**PRJ666 – List of project changes and deviations from PRJ566**

**Group: 1**

**1. Removal of User feature**

In the proposal we had the idea allow for the creation, deletion and editing of multiple user profiles. The intention was so that multiple users could use the app, each with a different set of saved data (scores). This feature was partially implemented and ultimately removed for a few reasons.

The first reason is that the feature was a “nice to have” but not an essential part of the user experience. It was a low priority compared to the more essential tasks and features that needed to be implemented.

Another reason is that it wasn’t necessary and proved to be a very rare and unlikely use case. In nearly all situations, the person using an app on a phone is the owner of the phone and the only user of the device. Having multiple users for a local app on a phone seemed pointless.

**2. Changed Periodic table game to Memory game**

The original 5 games were decided by the team in PRJ566. As the majority of the team didn’t continue on with the project into PRJ666, we decided to review the 5 game ideas with the new team members. The one game that we all decided on dropping or changing was the Periodic Table game.

In its place, we decided upon an implementation of the classic card flipping game “Memory”. This became the 5th game in the list.

**3. Changed analytics graph to table**

Implementing a visual graph to display the user scores and progress proved to be a very complicated endeavor and one that wasn’t straight forward to do in Android Studio. However, we still wanted to display the user scores and progress in a visual way within a specific module / activity in the app.

The workaround for this was to display the data for each of the games in a semi tabular format which allowed for the score data to be viewed all in a single screen at a glance. This proved to be much more straight forward to implement and likely even a better, clearer way of displaying the data than in a graph. This method achieves the same goal as displaying the data in a graph but in a different way.

**4. Removed speed and endurance scoring metric. Replaced with high score.**

There was originally going to be three scoring metrics we would calculate and save: Speed, Endurance and Accuracy. Endurance, which was a measure of how long a user played until they reached a game over state, seemed to be a redundant metric that didn’t really tell the user much about how well they are doing in terms of memory.

Speed, which as a measure of how quickly the user entered correct answers to questions, didn’t seem very useful either as a measure of memory. Additionally, the speed metric also proved to be a convoluted and hard to implement metric as it would need to be calculated slightly differently for each of the 5 games.

The final implementation tracks the users Accuracy, the measure of how many correct answers the user inputs compared how many questions they have answered in total. The other metric is a high score which is a more tangible and useful metric that the user can understand and try to improve.

**5. Removal of settings section / feature**

The settings section was very low priority and not integral to the user experience. The plan was to have a simple settings section where the user could mute in-game sounds and music as well as adjust the volume. As there is only a single game in the app that has any kind of sound and there is no music at all in the app, having these settings wasn’t useful.

**6. Switched from database to local file storage**

After further research into best practices for saving user scores and data for games similar to ours, we learned that using one of Android’s local file storage options was better than implementing an entire local database.

Additionally, it proved to be a move elegant solution because it didn’t require the constant accessing of a local database every time a CRUD operation was needed. Instead, simple comma separated values were saved locally on the device in text files.

Using this method had added benefits such as the fact that these text files aren’t accessible to the user and thus, they couldn’t simply change the values in the text file rendering the saving of data pointless and trivial. Additionally, these files are automatically deleted when the user deletes the app itself.