KIT305

Assignment3

TUTORIAL APPLICATION IOS

Wai Yeung Chan
476435

Contents

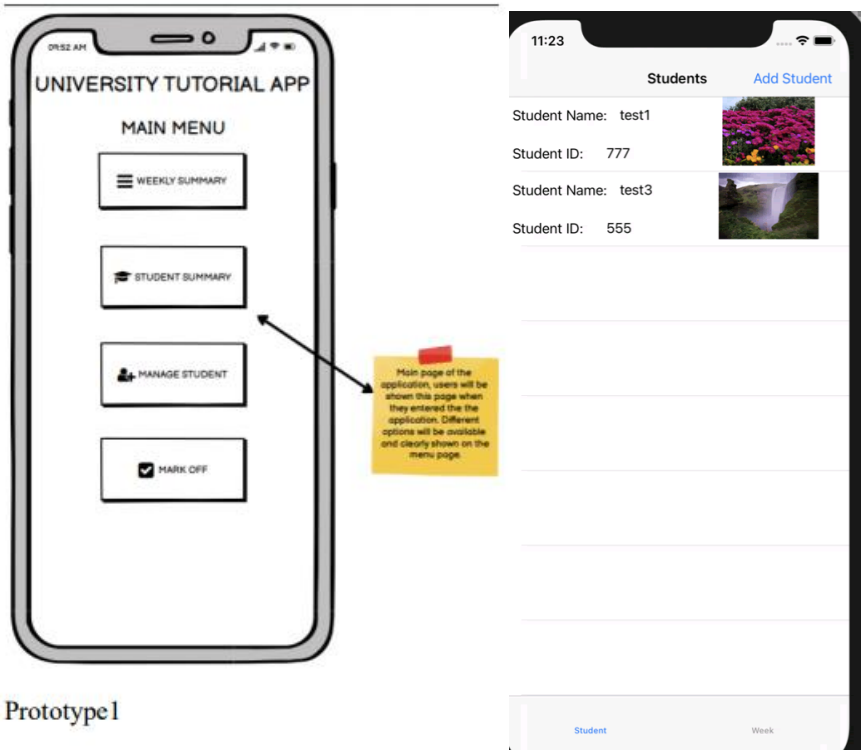
Introduction

A tutorial is one of the most important elements in university life. It helps us to make use of the knowledge we learned in the lecture and put them into practice. University encourages and monitors the attendance of the tutorial so the students will be able to consolidate their knowledge. In the ICT courses, tutorials include hands-on practical on the programming application, discussion among the students and small group consultation with the tutor. Tutorial attendance and marking are not united among the tutors, some of them are just using handwriting on a paper and some of them would have an excel recording all the information. Sometimes the tutors may even lose or forgot their attendance sheet or marking sheet that causes many troubles for the schools and students. To have a better working environment for the tutors and allow the school to record the marks that students attained in a tutorial, a united method of recording marks and tutorial attendance is necessary to reduce the possible mistake and potential problems happen in the tutorial. As iPhone is the company that owns most of the market share on the mobile phone market, other than the android application developed previously, an IOS based application is also developed with the use of XCode to facilitate the need of the market.

Differences to Assignment 1 Prototype

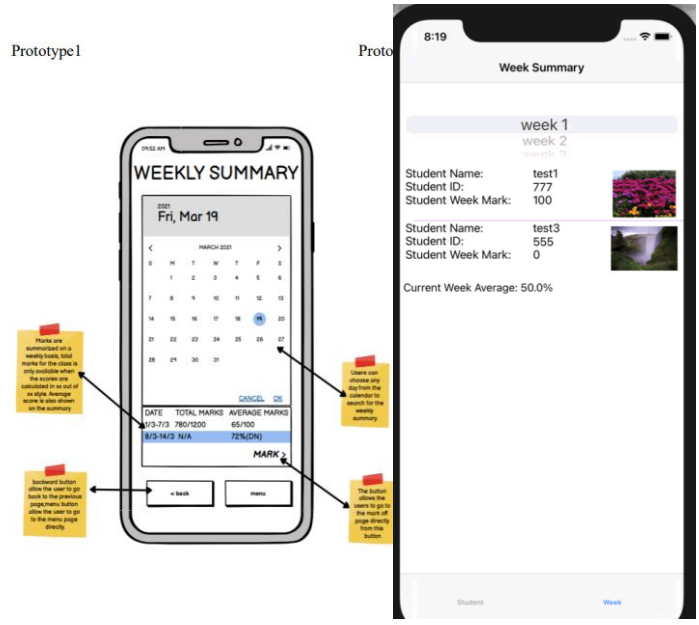
In the assignment 1 prototype, the interface of the application was divided into four, which is a weekly summary, student summary, manage student and mark-off.

The main menu page from prototype 1 is now replaced with the tap menu to increase the efficiency of the program. There are two tabs available including the student and the week which implies the student summary page and the week summary page.

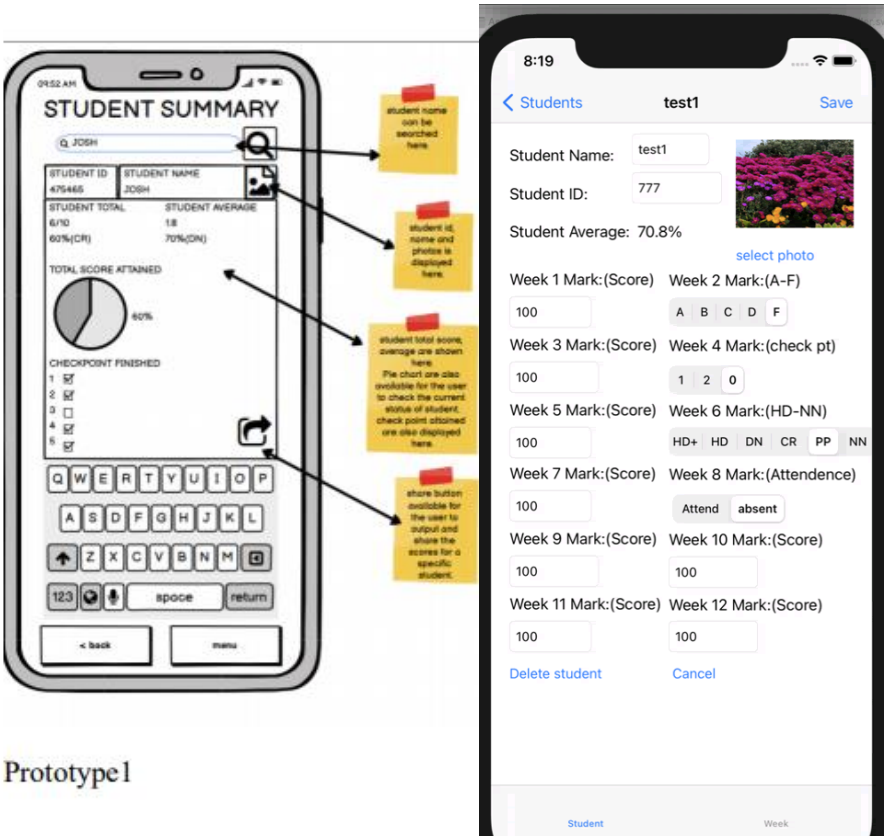


Prototype1

The weekly summary has changed from calendar picker to a table list working with the UIPicker in XCode to select the week detail the user want and display the student's mark and the weekly average of that week. The week average indicating the average mark the class obtained at certain week is also shown at the bottom of the week summary page.



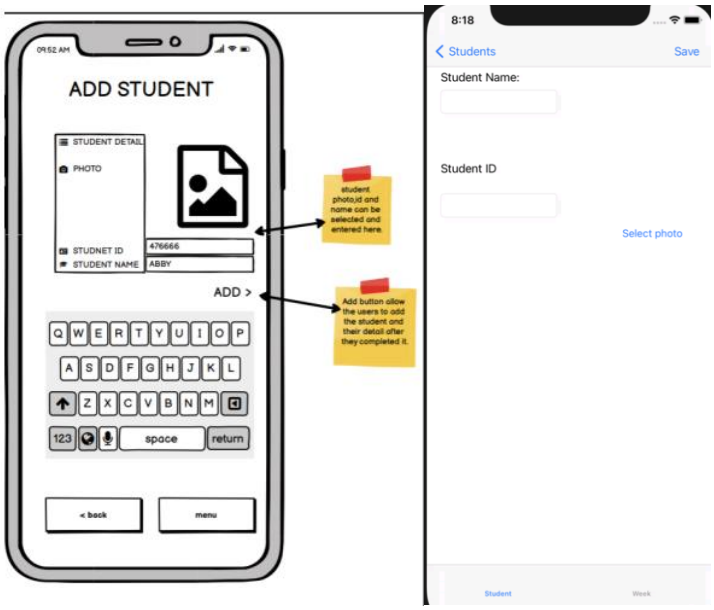
The student detail page is very similar to the prototype 1 page. They all include the photo selection, edition of student id and student name. The only difference is it is now combined with the weekly Markoff page to increase efficiency and reduce the need for directing to a different page. Student Average will also be shown on the student detail page.



Prototype1

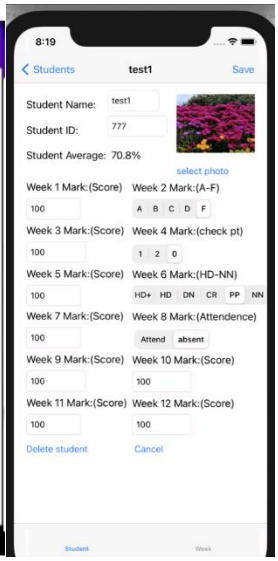
The add student page is the same as the prototype one. Users are required to type the student name, student id to add the student. They can also choose to select a photo from their photo library for the specific student. After they selected the photo the photo will be displayed on the top right-hand corner of the page and after they pressed the save button the student detail will be saved to the database and the photo will be uploaded to the storage.

Prototype1



For the Mark-Off page, instead of using the drop-down list to mark the student like prototype 1 and prototype 2, segmented control is used for the marking to display the options to the user clearly and more efficiently.

Prototype1



Usability Goals and Design Principles Self-Critique

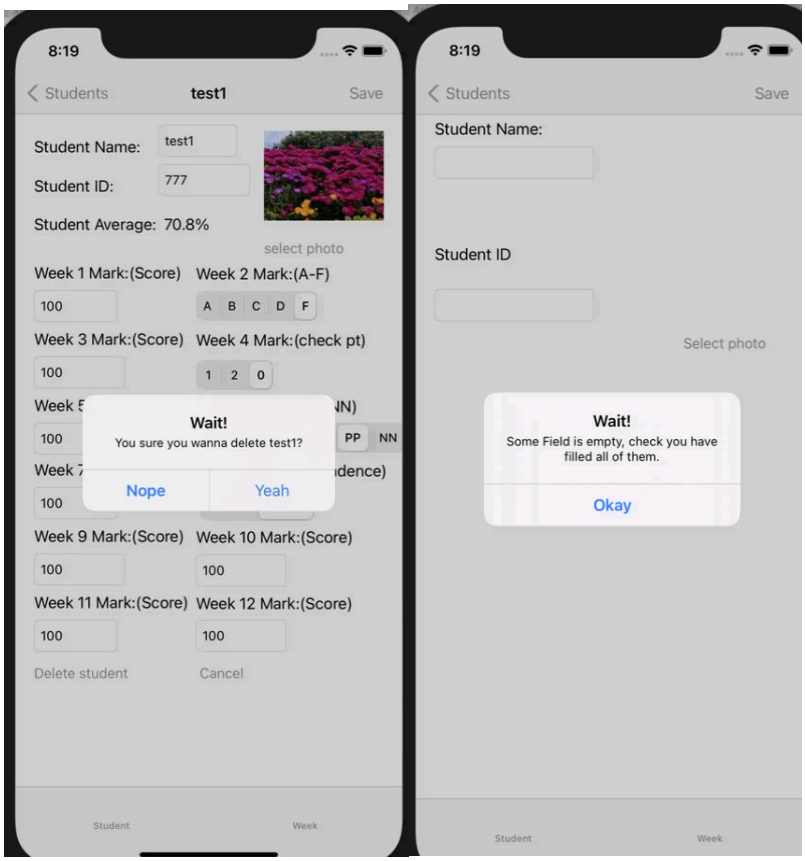
According to the Kit305 week5 lecture, there are 6 key aspects to a usable system which is called usability goal. They are Learnable on first use, memorable on repeat uses, efficient, failure-resistant, forgiving and satisfying. Prototype 2 aims to provide a working and usable application ready for the tutor to use. All the aspect needs to be fulfilled to provide a usable application to the user.

To make the system learnable on first use, the options available for the users are displayed with the description with text to show the users all the possible actions. For example, select photo, save, delete, and add student functions. It allows the user to understand the function of each button without testing.

To make the system memorable on repeat uses, the function of the application is now combined into a small amount of storyboard so that they can remember it easily. We have deleted the menu page from prototype 2 and combined the student summary and week summary into one page using the tab function. The marking function of the application is also combined with the student detail page to reduce the pages and redirection needed so it is easily memorable.

To make the system efficient, we have achieved the first two aspects of a usable system which is learnable on first use and memorable on repeat uses. With the two achievements, the system has reduced the operation required from the user and the simpler operation required. We have also reduced the use of the redirection and combined the functions into the pages to increase efficiency. A segmented Controller is used to allow the tutor to mark the student on that week with a single tap.

To make the system failure-resistant, alert dialogue is also used like prototype 2. Other than the user will be popped with the alert when they are trying to delete the student, they will also be popped an alert if they try to add the student leaving some of the field empty(except photo).



To make the system forgiving, failure-resistant is used to prevent the user from improper use. Also, every single page of the system is providing a cancel or back button except the main page so that the user can go back to the previous page whenever they want so to prevent improper use.

For Don Norman's Design Principles, there are also six main aspects for designing an application, including visibility, feedback, affordance, mapping, constraints, and consistency. To improve the usability of the application, the application is designed to achieve most of the principles.

To improve visibility, the system is designed with all the buttons shown with description so that the potential operation will be displayed to the user to avoid misunderstanding. The two main pages which is the weekly summary and student summary page are displayed with the tab function on the bottom to make the options more visible.

To improve the feedback performance, after the user added the photo to the student, a photo will appear directly above the select photo button. When the user changes the selected week in week summary, an update will automatically appear with the student list so the user will know what action has been done.

To improve affordance, we made our function the same as the perceived image of an object. For example, the button under the photo would be the add photo button. The save button will always on the right, these perceived images of an application will improve the affordance.

To improve the constraints, the marks with a score out of 100 is entered using the number pad instead of the normal keyboard to avoid the user type in a string to make the system crash. Alert for empty entry and deleting student also helps to improve the constraints to prevent the system from crashing. Some of the field like week 2, 4,6 and 8 are using segmented control so the user will not have a chance to type in a string which is reducing the chance of typo to make the system crash.

To improve the consistency of the program, the layout and style of the whole program are all the same with which background and black font. All the data from our database is the same and displayed the same to make sure the persistence of the database data and consistency of the program.

Code Documentation

The above IOS application is built and designed for the large screen device and tested with iPhone 11 simulator.

For the assignment 3 prototypes, there are 7 main classes used to make the system work.

1&2: StudentUITableViewCell and StudentUITableViewController->these two classes is used for storing and displaying the student list in the class for the tutor and responsible for the student summary page.

3&4: WeekSummaryTableViewCell and WeekSummaryTableViewController->these two classes is used for storing and displaying the student list and their mark on the selected week and responsible for the week summary page.

5:Student->> The collection class contains the database structure made for the whole application to interact with the database.

6: addStudentViewController-> this class is responsible for the add student page and allow the user to add the student after they have entered the student's name and student ID with the tap of the save button.

7:detailViewController-> this class is responsible for the student detail page and allow the user to edit the student detail including the student mark and save it to the database. It also provides the delete function for the user to delete the student from the database.

Most of the function in the application is constructed with the code from KIT305 lecture and tutorial slides, some of the function is constructed with the code from Firebase. The image upload function is constructed with the code from a YouTube channel Named IOS Academy. (Swift: Upload Photos to Firebase Storage (and Download, Swift 5) - Xcode 11 - 2020, 2020) The image retrieves function is constructed with the code from StackOverflow. (Swift, Dabus and Eduardo, 2015)

Conclusion

In the progress of working on assignment 3, I found the android studio and XCode way of interacting with the firebase database is very similar when we are using swift for XCode. I am proud of my achievement of making an efficient program for the use of tutorial in UTAS. However, there is more I can achieve which is sharing the function of the application and the function of changing the making scheme for a specific week. All the thing I learnt from this assignment is very useful and hopefully will be used in my future career. The next application I am going to make is the flutter program which is a cross-platform application and I hope I will be able to accomplish my target by achieving the sharing function and changing the marking scheme for a specific week. Looking forward to work on the next application.

Reference

Youtube.com. 2020. *Swift: Upload Photos to Firebase Storage (and Download, Swift 5) - Xcode 11 - 2020*. [online] Available at: <https://www.youtube.com/watch?v=TAF6cPZxmmI> [Accessed 21 May 2021].

Swift, L., Dabus, L. and Eduardo, L., 2015. *Loading/Downloading image from URL on Swift*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/24231680/loading-downloading-image-from-url-on-swift> [Accessed 21 May 2021].