Note

My app was tested and developed with the Pixel 3a API 30 emulator

Introduction

Keeping grades consolidated is not easy when there isn't on solution that ticks all the boxes. This leads to tutors using all sorts of different methods for keeping track of attendance, whether that is the convenience of just writing on paper but the downside of having to be digitised later, or using mylo with grade being entered electronically from the start this has the big benefits of consistent formatting and having all the grades in a central locations, but with it's clunky UI it isn't worth for some. So, how do we take grades in a way that's convenient, but also consistent and consolidated? Well, an app of course!

For this app to be a viable option for tutors they need to be able to enter and remove students, quickly and easily mark the students, be able to change what type of mark they are storing in each week and be able to share this data to other applications. So in this report I will detail my implementation of just such an app on the android platform.

Differences from Prototype

Overall, my final product is fairly close to my prototype, the layout for all the screens that were kept are almost identical to the prototype and anything that was removed is explained below.

One of the biggest differences between my prototype and the final application is that viewing a grade does not take you to a separate view/activity, instead just displaying the grade on the WeekDetails or StudentDetails activity. I did this mainly for efficiency as it means multiple grades can be changed very quickly instead of having to wait for a different activity or modal to load and having to go back for each grade you want to change. Another big difference is that the configure week screen was instead implemented as a much simpler modal over the WeekDetails activity instead of a separate activity. This was due to massively simplify how a week can be configured. In my prototype weeks could have multiple different types of marking schemes in the one week, whereas in the final implementation it was specified that each week only need one type of mark, so I just made it a simple modal instead of a whole other activity. The last big change was removing the starting screen in the prototype that could take you to the current week. I didn't implement this as I wasn't sure how to implement with the way I structured my app, and it could potentially just annoy users.

Apart from these changes rather large changes I made one smaller change with the layout of the StudentDetails activity. I changed this screen to have the picture of the student next to the student's name, student ID and grade average so that the students' details take up less vertical space so more weeks can be viewed without scrolling.

Usability Goals and Design Principles

My app was built with efficiency in mind. This is best shown in the StudentDetails and WeekDetails activities where all the grades are present straight from this screen as opposed to having to go to a separate page for each grade so multiple grades can be entered rapidly both for the one student in StudentDetails or for multiple students in WeekDetails. This does reduce the forgiveness of the app though as once a grade has been changed there is no way to undo this save for just re-entering the old grade, and it could be quite easy to accidently enter a grade. To try and help with this a made every grade change pop-up a toast to show you what the grade was and what it is now.

I also tried to make the app more visible and add in some mapping (MapInTime, 2013) with the over all style of the app. All the of buttons in the app are clearly label and can be seen straight away without cluttering up the UI. The tabs at the open of StudentFragment and WeekFragment help to make a clear distinction between the two fragments as well as make the user aware about the two separate browsing methods. I also tried to build on people expectations to make things easier to learn by making all the delete buttons red to coincide with the use of these colours on more dangerous actions as well as adding the floating action button to the StudentFragment with a plus symbol on it to help indicate to the user it's functionality.

I made the app more failure-resistant by making sure all the text fields are the right type and bring up the appropriate keyboard like on the score grades where they bring up the number pad and only except numbers as input. I also did similar things in the AddStudentModal where the modal cannot be submitted without entering a first name, last name and student ID which is 6 digits long. The student ID field much like the score fields for the score grade type only excepts numbers as an input and brings up the number pad instead of the traditional keyboard.

I made sure to keep my apps layout consistent with the general layout being details at the top of the screen with a list in the middle and the buttons at the bottom. I also keep buttons that do the same thing in different activities in the same place to help the users muscle memory not make them click the wrong button, This is best show with the share button in the StudentsDetails and WeeksDetails where the share button is on the left for both of the layouts.

To help with the visibility of the app and learnability for users who are used to Android I tried to stick to using material design (Android Developers, 2020) such as the app bar on all the screens and the floating action button on the StudentFragment to add students.

I introduced some forgiveness to the application by making the button to delete a student launch a pop-up to make the user confirm they want to delete this student to prevent misclicks from deleting students. In a similar vein the AddStudentModal can just me clicked off if this is accidently pressed.

I also introduced a lot of feedback into the app in the way of toasts pop-up after all actions that don't obviously change something. This is done in multiple places like when you are changing a grade in StudentDetails and WeekDetails, when you add or delete a student or when in the AddStudentModal you take a picture the add student button is disable while the image is uploading and there is also a toast informing the user when the upload has started and when it finishes.

I added constraits to the app in the form of the gradeList in StudentDetails and WeekDetails only display the type of grade that is configured for that week, and even more so for the score weeks as they make sure the user can only enter numbers and makes sure the number is less than or equal to the max score, changing a grade to big to the maximum score. As previously stated something similar is done in the AddStudentModal when the student ID has to be a 6 digit number.

I feel like in this app I had to make a lot of trade offs in the forgiveness of the app when chasing efficiency, as the most common way in my mind to add forgiveness is to add pop-ups to confirm important actions which I avoided in my StudentDetails and WeeksDetails pages so that grades could be change quicker. In the future I could potentially use toasts with undo buttons (which I honestly forgot existed until I was writing this report) like Gmail does as this allows me to be both efficient and forgiving. I also feel that users using this app for the first time may not find the app super intuitive so in a future version I would think of implementing a tutorial for the app on first time boot up like a lot of apps do these days, this would allow no changes to the UI while still telling the user how to use the app properly.

Code Documentation

MainActivity – The base activity that starts up the tabview and stores the global data like the student, each week's configuration, a list of all the weeks and other miscellaneous data that is needed globally.

SectionsPagerAdapter – This class handles starting up the different tabs when they are selected.

StudentsFragement – This is a fragment used in the main tab view to list all the students in the database and when a student is clicked on, it takes you to that students detail page.

StudentDetails. – This displays all the student's details (name, id, grade average, image and grades). Grades can be changed here as well as deleting the student and sharing all their grades in a csv format through the built-in share functionality of android.

WeeksFragment – This is the other fragment used in the main tab view, it lets the user select a week to see every student's grade for.

WeekDetails – This displays the grade for every student for the specified week. It also lets you change these grades, reconfigure what marking scheme that week has and share all the student's grade from that week in a csv format through the built-in share functionality of android.

AddStudentModal – This modal lets the user enter a new student to the database by specifying their first name, last name, student ID and upload a picture through the built-in camera functionality of android.

Grades – A data class used to initialise a new student's grades for the database.

Student – A data class that contains all of a student's details (first name, last name and student ID) as well as the name of their picture in the database to be fetch in the StudentDetails activity.

ConfigureWeekModal – This modal is used to change the type of grade in the week it was called from.

Nearly all of my code was sourced either from the tutorials (tutorial 5 mostly), the discord or android's own documentation. The code for the delete student modal came from https://www.journaldev.com/309/android-alert-dialog-using-kotlin, the code for the base of both the modals in the app came from https://blog.mindorks.com/implementing-dialog-fragment-in-android and the code to pass data from the WeekDetails activity to the ConfigureWeekModal and back came from https://code.luasoftware.com/tutorials/android/android-pass-argument-to-dialogfragment/ .

Conclusion

To round things out, I have made a faithful implementation of my prototype in an android application with minimal changes to the design. Like my prototype my android app was an efficient first design, making sure the most common tasks can be done with as little clicks as possible, while still staying intuitive. While I am happy with the outcome, I definitely learn how much of a balancing act it can be to try and make the app as easy to understand as possible while still being quick and minimal. There are definitely some features I missed and some bugs left unsquashed, but overall I am very happy with what I have managed.

References

1. Android Developers. 2020. Material Design for Android. [online] Available at: https://developer.android.com/guide/topics/ui/look-and-feel/ [Accessed 25 Apr. 2021].
2. MapInTime. (2013). Don Norman’s Design Principles. [online] Available at: https://williamgrimes12.wordpress.com/2013/01/22/don-normans-design-principles/ [Accessed 25 Apr. 2021].