## 2.3 Usability Features

### Learnability

To help the users learn how to interact with the app quicker, hints will be shown when the user clicks on different buttons. For example, when the user enters an equation that outputs a surd, the app will notify the user of the feature that converts values from surds to fractions or decimals. It also shows the user how to use the feature.

Another way the app will be made easy to learn is through the design of the user interface. The UI will take design features from other well-known calculator apps and real life calculators. This will increase the learnability because the user will find the design intuitive to use.

For example, it will use a “shift” button. This is a commonly used feature on calculators that enable buttons to double up on uses. The buttons can have a default function, and a shift function. Using this makes the app easy to learn because the “shift” button is often used in calculators and so it will be intuitive and not require much time for the user to learn.

### Efficiency

To help the users interact with the app as efficiently as possible, the majority of the actions the user can take are located in the first activity and layout of the app. Therefore for a large percent of the time that the app is being used, all the usage will remain in the first activity and therefore it will be efficient. It will be efficient because the user will not have to trawl through menus to perform their desired action.

Another way the efficiency is improved is by multiple layout options for different device dimensions. For example, a tablet layout will take advantage of the larger amount of space to include more buttons and functionality into the main layout. This means the space on the screen will be used very efficiently to enable the user to perform functions as quick and efficiently as possible

### Memorability

Using the app will be very memorable because it will closely mirror the design of real life scientific calculators. This means that most features will be labelled and have the same annotations as a real calculator. This will help with memorability because the userbase will likely already have experience with scientific calculators. Therefore they will be used to interacting with a similar design and so they will remember how to use it.

### Errors

My app’s layout and functionality will be designed to maximise the ease of use for the user. The most common error would likely be the ones caused by the user entering invalid data that could cause crashes. To prevent these types of errors, I will include a lot of validation for every method that the calculator uses. For example, when the trigonometry method is used, the input will be checked that it is between -1 and 1 for cosine and sine.

If the input is invalid, using it in the method would cause an error that could cause the application to crash or otherwise stop working. Instead, the validation rules can catch it and instead inform the user of the invalid input and prevent it from being used.

Other common errors that the application will encounter will be calculations have a result that is outside the possible range for the application to process. This will occur when the user’s expression results in an output that is too large to either display on the screen or the processor cannot handle the process.

The error of having a result that is too long can be solved by detecting when the error occurs by testing the length of the result and then changing the notation of the result if it’s too long. For example, converting the result from decimal to scientific notation will conserve space and enable the user to read the result with more ease.

The error of the user entering an expression that uses all of the CPU’s processing power which will cause the device to freeze or crash is more likely to occur on older devices, such as the one’s that run on Android 4.0 and older. This error will be much more difficult to prevent because different devices can handle different levels of processing power, therefore the application cannot be programmed to reject specific equations that are known to crash a device.

### Satisfaction

My app will be as satisfying as possible for the user to interact with. I am aiming to make it satisfying by creating the design based around mainstream calculator designs that are already widely used.

For example, I can base my calculator’s layout off the common calculator that students and schools use. This will make my app more satisfying to use because the user will have experience with other calculators that can be transferred over to my own.

Because of this design’s easy accessibility, the potential user base will be as wide as possible. Therefore the solution can be used by any student, teacher, or parent no matter their age or level of computer literacy. This means each user will be satisfied with the solution because the design will be specifically created so it is easily accessible to any user.

I will also carry out a survey targeted at GCSE and A Level students to ensure my design is aesthetically pleasing and can be easily interacted with. This will ensure that my final design is as satisfying as possible.