CS 401-001: Capstone Software Design Specification (SDS) Document

Tammy Busche

The Choice Is Yours, v.0.1

Android Mobile Application

May, 2023

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft 0.1	Tammy Busche	Preliminary draft outlining application design	06/01/23
Draft 0.2			

Review & Approval

Requirements Document Approval History

Approving Party	Version Approved	Signature	Date
Project Manager			
Professor Steve Sheehy			

Requirements Document Review History

Reviewer	Version Reviewed	Signature	Date
Tammy Busche	Draft 0.1		06/01/23
Angie Dickinson	Draft 0.1		06/02/23
Steve Sheehy	Draft 0.1		

Contents

1. Introduction

1.1 Purpose

The purpose of this design specification document is to describe in detail the CS 401 Capstone Project *The Choice Is Yours* Android mobile application functionality and user navigation. This document will serve as a guide to display the application's activities (program partitions with unique designs and purposes), use and user interface, appearance, and design strategy. Also included are application attributes, resources, assets, and images outlining design architecture.

1.2 Scope

The Choice Is Yours application electronic book (eBook) application. The unique aspect of this eBook app is that it is designed for the use of Gamebooks, a fictional genre of book that the series Choose Your Own Adventure is based on. These books are not read linearly where at the end of each section, referred to as *chapters* or *nodes* in this document, provide the user options as to how the story will continue.

Due to this story format, mechanisms are included in the program that displays to the user the chapter contents, images, next chapter options, selectable objects (buttons), and updating the chapter according to which option is chosen.

Aside from the main eBook reading functionality, the user will have the capability of visualizing different story branches that have been read. This requires the user to have a unique login ID (UUID) and database access. The database is used to retain data concerning story chapters and the *visited* status for each chapter. The database also retains chapters bookmarked. This bookmark is used for quick navigation to a particular chapter when accessing an eBook.

1.3 Intended Audience

The audience Gamebooks appeal to demographics from child to young teenager. However, this genre of book is also nostalgic for adult audiences. The eBooks for this app are written using the English language.

1.4 Definitions, Acronyms and Abbreviations

eBook – A book in an electronic format.
Chapter – A section of the eBook progressing either to a choice or to the end of the story.
Node – A chapter in context to its use to display Story Progression.
Firebase – An online application development service provided by Google.
Button – Interactive display objects allowing functionality.
Assets – Application files that are read only.
Recourses (Res) – Application static content used for app format and display.
Activity – Section of the application. e.g. Book Scrolling, Story Progression.
Toast – An on-screen message. The display is a small popup that disappears.

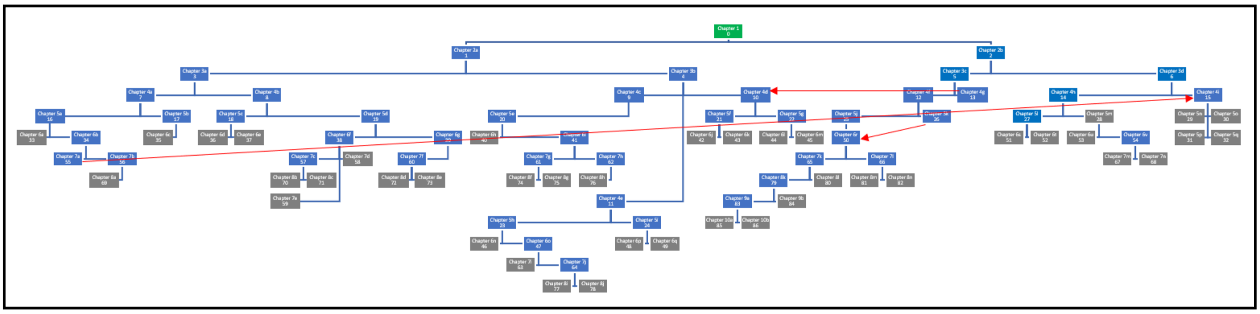
2. Architectural Design

2.1 General Description

The application design initially brings the user to a Main screen where the user can access their story library, login to and/or register for a user account, or read instructions for use of the app.

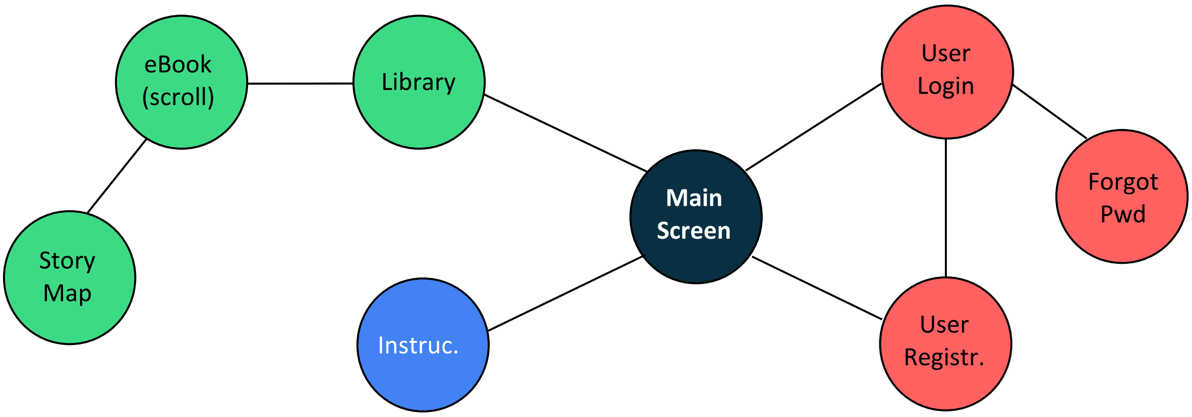
The current version of the application is relatively simple in content and operation where the user is provided five (5) *starter books*. Books are read in a scrolling page format (referred to in this document as the *Book Scrolling Activity*) where each page is composed of an image, chapter contents, and choices that determine what the next chapter will be.

In terms of login and registration, the utility of a user account is so that the user can review their progression for the stories they have read. For those who have read physical gamebooks, it may be difficult to determine whether they have read all of the contents of the book. Visualizing the various routes of the story shows a complex system of potential navigation for the reader. This is exemplified with the below image of the chapters for the volume 1 book of the library.



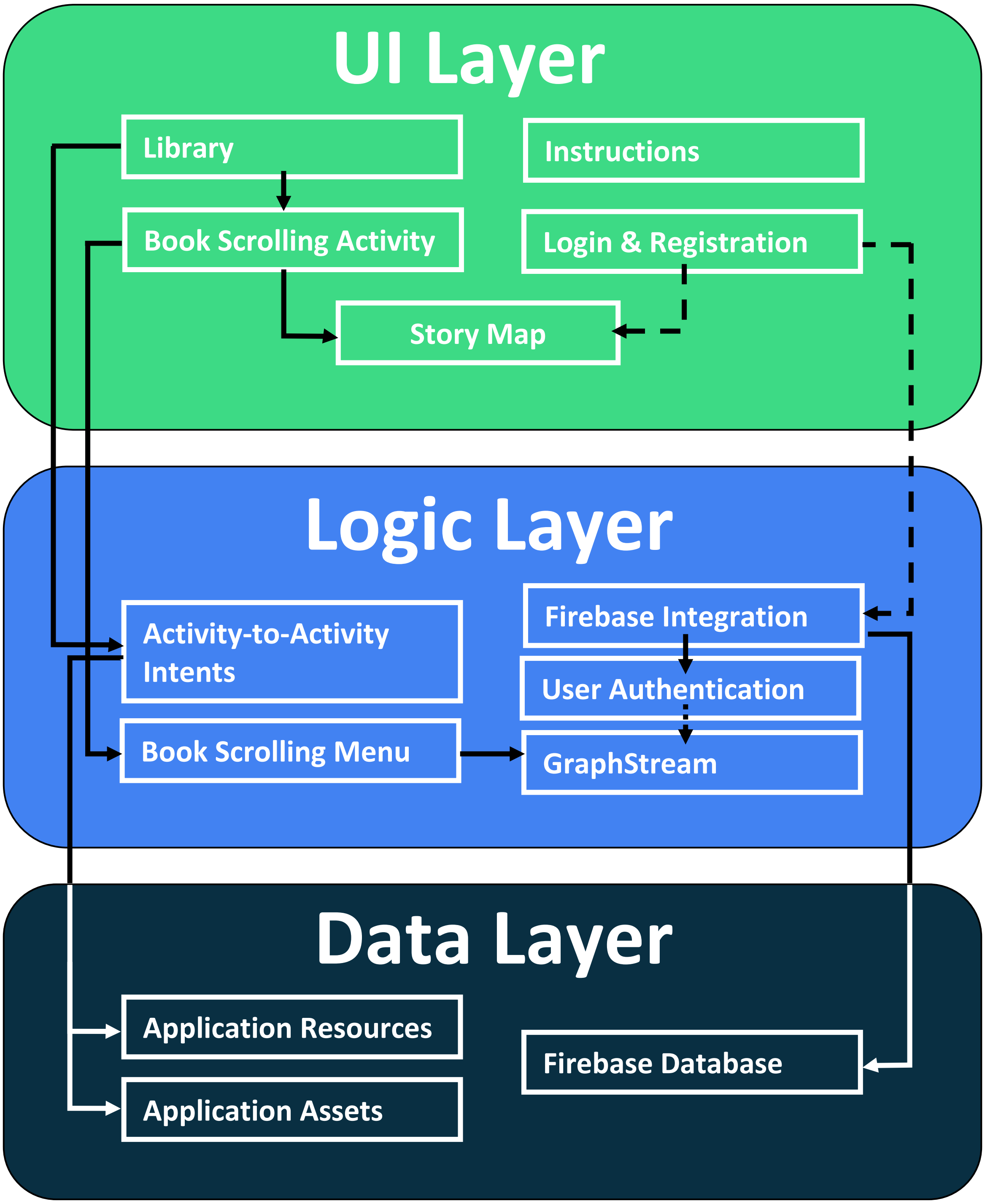
The eBook may be accessed and read with or without account authentication. As such, the application is operational offline. Meanwhile, if a user does login, a Story Map is assessable displaying accessed chapters.

In terms of use, aside from the Main Screen serving as a hub for app navigation and User Login & Registration, the usable components of the application are eBooks (as displayed in the Book Scrolling Activity), Story Progression Maps, and the application Instructions. These components will be displayed later in the document.



2.2 High-level Structure

Below is a diagram schematic illustrating the high-level structure of the application. Lines that are solid signifies component with a direct access to the application while dashed lines indicate transmission of data between components.



3. User Interface Design

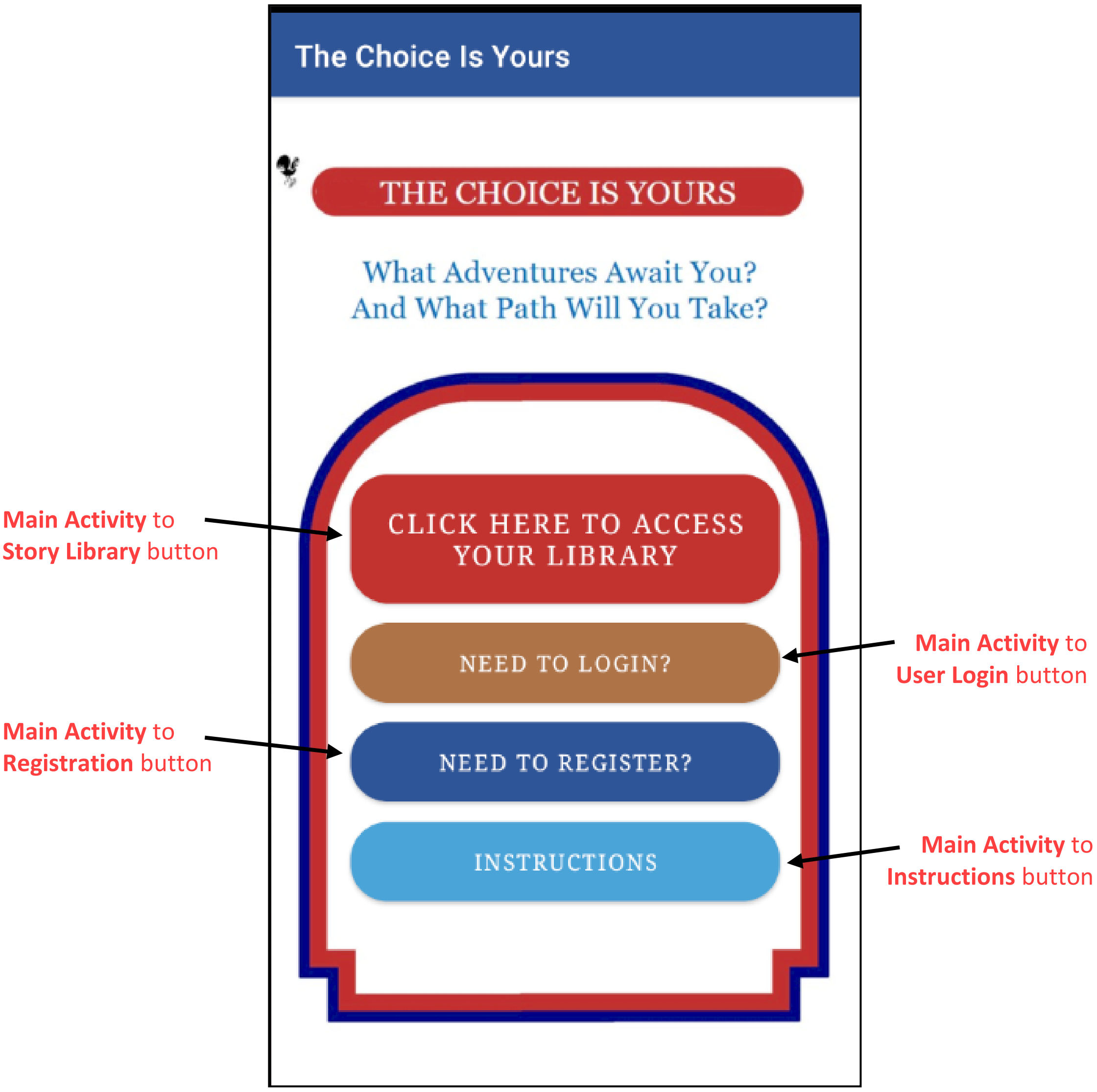
3.1 User Interaction and Inputs

As mentioned previously in this document, upon starting the application a Main Screen appears. Because the program is an Android application designed for use with Android phones and tablets, the user controls scrolling and activity navigation using on-screen *buttons*. For user login and registration, the application displays the device's on-screen keyboard for entry. The only user inputs used by the application are an email address and password for account authentication.

To navigate from activity to activity, the program uses *Intents*. As described by Google for Developers, "An Intent is a messaging object you can use to request an action from another app component." The activity screens and user-assessable on-screen objects are displayed below. Several on-screen buttons use Intents to begin an activity screen. Functionality beyond navigation intents are specified when applicable.

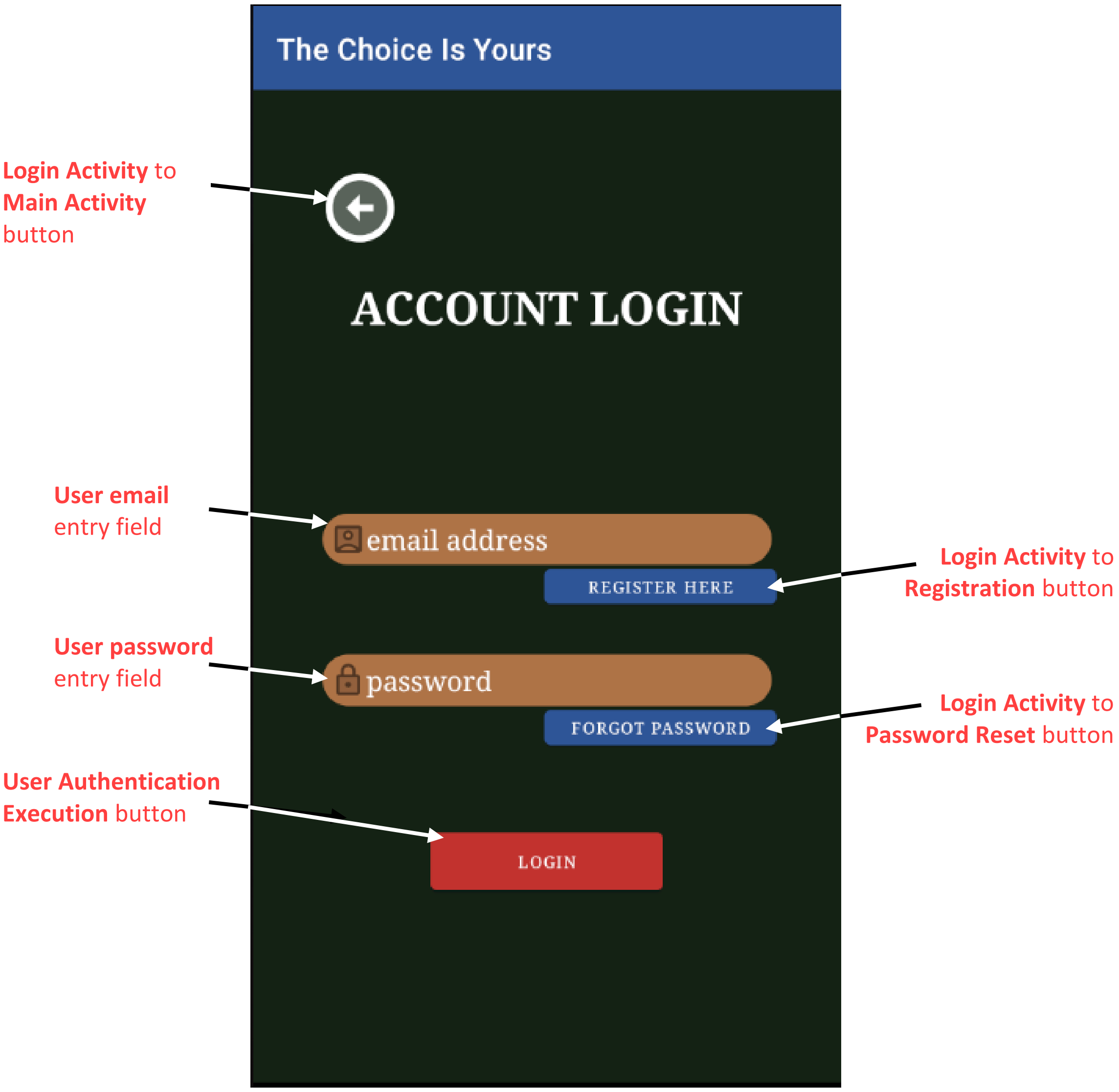
3.1.1 Main Activity

- Interactable objects: Four (4) buttons.
- Additional functionality: None

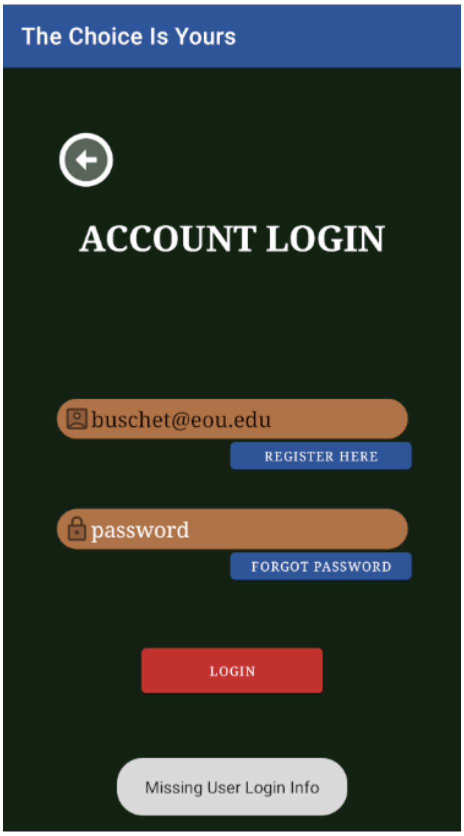

The Choice Is Yours
THE CHOICE IS YOURS
What Adventures Await You?
And What Path Will You Take?
CLICK HERE TO ACCESS YOUR LIBRARY
NEED TO LOGIN?
NEED TO REGISTER?
INSTRUCTIONS
Main Activity to **Story Library** button
Main Activity to **User Login** button
Main Activity to **Registration** button
Main Activity to **Instructions** button

3.1.2 User Login Activity

- Interactable objects: Four (4) buttons and two (2) entry fields.
- Additional functionality: Login – Performs user authentication with Firebase.


The Choice Is Yours
ACCOUNT LOGIN
email address
REGISTER HERE
password
FORGOT PASSWORD
LOGIN
Login Activity to Main Activity button
User email entry field
User password entry field
User Authentication Execution button
Login Activity to Registration button
Login Activity to Password Reset button

- Functionality Detail: Both email and password entries must be provided. The user credentials are compared to the user credentials stored on Firebase. There are three (3) possible outcomes when the registration button is selected:
  - If one or both of the user credential fields is missing an entry, a toast (a small transient popup) message appears prompting that user login information is missing.
  - If both credential fields have entries, but the credentials do not match user email and password data in Firebase an, "Authentication Failed," toast appears.
  - If both credential fields have entries and those matching credentials are recognized in Firebase, a successful login toast appears and the application automatically navigates to the Main Activity.
  - See images below. *Toast messages appear with a white background when the device is set to *Light Theme*.*

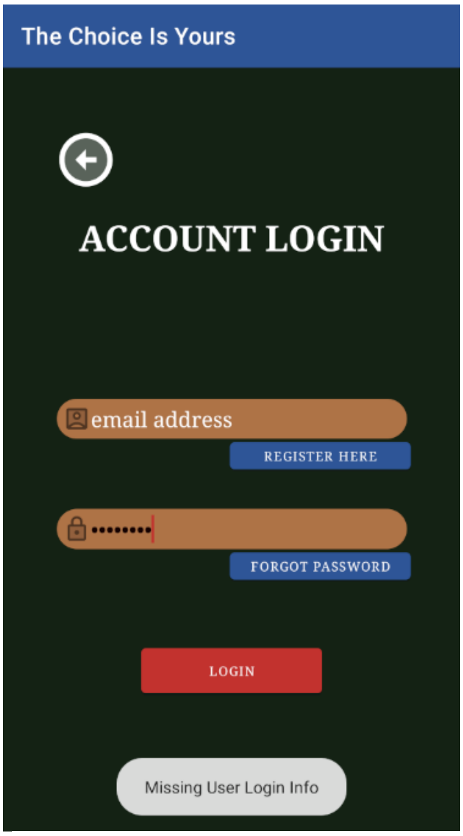


buschet@eou.edu

password

Missing User Login Info

Example: Missing user password entry

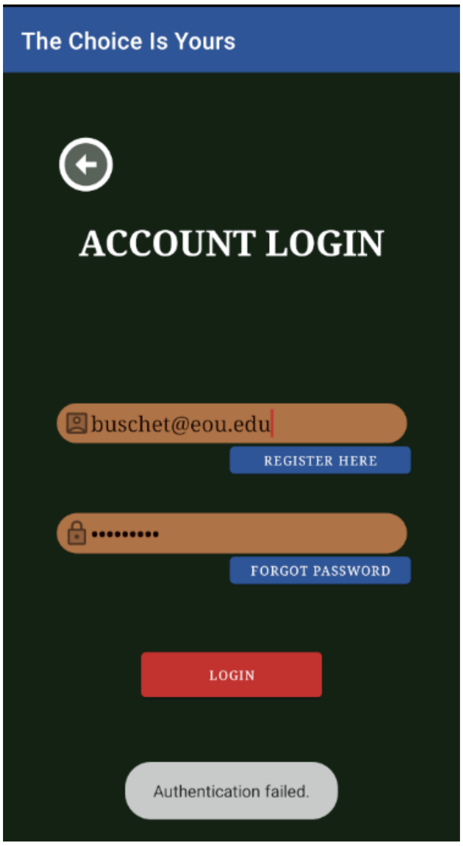


email address

••••••••

Missing User Login Info

Example: Missing user email entry

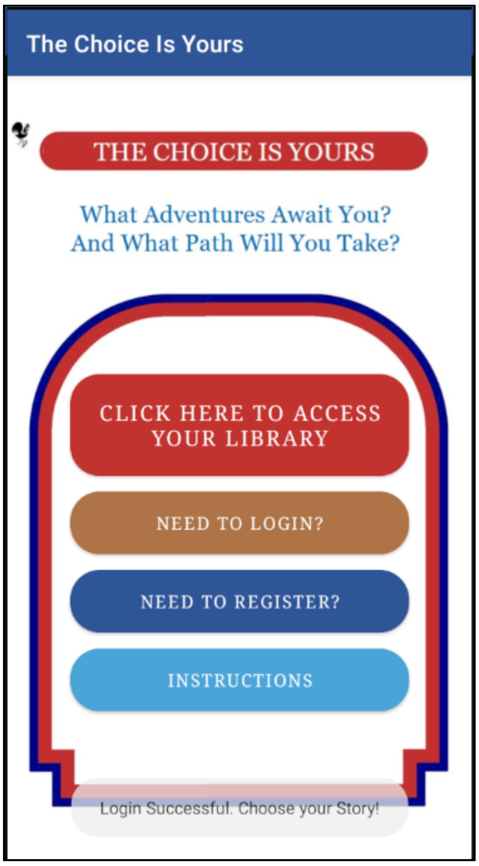


buschet@eou.edu

•••••••••

Authentication failed.

Example: Correct email, wrong password

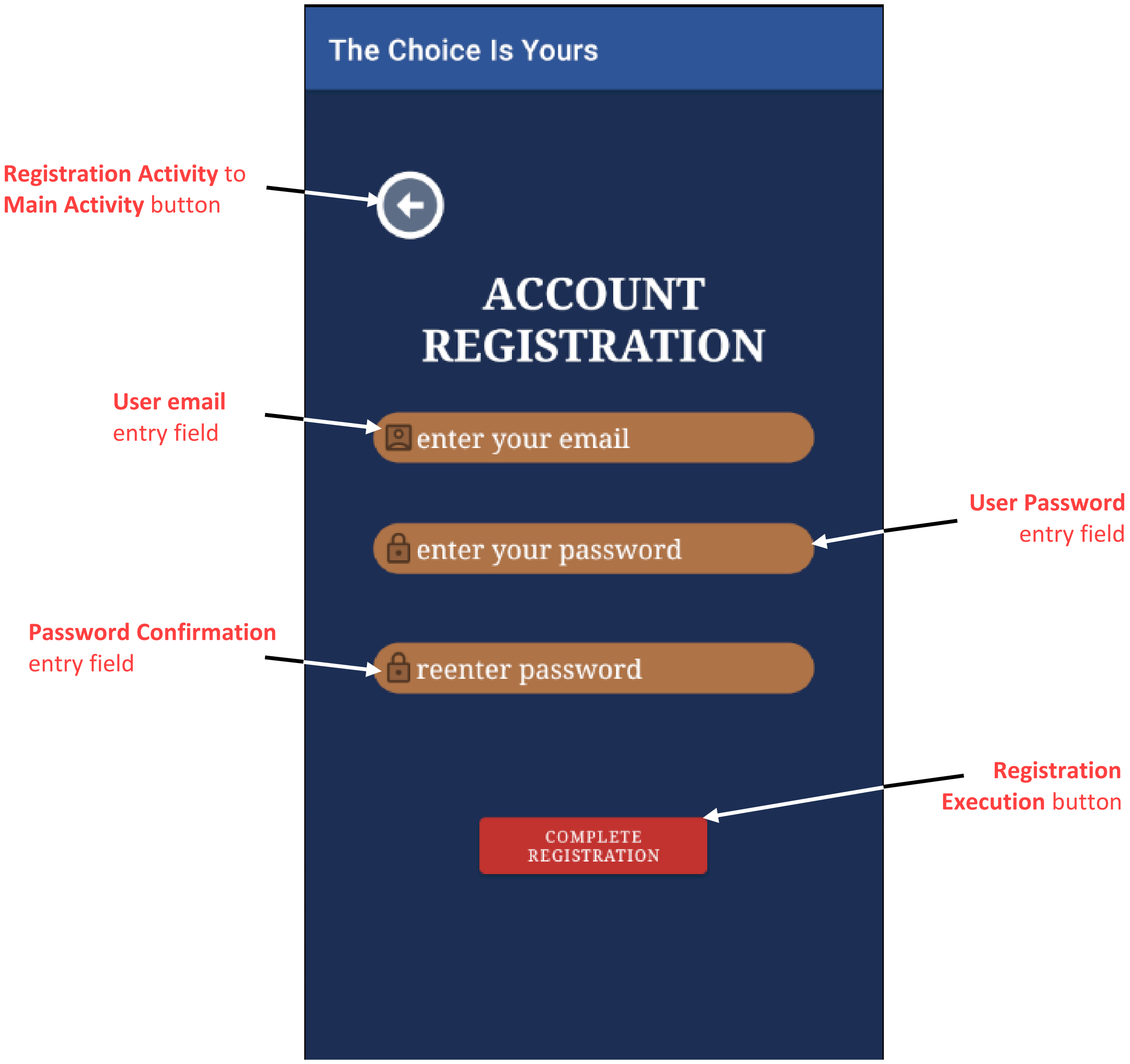

The Choice Is Yours
THE CHOICE IS YOURS
What Adventures Await You?
And What Path Will You Take?
CLICK HERE TO ACCESS YOUR LIBRARY
NEED TO LOGIN?
NEED TO REGISTER?
INSTRUCTIONS

Login Successful. Choose your Story!

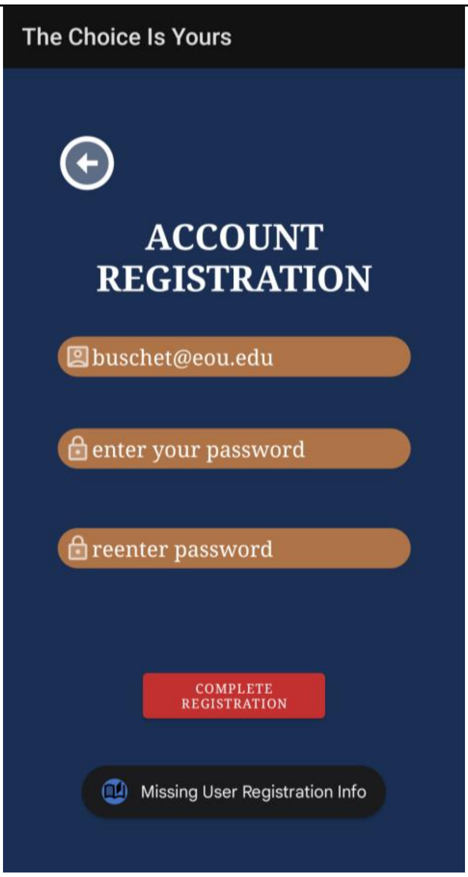
Example: Matching credentials, successful login.

3.1.3 User Registration Activity

- Interactable objects: Two (2) buttons and three (3) entry fields.
- Additional functionality: Registration – Creates new Firebase user profile.



- Functionality Detail: Email and matching password entries must be provided. There are four (4) possible outcomes when the registration button is selected:
    - If the email or both password fields are missing an entry, a toast (a small transient popup) message appears prompting that user-provided credential is required.
    - If all credential fields have entries, but the passwords do not match a toast appears specifying non-matching passwords and prompts reentry.
    - If all credential fields have entries, both passwords are matching, and a user account for the email does not already exist in Firebase, registration is successful and a toast message appears.
    - If all credential fields have entries, both passwords are matching, but a user account for the email already exists in Firebase, a registration failure toast appears.
    - See images below. *Toast messages appear with a black background when the device is set to *Dark Theme*.*



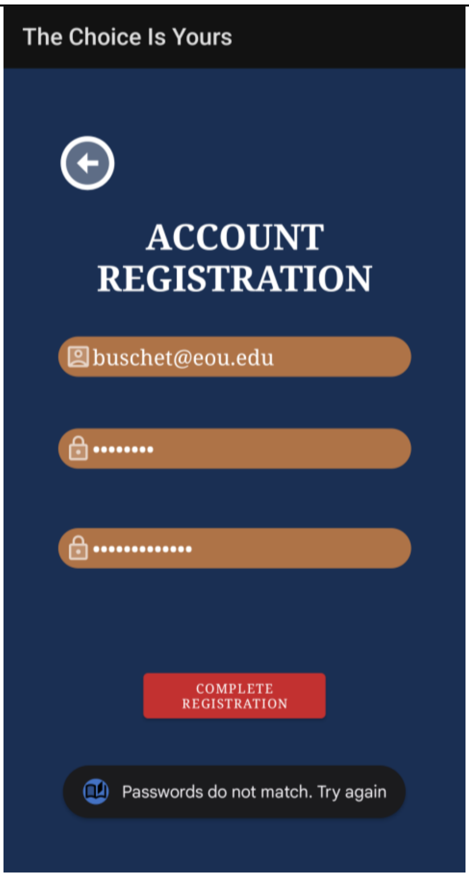
buschet@eou.edu

enter your password

reenter password

Missing User Registration Info

Example: Missing user password entries



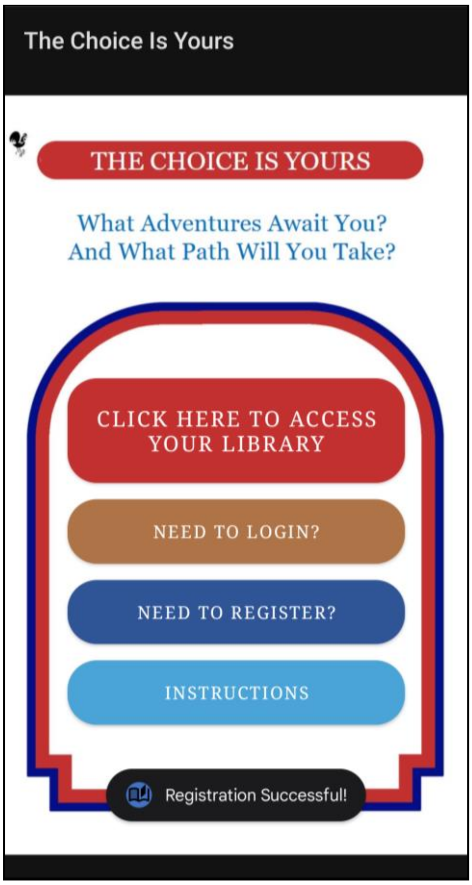
buschet@eou.edu

••••••••

•••••••••••••

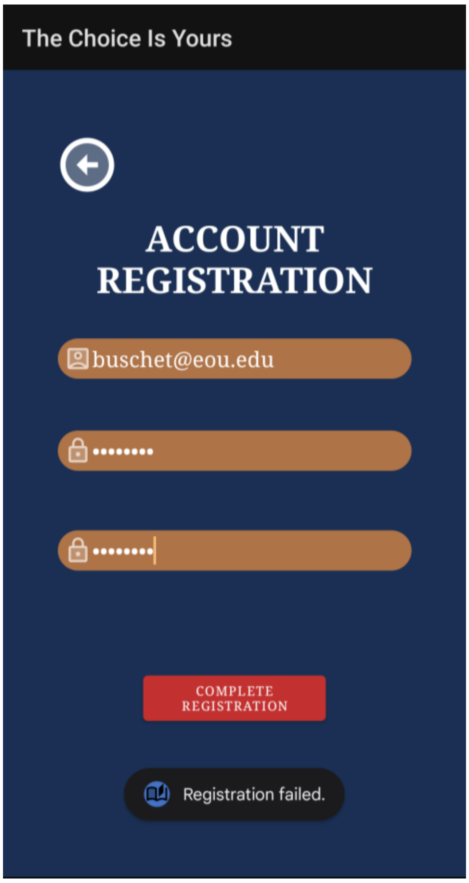
Passwords do not match. Try again

Example: Password entries mismatched


The Choice Is Yours
THE CHOICE IS YOURS
What Adventures Await You? And What Path Will You Take?
CLICK HERE TO ACCESS YOUR LIBRARY
NEED TO LOGIN?
NEED TO REGISTER?
INSTRUCTIONS

Registration Successful!

Example: New email and matching passwords



buschet@eou.edu

••••••••

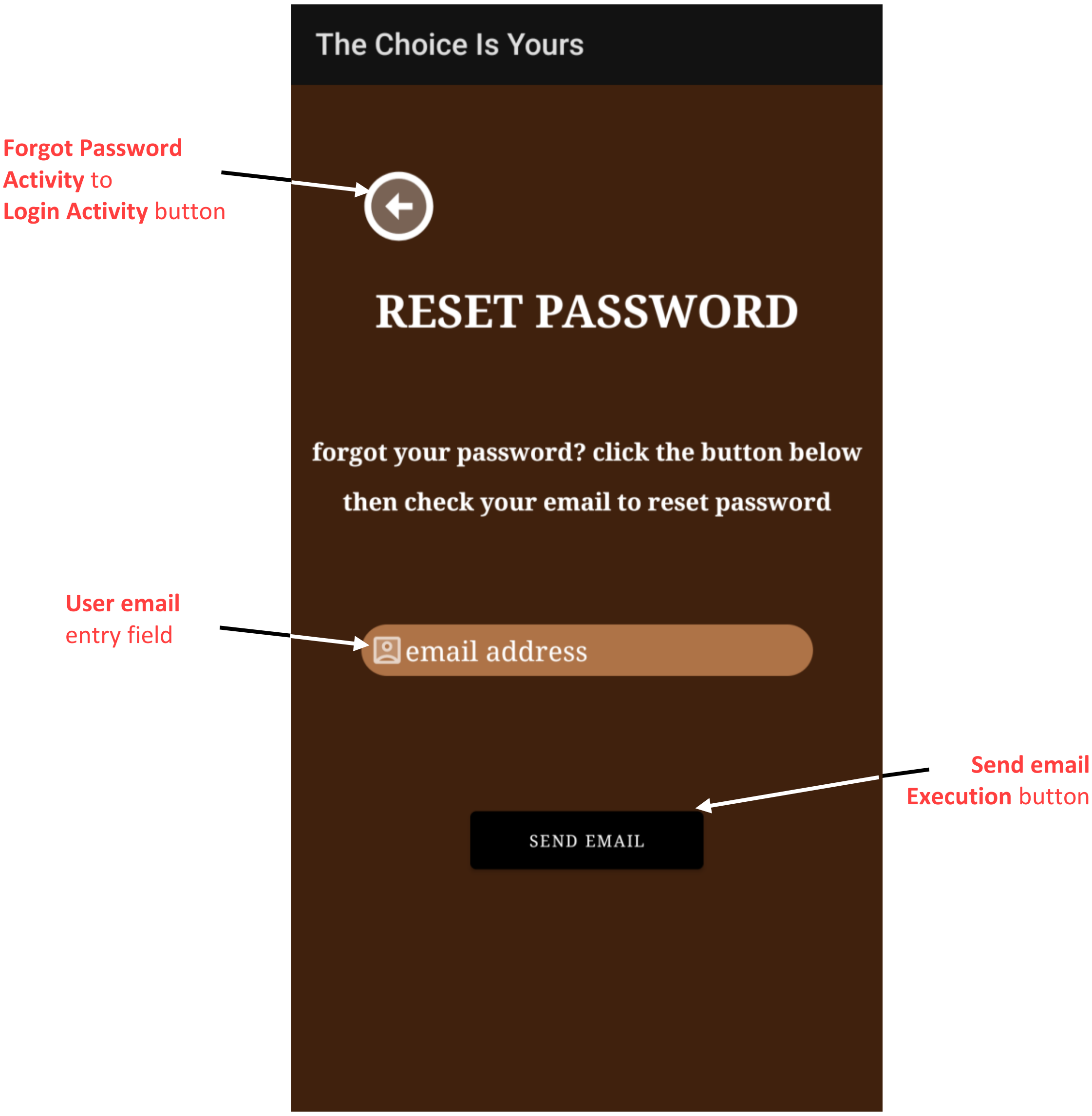
••••••••

Registration failed.

Example: Existing email and matching passwords

3.1.4 Forgot Password Activity

- Interactable objects: Two (2) buttons and one (1) entry field.
- Additional functionality: Send email – Firebase sends a message to the user so that the password may be reset.



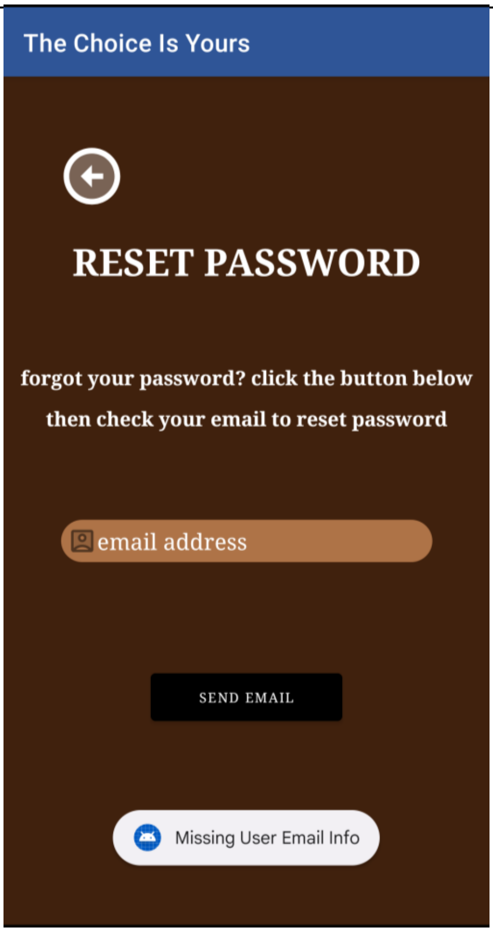
forgot your password? click the button below then check your email to reset password

Forgot Password Activity to **Login Activity** button

User email entry field

Send email Execution button

- Functionality Detail: An email address must be provided. There are three (3) possible outcomes when the send email button is selected:
    - If the email field is missing an entry, a toast message appears prompting that user-provided credential is required.
    - If the email field has an entry, but there is not a corresponding email in Firebase, a failure toast appears, and prompts for reentry.
    - If the email field has an entry and the entry matches an email credential in Firebase, the user receives an email for password reset and a message is sent to the corresponding email address. The screen returns to the User Login activity.
    - See images below.



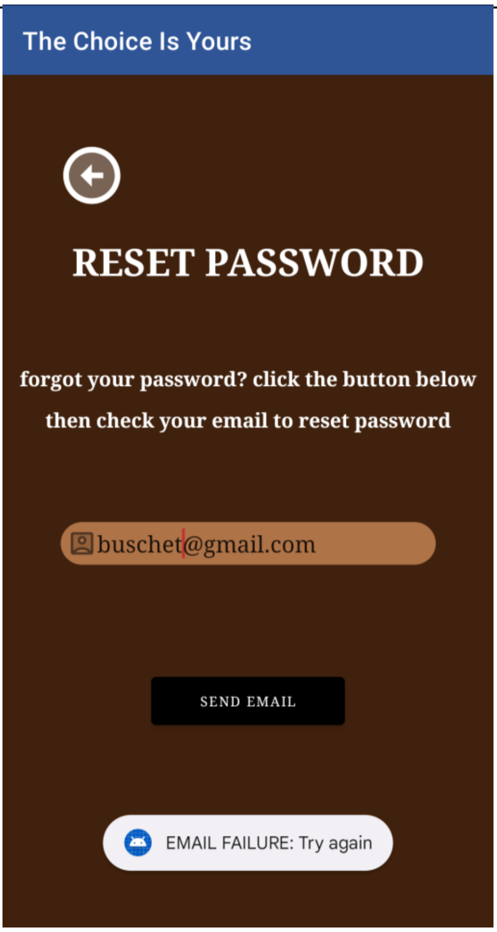
RESET PASSWORD

forgot your password? click the button below then check your email to reset password

email address

Missing User Email Info

Example: Email entry missing



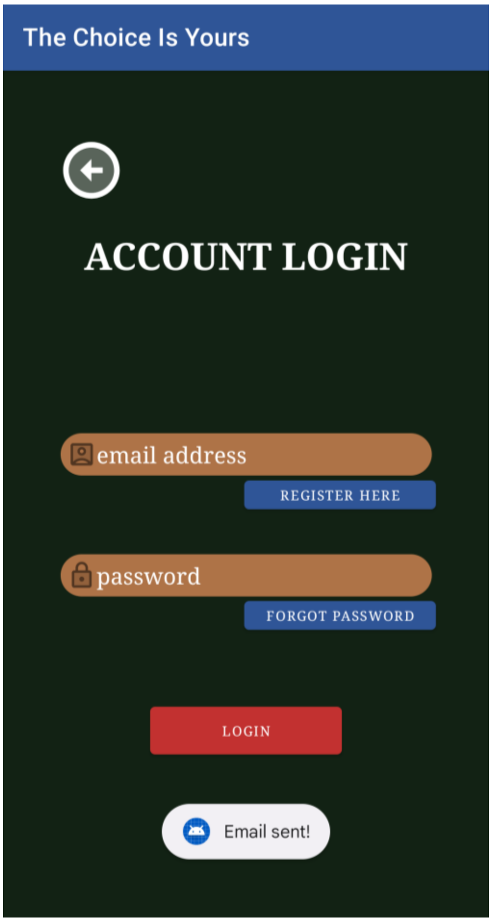
RESET PASSWORD

forgot your password? click the button below then check your email to reset password

buschet@gmail.com

EMAIL FAILURE: Try again

Example: Email provided not found in Firebase



ACCOUNT LOGIN

email address

password

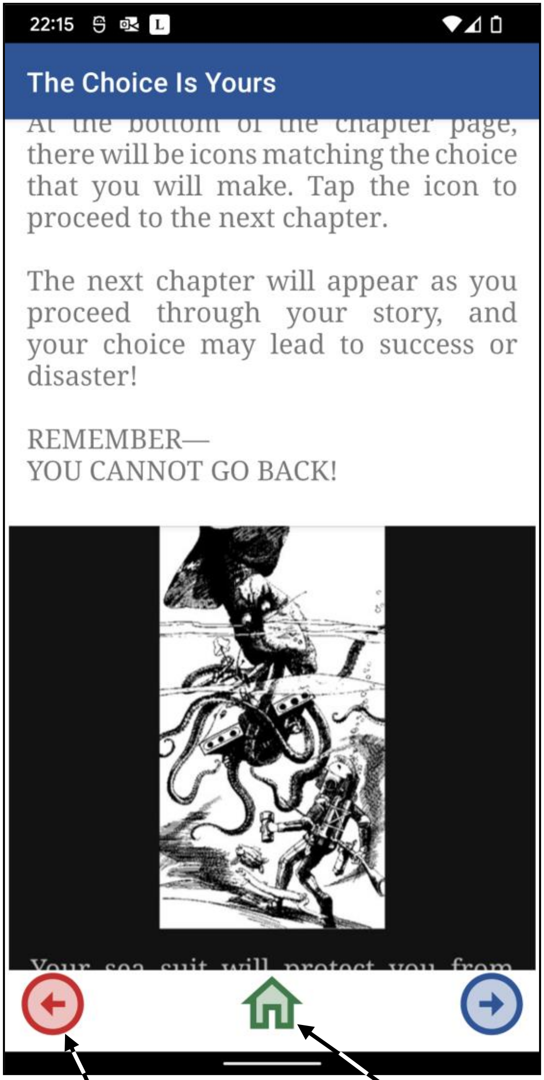
Email sent!

Example: Email address located in Firebase. Email sent to user and screen navigation to Login.

3.1.5 Instructions Activity

- Interactable objects: Three (3) buttons.
- Non-interactable objects: Instructions text, instructions images.
- Additional functionality: Instruction page navigation.

Default Light appearance



22:15

The Choice Is Yours

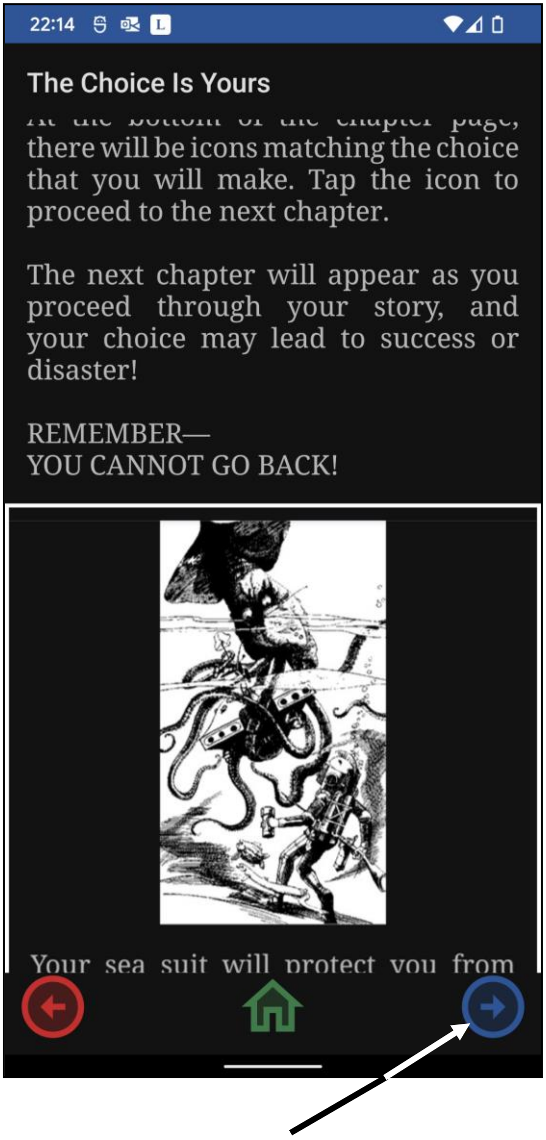
At the bottom of the chapter page, there will be icons matching the choice that you will make. Tap the icon to proceed to the next chapter.

The next chapter will appear as you proceed through your story, and your choice may lead to success or disaster!

REMEMBER—
YOU CANNOT GO BACK!

Your sea suit will protect you from

Dark Theme appearance



22:14

The Choice Is Yours

At the bottom of the chapter page, there will be icons matching the choice that you will make. Tap the icon to proceed to the next chapter.

The next chapter will appear as you proceed through your story, and your choice may lead to success or disaster!

REMEMBER—
YOU CANNOT GO BACK!

Your sea suit will protect you from

Previous page button

Instructions Activity to **Main Activity** button

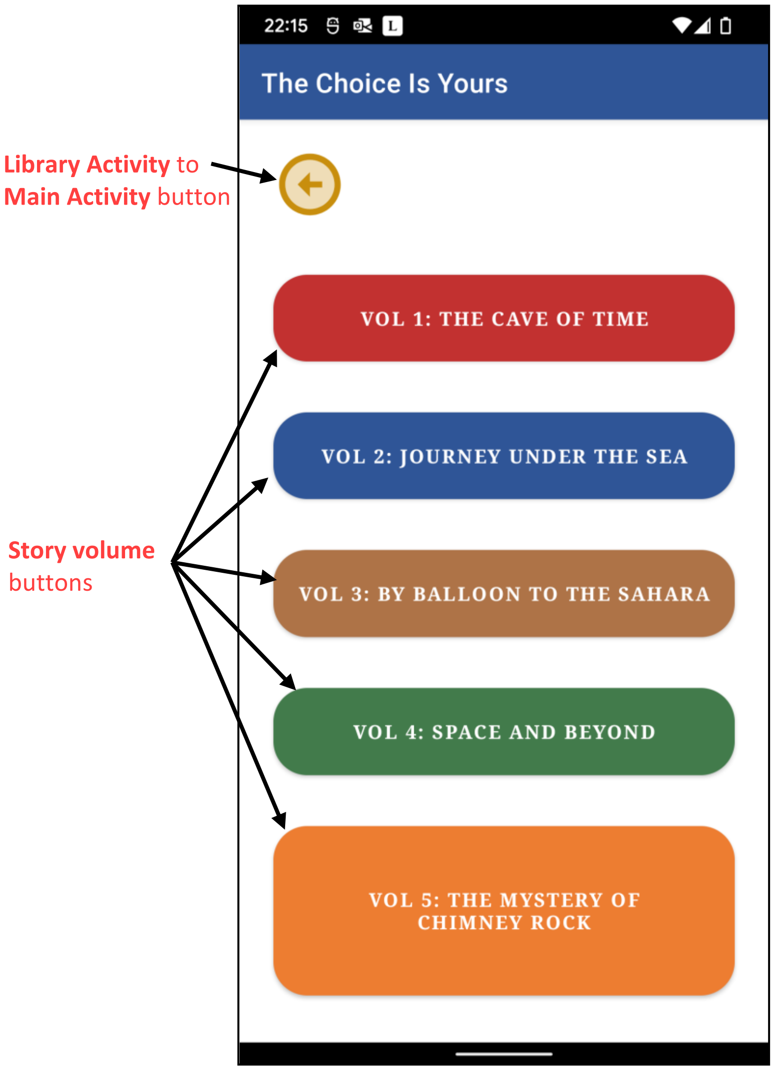
Next page button

- Functionality Detail: There are five (5) pages of instructions. The previous and next page buttons navigate the user over the instruction pages. A home button is provided so that the user can navigate directly to the main screen if desired.

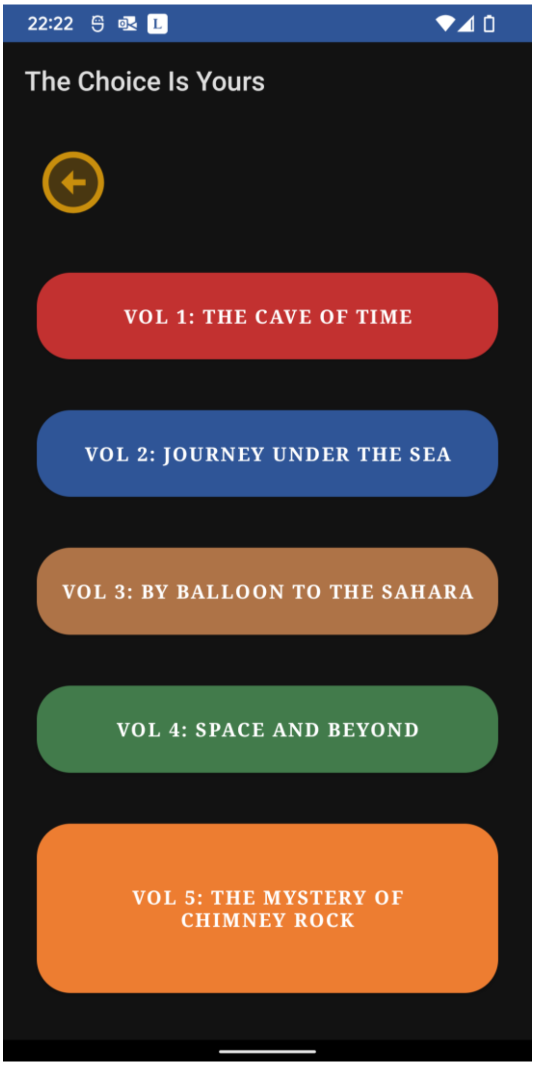
3.1.6 Story Library Activity

- Interactable objects: Six (6) buttons.
- Additional functionality: eBook choice.

Default Light appearance


22:15
The Choice Is Yours
Library Activity to Main Activity button
VOL 1: THE CAVE OF TIME
VOL 2: JOURNEY UNDER THE SEA
VOL 3: BY BALLOON TO THE SAHARA
VOL 4: SPACE AND BEYOND
VOL 5: THE MYSTERY OF CHIMNEY ROCK
Story volume buttons

Dark Theme appearance

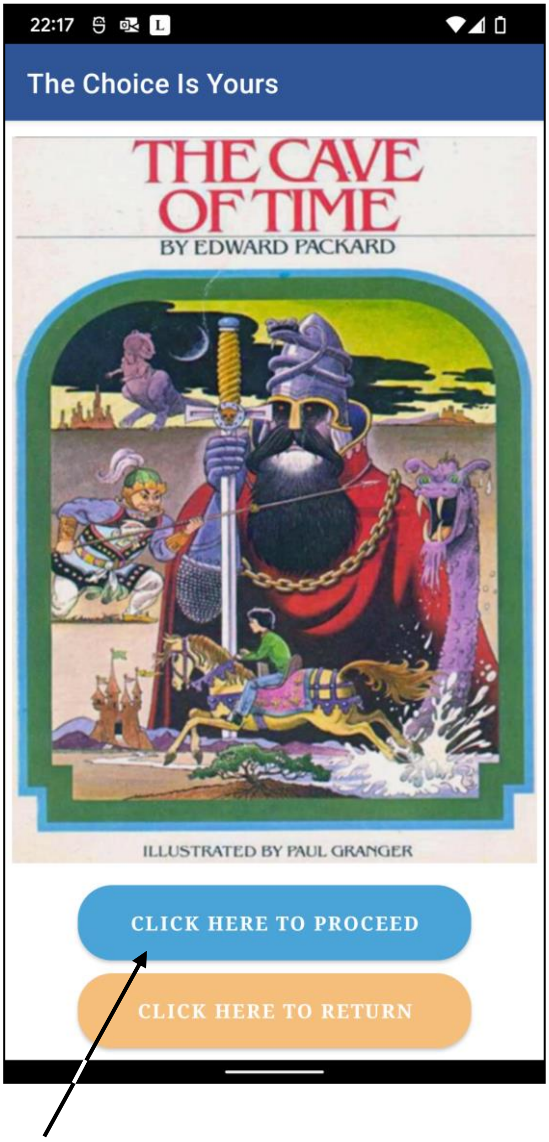

22:22
The Choice Is Yours
VOL 1: THE CAVE OF TIME
VOL 2: JOURNEY UNDER THE SEA
VOL 3: BY BALLOON TO THE SAHARA
VOL 4: SPACE AND BEYOND
VOL 5: THE MYSTERY OF CHIMNEY ROCK

- Functionality Detail: Each of the starter eBooks are displayed using a button with the volume number and title. Selecting one of the five buttons indicates which story contents to draw from. The data for each book is retained in the application's Assets directory in a subdirectory for each volume. Selecting a button indicates to the Library Activity which volume directory data to pass to the Book Cover and Book Scrolling Activities. It additionally triggers navigation to the Book Cover activity.

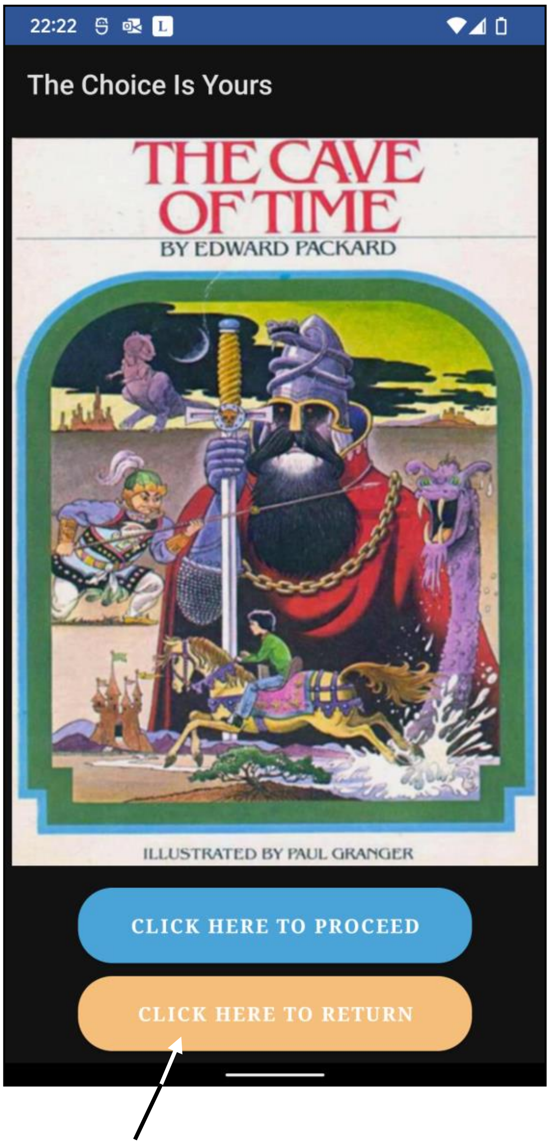
3.1.7 Book Cover Activity

- Interactable objects: Two (2) buttons.
- Additional functionality: None.

Default Light appearance



Dark Theme appearance

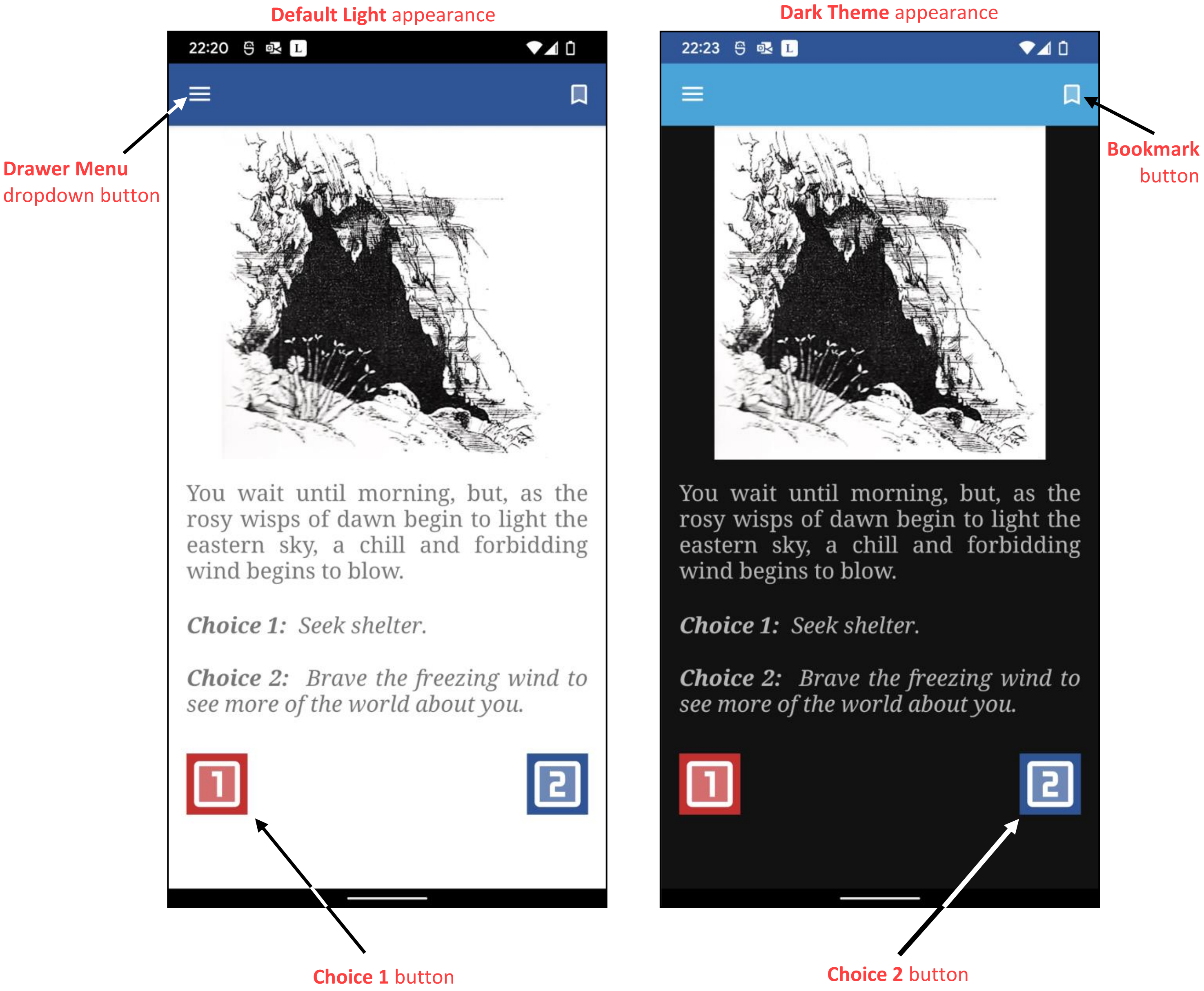


Book Cover Activity to **Library Activity** button

Book Cover Activity to **Book Scrolling Activity** button

3.1.8 Book Scrolling Activity

- Interactable objects: One (1) to four (4) story progression buttons, a bookmark button, and a drawer menu button.
- Non-interactable objects: Chapter images, chapter text.
- Additional functionality: Next chapter display.



You wait until morning, but, as the rosy wisps of dawn begin to light the eastern sky, a chill and forbidding wind begins to blow.

Choice 1: *Seek shelter.*

Choice 2: *Brave the freezing wind to see more of the world about you.*

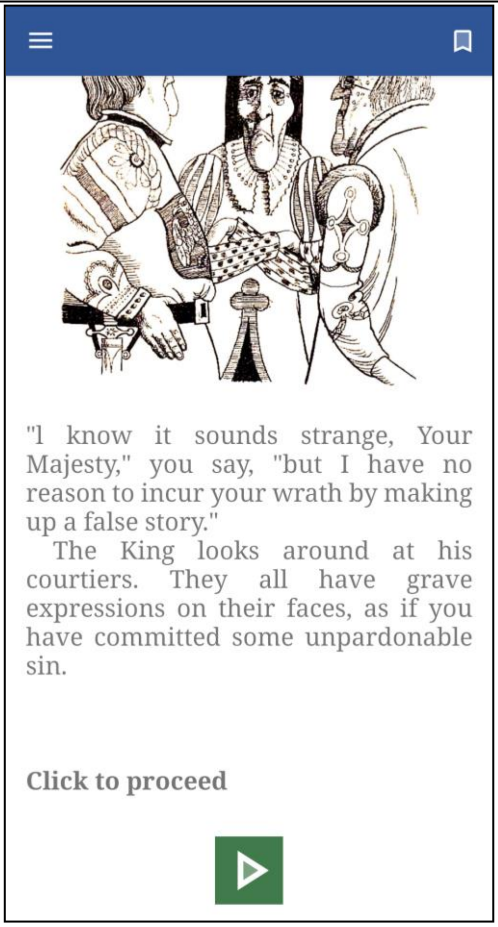
You wait until morning, but, as the rosy wisps of dawn begin to light the eastern sky, a chill and forbidding wind begins to blow.

Choice 1: *Seek shelter.*

Choice 2: *Brave the freezing wind to see more of the world about you.*

- Functionality Detail: The most frequent appearance of the Book Scrolling activity displays the chapter image on the top of the screen with the chapter contents displayed below. After the chapter narrative, the available choices are displayed. Below the choices are story progression buttons corresponding to the choices that the user has available for the next chapter to be displayed.

  Upon selecting a button, the scrolling activity is refreshed and updated with the contents of the new chapter, the chapter choices, and new chapter image (when indicated).



"I know it sounds strange, Your Majesty," you say, "but I have no reason to incur your wrath by making up a false story."

The King looks around at his courtiers. They all have grave expressions on their faces, as if you have committed some unpardonable sin.

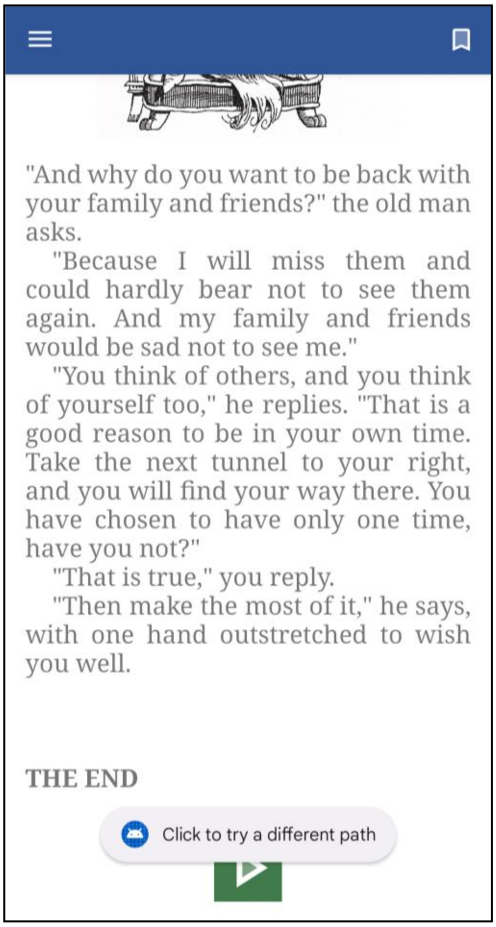
Click to proceed

Example no choice, predesignated chapter

As mentioned previously, one (1) to four (4) buttons appear depending on the next chapter/s available. There are two (2) scenarios when only one choice button appears; when there is a pre-designated chapter with no other choices or when *the end* of the story has been reached. For the former circumstance, the user selects the button and the display is refreshed and updated with the next chapter's contents per usual fashion.

For the latter circumstance, the last chapter contents are displayed and, "THE END," is presented on the screen. A toast message appears indicating the user to, "Click to try a different path." Selection of the button navigates the user back to the Book Cover activity.

When multiple choices are available, the choice buttons appear with numerals on them matching the options available for the next chapter contents. For two (2) choices, the buttons appear on the bottom left and right. For three (3) choices, buttons appear on the left, right, and center. For four (4) choices, and additional button below the 3-choice center button appears.



"And why do you want to be back with your family and friends?" the old man asks.

"Because I will miss them and could hardly bear not to see them again. And my family and friends would be sad not to see me."

"You think of others, and you think of yourself too," he replies. "That is a good reason to be in your own time. Take the next tunnel to your right, and you will find your way there. You have chosen to have only one time, have you not?"

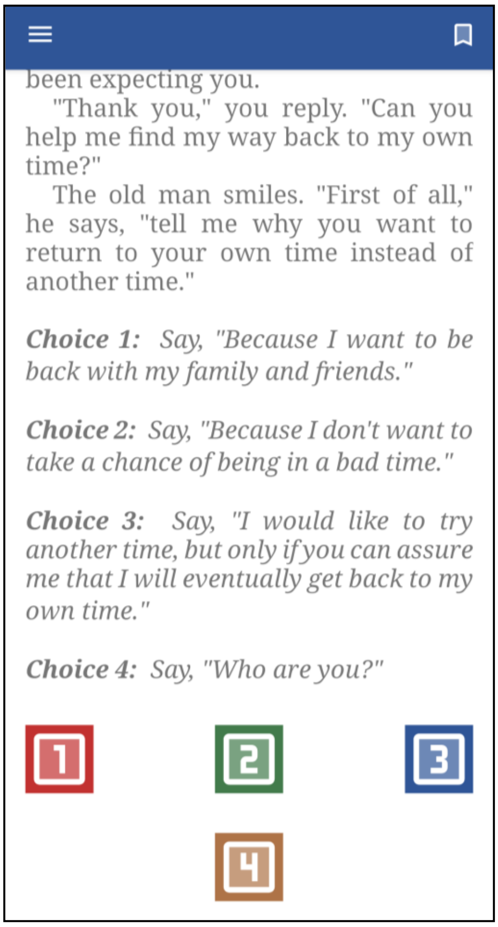
"That is true," you reply.

"Then make the most of it," he says, with one hand outstretched to wish you well.

THE END

Click to try a different path

Example THE END with toast message



been expecting you.

"Thank you," you reply. "Can you help me find my way back to my own time?"

The old man smiles. "First of all," he says, "tell me why you want to return to your own time instead of another time."

Choice 1: *Say, "Because I want to be back with my family and friends."*

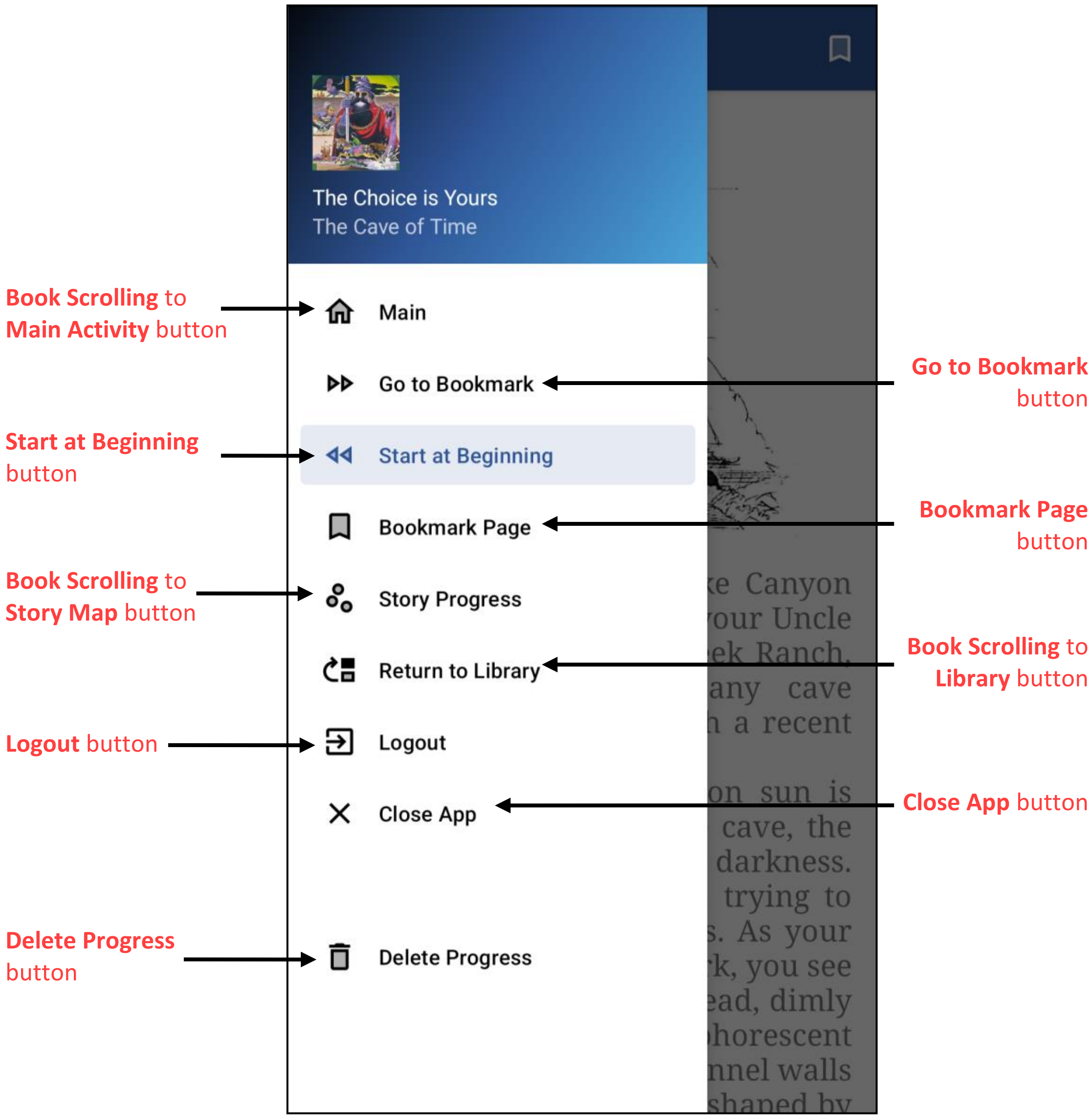
Choice 2: *Say, "Because I don't want to take a chance of being in a bad time."*

Choice 3: *Say, "I would like to try another time, but only if you can assure me that I will eventually get back to my own time."*

Choice 4: *Say, "Who are you?"*

Example Four (4) choices available

Upon selecting the bookmark button, an operation is performed transmitting the current chapter to the Firebase database. Finally, when selecting the menu drawer button, the menu display appears. The menu draw contains nine (9) operation buttons. The operation performed are described below.



Main: Performs a navigation intent from the Book Scrolling activity to the Main Activity.

Go to Bookmark: Accesses the Firebase database to determine the bookmarked chapter. The Book Scrolling activity then sets the display for the bookmarked page.

- If the user has not bookmarked a chapter for the current story, the default is bookmark is page 1.

Start at Beginning: Sets the chapter to page 1 of the book and refreshes the display to the chapter 1 contents.

Bookmark Page: Performs the bookmark operation the same as the main Book Scrolling activity performs.

Story Progress: Navigates the user to the Story Progress Map activity.

- If the user is not logged into their Firebase account, a toast message indicating that the user is not logged in and that a log in is required to access the story map. After the toast appears, the user in automatically navigated to the Login activity.

Return to Library: Performs a navigation intent from the Book Scrolling activity to the Library activity.

Logout: Performs a logout operation. Communication to Firebase for the logged in account is discontinued until the next login event.

- A logout successful toast appears. The user remains in the Book Scrolling activity on the same page. They are able to resume the reading.

Close App: The application is exited. It will not run in the background on the user's device.

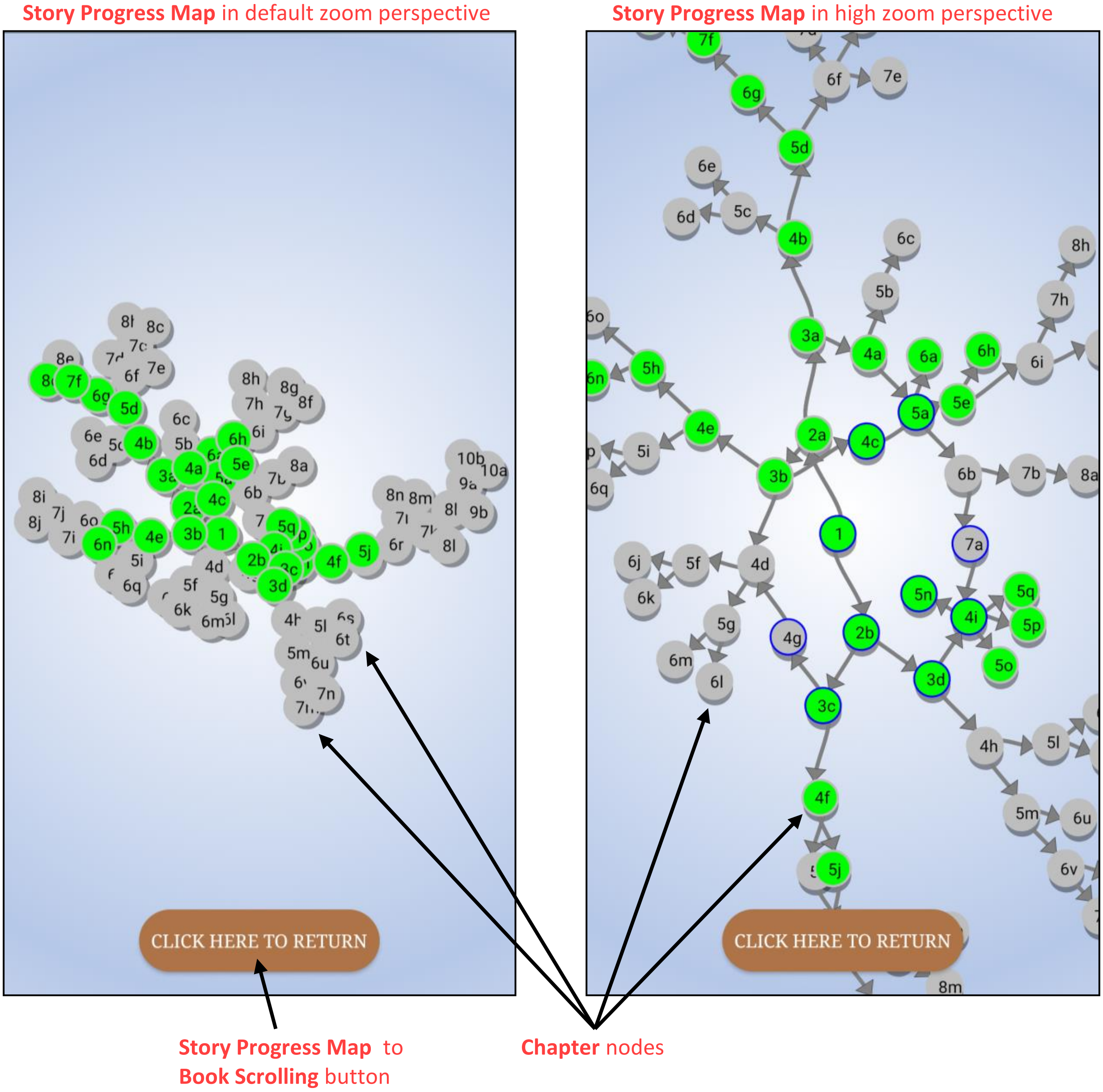
- A toast message saying, "Goodbye," appears, and the application returns to the user's device display.

Delete Progress: The user's Firebase database is accessed. For the book displayed (i.e. not every book in the library) booleans for each chapter *visited* are toggled to \`false\`. The bookmark is set for page 1.

- If the user is not logged in, a toast message appears indicating that a login is required to delete progress.
- If the user is logged into a Firebase account, upon selection a warning popup is displayed asking, "Are you sure?" with yes/no options. This is a safety check in the circumstance that the user has accidentally selected the button. If the user indicates *No*, the popup display disappears and the menu drawer remains open. If the user indicates *Yes*, a toast appears indicating that the progress has been reset.

3.1.9 Story Progression Map Activity

- Interactable objects: One (1) return to Book Scrolling activity button. Nodes representing chapters of the book.
- Additional functionality: GraphStream display controls.



- Functionality Detail: Aside from the single activity-to-activity navigation button, the screen display is controlled via GraphStream, and Java software program library that allows for dynamic controls of graphs. User controls are intuitive, for example two-point-touch pinching for zooming in and out the view perspective, selecting an object by touch, etc.

The graph’s nodes are represented as circles and each is selectable by the user. Nodes can be *dragged* where the selected nodes and those connected will be repositioned according to the user’s control. Nodes representing chapters that have not been visited are gray. Nodes representing chapters that have been visited appear as green.
The Firebase database stores boolean `true/false` values for each chapter of each book. These values are pass back to the Story Progression Map to determine color highlighting for visited chapters.

4. Functional Design

4.1 Sequence of User Events

The user workflow beginning with application startup has two possibilities. Either the user can log into their Firebase user account before proceeding to the library and subsequent story, or they can access the library and story immediately. In either instance, the user proceeds to the library and then selects which eBook they would like to read.

From the Library, the next activity displays the Book Cover of the story. The Book Cover, beyond being displayed as aesthetics for the user, serves as a return mark upon the user reaching one of the many endings of the book. From the Book Cover, the user proceeds to the first page of the book.

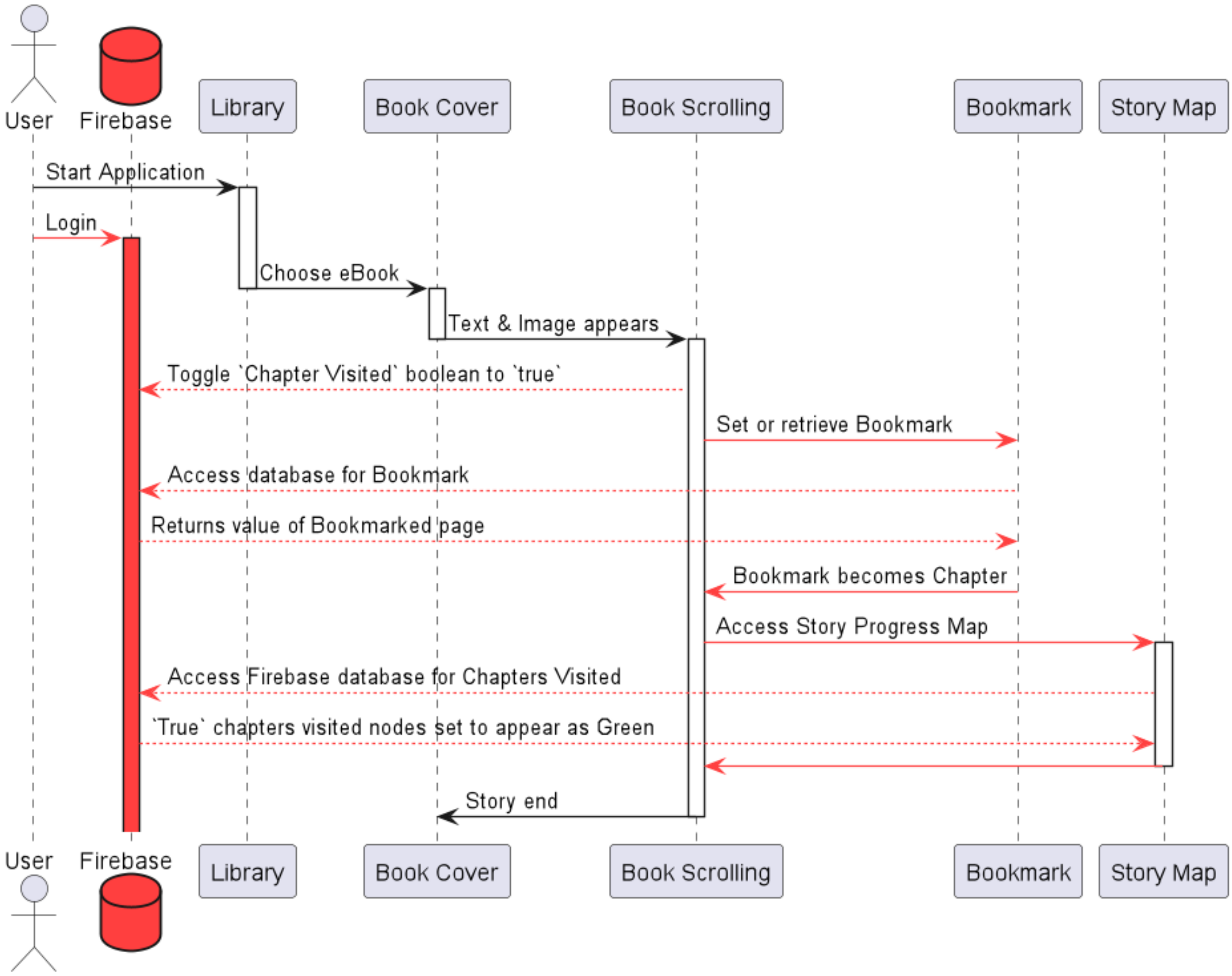
While the user accesses each chapter, if the user is logged in, the application transmits the current chapter value to the Firebase database and sets the *Chapter Visited* value to `True`. If the user is not logged in, there is transmission of data to Firebase, but otherwise story reading and control in unchanged.

For the Bookmark option, the user can either set or return to a bookmarked page at any time during the scrolling activity. For both operations, the Book Scrolling activity accesses the Firebase database. This means that the user must be logged in for Bookmark operations to occur.

Finally, within the Book Scrolling activity the user can access the Story Map at any time from the drawer menu. Because the values for whether a chapter has been visited is stored in the database, the user must be logged in. With an active login, the Book Scrolling activity transfers to the Story Progress Map activity which accesses the Firebase for the values for the chapters visited. For those values that are `True`, corresponding nodes are set to appear as green.

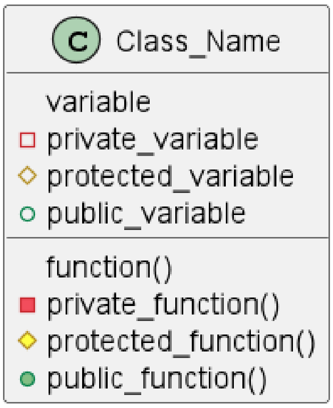
4.1.1 Sequence Diagram

The next image shows a sequence diagram for the above-described workflow. Colored items indicate user-login-required operation. Solid arrows indicate application functionality while dotted arrows indicate data transmission to the online Firebase database.



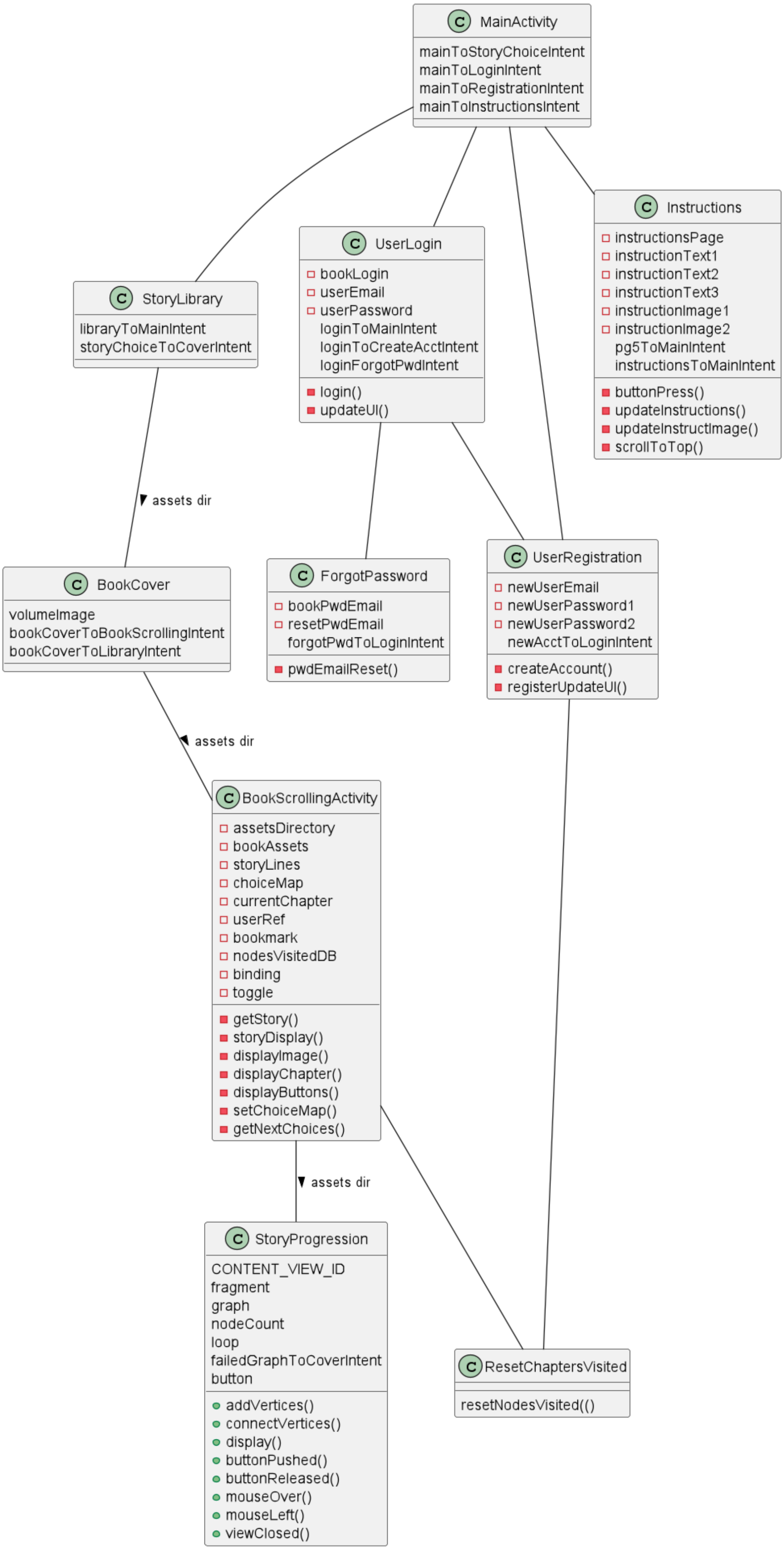
4.2 Class Diagram and eBook Assets

For an Android application, an *Activity* is a single-screen display that the user interacts with. As such, an Android activity class diagram does not follow the conventional class-relationships that typical class diagrams might display, such as inheritance, dependency, interface, etc. The class diagram representation provided in this document relates the various activities and classes as related by activity-to-activity intents. Meanwhile, some activities also have functionality beyond its display appearance, for example user login and registration or functions that transmit or receive data with the user’s Firebase database. Class variables and functions are indicated according to the key displayed.



Class key

4.2.1 Class Diagram and Class Relationships



There is one class of the application that is not an Android activity, `ResetChaptersVisited`. This class is accessed by either the Book Scrolling activity (via its drawer menu) or the User Registration activity.

4.2.2 eBook Assets

Each eBook has its own directory under the application's Asset directory. These directories contain the following file types with the described naming designations where the character, "X," indicates a volume value:

volX_contents.txt – The contents.txt file has the text for the eBooks chapters, as well as the choices for each chapter and the updated chapter image. An example of the text format appears as:

```
Part.1a  You've hiked through Snake Canyon once before while visiting …
Choice.1a.2a Decide to start back home.
Choice.1a.2b Decide to wait.
p1a  image vol1_files/vol1_picture1a.jpg
```

The Story Scrolling activity scans the contents.txt file for part (chapter) value and then uses that value, in this example `1a`, to determine the related choices for that chapter. Additionally, the chapter value is used to determine if there is a new image to set with the chapter text. For the chapter itself, all text is included on a single line. For choices the string, `Choice` designates the line as a choice, followed by the *current chapter value*, and then the *next chapter value*. When an on-screen choice button is selected, the corresponding *next chapter value* is used by the Book Scrolling activity to display the next chapter (by scanning contents.txt file for the matching chapter value). Finally when a line has a new image indicated, in this example `p1a`, the Book Scrolling activity accesses the Assets directory and sets the corresponding image. The file name and directory are used to indicate which .jpeg file will be opened and then placed.

volX_cover.jpg – This file is used for the Book Cover activity. It serves as the image displayed on the activity's background.

volX_header.jpg – This file is a thumbnail version of the cover.jpg. It is used for the image in the Book Scrolling activity menu drawer.

volX_nextChoices.txt – This file contains each chapter and the chapters that it navigates to according to the number of choices displayed at the end of a chapter. An example of the contents is:

```
4e 5h 5i
4f 5j 5k
4g 4d
4h 5l 5m
4i 5n 5o 5p 5q
```

This file is used to determine the number of choice buttons displayed in the Book Scrolling activity per chapter, and which chapters those buttons navigate to.

volX_picturexx.jpg – Images used in the Book Scrolling activity. Aside from the volume number, as demonstrated in the contents.txt example the file name indicates which chapter the image first appears.

volXNodeNames.txt – A file with the chapter values listed. For example:

```
1
2a
2b
3a
…
```

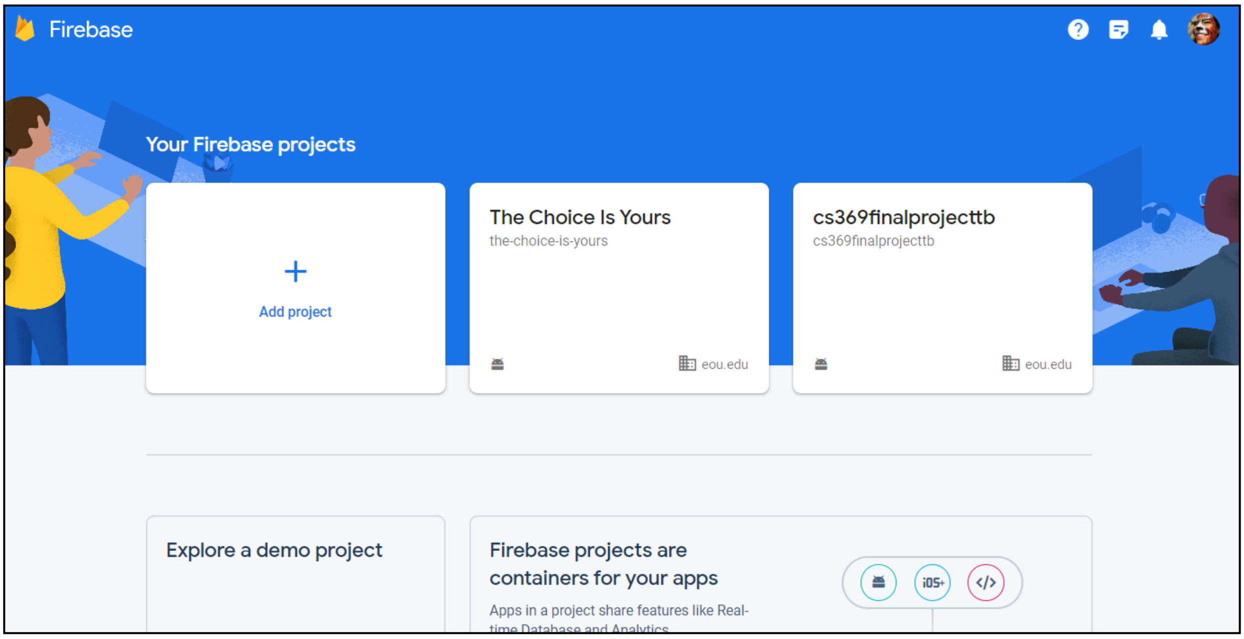
This file is used with the Story Progress Map activity to create the GraphStream map specific for that volume. When the Story Progress Map is accessed from the Book Scrolling activity, an *extra* data value for the volume directory in the Assets folder is included in with the Activity-to-Activity intent. Then the Story Progress Map activity uses the values in the file to name the nodes used in the GraphStream display

volXTree.txt – This file is used with the Story Progress Map activity. It's used to assign the book's chapter *nodes* and *edges* connecting each node. The Story Progress Map activity creates a directed graph where each edge has a direction according to the proceeding chapter and the next chapter. Edges appear as arrows in the direction of the story from beginning to an end node.

5. Firebase Authentication & Database

5.1 Firebase Authentication

The application utilizes the Firebase application programming interface (API) to perform program operations. The API has a number of functions that integrate the application with the online services devoted to the app. The below image is of my (the programmer's) Firebase webpage user console.



Your Firebase projects

The Choice Is Yours
the-choice-is-yours

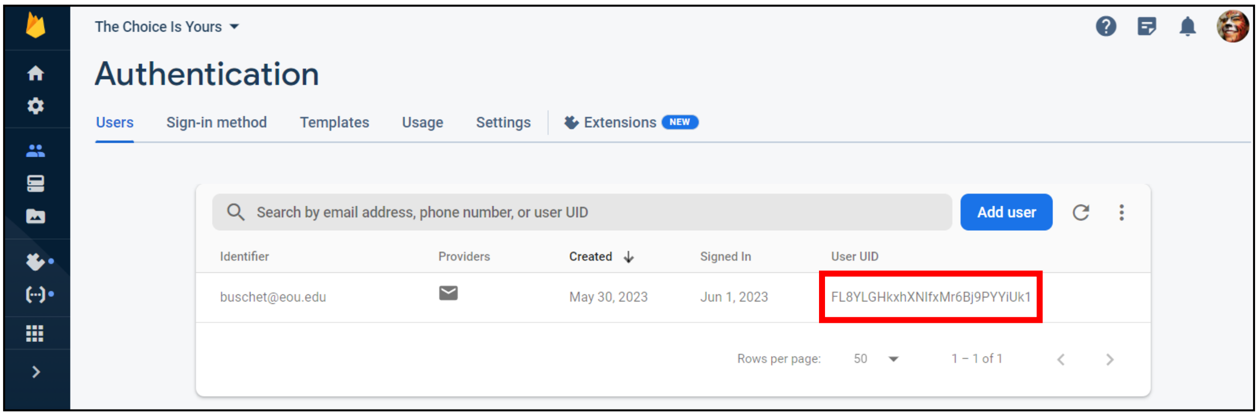
cs369finalprojecttb
cs369finalprojecttb

Explore a demo project

Firebase projects are containers for your apps

Apps in a project share features like Real-time Database and Analytics

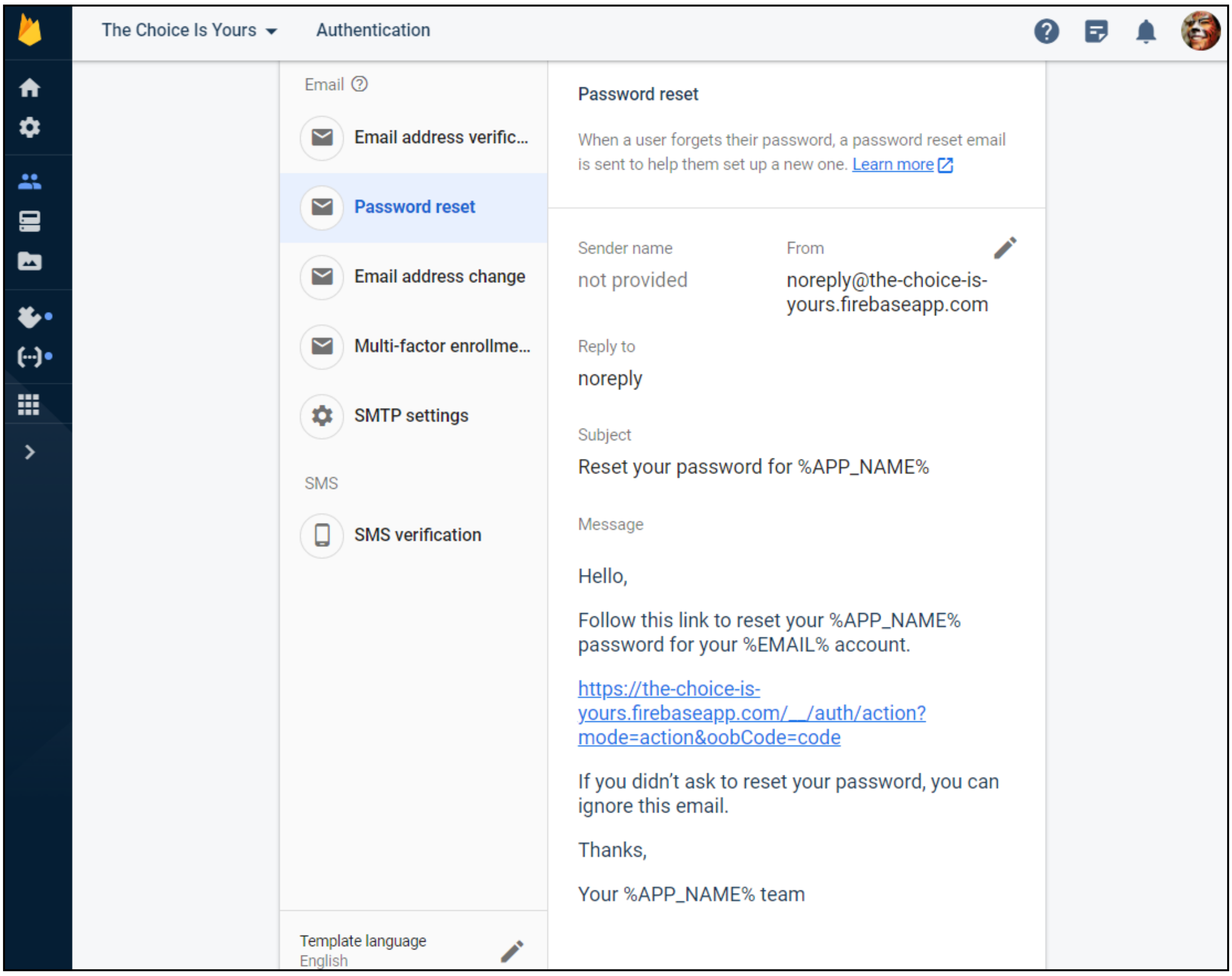
When a new user is created using an email and password, their credentials are retained in the online repository. The Firebase API generates a random 28-character User Unique ID (UUID), which is used in the application for the user's corresponding Firebase database repository.



Authentication

Identifier	Providers	Created	Signed In	User UID
buschet@eou.edu	✉	May 30, 2023	Jun 1, 2023	FL8YLGHkxhXNIfxMr6Bj9PYYiUk1

In the event an account user forgets their password, the Firebase API is again used to send the user an email to generate a new password. The template for the password reset message is displayed below.



Password reset

When a user forgets their password, a password reset email is sent to help them set up a new one. Learn more

Sender name
not provided

From
noreply@the-choice-is-yours.firebaseapp.com

Reply to
noreply

Subject
Reset your password for %APP_NAME%

Message

Hello,

Follow this link to reset your %APP_NAME% password for your %EMAIL% account.

https://the-choice-is-yours.firebaseapp.com/__/auth/action?mode=action&oobCode=code

If you didn't ask to reset your password, you can ignore this email.

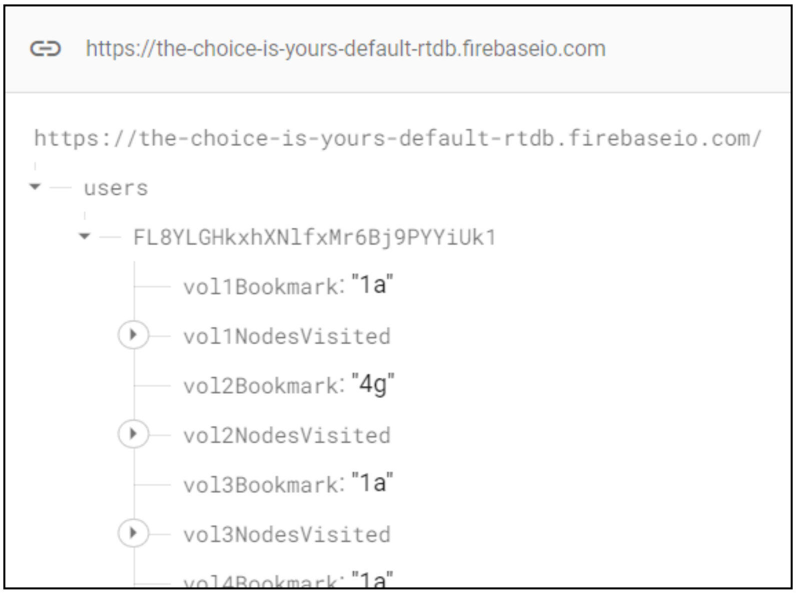
Thanks,

Your %APP_NAME% team

Template language
English

5.2 Firebase Database

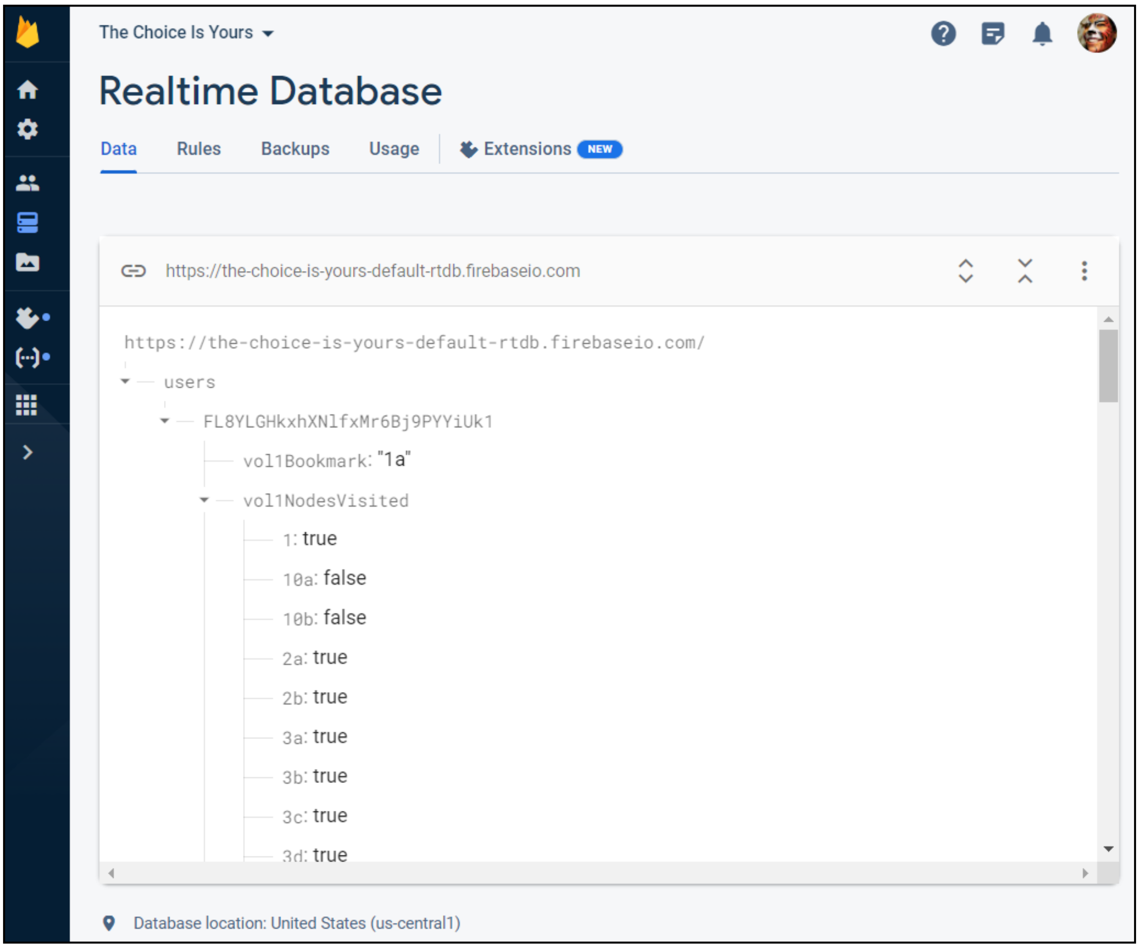
As mentioned, the Firebase database is used to retain the user's data concerning bookmarks and chapters visited for the contents of each eBook. Firebase uses a NoSQL mechanism for data storage.



```
https://the-choice-is-yours-default-rtdb.firebaseio.com

https://the-choice-is-yours-default-rtdb.firebaseio.com/
▾ — users
    ▾ — FL8YLGHkxhXNlfxMr6Bj9PYYiUk1
          — vol1Bookmark: "1a"
        ▸ — vol1NodesVisited
          — vol2Bookmark: "4g"
        ▸ — vol2NodesVisited
          — vol3Bookmark: "1a"
        ▸ — vol3NodesVisited
          — vol4Bookmark: "1a"
```

The `NodesVisited` values are a boolean, where the aforementioned volXNodeNames.txt is used to designate each chapter value used as a node for use in the Book Progression Map activity.



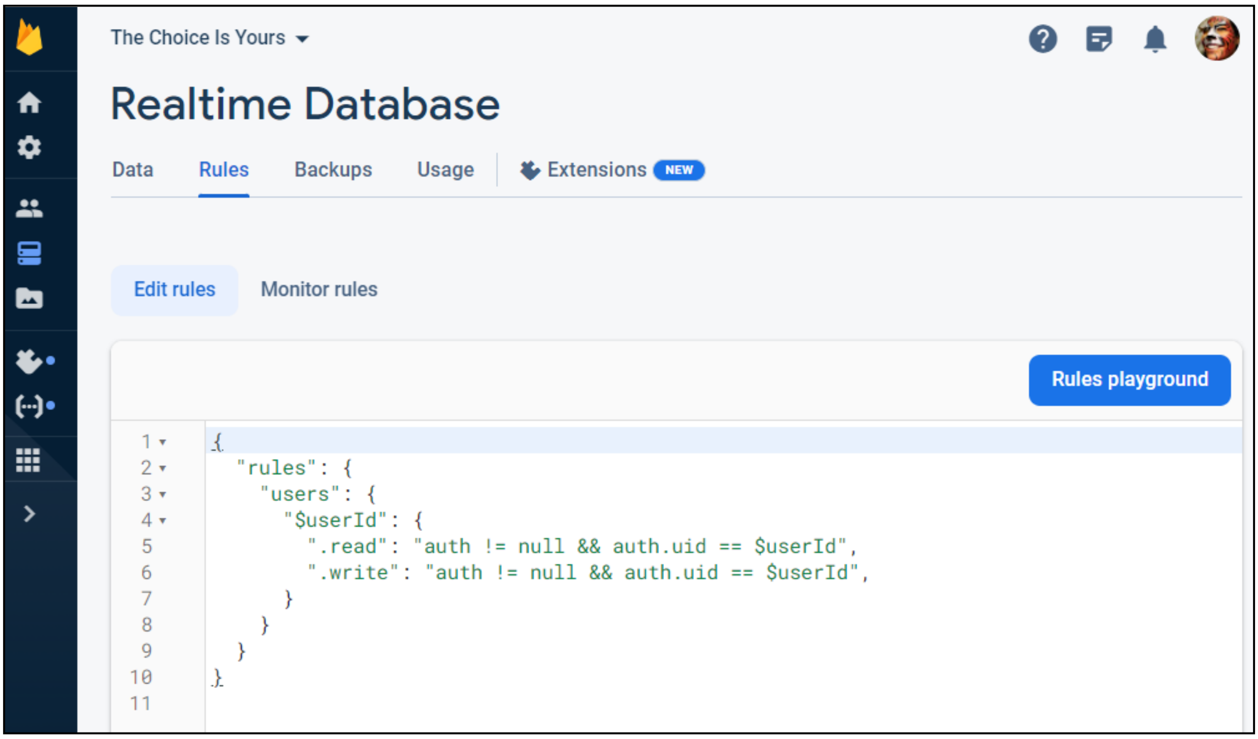
```
https://the-choice-is-yours-default-rtdb.firebaseio.com

https://the-choice-is-yours-default-rtdb.firebaseio.com/
▾ — users
    ▾ — FL8YLGHkxhXNlfxMr6Bj9PYYiUk1
          — vol1Bookmark: "1a"
        ▾ — vol1NodesVisited
              — 1: true
              — 10a: false
              — 10b: false
              — 2a: true
              — 2b: true
              — 3a: true
              — 3b: true
              — 3c: true
              — 3d: true
```

Database location: United States (us-central1)

5.2.1 Database Security

The Firebase database has been designed to use rules to prevent access to a user's data unless the user's account credentials and UUID match the user designation in the database. Only the application service owner (myself) has access to a user's email address, UUID, and database information.



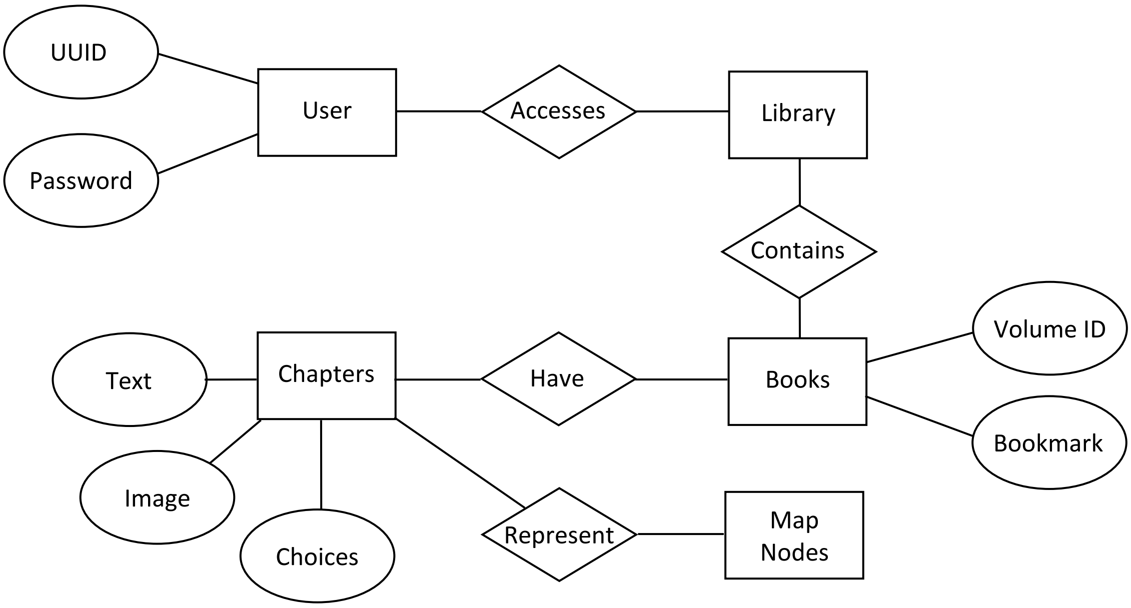
```
{
  "rules": {
    "users": {
      "$userId": {
        ".read": "auth != null && auth.uid == $userId",
        ".write": "auth != null && auth.uid == $userId",
      }
    }
  }
}
```

That said, I do not have access to a user's application password, and I am not able to manipulate data other than to disable an account, delete an account, manually send a new password request to the user's email address, delete database data, or add database data. I cannot change existing database data.

Given that this application is not being distributed for public use, users do not need to provide sensitive information for use, such as financial account data or personal demographic information. As such, there is a certain amount of anonymity on the part of the user.

5.3 Entity Relationship Diagram

Interaction and utility of data between entities are displayed below.



6. System Design

The application was created using the JetBrains integrated development environment (IDE) software IntelliJ IDEA Community Edition 2021.3.3 (Community Edition). The IDE has additional plugins for Kotlin (213-1.8.10, API version 1.8), Android (API 33 Platform), Gradle (Android-Gradle SDK Plugin Version 7.0.0), UI Designer, Git, GitHub. The software development kit (SDK) OpenJDK JDK 19.0.2 was used for program configuration settings.

The Gradle build tool is used to configure application dependencies for implementation using the Android SDK, integration with Firebase, and use of GraphStream Android-compatible libraries (gs-algo-2.0.jar, gs-core-2.0.jar, and gs-ui-android.aar).

Git and GitHub has been used as a version control system while creating the application.

7. Error Handling and Exception Handling

Most error and exception handling uses toast messages to indicate the app’s inability to complete operations (such as those requiring the user authentication). A characteristic of the Kotlin programming language is the use of *Null Safety*, a feature where a reference cannot be assigned a null value (no value) risking program failure. Some errors, handled by the Android SDK, causes the application to automatically return to the Main activity or close the application.

8. Testing and Quality Assurance

As the application does not manipulate or create data beyond transmitting simple values retained in the Firebase database, unit testing does not serve much applicability. The bulk of testing performed involves executing the application itself and checking functionality. Foreseeable application use that may introduce error (e.g. attempting login without a password or accessing the Story Progress Map without a login) has been experimented with during program execution. Most bugs observed during testing is due to inadequately formatted book Assets, namely text files such as volX_contents.txt, volX_nextChoices.txt, etc. These are corrected as they are identified.

9. Appendices

UML Images generated using a IntelliJ IDEA using the PlantUML plugin: https://plantuml.com/

9.1 Software Design References

Google for Developers. Guide to app architecture.
https://developer.android.com/topic/architecture

Google for Developers. Intents and Intent Filters.
https://developer.android.com/guide/components/intents-filters

Firebase website URL.
https://console.firebase.google.com/

Package Summary – Firebase SDKs for Android that have Kotlin extension (KTX) libraries.
https://firebase.google.com/docs/reference/kotlin/packages

Firebase website URL.
https://console.firebase.google.com/

GraphStream Tutorials – Graph Visualization.
https://graphstream-project.org/doc/Tutorials/Graph-Visualisation/

GraphStream for Android.
https://github.com/graphstream/gs-ui-android

GitHub The Choice is Yours application URL.
https://github.com/Tamsters24/The_Choice_Is_Yours

Sheehy, Steve. “CS 401: Capstone - Course Syllabus” 2022. PDF file.

9.2 eBook References

Packard, Edward. “The Cave of Time.” Bantam Books. 1979.

Montgomery, R.A. “Journey Under the Sea.” Bantam Books. 1979.

Terman, D. “By Balloon to the Sahara.” Bantam Books. 1979.

Montgomery, R.A. “Space and Beyond.” Bantam Books. 1980.

Packard, Edward. “The Mystery of Chimney Rock.” Bantam Books. 1980.