

# CA357 Report - Group 5

## Introduction

In this report we will analyse the usability of the User Interface of our WebApp CampusMeet. We will detail how we implemented the requirements which we established in our presentation, we will evaluate our prototype, analyse the results of our prototype and look at potential problems and how we would fix them.

## Implementation of Requirements

One of the first requirements we established was that our WebApp would have to be clean, clear and simple to use. This was in order to draw students and lecturers away from traditional technologies such as DCU Loop and email. We implemented this requirement by keeping the webapp clutter free, informative and easy to use. We used a variety of monochromatic colours to ensure that different aspects of the webapp stand out and the user isn't assaulted by a barrage of colour. We kept things simple for students by having dropdowns of their lecturers and their free slots as the only ones available for selection. Students may not enter messages and there are no external links to other webapps to distract them. When a meeting request has successfully been placed the student is informed to make sure they don't spam the lecturer with requests. They can then see the status of their meeting, "Accepted" or "Rejected". To keep things simple for lecturers, all they see upon logging in is a list of meeting requests and a choice to accept or decline them. The lecturer is also able to give the student a reason, should they decline a meeting. Another requirement we discovered was a way of coping with the cultural differences which are present in DCU. The steps we took to ensure this requirement was met included different formats for date and time and universal symbols. If we had more time we would look at implementing various languages, with the option of choosing your preferred language upon login.

From our interviews we discovered that students often felt like their messages weren't even being read quick enough. To ensure that students who needed a meeting quickly got what they needed we decided to add an urgency tag to the students meeting request. This would allow the lecturer to see their meeting at the top of the pile upon login and hopefully lead to a quicker response and ultimately meeting.

The webapp had to be accessible to a wide variety of people from first years using new websites for the first time to lecturers who have busy schedules and not a lot of free time. This information was paramount in our decision to create a simplistic, accessible and efficient design.

The webapp also had to be compatible with both computer screens and phone/tablet screens. From our survey we deduced that some students can sometimes use one type of device exclusively. This would increase our accessibility.

In order to ensure that our WebApp is accessible to users who are visually impaired we made sure the WebApp was screen reader accessible. We also avoided using colours which would cause problems for colour-blind people. This lead us to using the monochromatic colours which don't cause problems.

We also had to make sure lecturers can change their availability at any given time. Our interview with a DCU lecturer proved to us that they are very busy people and that new, urgent commitments can arise at any given time.

Students also have the option of cancelling a lecture as long as it is in the pending stage. This would avoid problems where students accidentally book the wrong lecturer or a meeting at an unsuitable time.

## Prototype evaluation

To make sure our implementation was as robust and accessible as possible we made sure to include users at every step knowing the importance of getting user input to the delivery of a much superior product than without. In this section I will discuss the variety of ways we tested and evaluated our project with user and with ourselves for very early on in the product life cycle. The finished WebApp we have delivered today is the product of many iterations of the design as we had frequent group meetings to refine the final look of the WebApp.

Our goal for the first round of evaluations was simply to learn and get a better feel for what users expect and prefer from our design. We started off by showing a group of users who are 3 students in DCU our initial sketches of the pages in our WebApp, looking back this was very critical and informative as it gave us a much better idea of the user needs.

- Softer colours
- Better access to the most important functions
- An about section
- Less clutter

These were some of the suggestions on our initial sketches which is directly reflected in the final implementation.

Once we had an idea of the direction of our project we performed a heuristic evaluation of our project and focused on expanding and refining our functions. After this we wasted no

time and included users yet again with a quiz. We received some suggestions which have now been implemented into the final product such as;

- Calendar
- Forgot password facility
- Contact page to reach out to us

After we gathered as much user input as possible we began working on the implementation. After we were happy with the stage it was at we felt it was time to do a heuristic evaluation before involving users. We used the criteria recommended by Jakob Nielsen<sup>1</sup>.

<u><b>Heuristic</b></u>	<u><b>How system complies</b></u>
Visibility of system status.	The system does this adequately fine, there is a live background that moves when a user scrolls in the page, this is not universally visible and somewhere our product can improve.
User control and freedom.	The WebApp is not very dynamic originally, it is intended to be as accessible and straightforward as possible. Therefore user is almost guided from one function to another.
Consistency and standards.	Following on from initial user input we have opted for the easiest experience. This means our implementation is constince in colours and button and placement of objects. Every page follows the same colour scheme and layout.
Error prevention, help users recognize, diagnose and recover from errors.	The users are generally walked through the WebApp so the likelihood of them causing unexpected errors is slim due to the simple nature of the implementation. There is nothing that will loop endlessly or anything of the sort. The user can always back out of any error using web browser navigation or simply exiting the website. Due to the time constraint we haven't done extensive error testing and something we would do given more time.
Recognition rather than recall.	While the system is very self explanatory

	and very little for the user to learn. However the method in which a meeting is scheduled may require learning.
Flexibility and efficiency of use.	The WebApp is extremely efficient with its features with very little clutter due to this there isn't very much flexibility. The only place flexibility is present is the scheduling of the meetings as they are unique and personalised by the user student/lecturer.
Aesthetic and minimalist design.	Given the time and resources constraints we felt as a group that a minimalist approach was best, however in my opinion this approach should be adopted in any project as it is the best way to appease the users and reduce bugs and errors. Our WebApp uses soft colours and few buttons to achieve its goals creating a very aesthetic website with minimalist design.
Help and documentation.	There is an about section in the website that gives a brief synopsis of the functions. It is not in the scope of this project however given more time a fully fledged about page would be included to allow any user to easily understand our website.

Finally after we had gathered as much info and testing as time permits. We decided to leave the final wrap up of our project to a group of three peers using the Empirical method of evaluation we set up a session where we would observe the chosen users navigating and using our WebApp. This was extremely valuable as we could observe their train of thought and the sections they had difficulty with the most. We observed the users navigating the website and tried our best to pay attention to their reactions. This was extremely valuable and we were able to make the following deductions;

- Larger buttons and text as they are easier to see
- Clearer 'meeting scheduling' page as the users were getting stuck and confused on how to schedule a meeting.
- A clearer about section so that they would better understand the need for the WebApp.
- A small description of the meetings so that they may remember why they scheduled them.

To conclude the evaluation of our project, we all strongly believe that involving the user from the first step made our job much easier. It resulted in a clear and refined website that is easy to use and navigate. Given more time we could have used more tried and tested methods to get more user input and testing. However we are very happy with the final product which is a direct reflection of the advice and input we received from our users.

*'Sources : <https://www.nngroup.com/articles/ten-usability-heuristics/>*

## Results and implications of prototype and tests.

After conducting our user tests we received a variety of useful information in regards to our websites current state and what it should improve on. Users mentioned that they appreciated the features we had already such as a neat navigation bar, a working login and a nice color scheme.

Some of the users had some interesting suggestions that we ended up implementing in our final version. One of the users had made a pseudo account but after logging out forgot their password. He suggested that it is extremely important for us to have a facility to retrieve lost passwords and so we ended up implementing that feature immediately after as we realised the importance of it.

The production of our idea has resulted in a functional and accessible web app that speeds up and smoothenes the process of lecturer-student communications.

The following are the results of our implementation of accessibility features:

1. Accessible colors used on website: Through research we discovered that a major requirement for an accessible website is that it's content is readable by people who might not be able to see all colours on the spectrum(e.g color blind people). To do this we decided to use a simple black and white color scheme. While this may not be the most flashy or creative color scheme, choosing such a scheme allows us to focus more on the functionality side of our website which is our main priority.
2. Structuring content: The second feature implemented to make our content more readable was structuring our pages, as good structure makes a lot of difference in user experience. Good structure means easier navigation and so we did this through the use of H1 headers for our page titles.
3. Another vital feature that our website has enabled is keyboard accessibility. Our site can be used via keyboard only. This is important for a number of reasons, such as the fact that

many assistive technologies use only the keyboard for navigation. An example of this in our webpage is using the tab key to jump between login fields.

4. After noticing that we had a few images in our design, we quickly realised another feature we needed to be implemented. This was the use of alt-text attributes. Our alt-text attributes provide a description of images for people using screen-readers.

Implications of the results of our project:

Students now have an easily accessible platform for scheduling meetings with a lecturer, emphasis on the accessibility. On top of email being a slow and inefficient process of organising meetings in general, it also lacks the depth of accessibility features that our website offers.

One major implication will be the improvement of quality of year long projects that require frequent meetings with particular lecturers. A specific example of this would be 3rd year computing students working on their third year project. They now have the ability to schedule meetings without delay and work around those time schedules much easier. Also, this will result in less stress when students are in desperate need from some guidance on some issue and don't have to wait in anticipation for hours or even days for a response from a lecturer.

Results of testing our code:

To test our code we used the w3 markup validation service and looked for areas in our design that may be faulted or just lacking the optimal implementation. A couple of issues found were:

1. Unclosed div class syntax errors.
2. Lack of alt attributes attached to some image files.

These issues were noted and resolved soon after discovery.

## How We Would Fix Things

Due to time constraints, we didn't have the opportunity to go back and fix the webApp. Due to this, we would have changed the following:

- **User scroll:** The automatic scroll is very beneficial and helpful. However, it's bad for those users who are visually impaired. It doesn't exactly tell the user where they are

on the page when the scroll has stopped. To improve on this design, we could make an about and contact page.

- **Links:** The name links within the meeting requests (in the lecturer Dashboard) and organising a meeting (in the student dashboard), could be more visible in order to tell that it's a link rather than a list of names.
- **Login and organising a meeting form:** Implementing a login form that actually works by using a username and password would improve the security and it'll allow a user to access their specific dashboard based on their name.
- **Description:** A meeting description would have been really beneficial if implemented in order to remind both the student and the lecturer of what the meeting is about.
- **Search bar:** implement a search bar that actually links the user to a specific lecturer/student names and gives you a list of recent meetings you had with that user.
- **Tutorial:** Implement a tutorial that shows both users how to iterate over the site.

In conclusion to this, more user input and testing would have been beneficial however, from the results we gathered, users really liked the styling, how clean the webApp is and how easy it is to retrieve what they're looking for by just a few clicks.