**The Problem**

The client is my **uncle, U**. U owns a pharmacy that is losing sales due to online services such as Amazon and Wish providing the desired products for customers more conveniently[[1]](#footnote-0). Additionally, U avoids selling products on those online services due to their undesirable drawbacks, such as the referral fees, being unable to limit customers who can purchase his products to his target audience; the residents of the village he lives in, which also adds import tax, and having to wait for Amazon workers to transfer his products which could be done much quicker by U himself if he knew the address of the customer[[2]](#footnote-1).

**Proposed Solution**

The solution was agreed to be an android application made using Android Studio[[3]](#footnote-2). It will function similarly to Amazon by providing only users in U's village with U's products and emailing U the product name and price, users name, and time of purchase, as U specified[[4]](#footnote-3). The address and name of each user will be received through their registration and will be stored in Firebase Authentication (FA). The title, description, price, and image of each product will be stored in Firebase Realtime Database (FRD) and Firebase Storage (FS). Users will be provided with a recommendation page which uses collaborative filtering (CF). A search bar using linear search is provided to search for specific products.

**Rationale for Solution**

U specified the solution to be an android application due to all of his customers using android which Android Studio achieves[[5]](#footnote-4). It was chosen over other softwares such as Eclipse, IntelliJ IDEA, and LeakCanary due to having more useful features than them. Examples are provided below:

* Code templates, sample applications, and the APK analyzer help reduce file sizes and tiresome tasks, increasing the speed at which code can be focused on, also making the application run quicker, and reducing the likelihood of buffering[[6]](#footnote-5) [[7]](#footnote-6)
* All types of Android devices, such as tablets, phones, and more are included, which U specified was preferred7 8
* Provides advanced code completion, refactoring, and code analysis, which increases programming speed and efficiency, helping develop the solution quicker[[8]](#footnote-7)
* Emulates android devices, increasing the speed at which code can be tested and debugged[[9]](#footnote-8)

FRD, FA, and FS were chosen for the following reasons:

* Realtime storage is provided, meaning that values get updated as soon as they are changed, avoiding purchases occurring before a user’s data is stored or delays between when a new product is published and when a user can purchase it[[10]](#footnote-9)
* The data is private, and I control who has access to the data to avoid potential outsider malice and allow U and other trustworthy people who can help evaluate the success of a product to use the data, as U wanted11 16
* FA encrypts users’ data17, so even I cannot see users’ passwords, which is vital for the data’s security and the ethical right of users being the only ones who should know their own passwords
* FRD has been designed to function with FS to be able to storage images and the rest of a products data, allowing for the cloud to store large amounts of data of images of products sold on the app18

Android Studio has built-in storage systems which I considered instead of FRD, including app-specific storage, shared storage, and preferences[[11]](#footnote-10). However, app-specific and shared storage relates to multiple applications, which is not relevant for the product being made11. Preferences only store primitive data11. This is undesirable because each product's data will be stored as an object to reduce chances of stackoverflow and incorrect indexing of users data.

CF will be used for the recommendation algorithm because it is the most commonly used algorithm which has been proven to be successful for companies like Netflix[[12]](#footnote-11). A recommendation algorithm is important because if it is effective, the likelihood of a purchase increases, increasing sales.

A search algorithm is important to find a users’ desired product as quickly as possible. The longer it takes for a user to find their desired product, the less likely they are to purchase one, decreasing sales. Hence, I will use a linear search algorithm as it is the most efficient search algorithm for unsorted data19.

From all of the languages Android Studio offers (Java, C++, Kotlin), I chose Java over the others because it is multi-threaded (Kotlin and C++ aren’t multi-threaded), which is needed when multiple programs must run simultaneously such as to show a loading screen while loading a process or for the recommendation algorithm to calculate the most suitable product to display while other programs, for example, display icons[[13]](#footnote-12) [[14]](#footnote-13). Being an Object Oriented Programming (OOP) language is also important for reasons provided below:

* Encapsulating data, such as users' account details, and the details of each product’s image, title, description, and price, which will increase data safety by protecting mutable data from changing unintentionally[[15]](#footnote-14)
* Inheriting methods and variables, such as the methods and variables used to create each product, and the methods and variables used for each user’s account details, which increases memory efficiency15

**Success Criteria**

* Users can register, providing their address (only addresses in U’s village accepted), name, and password which gets stored on Firebase Authentication
* Users can login by providing their username and password which they specified when registering
* A special account with a prescribed username and password created by me will be made which allows U and others he trusts to create purchasable products with a title, picture, description, price, and buy button; the products will be stored in Firebase Realtime Database and Firebase Storage
* Items can be scrolled through and purchased
* When an item is purchased, U is emailed with the product name, product price, customer name, and time of purchase
* The order of products are displayed based on a collaborative filtering recommendation algorithm which uses ratings users gave for products that are stored in Firebase Realtime Database
* A search bar is provided, displaying all products whose characters in the title, description, or price match the inputted text

1. see Appendix, Interview 1 (Line 3) [↑](#footnote-ref-0)
2. see Appendix, Interview 1 (Line 7) [↑](#footnote-ref-1)
3. see Appendix, Interview 1 (Line 31) [↑](#footnote-ref-2)
4. see Appendix, Interview 1 (Line 16) [↑](#footnote-ref-3)
5. see Appendix, Interview 1 (Line 31) [↑](#footnote-ref-4)
6. <https://developer.android.com/studio/debug/apk-analyzer> [↑](#footnote-ref-5)
7. [https://developer.android.com/studio/features](https://developer.android.com/studio/features#:~:text=Android%20Studio%20offers%20build%20automation,shrinking%20and%20app%20signing%20configurations) [↑](#footnote-ref-6)
8. see Appendix, Interview 1 (Line 38) [↑](#footnote-ref-7)
9. <https://developer.android.com/studio/run/emulator> [↑](#footnote-ref-8)
10. [https://blog.mindorks.com/firebase-realtime-database-android-tutorial](https://blog.mindorks.com/firebase-realtime-database-android-tutorial#:~:text=We%20have%20to%20connect%20our,will%20be%20shown%20to%20you) [↑](#footnote-ref-9)
11. <https://developer.android.com/training/data-storage> [↑](#footnote-ref-10)
12. <https://realpython.com/build-recommendation-engine-collaborative-filtering/> [↑](#footnote-ref-11)
13. [https://www.guru99.com/kotlin-vs-java-difference.htm](https://www.guru99.com/kotlin-vs-java-difference.html#:~:text=Kotlin%20combines%20features%20of%20both,and%20Java%20supports%20implicit%20conversions) [↑](#footnote-ref-12)
14. <https://www.educba.com/c-plus-plus-vs-java/> [↑](#footnote-ref-13)
15. <https://stackoverflow.blog/2020/09/02/if-everyone-hates-it-why-is-oop-still-so-widely-spread/>

    15 see Appendix, Interview 1 (Line 104)

    16 <https://firebase.google.com/support/privacy>

    17 <https://www.youtube.com/watch?v=9-oa4OS7lUQ>

    18<https://www.codecademy.com/learn/apcs-algorithms/modules/apcs-searching-and-sorting/cheatsheet> [↑](#footnote-ref-14)