**CSC3721 – Coursework2**

**B9052919 Ryota Fuwa**

1. Peer Evaluation

We used cooperative evaluation. And as evaluation criteria, we tried to asses other’s prototype based on 10 usability heuristics for user interface design [1]. The observer briefly leads the evaluator to provided functionalities, and the evaluator interacts with the prototype and give feedbacks based on the 10 usability design guides.

He pointed out that my prototype didn’t provide ways for users to recover from errors when error messages popped up. After this, I listed suggestions of what users can do to deal with the situations in the error popups. And also, I forgot to include help documents on the website which users can refer to when they don’t know what to do, which should never happen though.

he suggested that I include a calendar with which users can set a date for each lecture. I didn’t have any functionality about setting dates for lectures even though I divided curriculum resources into each lecture. He found that little inconvenience.

He also pointed out to me that it was inconsistence that users had to click “Back” when they finish editing (figure1 Left).

A screenshot of a social media post

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Figure1 Editing Page (Right Before, Left After)

I installed another button on top that says “Done” to circumvent this inconsistency even though both of them do the same thing (figure1 Right).

Moreover, He also recommended that I have a notification functionality. When someone responded to the questions a user posted, it is helpful if the website send notification to the registered email address. This functionality can also be applied to the calendar. For instance, if the lecture will be hold tomorrow, the website notifies it so that users (teachers) can prepare for it well beforehand.

While I was working on my prototype, I paid attention to usability designs from practical 5. However, he found out some inconsistencies and inconvenient functionalities in my prototype. I realized that evaluations from other perspective is really important in design process.

1. Design Process (1500)
   1. Representation of Curriculum Resources

* **Design Space**
* **Just-a-folder representation**. User can freely organize the structure and contents by creating folders inside and adding/deleting files.
* **Semi-structured representation**. User has to include required file such as document for lesson plan, yet user still have an option to add resource they want.
* **Fixed-structured representation**. User has no options to add another resource although they still can modify each pre-defined file.
* **Design Criteria**

Internal Sources:

* The usage of the resources is helping users(teachers) to run Micro:bit curriculum.
* The resource should be able to be modified according to teachers’ needs.

External Sources:

* Flexibility and efficiency of use ([1] #7)
* Aesthetic and minimalist design ([1] #8)
* Exploit the power of constraints ([2] #5)
* **Final Design**

I chose semi-structured representation as the final design. Curriculum resources don’t have to be versatile since the objective of the resources is only teaching guideline and resource management for Micro:bit. Therefore, rather than making the representation unnecessarily flexible, giving structure makes it better since users can understand easily and also organize the resources more simply. However, if the available resource were completely fixed, users might not be able to fill their needs, which is also not desirable.

I divided resources into each lecture so that teachers can organize and modify it for each lecture (figure2), otherwise it would be a huge effort to download all resources only for one lecture.

Moreover, I made lecture plan documents required. All versions have to include them for each lecture. This makes sure that users can find lecture plans in all versions. This might sound demanding since users have to include a plan document for each lecture, but almost all the time lecture plan is necessary in order to run a curriculum well. Furthermore, users don’t have to create one by themselves even if they don’t need them because all versions come with default lecture plan documents. if that’s the case, users can just ignore them, and let them alone.

A screenshot of a cell phone

Description automatically generated

Figure 2 Home Page

Figure2 also demonstrates that the users can modify the content of lecture documents online. This is possible since the lecture document is required, and file format is fixed.

As to the web page design, I tried to make is as simple as possible by using straightforward icons instead of letters. Also, I used Drop-down lists for Other Materials so that users don’t have to deal with a lot of information unless they need them (figure3).

A screenshot of a cell phone

Description automatically generated

Figure3 Drop-down list

I’m sure that users would like to add files other than lecture plan documents such as lecture slides. It can be archived only by clicking plus and minus buttons. Those are quite straightforward, so users can easily organize additional materials by themselves. There are no constraints about those additional materials: file formats, sizes, etc.

* 1. Version Control
* **Design Space**
* **GitHub-Like Version Control.** Public Repositories working with Git like GitHub, so users develop their versions locally using Git, and also synchronize it with public repo by pushing. Other users can clone the repo, modify it, and update the public repo. It is okay for others to update the repo since it can have different branches.
* **Just a Pool of Versions.** Just database of versions. So, users can download others’ versions, but users can update only their own versions, so there are no complexities and no fear that others update your versions.
* **Tree Structured Version Control**. A new version has to be derived from an existing version. A new version takes it as base version. In other words, when a new version is created, the selected version will be copied, so the new version starts as the exact same as the selected version.
* **Design Criteria**

Internal Sources:

* Users can contribute their versions of the curriculum, and others can make the most of them.
* This website serves not only for IT experts, so probably some might not have any IT knowledge and experiences.

External Sources:

* Exploit the power of constraints ([2] #5)
* Recognition rather than recall ([1] #6)
* **Final Design**

I decided that Tree Structured Version Control is the best suited for this website. First, it is easy to use. Users don’t have to know anything about how version control works since what users do about it is just to create their own version based on another one. Users don’t even have to know tree structure behind the interface (figure4 [3]).

A drawing of a face

Description automatically generated

Figure4 Tree Structure [3]

Users can’t go back to old state of their version unlike Git. However, what users will do is not assumed to be serious application developments. Instead, they will just modify lecture plan documents, which can be done online pretty much the same as Microsoft Office Word works and will add or delete additional resources. Nevertheless, if all versions are organized with database, which doesn’t provide any inheritances between them, it will be hard to take advantage of other versions from others. Also, it is the most likely that lots of versions will be similar to others considering the curriculum is always about Micro:bit even though the details differ from one another. Thus, creating new version is not ideal.

* 1. Q&A Forum Evaluation
* **Design Space**
* **Thumbs up and down**. Like YouTube videos, users can say good or bad to a response.
* **Rating on the scale of 5**.
* **Turn a bulb on and off.** When a user thinks a response is sufficient, he can turn on the bulb, and others can turn it off again if they are not satisfied with it (figure 5).

A screenshot of a cell phone

Description automatically generated

Figure 5 Bulb Evaluating System

* **Design Criteria**

Internal Source:

* This forum is intended to help users solve problems with curriculum, **NOT only the place they can** **refer to.**

External Source:

* Visibility of system status ([1] #1)
* Match Between system and the real world ([1] #2)
* Aesthetic and Minimalist design ([1] #8)
* **Final Design**

I chose **Turn a bulb on and off Evaluation.** Based on design criteria, this forum should not only be the place where users can ask questions, and others answer to them. The primary point of this forum is that the users can **solve** their problems and questions related to the Micro:bit project. Therefore, Rating system is not appropriate here. It is helpful to know which response is the most helpful, but it doesn’t tell users whether the response solves the questions or not. The same argument is applied to Thumbs Up Evaluation. I assume a lot of users will have same or similar questions after some time. Imagine that a user found the same topic he wanted to ask was already posted in the forum, but only thing he could see were a number or a count of thumbs up. I’m sure that wouldn’t help him. On the other hand, if he sees a bulb lighted up, he can easily guess that someone answered to the question quite well. Also, since the state will be either on or off, the website can automatically sort the order of questions, and put unsolved ones to the top.

Everyone can turn on and off the bulb for each question even if they are not the very person who posted the question. The answer that was sufficient to some might not be to others. In that case, they can just turn off the bulb, and rise a question to ask a different explanation.

Reference

[1] Nielsen Norman Group. (2019). *10 Heuristics for User Interface Design: Article by Jakob Nielsen*. [online] Available at: https://www.nngroup.com/articles/ten-usability-heuristics/ [Accessed 12 Dec. 2019].

[2] Sites.google.com. (2019). *Norman’s 7 Principles - HCI-06129*. [online] Available at: https://sites.google.com/a/nu.edu.pk/hci-060129/lectures-1/norman-s-7-principles [Accessed 12 Dec. 2019].

[3] Encrypted-tbn0.gstatic.com. (2019). [online] Available at: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRmBnCncJc2gaZISH0oYvYCfwNlu0jJfU1nzyytQFS9\_T0vR1DWHg&s [Accessed 12 Dec. 2019].

Appendix

|  |  |  |
| --- | --- | --- |
|  | Description | Rational |
| F1 | Organizable lecture by lecture (i.e. Curriculum are divided into each lecture) | Most of the users use this website for teaching in school, so it would be much easier to organize stuff for each lecture than everything as an entire curriculum |
| F2 | Download course schedule and all other resources for the curriculum as a zip file with one click | For older or non-IT specialist teachers, who are not quite comfortable with lots of information on a website, it’s much easier if all resources they need are downloaded with just one click. |
| F3 | Able to edit lecture plan documents online without downloading and uploading. | Users are supposed to make small changes to their teaching schedule a lot according to how it goes, so downloading and uploading for modifying contents are not desirable. |
| F4 | Q&A forum where users can ask questions related to the curriculum. | If they don’t know something about the curriculum, they can ask anytime they want. The users can also be the first time in the curriculum and this website, so they might have questions and would like to have a place to solve it. |
| F5 | Response to Questions with resources | If the question is fairy complicated or difficult concepts, other users might want to attach extra resources to make explanations easier. |
| F6 | Search Q&A by user’s names, updated dates and keywords | The Q&A forum becomes big if the websites are used for a while, which makes finding a specific question by scrolling unbearable. |
| F7 | Link to tutorials on YouTube | To make sure new teachers for the curriculum don’t have to be IT expert, study IT from scratch or read manuals for all tools and devices used in the project. |
| F8 | Search Tutorials by topics or keywords | Users might be completely a newcomer in IT subject, so they might not know even what they’ll have to teach in their school before they watch tutorials. Perhaps they only know keywords. |
| F9 | Able to upload Materials of any types | Different users might want to use resources whose type is not Word or PowerPoint. |
| F10 | Able to find different versions with ease | Stored resources should be organized in a way that users can find their suitable version of the course resources with ease. |
| F11 | Calendar allowing users to set date for each lecture | Resources are divided into lectures, so setting date for each lecture helps teachers to run the curriculum smoothly. |
| F12 | Exit button on editing page | Clicking “Back” implies that the editing will be aborted. So it is for consistency. |
| F13 | Notification | User can get notification to their email address. |