## 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 behaviour, increase self-esteem and improve the relationship 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 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 available 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 practical 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.

ICT aspects and resources

|  |  |
| --- | --- |
| Resource/ICT Aspect | Uses/Description |
| Android Studio | This is Android official IDE for mobile app development. It covers the front end and back of the mobile development. Android gives the developer a visual representation of the app through the layout activity that brings an ease to front end development that other technologies like Cordova lack |
| Java | Java is an Object-Oriented Programming language that can be used within android studio. This covers the functionality of the App and connects aspects of the front end(Buttons etc) together |
| XML | XML is a mark-up language used for the front-end design and development of the mobile application. Is used within android studio and can be autogenerated for some elements. For example it is possible to drag and drop buttons onto the activity layout. |
| SQLite Database | SQLite provides the relation database functionality within the app. This will be essential for storing user details. |
| Gradle | Gradle comes installed within Android studio and is essential to it’s functionality. It facilitates the build process as well as testing and debugging. |
| Native Mobile Application Development(Android) | This is a combination of all the aspects above, bring them together into a functional app that serves a novel or important purpose. As it is Android only being used this is classed as native development. The benefits include better performance, security and access to the platforms full set of features(ClearBridge Mobile no date) |
| Object Oriented Programming | This is essential as Java uses the classes system. This helps break code down into smaller more manageable sections. within Classes methods are used to break down the code further. |
| Version control(GitHub) | GitHub is a form of version control and allows developers to upload projects into repositories. This is a fairly safe way to store the project. |

## Description and scope of mobile app

Project output

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 child’s 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 behaviour.
* 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.

## Stakeholders

* My wife, and myself will be the main users of the app
* My Daughter, as her personal development is what the app will be used for. She will also participate in the use of the app.
* Potentially other members of the family
* Potentially members of the public if I decide to release it on the app store.

## 

## 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 literature and tutorial being essential to good understanding of its complexities

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 tutorials will be necessary 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 tutorial this would have perhaps taking a considerable amount of time. After an extensive 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.

In terms of navigating and understanding how to use Android studio as well as initially setting up a project Tim Buchalka’s Android Java Masterclass(online tutorial/no date) has been a valuable resource. This has taught me the basic aspects of setting up a project and how deal with the often-fiddly 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 activity(Screens) not linking together, only to find out I had not included them in the manifest. Although I did not complete the whole tutorial as it is vast, I did complete enough to become confident using Android Studio

After trying unsuccessfully in getting the login and register activity to work I again looked for a tutorial. It took me some time to find an up to date one. Android 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 variables that did not make any sense. Lucky in android studio there is a refactor setting in the menu that allows you to rename a variable 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………………………….TODO

<https://developer.android.com/training/data-storage/sqlite>.......TODO

# 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 the task list activity. In the second activity amendments were made and the Math quiz and calendar were added.

The Activity diagram below shows the basic flow of the app. Although each activity will have more functionality than is shown, 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

## Outline of the development process

The process of developing this app consisted of designing and developing the UI with drag and drop elements mixed with XML code. I would develop one activity then code the java class to add the functionality. After a quick test on a physical android device development began on the next activity, again creating the front-end UI then coding the Java classes and quickly testing.

Graphical user interface

Description automatically generated

Screen shot of the activity layout for the math quiz.

Above is the activity layout for the math quiz activity. On the left is the XML code. Some of the XML code is auto generated if elements are dragged and dropped onto the design interface on the right-hand side. On the design screen you can see the two rectangle boxes which will provide the image view for the numbers, the text view below for the input and the button at the bottom. The elements are held together with a constraint layout.

Text

Description automatically generated

The Java code

Above is the java class for the math quiz. this is where the logic is performed and the image view elements are linked to values.

## Evidence, description and explanation of the app

Below are screen shots of the working app so far. I developed the user interface with layout resource files which includes XML coding with drag and drop elements within the Android studio IDE. Java was then used to code the functionality. Other essential elements 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 activities. 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

(Above)Login Activity. Background picture free to use(@viagalactica 13/08/2019)

The Login Activity is the first screen shown upon opening the app. This was made using a layout resource 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

(above)Register Activity

The registration activity will enable registration using basic authentication. This uses a database through SQLite to store user details. Upon successful 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 be 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 preferences. 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 considered unnecessary for a single integer.

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 multiplication. 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 milestones.

Below is a table of the java classes implemented with a brief description and the purpose of them.

|  |  |
| --- | --- |
| Java Class | Description and purpose |
| AddNewTask | This is a class to add a new task that extends 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 implements CalendarAdapter.OnItemListner. This class sets the calendar views including monthly and weekly. as well 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 Calendar 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 calendar. It displays the days of the month in the cells. |
| DatabaseHelperLogin | The database helper in SQLite for the login activity. it declares the database name as 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 functionality by declaring button variables and has added Toasts tell the user if login was successful 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 functionally for the maths quiz activity. A number array is used and linked to number pictures through the drawable resource files. The numbers are randomly generated to create a solvable math question. The operator is also randomly generated. |
| Menu | A class that linked the menu buttons to other activities. |
| RecyclerItemTouchHelper | A class that allows the swiping of the task activities to edit or delete. |
| TaskActivity | This Class displays all the tasks as well as the editable kudos. |
| TaskAdapter | Facilitates 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 activities 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 activities to link up to each other from the menu, even though I had initiated it in the java code(see below)

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 activities in this. See the updated Manifest below that includes All the relevant activates under the <activity/> tag. The menu, TaskActivty, mathQuiz and calendar activity are all included there.

Text

Description automatically generated

(above)Updated Android Manifest

The other issue encountered 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 preferences interface locally when it should have been a class variable.

Text

Description automatically generated

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

Graphical user interface, text

Description automatically generated

(Above)Implementation of shared preferences

Text

Description automatically generated

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

Another issue during the first iteration was the use of poor naming conventions. This led to confusing code and classes. It also involved the deletion in two classes and the refactoring on others. Effort has been made to carefully name future classes and variables in a more descriptive manner resulting in a far more successful second iteration.

## Testing the app

User acceptance tests after the second iteration.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Sequence | Tested | Description | Outcome |
| 1 | Register activity button | Upon tapping the register button the user is taken to the register activity | Accepted. The register activity is displayed |
| 2 | Complete register | Fill out the register form with matching passwords. | Accepted. Register dialog box displayed |
| 3 | Unsuccessful register | Password doesn’t match with the confirm password | Accepted. “Password does not match” is displayed in a dialog box |
| 4 | Username already taken | Username already taken | Accepted. Upon trying to register with an already in use username a dialog box appears “Username already taken” |
| 5 | Login | Enter username and password to login | Accepted. “Login successful” is displayed |
| 6 | Failed log in | Wrong details entered | Accepted. “Invalid username or password” is displayed. |
| 7 | Task List Activity | Button takes user to the task activity | Accepted. Task activity is on display |
| 8 | Add Task | Add task button if pressed | Accepted. Task successfully added and saved |
| 9 | Delete Task | Delete task by swiping left on the task | Accepted. Task successfully deleted from list |
| 10 | Edit Task | Edit task by swiping right on the task | Accepted. Task listed was edited successfully |
| 11 | Add Kudos | Press the +Kudos button | Accepted. Kudos added |
| 12 | Remove Kudos | Press the – Kudos button | Accepted. kudos subtracted |
| 13 | Maths Quiz Button | Button takes user to the math quiz activity | Accepted. User is taken to the Math quiz activity |
| 14 | Correct answer | Answer a question correctly displays an alert tell the user “correct” and increments the progress bar by 1. | Accepted. displays correct alert and progress bar moves by 1 point. |
| 15 | Wrong Answer | Answer a question wrongly and alert tells the user the answer is “wrong”. The progress bar does not move | Accepted. displays the “wrong” alert and progress bar does not move |
| 16 | Finish the game | Progress bar reaches the end after 10 correct answers. A dialog box asks to try again with “yes” or “No” option. Yes starts another r game and no takes user back to the menu screen | Accepted. Both options at the end do as intended and all question logic appears to be correct. |
| 17 | Calendar button | Button taking the user to the calendar activity. | Accepted. User successfully taken to Calendar |
| 18 | Weekly view | Weekly tapped to access weekly view | Accepted. Weekly view accessed by user |
| 19 | New daily event | Add a new Daily event/reward | Accepted. Daily event/reward added |
| 20 | Change month | Toggle through months with arrow buttons | Accepted. User can change months. |
| 21 | Change week | Toggle through weeks with arrow buttons | Accepted. User can change weeks. |

## User Feedback

User feedback has been generally good and they are happy with the general set up of the app. There are however some changes that will be made following feedback. These will be made in the third and final iteration.

|  |  |  |
| --- | --- | --- |
| Feedback issue number | User feedback | Proposed solution |
| 1 | The input text for login and register activity’s clashes with the background | Change the text colour to white. |
| 2 | Did not notice the progress bar at top of the screen on the math quiz activity | Widen the thickness of the progress bar and add text to indicate what it is. |
| 3 | Calendar view needs to be intuitive on what its function is. It’s not obvious on how to get to the weekly view and to add daily event/log/reward. | Add more descripted text to the app and make responsive points more obvious. |
| 4 | Change the background. The back ground clashes with certain activities. | Change the back ground to a generic single colour. |
| 5 | Would be nice to have a welcome screen as app loads abruptly. Would give the app a more professional feel | Add a splash screen to start the app. |
| 6 | Change register function to valid email address only to prevent child access | Modify the code to implement valid email address only. |

## 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 number 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. The user feedback issues 1 to 6 will be addressed/fixed and it will go through the last set of testing to ensure the final app is fit for purpose. Lastly the ability to change the difficulty setting on the math quiz app should be implemented. The multiplication question are out of the grasp of a 4-year-old, so a toggle between multiplication and subtraction as well lowering the range of numbers are options.

# Review

## Review of current stage of project work

I am currently pleased with the progress I have made on the development aspects of the project; however this has not come without problems. I had significant problems getting to grips with SQLite and successfully implementing it into the application. Although the Android documentation was helpful I had to seek out other learning resources to help men implement it. I did however get there in the end by implementing by setting up two database helpers for two different tables.

I have also started to think that the Login and register aspect may be redundant for the app, but I have been reassured from user feedback that it would prevent being tampered with by the users child. I should however change the register activity to only accept a valid email address instead of a username. It is very unlikely that a child of the ages 4 to 8 would have an email address. This would need to be something I thought more about if I was to release the application to the public.

I think my lack of design experience shows in the app as well. I’m in agreement with the users that the look and feel can be improved to make the user experience more intuitive. Regardless of the problems I’ve encountered I am relatively happy with the current progress and without too much work the app can be improved in the next iteration.

## Project Management

I believe project management is one of my main weaknesses when it comes to this project. Although I developed a schedule I seldom stuck to. I instead tended to do work in sprints. I would work intensely for 3 or 4 days then stop for a period and repeat. This was mostly due to personal circumstances as I didn’t take into account certain scenarios. My daughters nursery closed for 10 days due to a Covid outbreak meaning I was the primary child carer. This made project work very difficult during that time. This resulted in periods of intense work(sprints) to catch up. I also had a delay in starting the first iteration of the project development due to an EMA of another module. I also failed to take this into account resulting in a sprint for the first iteration.

I am however pleased that I finished the first and second iterations of the project on schedule. I have also successfully kept notes of problems and difficulties during the development process which has helped me plan out the TMA and evaluate how the project work has progressed.

## Assess risks to project completion

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Impact | Likelihood | Mitigation |
| 1.The Users reject the app | H | L | Risk can be managed by an early prototype and user testing. This will ensure the app is designed around user preferences.  Testing after each iteration will be essential to improve UX and to gain valuable feedback from the stakeholders and implementing it will increase chances of users embracing the app. |
| 2.Loss of project data through personal hardware failure | H | L | Loss of project data through personal hardware failure  This is a very possible risk, so it is essential that I have everything backed up. It is something I will do after each work session. A flash drive and perhaps a cloud storge back up too. |
| 3. Personal illness or family issues | M | L | Common illness or other more serious issues may arise, So I have scheduled more time than I usually would for certain tasks and iterations to ensure I have some leeway if the worst were to happen. |
| 4.Not enough time to fulfil project scope | M | M | As unlikely as this is to happen it can be solved by adjusting the scope. I will likely know for sure by the time I have completed the first iteration. One way to mitigate this will be to cut back on other commitments temporally until project work has be caught up with or is back on course. |
| 5. Project Scope is unmanageable | H | L | The scope of the project has been researched and defined. Nothing has been considered to advanced or beyond the capabilities of knowledge. |
| 6. Scheduling clashes and other modules to complete. | L | L | Majority of other module work should be completed in time to concentrate on project development |

Addressing the risks above

1. At this stage the primary user has had the opportunity to test the app and give feedback. This has dramatically reduced the risk as feedback is being received. The app is being tailored to the users wants and needs.

2. This risk has been reduced and managed by using version control. the project has been uploaded to GitHub at regular development milestones, as well as having a flash drive safely stored away from the computer.

3.This risk actually occurred in some sense that my daughters nursery was closed for 10 days due to a Covid outbreak. I managed the risk by working intense sprints to catch up with project work.

4.This risk has significantly eased as the majority of the development work has been completed. The last iteration requires development work than the first two.

5. Although the project is very challenging, nothing has seemed completely out of reach in terms of scope.

6. A clash of modules was an issues at the start of the development process meaning work on the app was delayed by two weeks. Intense periods of work followed to get the project back on schedule

## Review of personal development

(@viagalactica 13/08/2019) <https://unsplash.com/photos/0SqUthjger8>

(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>)

(ClearBridge Mobile no date) <https://clearbridgemobile.com/benefits-of-native-mobile-app-development/>