## The Title

**An android mobile application for a child’s personal and behavioural development**

**Craig A Corstorphine**

**TM470 Computing and IT project**

## The Problem

My daughter has always responded better to being rewarded for good behaviour than being punished for bad behaviour. It is very well documented that a reward system can have great benefits on child behaviour. A document on the CDC website lists the benefits as improved behaviour, increased self-esteem and an improved relationship with the child(CDC 2019).

Rewards can encourage good behavours, increase self esteem and improve the releationship with the child. Although children can respond better to specific types of parenting, I am going to develop a mobile app to help compliment my parenting approach.

It can be hard to keep track of a child’s behaviour and development over the weeks to a month, and an app can help track that. In the past I have used a paper chart and sticker system, but without a reminder or notification it was often forgotten about. The paper chart would also become a mess if mistakes were made and it was harder to amend. A plastic chart involves add and removing hundreds of stickers every week which over a long time can become costly. A mobile application will solve these issues with various features and an easy to edit profile.

Text, whiteboard

Description automatically generated

*The picture above is a sticker chart used used by my family. After a week it takes considerable time to remove all the stickers*

Although this app is to be developed for the benefit of my own family, there is the potential for other parents to download and use the app. Now that Covid19 restrictions are easing It will be easier to test the App with other potential users/parents. I will consider other parents as stakeholders although some alterations may be made to the app before it is made avaible to the public. This would include a more generic layout and look that is not so personal to my family.

## Benefits of solving the problem

There are multiple benefits to solving this problem which include improved child behaviour and reduction of tantrums. This can be through the impact of rewarding good behaviour and making it a habit. A happier child is also likely as there will be a decrease in punishments that may affect self-esteem. The child can also become familiar with a delayed reward system as they earn points and rewards over the short to medium term periods. An example would be a certain reward after a day or week of satisfactory behaviour.

It may also benefit and help with educational development as rewards can be earned for schoolwork and/or grades. In a practicle and logistical sense, money and time can be saved. There is no longer a need to buy stickers and new charts, as well as not having to buy or subscribe to another app.

## 

## Knowledge of problem

My knowledge mostly comes from personal experience with my daughter, and a lot through trial and error. Like most parents we went through the process of reading the recommended books on toddler development. I know that a reward system works for my daughter as I’ve implemented one with sticker charts before. There is also a lot of research done on reward systems for children as well as Gamification. “Gamification is the application of game mechanisms in non-gaming envir0ments(Kalpana Nand, Nilufar Baghaei, John Casey, Bashir Barmada, Farhad Mehdipour & Hai-Ning Liang, 2019)

Gamification may be a method I can successfully implement into the app, however as the app is predominantly used by the adult, it would need to be implemented into the Maths quiz section of the app.

I have made some amendments to the scope of this project. The app will focus more on short term daily tasks and rewards as opposed to weekly or monthly ones. Long term goals don’t seem to work as efficiently with younger children, especially before school age. Again as this app is firstly for personal use it will be configured to suit my situation. It will however be easily modified for public use. A future feature may be making the App customisable by the childs age.

## Description and scope of mobile app

I have made some minor amendments to the scope of this project. The app will focus more on short term daily tasks and rewards as opposed to weekly or monthly ones. Long term goals don’t seem to work as efficiently with younger children, especially before school age. Again as this app is firstly for personal use it will be configured to suit my situation. It will however be easily modified for public use. A future feature may be making the App customisable by the childs age.

The app will be designed for parental use mainly but will feature aspects that the child can use. It will work by scheduling chores and tasks with a daily calendar system and points can be earned by completing tasks or quiz’s.

* A mobile app to reward good behaviour through a kudos system(Reward points). Kudos can be earned through completing chores, tasks and good behavour.
* A login or register new user function
* I will implement a calendar system so it is possible to track past present and future tasks, events and rewards.
* Accumulation of kudos will translate into certain rewards that can be listed on the app.(Example, 5 kudos may earn extra TV or tablet time)
* Physical tasks that are completed outside of the app will require a password entered by an adult to add reward points(kudos).
* The higher number of Kudos, the better the reward.(agreed by child and parent)
* Tasks on the app itself, like a Multiplication test can be completed by the child.
* Kudos will be reset daily and new rewards and tasks can be entered into the app manually.
* Kudos can also be removed or edited via password by parent.
* The difficulty of the Math test can be set
* The App will give a daily reminder in the form of a notification.
* The App will give a notification when certain milestones are reached(accumulation of kudos for a certain reward).
* The App should be designed in a simple but aesthetic manner that is intuitive to use.
* SQlite will be used for the database to store user details, tasks and kudos
* The App is primarily for parental use.

## Architecture

The app will be built using Android studio for the Android platform, which will help me develop the front and back end of the application. I will be using java for the back end(ViewModel), XML(UI Controller) for the front and SQLite for the database. Below is a diagram of the components of the android platform architecture (Android platform documentation, figure 1, No date)

Diagram

Description automatically generated

(Android platform documentation, figure 1, No date)

## ICT aspects

The mobile app will be developed on the Android platform using Android studio. I have chosen this as the development environment contains almost everything I need to develop the app. It covers the front end and back end of the app. Android Studio it’s self is a powerful tool with litatrue and tutorial being essential to good understanfing of it’s complexites

Java will be used for the functionality of the app. The classes and methods will be written in java using Object Oriented Programming skills. For the UI XML will be used. This is fairly easy to link to the java classes using simple code.

For storage I will be implementing SQLite. I will Use SQLite to implement database storage for the user details for logins and to save registration details. It will also be used to store tasks, rewards and kudos. SQLite is new to me so literature and tuturals will be nessessery for me to learn how to use it effectively.

## Account of related literature

One of the biggest challenges in building this App was implementing the database using SQLite. Without a tutoral this would have perhaps taking a considerable amount of time. After a thourgha internet search I discovered a tutorial that seemed to match my needs. Anupam Chugh(no date) This walked me through the use of SQLiteOpenHelper Class helping me understand how I can implement it into my App. It was very clear on how to create and insert data into tables.

Interms of navigating and understanding how to use Android studio aswell as initially setting up a project Tim Buchalka’s Android Java Masterclass(online tutorial/no date) has been a valuble resource. This has taught me the basic aspects of setting up a project and how deal with the often fidily constraint layout with XML. It was also very helpful in understand what the android manifest does and what to include. I have had problems with activitys(Screans) not linking together, only to find out I had not incuded them in the manifest. Although I did not complete the whole tutoral as it is vast, I did complete enough to become confident using Android Studio

After trying unsuccessfully in getting the logup and register activy to work I again looked for a tutorial. It took me some time to find an up to date one. Adroid studio is constantly changing and updating so many old tutorials from 4 or 5 years ago are very outdated and don’t fully work. I eventually found (razormist  4, 2021) and it helped guide me through the process, although it was still slightly challenging putting it into my App.

As the project was developing, the code I was writing tended to get messy as perhaps I was becoming lazy with naming conventions. I decided to go back and look at Clean Code(Robert C. Martin(2009) and rename certain classes and veriables that did not make any sense. Lucky in android studio there is a refactor setting in the menu that allows you to rename a varible or class once and it will change it in every part of the code or file that it features, saving the time of going through all the code again.

Anupam Chugh shared pref tutorial no date

## Account of project work and its outcome

The Project has proven to be challenging with many new concepts such as SQLite and creating a calendar. Progress however is going well and the second iteration of the project has been completed. The first Iteration consisted of creating the login activity, register activity, and thetask list activity. In the second activity amendments were made and the Math quiz and calendar were added.

The Activty diagram below shows the basic flow of the app. Although each activity will have more functionality than is showen, it would quickly become a mess to include it all. Following on from this I will give a detailed description of each activity and the classes it holds.

Diagram

Description automatically generated

Activity Diagram

Bellow are screen shots of the working app so far. I developed the user interfece with layout reasorce files which includes XML coding with drag and drop elemnts within the Android studio IDE. Java was then used to code the functionality. Other essential elemements include the Android manifest which is an XML file which includes essential information such as tools, information from the gradlebuild files, the apps name and it’s actitvitys. I encountered a few problems with this as the project went through the first iteration. I have described the problems I encountered in a later section.

Graphical user interface, website

Description automatically generated

Login Activity

The Login Activity is the first screen shown upon opening the app. This was made using a layout reasource file which allows you to drag and drop various elements on the design such as button and text input fields. The login page will take you to the menu activity on successful log in or it will take you to a registration page.

Graphical user interface, website

Description automatically generated

Register Activity

The registration activity will enable regristion using basic authentication. This uses a database through SQlite to store user datails. Upon succesfull registration the user can then log in.

A screenshot of a cell phone

Description automatically generated with medium confidence

Menu Activity

The menu activity displays 3 buttons, each taking the user to an activity.

A screenshot of a cell phone

Description automatically generated with medium confidence

Task list page

The task list page allows the user to add and remove tasks from the list. Completed tasks can be checked by a tick. Tasks can be added by tapping the green button on the bottom left. A recycler view is used for the list and is kept in the activity\_task\_list.xml file. Tasks can edited by swiping left and deleted by swiping right on the task(recyclerview). The + and – Kudos buttons are for manually adding and removing kudos. Kudos are currently saved using androids shared prefernces. This locally stores key-value data on the device and is ideal for small amounts of data, as a new table in a database can be consiredered unessecary.

A screenshot of a video game

Description automatically generated with medium confidence A screenshot of a game

Description automatically generated with medium confidence

Math Activity Math Activity showing progress bar.

The math activity is a simple maths test with random generated numbers as well as a random sign of addition and multiplacation. Correct answers move the progress bar shown the top of the page to the end, at that point the game ends and asks the user if they want to continue or finish. Successful completion of the game will add 1 kudo to the daily total.

Graphical user interface, calendar

Description automatically generated A screenshot of a video game

Description automatically generated with medium confidence

Calendar activity Calendar Weekly view

A picture containing graphical user interface

Description automatically generated Graphical user interface

Description automatically generated

Calendar add event New event added

Above is the calendar activity. Here the user can see a monthly view. By tapping on weekly the weekly activity will be displayed. Here the user can add a new event under name and then save it with the button below. As the intended user of the app, this is where I would add milestones and rewards. It will also keep a record of previous mile stones.

|  |  |
| --- | --- |
| Java Class | Description and purpose |
| AddNewTask | This is a class to add a new task that extends a bottomSheetDialogFragment. This means a sheet pops up from the bottom of the screen that will contain a box to write the new task in. |
| CalendarActivity | This is the calendar activity that extends AppCompatActivity and impements CalendarAdapter.OnItemListner. This class sets the calendar views including montly and weekly. aswell as previousMonth and nextMonth methods for changing monthly views. |
| CalendarAdapter | CalendarAdapter extends a RecyclerView Adapter with the CalendarViewHolder. This class creates an Array list to hold the days and display what day of the month it is. |
| CalendarUtils | This Class formats the Calander to local date and time. The library of dateTimeFormater is used to format the days, months, years and time. |
| CalendarViewHolder | This is the view holder for the calander. It displays the days of the month in the cells. |
| DatabaseHelperLogin | The database helper in SQLite for the login aqctivty. it declares the database name a well as what the content values will be with the insert method. It also has checkUsername and CheckLogin Boolean methods. |
| DatabaseHelperTask | Just like the DB helper for the log in, but it helps set up the database for the task list activity. It contains methods update task and delete task. |
| DailyEvent | A Class to add a daily event such as a reward and kudos earned that day. |
| DailyEventAdapter | Adds the event into the daily cell. |
| DailyEventEditActivity | Sets the date, time and name of event |
| Login | Links up to the activity login view to give it functionalty by declaring button variables and has added Toasts tell the user if login was successuuf or invalid. |
| MainActivity | The main activity in this app is for the register function as it was the first class that was built. This Class allows the user to register with the help of the DataBase helper Login Class. |
| MathQuiz | The class with all the functionallty for the maths quiz activity. A number array is used and linled to number pictures through the drawable reasource files. The numbers are randomly generated to created a solvable math question. The operator is also randomly generated. |
| Menu | A class that linked the menu buttons to other activitys. |
| RecyclerItemTouchHelper | A class that allows the swiping of the task activitys to edit or delete. |
| TaskActivity | This Class dispalys all the tasks aswell as the editable kudos. |
| TaskAdapter | Facititates editing, setting and deleting of tasks |
| TaskModel | This class has the getter and setter methods used in the task classes |
| WeekViewActivity | This class has the functionality of the week view activity. |

A picture containing text

Description automatically generated

(Above) These are the layout files I created for the project. For most of the project the layout files were created first with the functionality add after with java.

A picture containing text

Description automatically generated

(Above) A list of All the java classes that have been implemented.

SQLite proved to be challenging at first, and although many problems were encountered at first, the project finally came together. There is perhaps not the need for two database helpers, but by that point the code was starting to get messy and thought best to implement two separate database helpers to keep the classes clearer. The DatabaseHelperTask Class is for implementing and saving the tasks in the task activitys and the DatabaseHelperLogin is used to store the registration details and have the login details checked with the database upon logging in.

## Problems encountered

The first major issue turned out to be a simple fix. I could not get the activitys to link up to each other from the menu, even though I had initiated it in the java code(see bellow)

Text

Description automatically generated

Above(Java code used to move from the Menu activity to the Task Activity)

The problem was not with the Java code, but within the Android Manifest. I had not included the new activitys in this. See the updated Manifest below that includes All the relevant activites under the <activity/> tag. The menu, TaskActivty, mathQuiz and calander activity are all included there.

Text

Description automatically generated

Updated Android Manifest

The other issue encounterd was getting shared preferences to work correctly. I could not quite figure it out from the android documentation. The problem was I had declared the shared prefernces interface locally when it should have been a class variable.

Text

Description automatically generated

This is where the shared preference interface was declared as kudoCounter.

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Above is the implementation of it into the onClick listener. Log.i is included for testing.

## Final iteration work to do

The third iteration will involve implementing a daily notification system. This will come in for of a reminder that can be set for a specific time to check off tasks or add new ones. This will help with consistency for the user. It should also prompt the user to add daily events which should be used to document rewards, passed rewards and the amount of kudos earned on that day. Another implementation will be a reset feature for the kudos. This will be a function to rest the kudos at the end of every day.

(Android platform documentation, figure 1, No date) <https://developer.android.com/codelabs/android-room-with-a-view?index=..%2F..%2Findex#1>

Anupam Chugh(no date)(<https://www.journaldev.com/9438/android-sqlite-database-example-tutorial>)

([razormist](https://www.sourcecodester.com/users/razormist)  4, 2021) <https://www.sourcecodester.com/android/12151/android-simple-registration-and-login-application.html>

Anupam Chugh shared pref tutorial no date( <https://www.journaldev.com/9412/android-shared-preferences-example-tutorial>)