1). Most of you are aware that the assessment allocation for this course has changed slightly from the outline and was agreed upon by all students that have been coming to class. It has changed in that instead of 2 labs worth 15% each, in this session we had 4 labs worth 12.5% each, for a total of 50% of the course mark. Most of you have completed all 4 of the labs and have done very well. If you did not hand in a lab then you got a 0% for that lab.

2). The assessment allocation for the project is still the remaining 50% of your course mark, made up of 30% for the handed in source code via moodle, and 20% for the presentation of your project to other class members in the last week. I would like to clarify this with several potential scenarios:

* If you do not participate in a presentation, then the most you can get for the project mark is 30%.
* If you or your group do not hand in via moodle, a zipped file containing the “lib” directory of your project source code, you will not get any of the 30% for the handed in code portion of the project mark.
* This means that if you do not participate in a presentation and do not hand in your code, you will end up with a project mark of 0%. Most students will NOT fall into this category, but I feel that I must clarify this, just so we are all clear on the consequences of not doing a project!!
* If you do participate in a presentation and hand in your source code, you will be guaranteed a project mark of between 70% and 90%, depending on my subjective ranking of your groups project, relative to the other projects in this class. This means that a very good overall mark is instore, for those students that have participated in the labs and project.

3). The criteria that I will use to differentiate or rank the projects are as follows. The more of these bullets that are included in your project, the higher your project will be ranked, relative to a project that does not include as much (unless you are programming a game, then see the last bullet).

* Use of a list of objects, and a class to define the objects.
* Use of the listname.map function to convert the list of objects into a gridView/listView/ListTile widget.
* Use of some form of Navigation between a minimum of 2 screens, either with “unnamed routes”, “named routes”, or a ”tabView controller”.
* Use of State Management with either the “scoped\_model” package, or “stateful widgets/setState”.
* Use of the “http.get” package to retrieve information from the web, or any mechanism to get info from the cloud. This implies that the JSON returned from the http.get request, must be parsed.
* Use of some form of animation using an animation controller.
* Use of local persistence, using the “sqflite” package.
* Use of cloud persistence using AWS or Firebase would be a bonus, even though we ran out of time to cover this topic in class.
* Use of “Firebase Authentication” would be a bonus, even though we ran out of time to cover this topic in class.
* Use of “google maps” would be a bonus, even though we ran out of time to cover this topic in class.
* If your project involves a game of some sort, then you do NOT have to do any of the above, except what is necessary to make the game an enjoyable user experience. By this I mean that you will have to use some kind of game engine integrated with flutter.