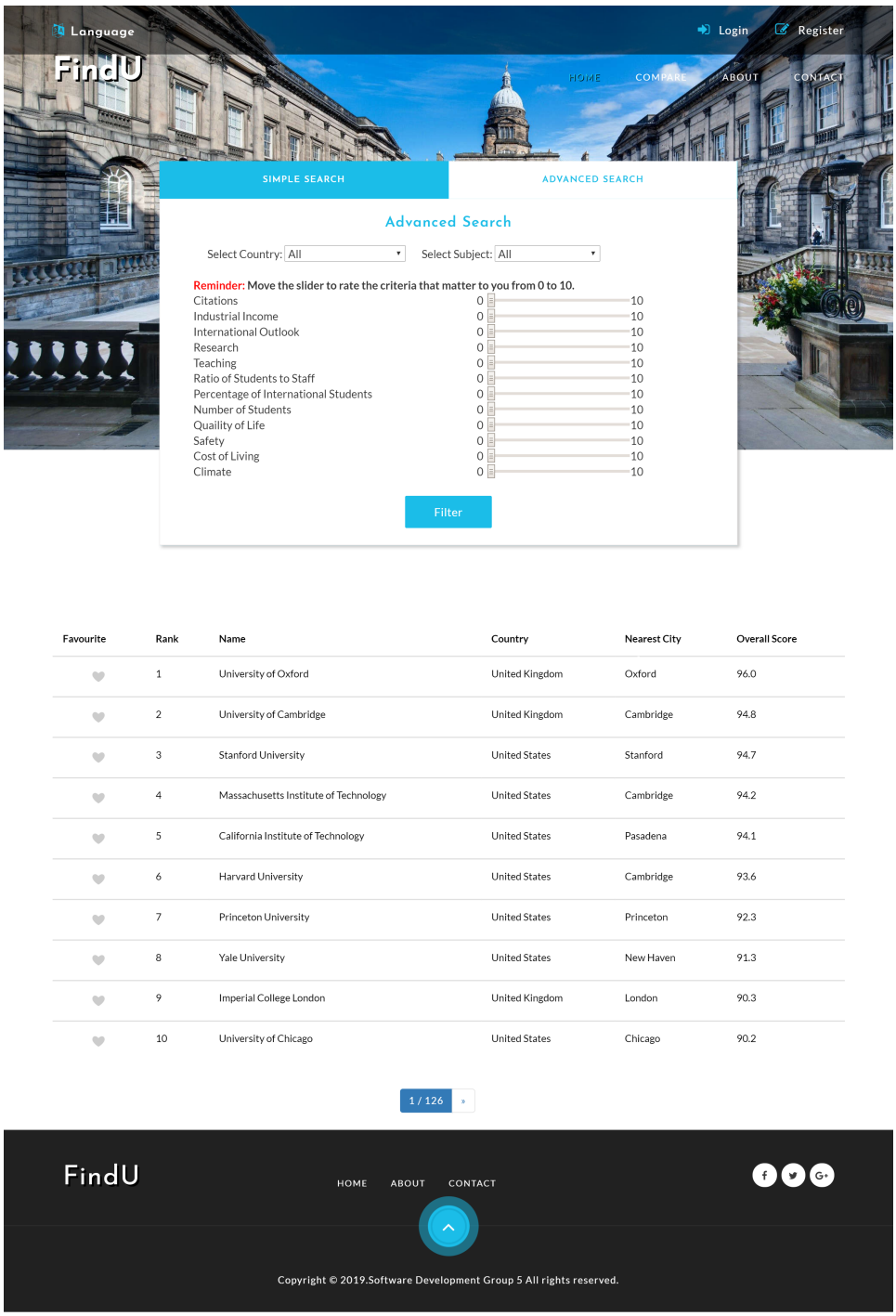


Figure 5: The final version of the prototype, showing the fixes to the sliders bars and the change in table colour.

2.4.1 Fix pagination

A bug in the pagination was causing the results to refresh to unfiltered after clicking to the next page. This was a cause of confusion in the usability tests. This bug was fixed to allow the correct results to show.

2.4.2 Added numbers to slide bar

Some users were confused about the slider scale and what moving it meant to the results. To solve this word descriptors such as "important" or "not important" were replaced with numbers. This is to make it less confusing and simplify the design. In the future we could add the option to show the value that is currently selected on the slider.

2.4.3 Changed table font colour

Some users found the light grey colour of the results table too difficult to read. To fix this we changed the colour to black.

3 Prototype description

To create this database-driven website easily, we use Django, a python-based web framework to build the prototype. The back-end code was written by Python and the front-end by html, decorating with css and js codes.

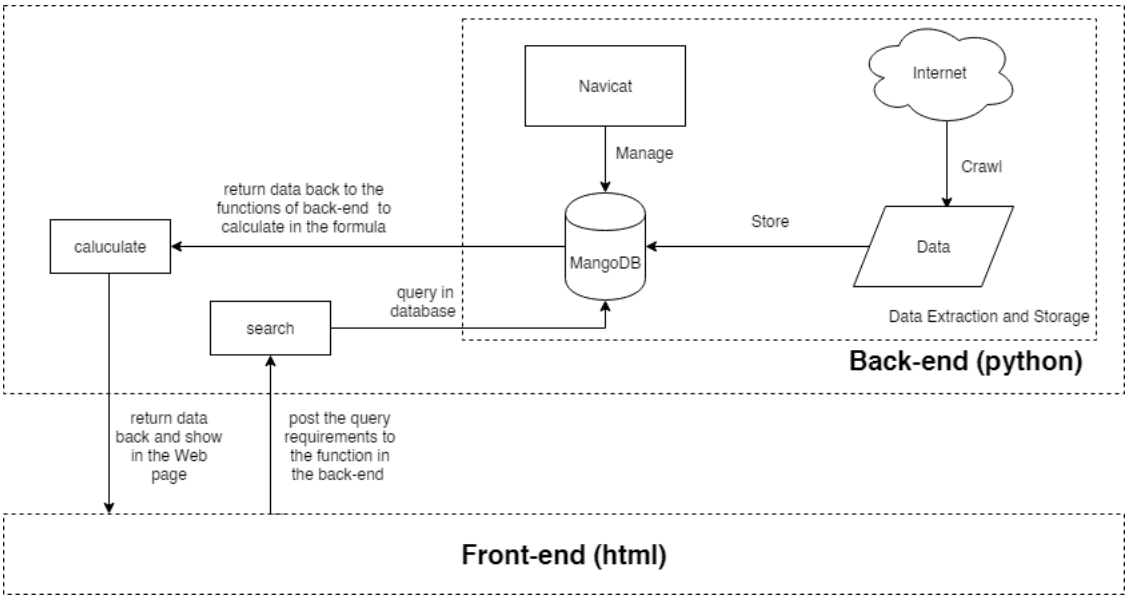


Figure 6: Front-end and Back-end of the Prototype

3.1 Framework Structure

We can see the Django framework for this project in the Figure 2.

For *findu* folder, the file *settings.py* contains settings which are shared between development and production settings. The database configuration is done in this file to state the engine package in python and database name. In *urls.py*, urls which are configured in *results* folder are included in it.

Inside *results*, there are three sub-folders to save different kind of files. *Migrations* are Django's way of propagating changes made to models into the MongoDB. They are designed to be mostly automatic. In *static*, css and js files, images and fonts using in the html files are all stored here. This directory is used to provide our project wide static assets. However, the html files would be saved separately in the *templates*, which static files are linked to with the statement between the *header* tags. The code is presented as below.

```
{% load static %}
<link href="{% static 'css/bootstrap.css' %}" rel="stylesheet" type="text/
    css" media="all" />
```

There are also four configuration files inside the *results*, *models.py*, *urls.py*, *views.py*,*admin.py*, to be listed. *models.py* contains the essential fields and behaviours of the storing data, each of three models map to a single database table in MongoDB. Some functions are written in *views.py* to get the queried data from database and calculated according to the formula before passing it to the front-end.

The complete Django file structure is as below.

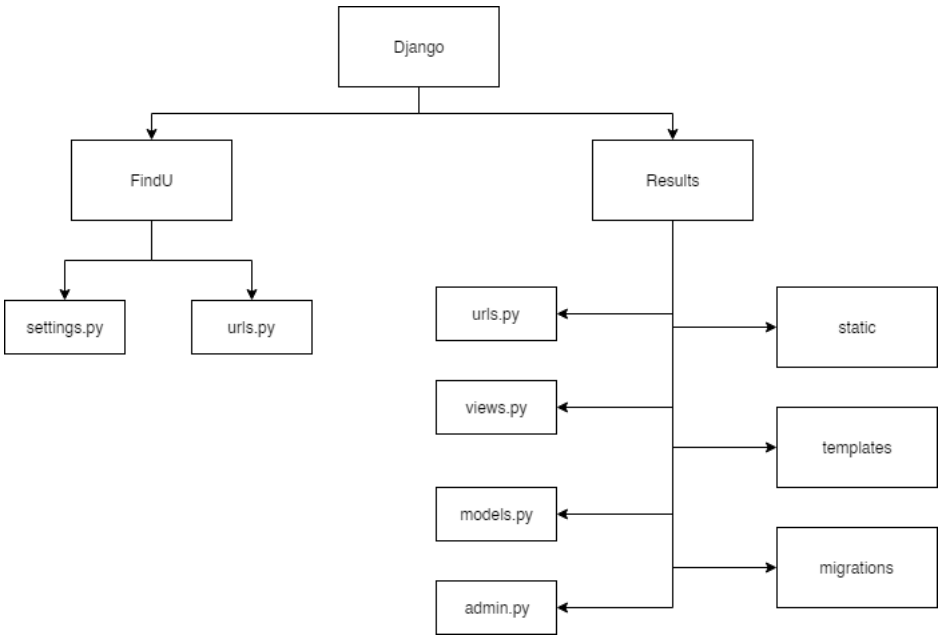


Figure 7: Django file structure

3.2 Code Implementation

The project consists of three webpages: homepage, 'compare' page and 'about' page.

For all three pages, they have the same header and footer to maintain UI design consistency. Language transfer, Login and Register buttons are placed on the top of the webpage. Meanwhile, a navigation bar is located in the header to show a clear visual hierarchy, which is made up of a link to the 'compare' page, 'about' page and the mailbox, which is for users to contact us with any feedback for this website. For the footer, we also place the link to the 'about' page and the mailbox, plus buttons to share our website to Facebook, Twitter and Google Plus. BAn up-arrow button is located here to provide users with a method to go to the top of the webpage.

In the homepage, we make the main search function of this project prominent by putting the search box in the centre. Once users enter into the homepage they can start searching and ranking universities. To avoid users having extra effort to go back to homepage for another search, the search results are placed below the search box rather than opening another page.

As detailed in the coursework one, we put the Simple Search and the Advanced Search in one section with two different tabs to switch from one an other. The Simple Search is implemented by filtering the data from database by the search conditions from the front-end. The layout for the Simple Search is shown in the figure 3.

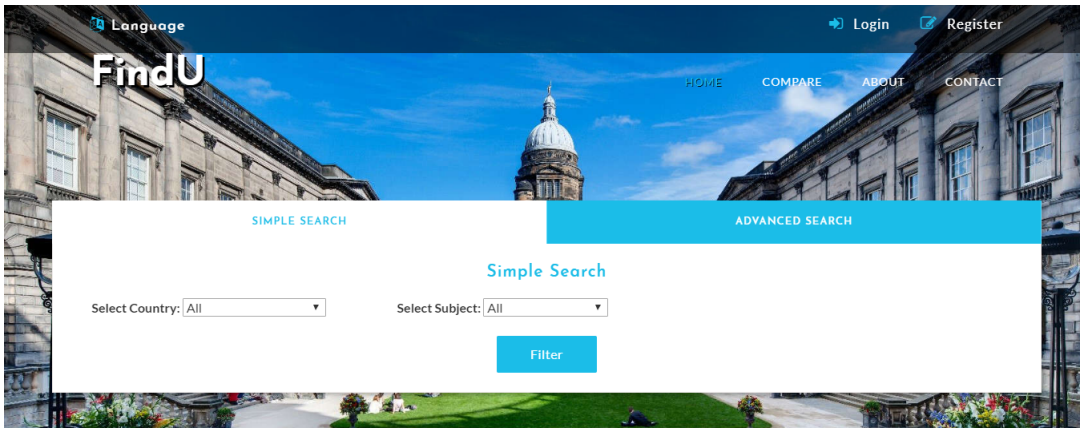


Figure 8: The Simple Search Part in the Index Webpage

The Login and Register feature is implemented by the javascript code. This this function is not linked to the database, which means the database can not not save the user data. We set some restrictions for Register function. First, for the email, the content users fill with should be the email format. Second, the password and confirm password should be the same. Third, all the blanks should be filled. If any condition is not satisfied, some reminders would come out to tell the users what errors occurs, leading to stopped registration. The Register box is shown as below.

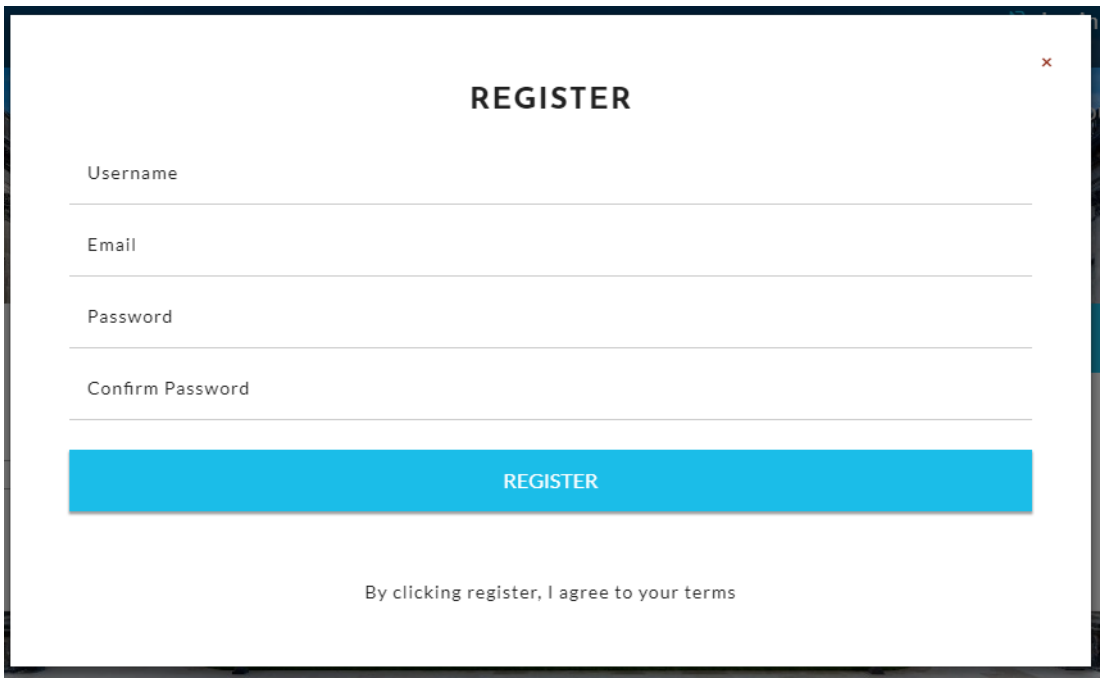


Figure 9: The Register function

For Login Function, likewise, the email should be input as the email format.

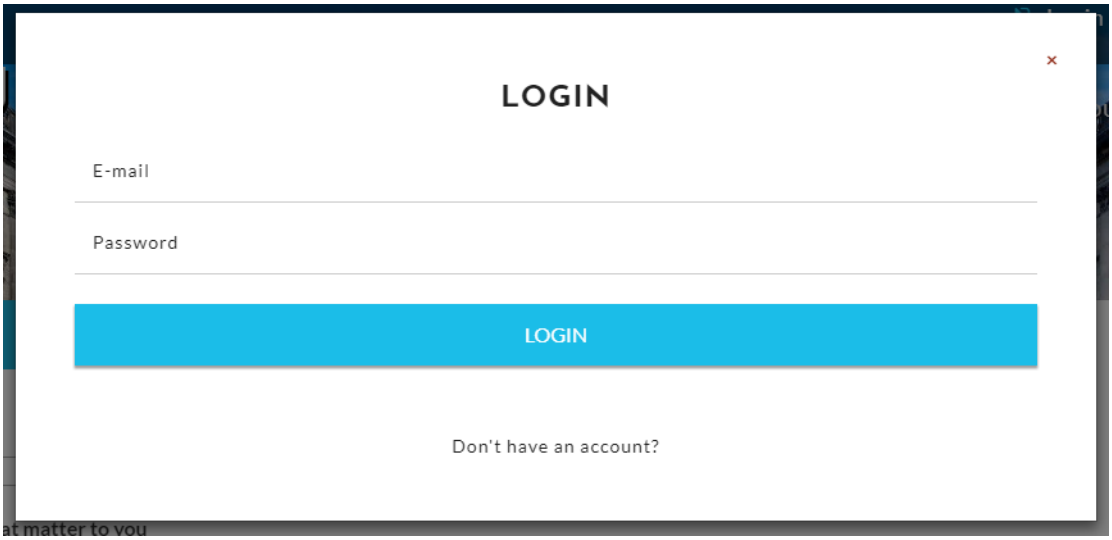


Figure 10: The Login function

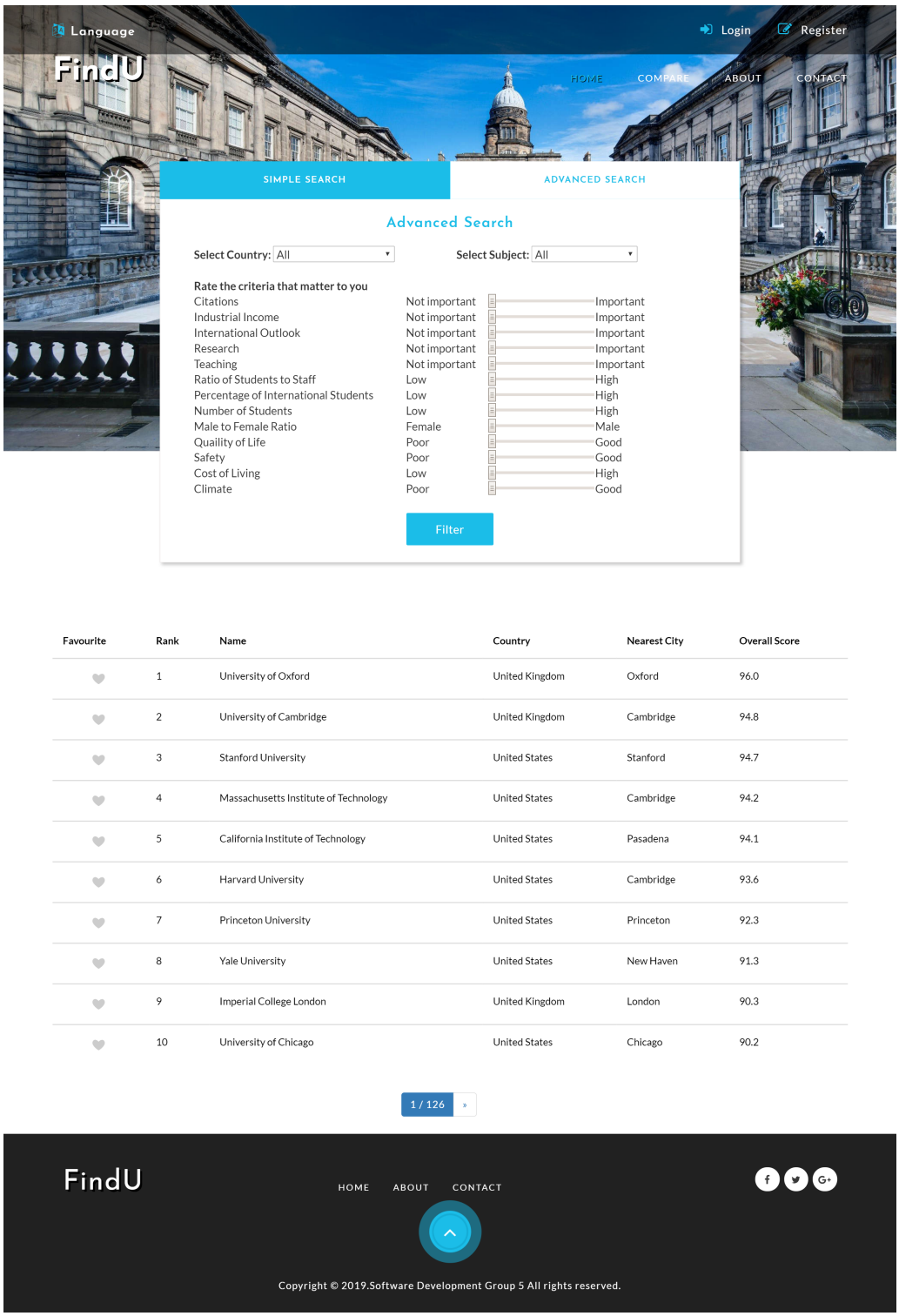


Figure 11: The index webpage with advanced search

To implement the Advanced Search, we need to add a calculating step between getting the data and pass the results from the back-end to the front-end. The formula is shown as below.

$$new_score = \sum_{n=1}^{13} score_n \times weight \tag{1}$$

For this formula, $score_n$ stands for scores of 13 kinds of criteria to filter the universities, which are 'scores_citations', 'scores_industry_income', 'scores_international_outlook', 'scores_research', 'scores_teaching', 'stats_student_staff_ratio', 'stats_pc_intl_students', 'stats_number_students', 'stats_female_male_ratio', 'quality_of_life_index', 'safety_index', 'cost_of_living_index', 'climate_index'.

The results located below the search box have been paginated by 10 records, which helps to shorter the page length and help users to find what they want more quickly. We use the *paginator* package in Django to implement this function.

The 'compare' function has been taken out and put in a single page for users due to the fact that it is not the main aim for people who enter into this website. We regard it as one additional function to provide users more methods when users confronting the problem of choosing the future universities.

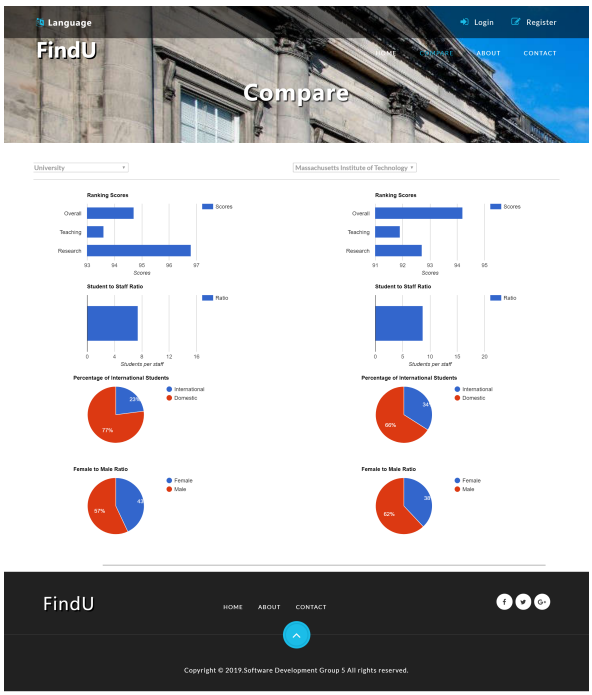
The 'about' webpage is necessary to declare the data sources.

4 Prototype Evaluation

4.1 Design alternation/replacement

In the original UI design, we planned to draw statistics as grouped bar charts instead of pie charts for female to male ratio and percentage of international students in the 'compare' page. This would allow users to compare the statistics of two universities more interactively. Nevertheless, this would require the front end to load the statistics instantly before plotting and displaying the results. In order to avoid slow loading in the prototype the statistics are plotted separately for each university. This allows the front end to load statistics of each university in parallel. Figure 12a illustrates the original design, where all categories are plotted as bar charts. Figure 12b shows the 'compare' page in the prototype, where the key statistics of each university are displayed in a combination of bar charts and pie charts.

(a) 'Compare' page in the original design.



(b) 'Compare' page in the current prototype.

Figure 12: Screenshots of 'compare' pages in the original design and the prototype.

4.2 Requirements met

Table 1 shows the list of requirements for the product identified in the first report and whether we were successful in meeting these.

Overall we were successful in meeting all of the MUST criteria apart from the responsiveness. We did not have enough time to test the website on different platforms such as mobiles and tablets. The website does work correctly on Chrome and Safari web browsers. It works on Internet Explorer, but the slider in the advanced search does not work.

Other requirements we failed to meet were the user log in page, showing a list of available courses, allowing users to provide feedback and having multiple languages. However none of these requirements affect the overall functioning of the prototype and it is usable in its current form.

4.3 Future work

4.3.1 Basic search

For the basic search we did not manage to implement any functionality to show the courses on offer at a university. This was because finding the data for this in an easily importable

Requirement	Priority	Met?
The product will be a website	MUST	Y
The product will use a database	MUST	Y
The database will draw data from multiple data sets	MUST	Y
Log in and save previous searches and favourites	COULD	N
Mix and match search criteria	MUST	Y
Filter the results by location and subject	MUST	Y
Customise the search criteria for an advanced search	MUST	Y
Ordering universities based on criteria	SHOULD	Y
Show what courses are on offer at each university	SHOULD	N
Compare universities	COULD	Y
Access to different information about a university	MUST	Y
Provide feedback about the tool	COULD	N
Usability - Easy to use and navigate	MUST	Y
Readability	MUST	Y
Responsiveness	MUST	N
Multiple languages	COULD	N
Performance	MUST	Y

Table 1: The product requirements and whether these were met.

format was not possible. It may have been possible to add this information by hand for some universities but this was not done due to lack of time. In the future we would work to add this into the results table in the form of a link to the information. We also did not implement anything to change the ordering of the results in the basic search. In the future we would add this so that the users could click the top of a column and the results would become ordered by this value.

4.3.2 Advanced search

To improve the advanced search more data and results would be added to the database. Currently the advanced search becomes very slow when searching on lots of data as it has to write the new ranking to the database. For this reason we limited the search to only universities in the UK or US. In the future we would like to find a more efficient way of handling the data so that more universities can be added.

4.3.3 Compare page

In the current prototype, data used in 'compare' function does not stored in the online MongoDB database as the the data used in 'advanced search' function. Instead it is hard coded at the front end. The choice of this data storage is to improve the efficiency of data visualisation and allow users to test the functionality. Therefore, users can only select from ten universities

in a drop-down list. In the future, data for 'compare' function will be stored in the database and loaded instantly once the 'compare' function is requested from the front end. The change of data storage will offer a much larger range of university selection for the compare function.

Another direction of extension for the compare page is to enable the website to compare multiple universities simultaneously. Currently the website only allows up to two universities for comparison at a time. Users can first generate a list of universities they wish to compare, then request to display the statistically analysis across the university selected.

In addition, the plotting styles can also be modified once the data can be loaded from the database. Currently the 'compare' function will plot statistics of two universities selected separately as two series of plots. In future releases, the front end will first query corresponding data as the two universities are selected. Then the database would present data to the front ends for plotting. Next the plots will be generated as grouped bar charts for each criterion. This will allow users to visualise the differences between the universities for each categories.

4.3.4 Multiple languages

Originally we planned to give users the ability to change the language of the website. This is something that could be added in the future in order to appeal to a global audience.

5 Project Evaluation

5.1 Schedule evaluation

The completed Gantt chart for the project is shown in the Appendix. As seen in the Gantt chart the task times took longer than estimated. The prototype development phase of the project ended up taking a week longer than estimated. This meant that other parts of the project, such as the usability testing and analysis, as well as writing the report had less time to be completed.

The extra time needed for the development of the prototype is likely due to multiple factors. Firstly most of the team members had a lack of experience in developing websites, and using the particular codes and modules. This meant that extra time was needed to learn how to do tasks before any progress could be made. Some tasks took longer than anticipated due to their complexity being underestimated. For example the login page, and user favourites.

The tasks and their role assignment may have also affected the time keeping. Some tasks had dependencies on others which had not been realised until during the development. For example it is difficult to implement the search functionality in the back-end without having some of the UI in place to input and output the data. This meant that extra time was needed for tasks to be completed before others could start. Other tasks were found to be difficult to assign to team members because they were directly linked to other tasks.

It may have been possible to stop work on the prototype a week earlier, however this would have left some of the features unimplemented and some errors in the website. It was decided that allowing the schedule to slip by a week would be better as the prototype would be almost completed and easier to use for the usability testing. In hindsight maybe the usability testing should have been performed on the prototype at this earlier stage in order to allow more time to make any required changes that were discovered during the testing.

5.2 Risk evaluation

Overall the risks presented in the previous report were a fairly accurate representation of the problems that occurred during the development. The biggest risk in the project turned out to be lack of experience as predicted. Poor planning ended up being more detrimental to the project than poor time management. As mentioned previously, the planning of tasks and assignment contributed to the schedule slippage.

5.3 Product development model evaluation

In the planning stage we decided to opt for the Design to Schedule development strategy. This meant that we would concentrate on implementing the product in stages, where each stage could be a working release. Features are added to each stage in order of importance. Overall we mainly stuck to this strategy, the main features of the product were implemented first, giving a working release at the time of the deadline for finishing the prototype. Some less important features were not implemented due to time constraints, such as the user log in page and user favourites.

6 Appendix

6.1 Data sources

The main university data is taken from the Times Higher Education website [1]. This data is imported into the database using the `Datainserts/insert-uni.py` python script which pulls the data directly from the `json` file used on the Times Higher Education website.

The data for the universities nearest city is found on the UniRank website [2]. The city data is imported into the database where the university name matches with that in the database (using `Datainserts/insert-cities.py`).

Finally the quality of life index, climate index, cost of living index and safety index are taken from numbeo [3]. The data from this website is copied into a .csv file which is then imported into the database. The data is matched on both the city and country in the database in order to

ensure that the correct data is used (for example Cambridge is found in the US and the UK). The script for this is `Datainserts/insert-city_data.py`.

6.2 Test cases

Testcase ID	Test Scenario (Steps)	Expected Result	Actual Result	Test Priority	Pass / Fail
Nav_1	Select "Home" from the navigation menu.	Redirect or refresh to the Home page.	Redirect or refresh to the Home page.	High	Pass
Nav_2	Select "Compare" from the navigation menu.	Redirect to the Compare page.	Redirect to the Compare page.	High	Pass
Nav_3	Select "About" from the navigation menu.	Redirect to the About page.	Redirect to the About page.	Medium	Pass
Nav_4	Select "Contact" from the navigation menu.	Redirect to the mailbox to send an email .	Redirect to the mailbox to send an email .	Medium	Pass
Base_search_1	Select Location: "Oman" and Subject: "Engineering"	Display a table contanis infromation about universities available in Oman with engineering subject.	Display a table contanis infromation about universities available in Oman with engineering subject.	High	Pass
Advan_search_1	Select Location: "United Kingdom" Subject: "Computer Science". Scale "Citations" and "Research" to important	Display a table contanis infromation about universities available in United Kingdom with Computer Science subject and citations and research scores	Display a table contanis infromation about universities available in Oman with engineering subject.	High	Pass
Perfor_1	Run the website in Google Chrome	Display the full website	Display the full website	High	Pass
Perfor_2	Run the website in Safari	Display the full website	Display the full website	High	Pass
Perfor_3	Run the website in Internet explorer	Display the full website	Some of the design functionalities were missing	Low	Fail

Figure 13: Test cases

6.3 Usability questionnaire

The questionnaire used in the usability testing can be found at:

https://www.surveymonkey.co.uk/r/CNGVKWP

6.4 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	4	100%
5.2	Perform usability tests	Product managers	3	3	100%
5.3	Analyse usability tests	Product managers	2	2	100%
6	**Create Prototype**				
6.1	Create Database	DB	5	8	100%
6.1.1	Gather data and design database	DB	2	4	100%
6.1.2	Process data and save in database	DB	3	4	100%
6.2	Implement Back End	BE	18	19	100%
6.2.1	Implement basic search functionality	BE1	5	5	100%
6.2.2	Implement data sorting	DB	2	0	0%
6.2.3	Implement advanced search functionality	BE1	5	7	100%
6.2.4	Implement compare universities function	BE2	6	7	100%
6.3	Implement user page	BE1/BE2	5	4	100%
6.3.1	Log in/Sign Up	BE2	3	4	0%
6.3.2	Favourites and search history	BE1	2	0	0%
6.4	Create UI	FE	10	15	100%
6.4.1	Search page	FE	5	6	100%
6.4.2	Compare universities page	FE	5	6	100%
6.5	Make any changes to prototype design	All	4	3	100%
7	**Evaluation**				
7.1	Evaluate prototype	All	5	4	100%
7.2	Evaluate project	All	5	3	100%
8	**Write second report**				
8.1	Write usabilty testing section	Project manager	7	7	100%
8.2	Write prototype section	Main developers	7	5	100%
8.3	Write evaluation section	Project manager	7	6	100%
8.4	Proof read report	All	2	1	100%

6.5 Source code

The source code can be found at:

https://github.com/RuwaidaA/Software-Development

References

[1] 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.

[2] uniRank. https://www.4icu.org/gb/.

[3] Cost of living. https://www.numbeo.com/cost-of-living/.