# Executive Summary

Managing a personal library can be a tedious task, especially with a community that continues to rapidly expand as each academic year passes by.

# Usability Requirements

# Guidelines

## The Developer’s Society Brand Guidelines

As this app will be designed for the society it is crucial that its aesthetics are in line with existing ‘brand’ design guidelines. This primarily includes the use of the society’s logo, preferred font and colour scheme. All of this can be found within the publicly available assets repository (link here) of the society.

## Google’s Design Guidelines

Through a quick survey of the user group within DevSoc it was found that about 70% of its current activate member base are android device users. Thus, this project will focus primarily on adhering to Google’s Design Guidelines for android development. Included below are key component guidelines that the project will use to ensure ease of use and better accessibility within the application.

### Buttons

This app will implement the use of buttons within most of its NFC interaction screens. This includes a mixture of icon buttons and text buttons. The text buttons will be used for yes or no responses from the user when asked if they’d like to borrow or return books. The icon buttons on the other hand will activate the NFC scanning feature.

As outlined in Google’s guidelines buttons should be identifiable, findable, and clear. They must clearly indicate what they’re meant to do and what its current state is. Moreover, navigating to a button should be easy for a user when it’s incorporated alongside other elements. Buttons must also be responsive to changes in the layout. This is especially crucial for android app development since the types of android mobile devices vary greatly in screen size. Additionally, the button’s container should have a dynamic width that adjusts to the text content’s length and the container must have rounded corners.

### Top App Bar

For ease of use when navigating the app, a top app bar presents itself clearly to the user rather than a bottom app bar. Moreover, by displaying the name of the currently active screen via the app bar, it summarises and highlights what the current screen’s contents are to the user.

The app will incorporate a centre aligned top app bar. Google’s guidelines outline the most ideal anatomy for the top app bar contents wherein the application’s hamburger menu and logo will be positioned in their corresponding sides.

The hamburger navigation menu must be placed in the left most corner while the interactive logo icon will be placed on the right most corner. Any text detailing what screen the user is currently on i.e., landing page, settings, etc. will be centre aligned.

### Navigation Drawer

As this application will have multiple different screens available for the user to navigate to, they need to be displayed in an accessible manner. With a navigation drawer, upon interacting with the top app bar’s hamburger menu, it will unlock the drawer which will display the different possible screen names to the user in which they can then pick one to navigate to.

The guidelines highlight that the navigation drawer must open from left to right and that labels for screens are concise. The labels need to be descriptive enough for the user to understand what a screen’s contents might include.

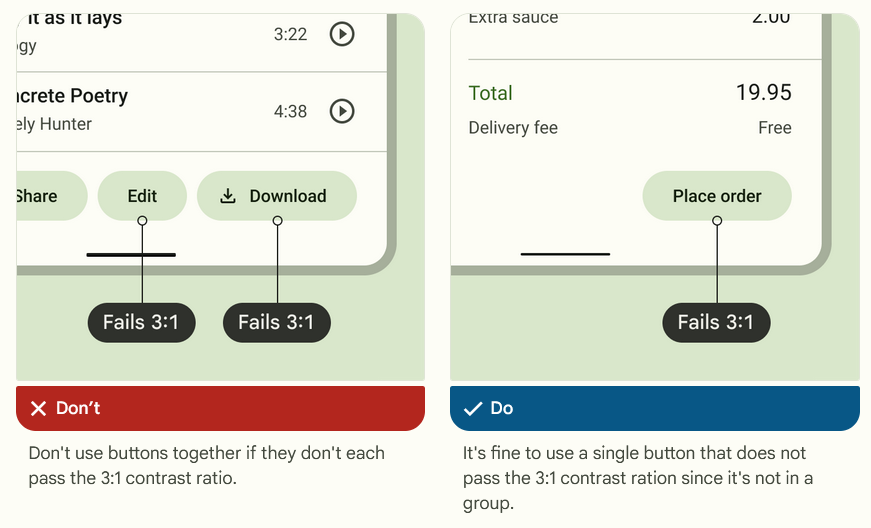
### Text

## Web Content Accessibility Guidelines (WCAG)

This project aims to be accessible to a wide array of users and as a result it will adhere to accredited accessibility guidelines in relation to the design of the application’s features. The Web Content Accessibility Guidelines (WCAG) are recommended within Google’s Guidelines as a source to be considered even for android app development.

### The 3:1 Contrast Ratio

The purpose of this guideline is to ensure that there is enough contrast between graphical elements that individuals with visual impairments can identify an element correctly. This primarily applies to the app’s use of buttons wherein the button’s text and container should have enough contrast between them that a user is able to identify and read the text with ease. Google’s guidelines further highlights that this must be followed in cases where there are multiple buttons next to each other.



https://m3.material.io/components/buttons/accessibility#71afc060-7055-4c40-a432-34b2de288eb8

# Constraints

# Proposed Idea and Impact

The Developers Society (DevSoc) has an ever-growing collection of literature stored in a cabinet within their office. The mini library is open to the general member base to borrow, read and return which has given way to a rather challenging problem to manage. As the community size has now grown well beyond 800 members it has become increasingly difficult to try and keep track of who has taken a book and not returned it.

The solution to this would be a mobile app that’s easily accessible that allows users to scan a book via an NFC sticker before registering themselves as the current borrower. This information would then get stored on a cloud database that’s available for the current society’s committee to access.

From a committee perspective, as noted in Persona C, this system creates a more formal borrow and return process. The database would allow them to keep track of any book within their library without having to consult a series of different emails or paper records. With a simple scan of the NFC committee access card, the app would unlock the hidden database view that’ll display all the books and the user that has borrowed them.

For members within the society, the application would solve a few different issues. As discussed in Persona A, users would be able to open the app and easily check the status of a book before making the trip down to Clifton Campus to collect it. Moreover, as highlighted through Persona B the borrowing and returning process would be simplified as it removes the need for the ‘middle-man’ committee member who currently must be present whenever a book is returned or borrowed so they can manually update the society’s records. With the NFC scanning feature, members simply need to open the app, activate the scanner, and scan a book. They’ll then just need to confirm if they’d like to borrow or return a book and the database will be updated accordingly in real time. Furthermore, with the use of a database members can view additional details about a book (i.e., author’s name or a summary) without having to look at the physical book. Finally, as the database stores data relating to the borrowing window/time limit the application will be able to automatically notify a user of when they’re due to return the book without the committee having to chase the individual up themselves.

# Application Features

|  |  |
| --- | --- |
| **Feature** | **Purpose** |
| NFC Stickers on Books (NFC Scanning) | Allows the use to easily scan a book to borrow or return it via the application |
| NFC Committee Access Card (NFC Scanning) | Unlocks hidden admin features within the application i.e., the ability to view the exact user that has borrowed a specific book |
| NFC Scanner Button (NFC Scanning) | This will activate the NFC scanning logic to ensure that we don’t need to have the app constantly checking and scanning in the background. This will also result in a pop up so the user is aware of what changes are being applied to their accounts |
| Hidden NFC Scanner Button (NFC Scanning) | This will be disguised as the logo icon button within the top app bar and is a way to unlock the hidden scanning feature of NFC committee access cards. Only with a success will the app unlock the additional database information. |
| Firebase Database | Stores data on the users and books via the cloud so that it is easily accessible |
| The ‘Backpack’ | Allows the user to view all their currently borrowed books along with the remaining time left before they need to return it |
| The ‘Library’ | Displays all the books stored within the database alongside additional information i.e., author name and current availability status |
| Settings | Allows the user to easily sign out of the app or choose to delete their account and data from the database |
| Login Page (User Authentication) | Allows the user to securely log into the application and store all their app activity in relation to their account. Enables easy transition between devices as all user data is hosted via the cloud database |
| Landing Page | Displays key data that a user might want to see immediately after a successful login i.e., their currently borrowed books and the activate NFC scanner button |
| Notifications | When a user’s time limit to borrow a book is about to end, the app will automatically notify them to prevent the need for the committee chasing after individuals themselves. |

# Evaluation of Paper Prototype

## User Testing – Session 1

During this session, the user noted that they’d prefer if there was a clear distinction of when the NFC scanning feature has been enabled or not. Currently, the app is set to constantly scan for the relevant NFCs though this might become too resource intensive for the device.

Instead, with a dedicated button to trigger the scanning feature it ensures that the scanner isn’t running constantly in the background when it isn’t needed, and that the user is explicitly aware of when the scanner is running.

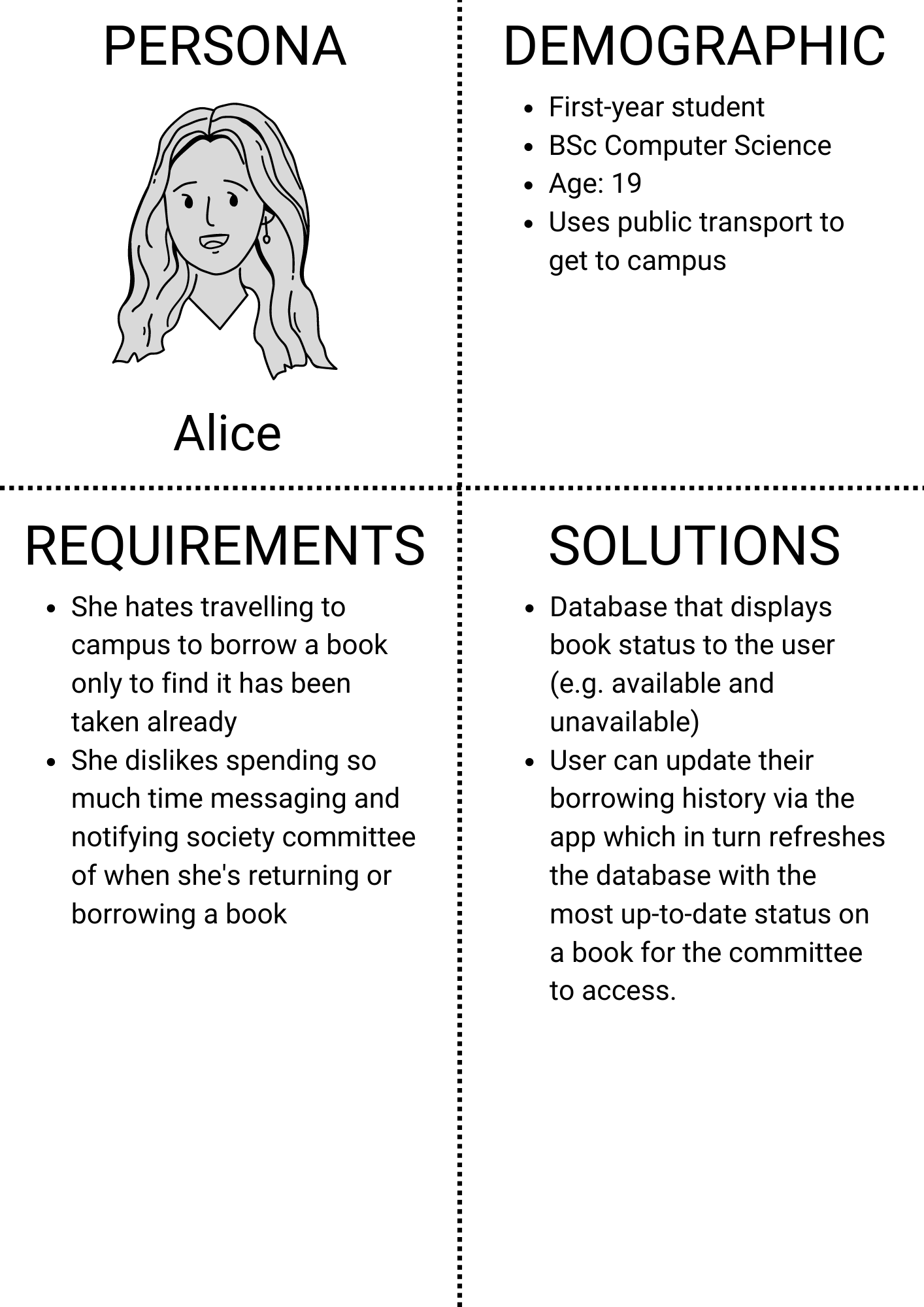
## User Testing – Session 2

In this testing session the user highlighted the need for a notification system. They noted that while the app itself does inform them of how long they have left to borrow a book, if the user doesn’t constantly check this app they might not know about the deadline. Thus, it would be helpful to be able to get a pop-up notification that acts as a reminder outside of the app in addition to what’s available within the app.

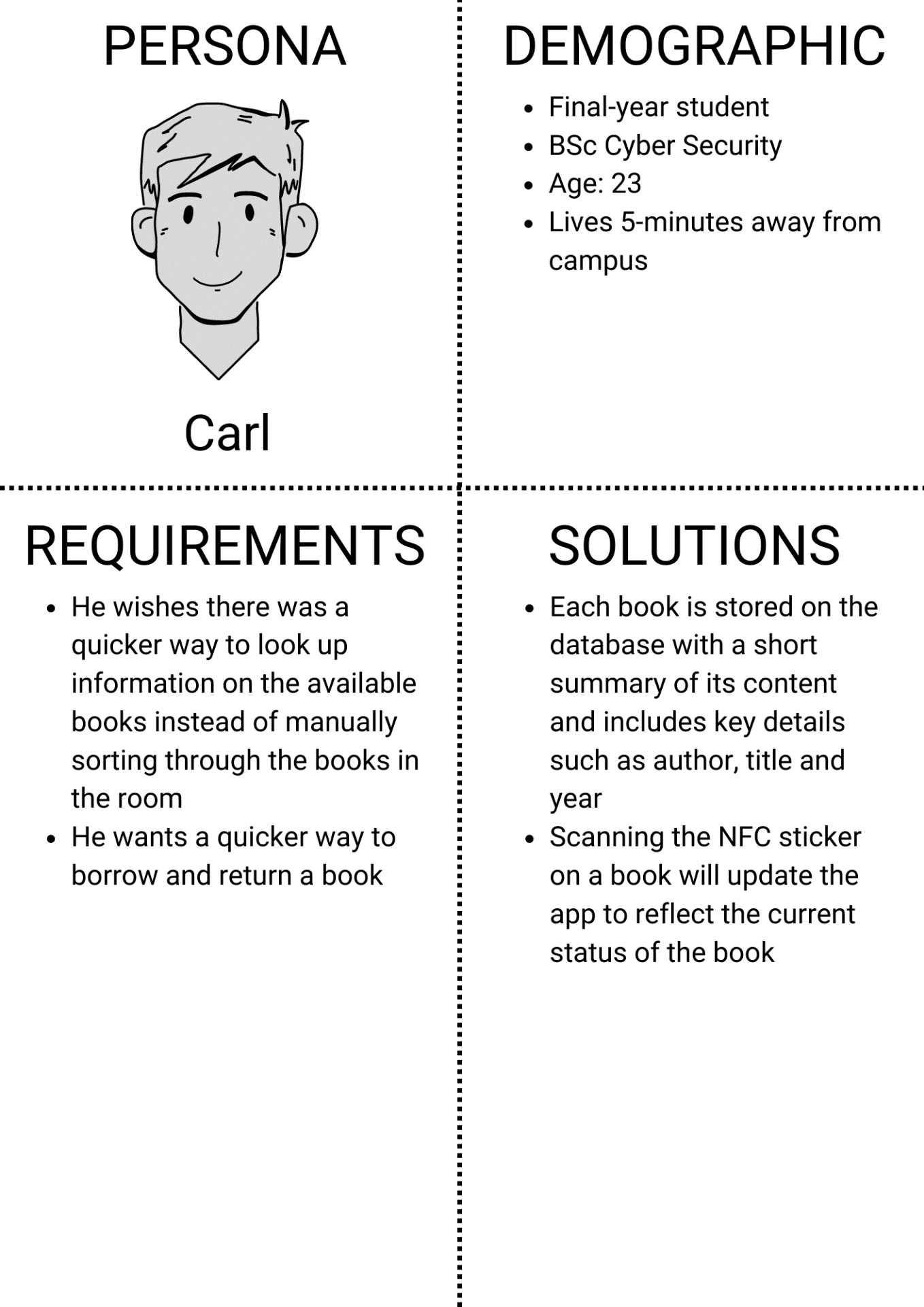
In addition to the above, the user also noted that when in admin/committee view of the app, the ‘library’ and the ‘database’ screens both serve the same purpose and thus results in unnecessary duplication of data. The solution to this would be to remove the additional ‘database’ screen and just display more information via the ‘library’ screen.

# Appendix

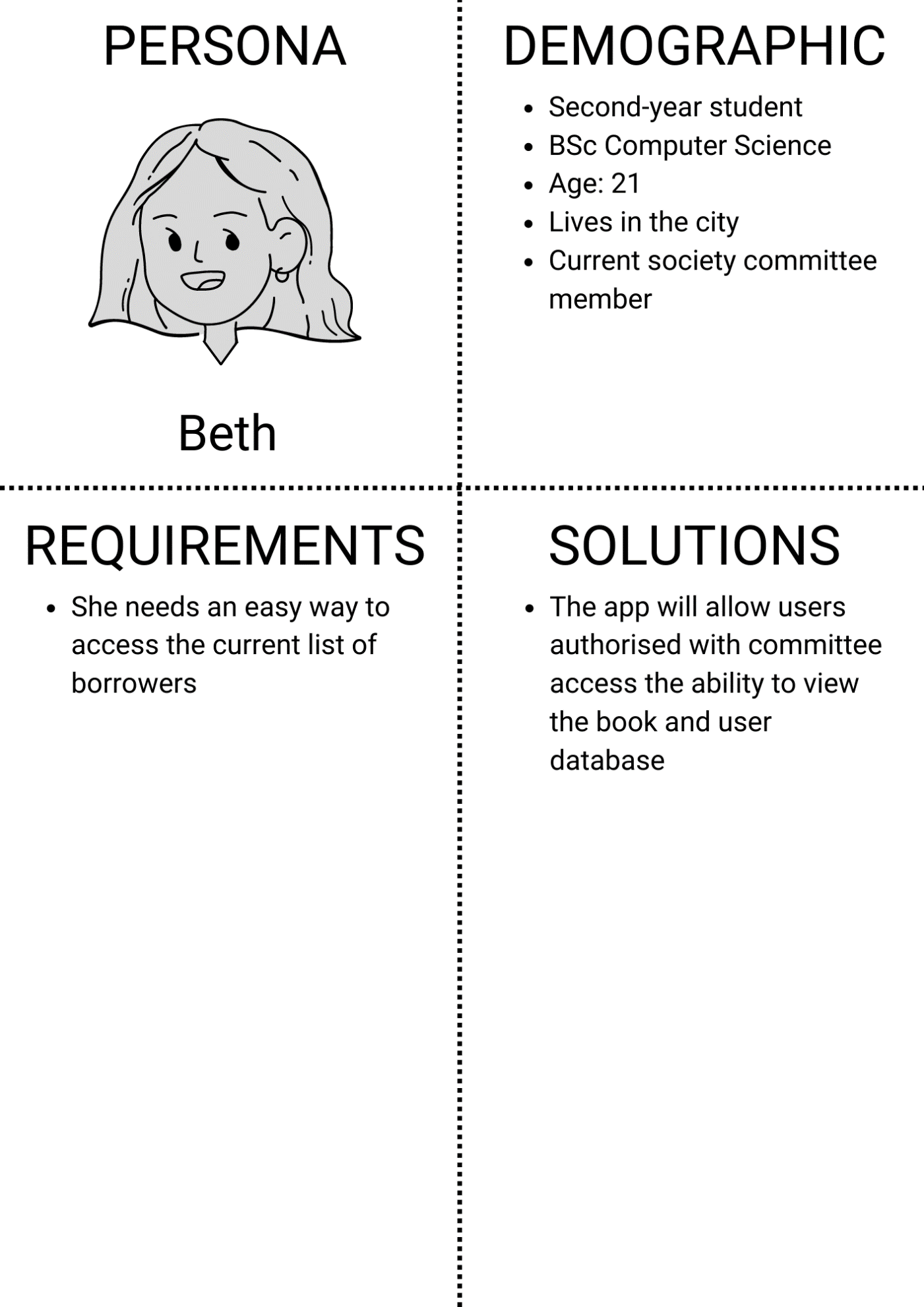
## Persona 1 – Alice



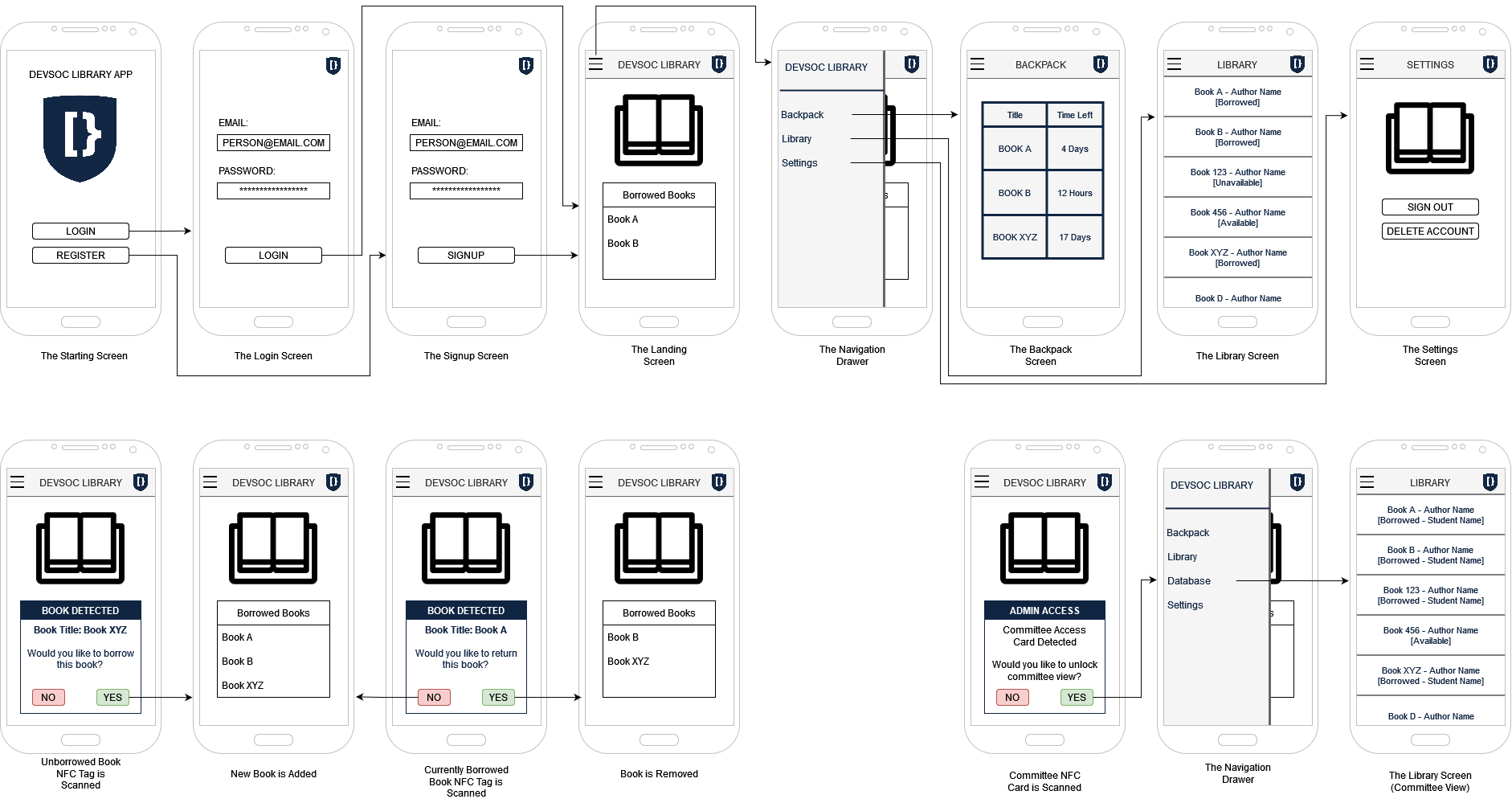
## Persona 2 – Carl



## Persona 3 – Beth



## Paper Prototype (Before user testing)



## Paper Prototype (After user testing)

