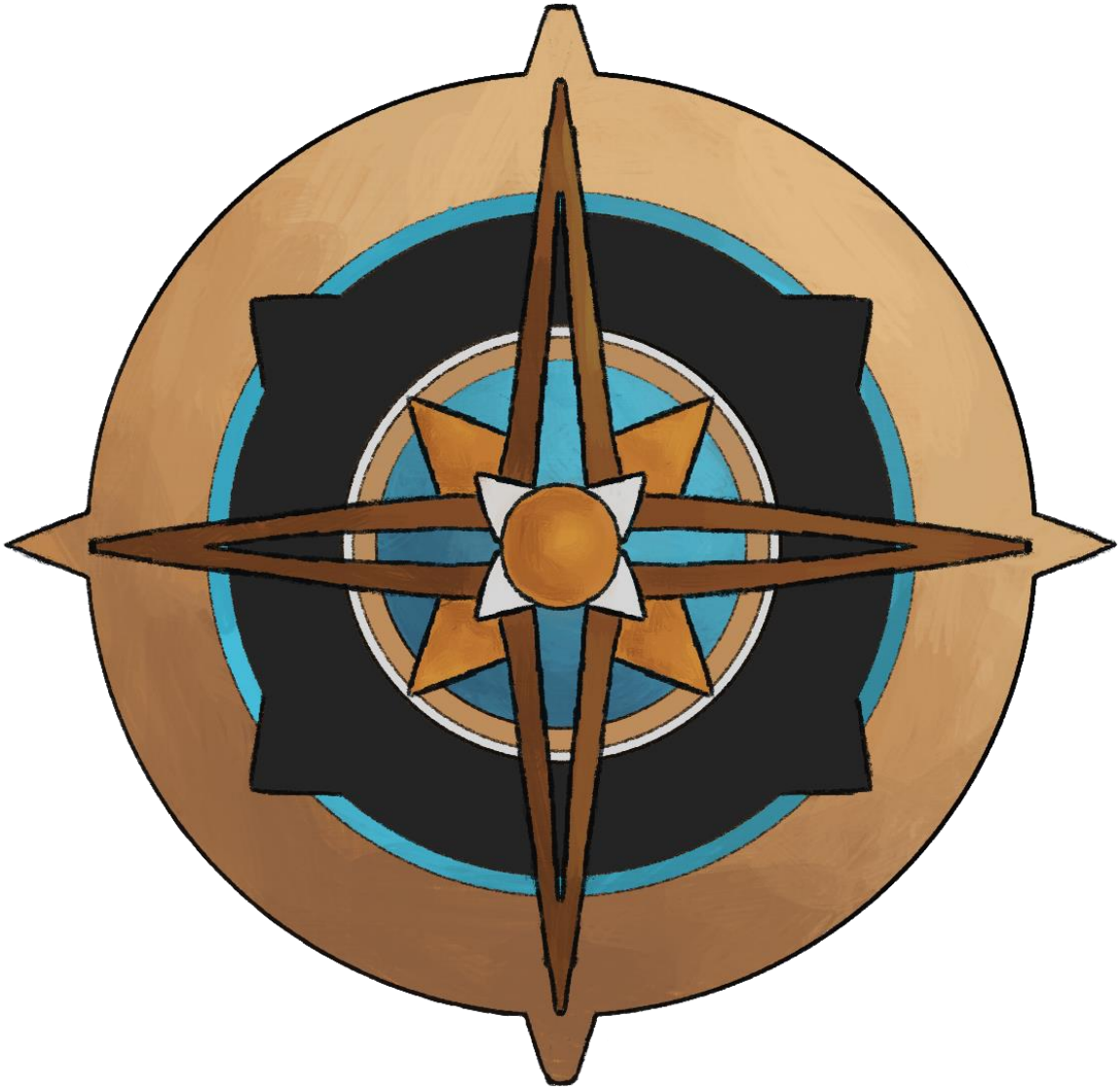
March 2021

Installation and User Instructions

Starfinder Companion 2020

Gitlab: https://gitlab2.eeecs.qub.ac.uk/40178070/starfinder-companion-2020



March 2021

Contents

March 2021

Installation

The installation of the application is straightforward. There is a developer install process and upon release there will be a release install process. Both processes are only certified for the Windows platform.

1. Developer Install

1.1 Installing Flutter

In this section, we will focus on the Flutter install process. This guide will follow the official Flutter install process found here (https://flutter.dev/docs/get-started/install/windows).

1.1.1 Minimum Requirements

Operating Systems	Windows 7 SP1 or later (64-bit), x86-64 based
Disk Space	1.64 GB (does not include disk space for IDE/tools).
Tools	Flutter depends on these tools being available in your environment. ○ Windows PowerShell 5.0 or newer (this is pre-installed with Windows 10) ○ Git for Windows 2.x, with the **Use Git from the Windows Command Prompt** option. If Git for Windows is already installed, make sure you can run git commands from the command prompt or PowerShell.

1.1.2 Get the Flutter SDK

We will need to download the following Flutter SDK bundle found here: https://storage.googleapis.com/flutter_infra/releases/stable/windows/flutter_windows_1.22.6-stable.zip

While it is very likely that the program will be functional with newer Flutter releases, this Flutter release is last release certified by the developers. Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK (for example, C:\src\flutter). Please do not install Flutter in a directory like C:\Program Files\ that requires elevated privileges.

If you wish to run Flutter in the regular Windows console, please see here: https://flutter.dev/docs/get-started/install/windows#update-your-path

Next, we will check if there are any platform dependencies are needed to complete the setup. Run this command in a console window with the Flutter directory:

C:\src\flutter>flutter doctor.

This command checks your environment and displays a report of the status of your Flutter installation. Check the output carefully for other software you might need to install or further tasks to perform.

Once everything has been verified to be set up correctly, we can move on to the next section.

1.2 Installing an IDE

To start, a full Android Studio installation is required to supply Flutter with its Android platform dependencies. For this reason, we recommend using Android Studio as your IDE, as Flutter themselves do (https://flutter.dev/docs/get-started/install/windows#install-android-studio). Other IDEs for Flutter are available; however, this guide will only cover Android Studio. You can find various IDEs here:

IDE	**Install Guide**
Android Studio	https://developer.android.com/studio/install
Eclipse	https://www.eclipse.org/downloads/
Visual Studio Code	https://code.visualstudio.com/
Xcode	https://developer.apple.com/xcode/

March 2021

1.3 Choosing A Test Device

You may want to run the application on either an Android Device or an Android Emulator. Both are covered in the following sub-sections but using an Android Device is recommended to save storage space and resource usage.

1.3.1 Setting up an Android Device

The minimum API for any Android device to run Flutter is API level 16, which depending on your device model will be found under “Software Information” or “System Information.” If your device lacks these sections, it will need to be investigated personally. After this, follow these steps:

Step 1. Enable **Developer options** and **USB debugging** on your device. Detailed instructions can be found here.

Step 2. Install the Google USB Driver.

Step 3. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.

Step 4. In the terminal, run the flutter devices command to verify that Flutter recognizes your connected Android device. By default, Flutter uses the version of the Android SDK where your adb tool is based. If you want Flutter to use a different installation of the Android SDK, you must set the ANDROID_SDK_ROOT environment variable to that installation directory.

1.3.2 Setting up the Android Emulator

Follow these steps:

Step 1. Enable VM acceleration on your machine.

Step 2. Launch **Android Studio**, click the **AVD Manager** icon, and select **Create Virtual Device…**

- In older versions of Android Studio, you should instead launch **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device**…. (The Android submenu is only present when inside an Android project.)

- If you do not have a project open, you can choose **Configure > AVD Manager** and select **Create Virtual Device…**

Step 3. Choose a device definition and select **Next**.

Step 4. Select one or more system images for the Android versions you want to emulate, and select **Next**. An x86 or x86_64 image is recommended. Android SDK, you must set the ANDROID_SDK_ROOT environment variable to that installation directory.

Step 5: Under Emulated Performance, select **Hardware - GLES 2.0** to enable hardware acceleration.

Step 6: Verify the AVD configuration is correct and select **Finish**. For details on the above steps, see Managing AVDs.

1.4 Set up an editor

The full guide can be found here which will be useful if you want to use an editor besides Android Studio, but this section will cover Android Studio as a short-hand reference. Follow these steps:

Step 1: Open plugin preferences **(File > Settings > Plugins).**

Step 2: Select **Marketplace**, select the Flutter plugin and click **Install**.

1.5 Project Install

To install the project, follow these steps:

Step 1: On the **Welcome to Android Studio** screen, select **Get from Version Control.**

Step 2: Navigate to this page in a web browser, you may need to log in.

Step 3: Choose **Clone,** the copy the link under either **Clone with SSH** or **Clone with HTTPS.** The SSH cloning method is recommended.

Step 4: Return to Android Studio and copy the link into the **URL:** field. If you want to use a custom directory, change the **Directory:** field.

Step 5: Click **Clone**.

We have now installed a working Starfinder Companion 2020 project, which can be edited as a developer.

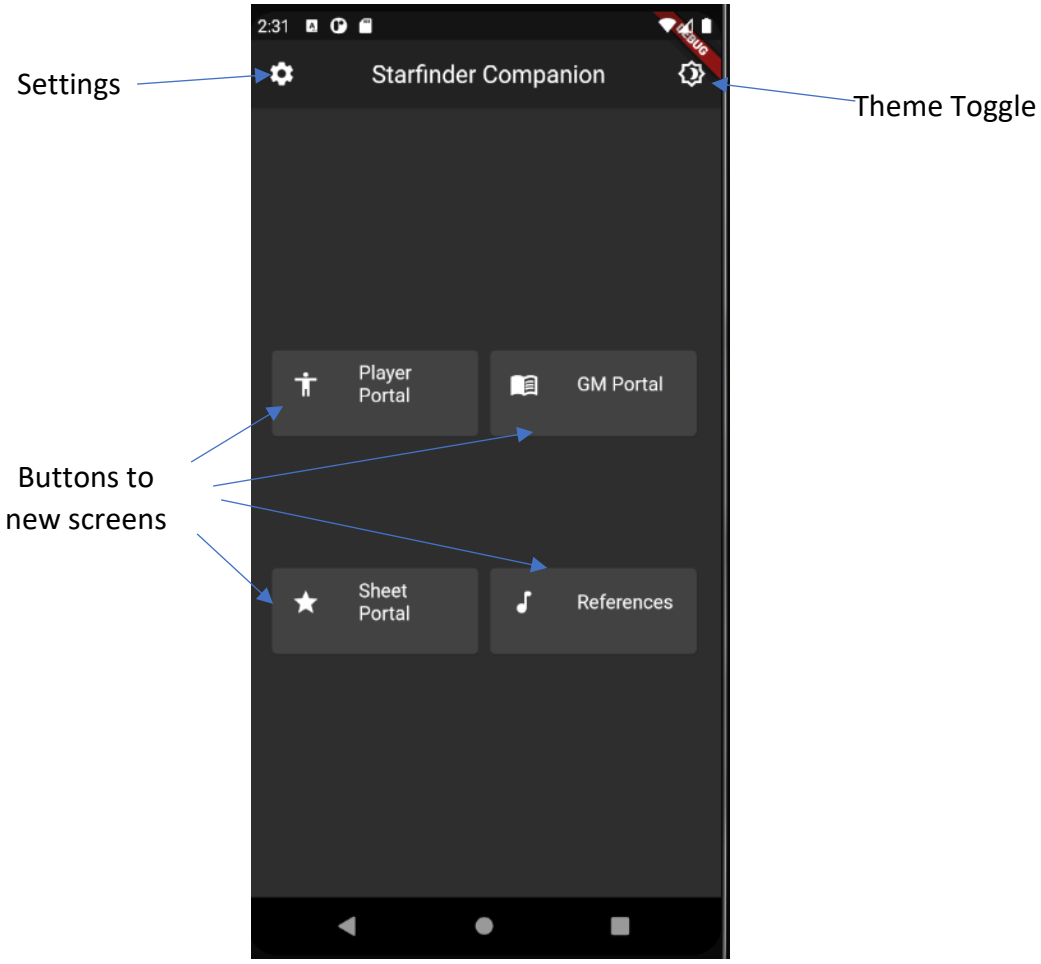
March 2021

2. Application Guide

This section will cover general usage of the application, as well as some technicalities at this pre-release stage. It will be a visual guide for the most part, with explanations found below each Figure.

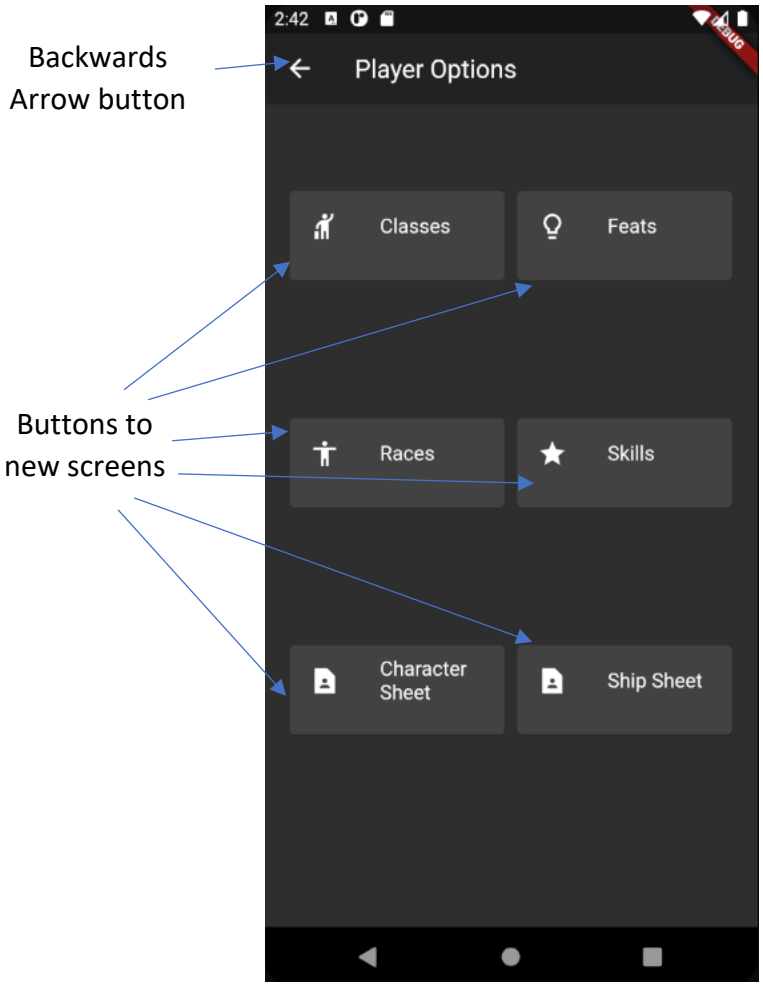
2.1 Navigation

2.1.1 Main Screen



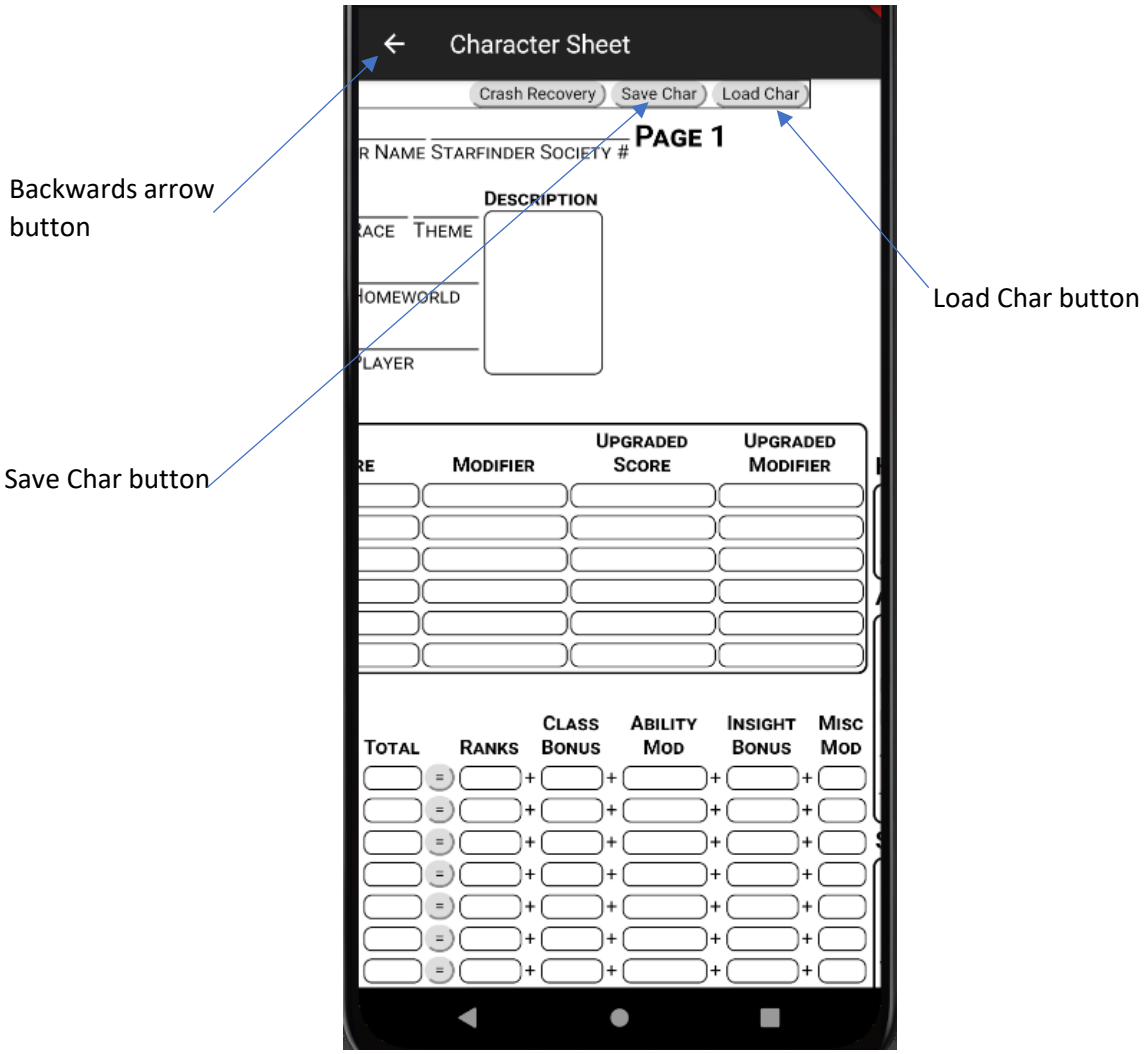
- The Settings menu as of current release is not fully implemented, holding only licensing agreements.
- The Theme Toggle has two settings: light mode and dark mode. Whatever setting you leave it on will be saved.
- Each of the buttons will open a new screen, which will be explained in turn.

2.1.2 Player Options Screen



- The Backwards Arrow button will return you to the main screen. These are found standardised across all screens.
- Each of the buttons are detailed here:
    - The classes and races buttons are implemented indexes, which stores details for players needing those details to create a character.
    - The Feats and Skills buttons remain unimplemented.
    - The Character Sheet button loads up the Starfinder Character Sheet shown below.
    - The Ship Sheet opens the Starfinder Ship Sheet shown below.

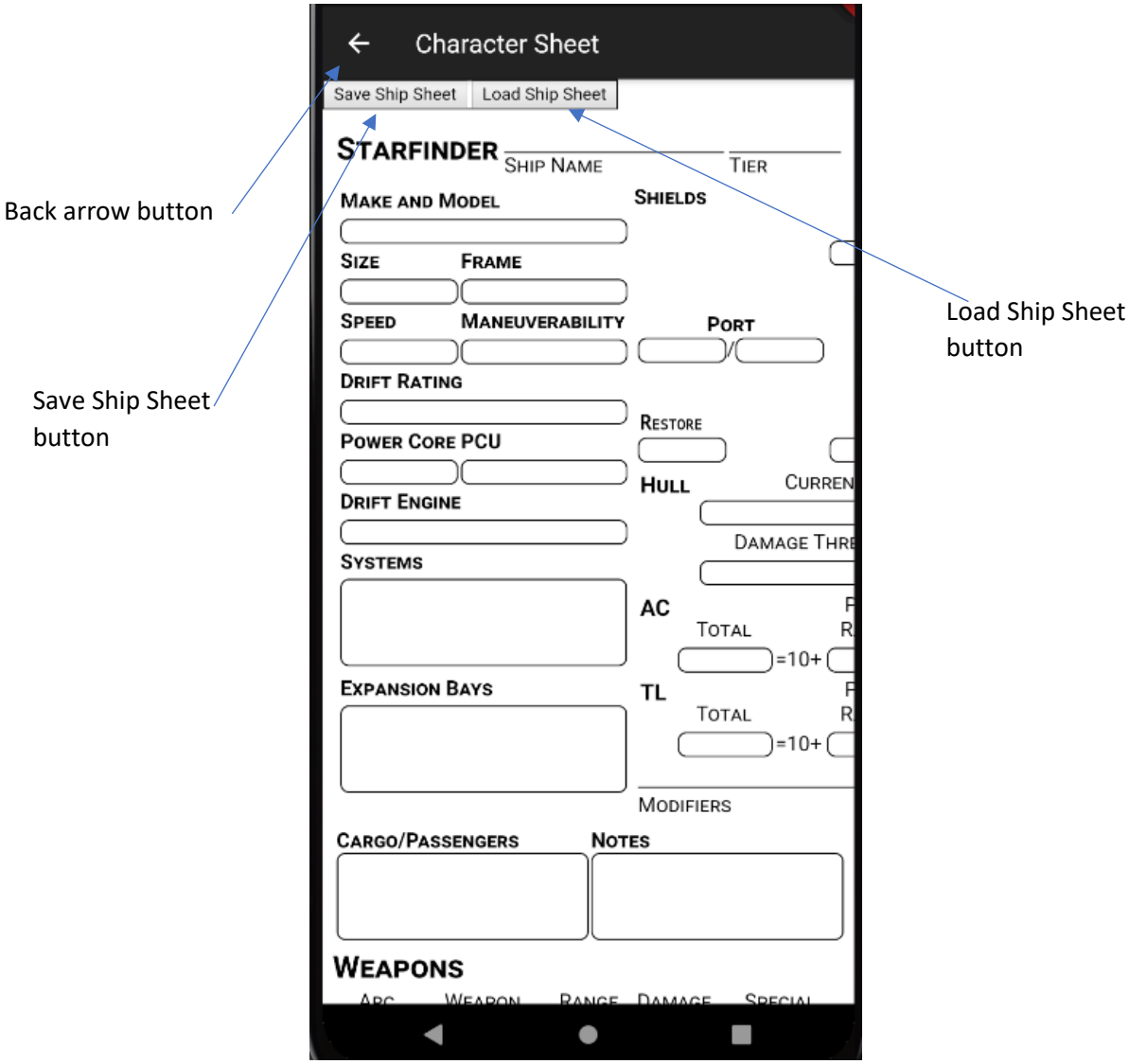
2.1.3 Character Sheet Screen



- The boxes that can be seen in these screenshots are all HTML input fields, each one is interactable and capable of storing text.
- When the Save Char button is tapped all the values on the character sheet are compiled into a file and saved to the user's phone.
- When tapped, the Load Char button takes the user to a new screen with all saved sheets in a list of tiles. The user can choose to tap a tile and load the associated sheet, or not choose any tile at all by clicking the backwards arrow.

March 2021

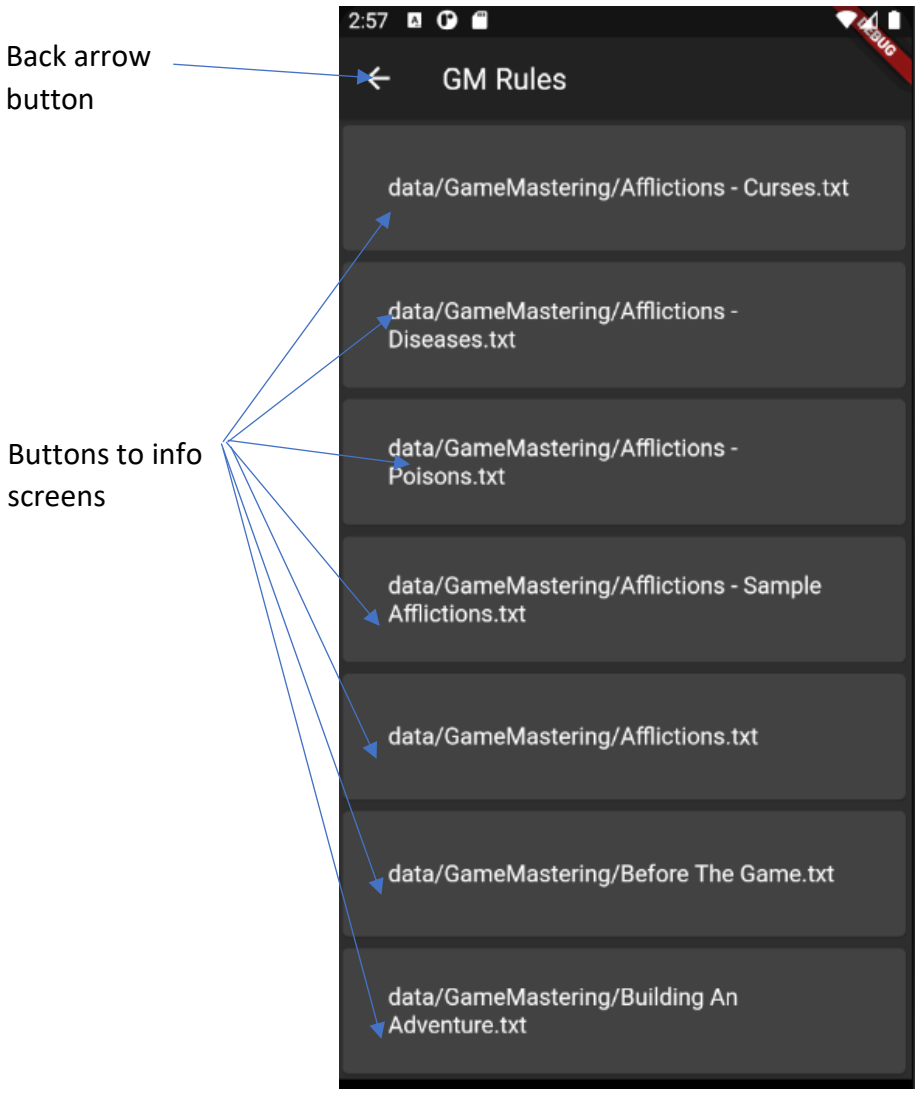
2.1.4 Ship Sheet Screen



- The backwards arrow button will take the user back to the Player Screen
- The boxes that can be seen in these screenshots are all HTML input fields, each one is interactable and capable of storing text.
- When the Save Ship Sheet button is tapped all the values on the character sheet are compiled into a file and saved to the user's phone.
- When tapped, the Load Ship Sheet button takes the user to a new screen with all saved sheets in a list of tiles. The user can choose to tap a tile and load the associated sheet, or not choose any tile at all by clicking the backwards arrow.

March 2021

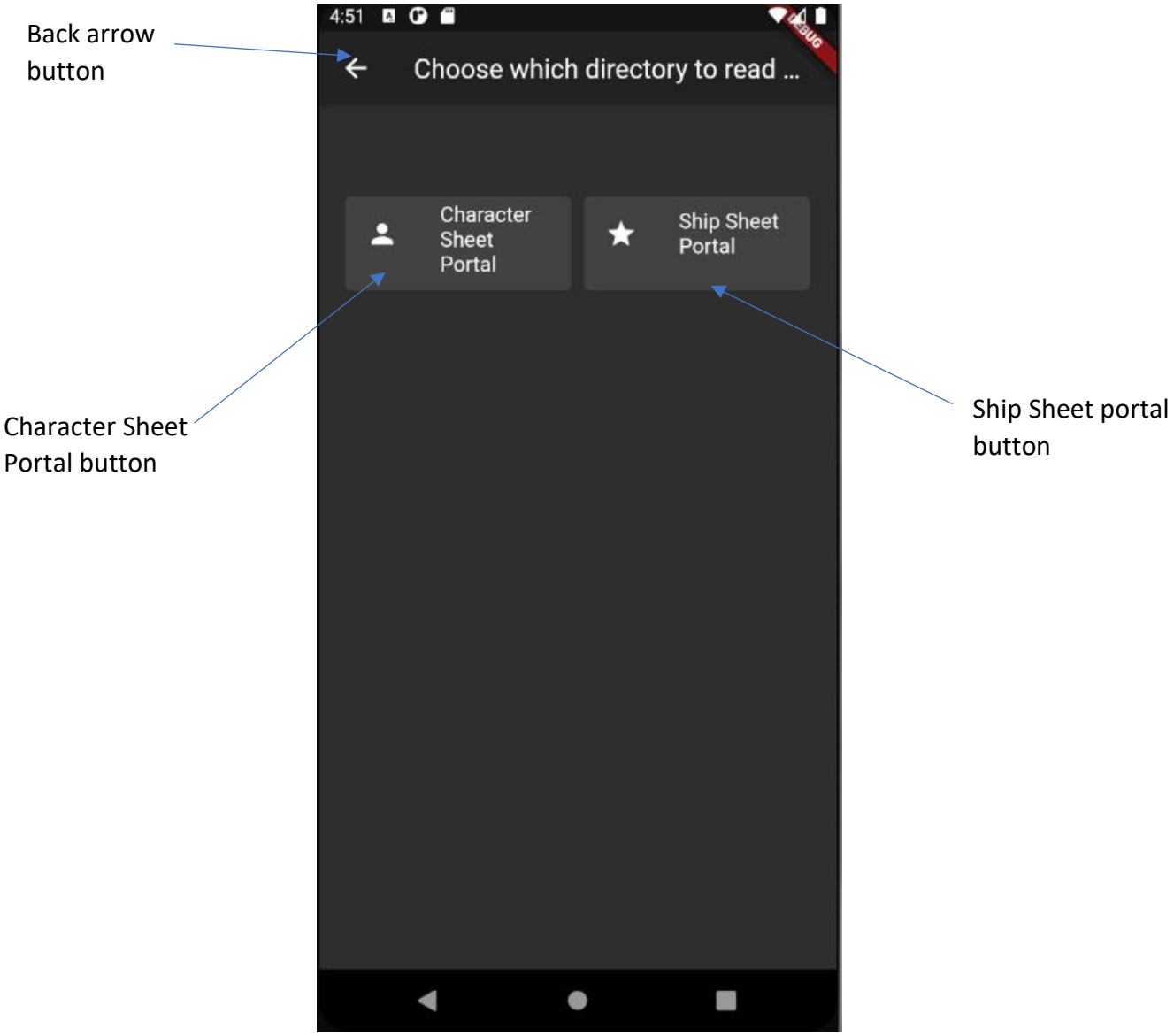
2.1.5 GM Rules Screen



- The back arrow button returns the user to the Main Menu.
- Each of the List Tiles on the screen opens a file containing the associated snippet from the Game Mastering section of the core rulebook

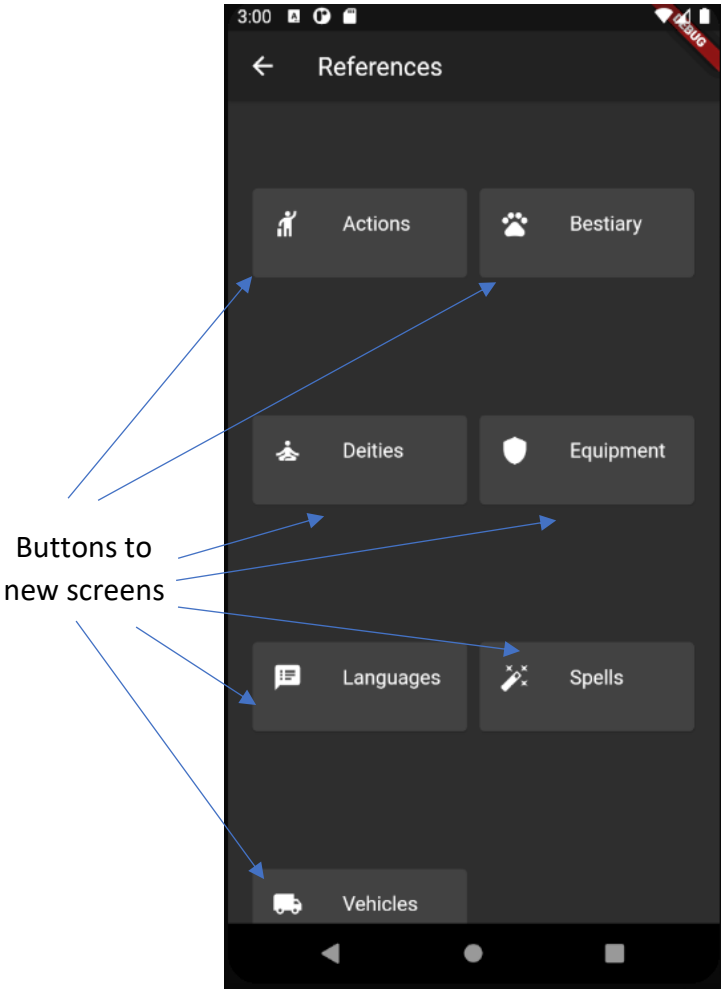
March 2021

2.1.6 Sheet Portal Screen



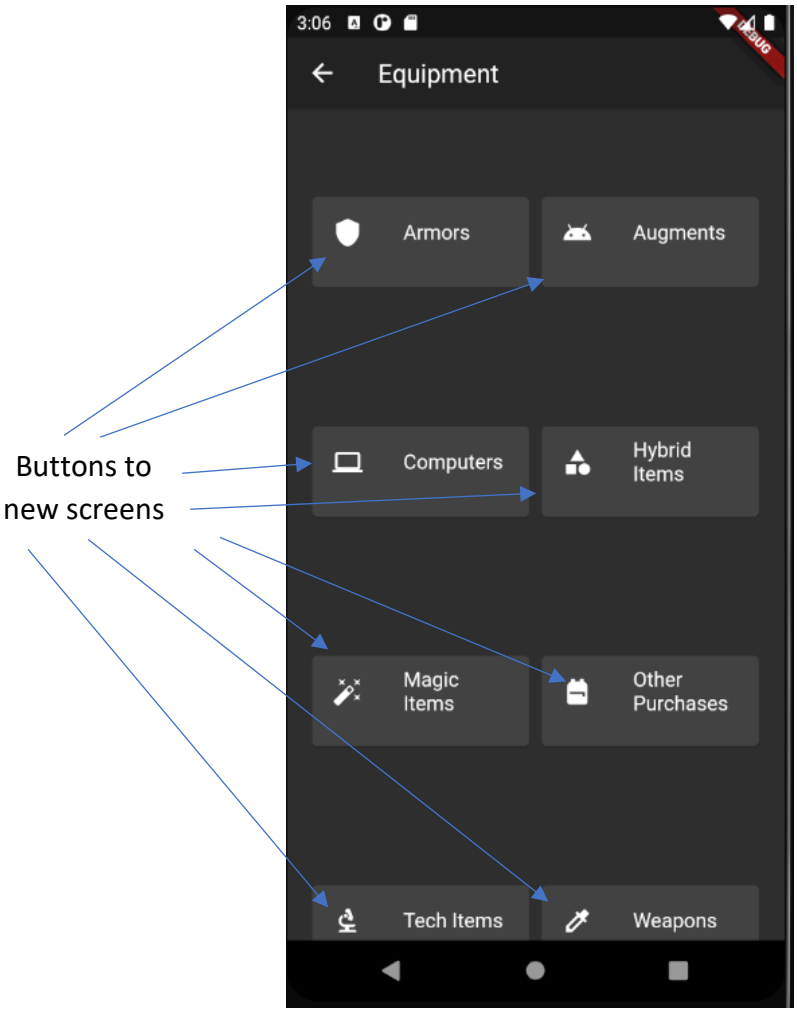
- The backwards arrow button will take the user back to the Main Menu
- The Character Sheet Portal will take the user to a screen displaying the list of saved character sheets the user has made. If there are no sheets saved a message is displayed instructing the user on where to find the initial character sheet.
- The Ship Sheet Portal will take the user to a screen displaying the list of saved Ship Sheets the user has made. If there are no sheets saved a message is displayed instructing the user on where to find the initial ship sheet.

2.1.7 References Screen



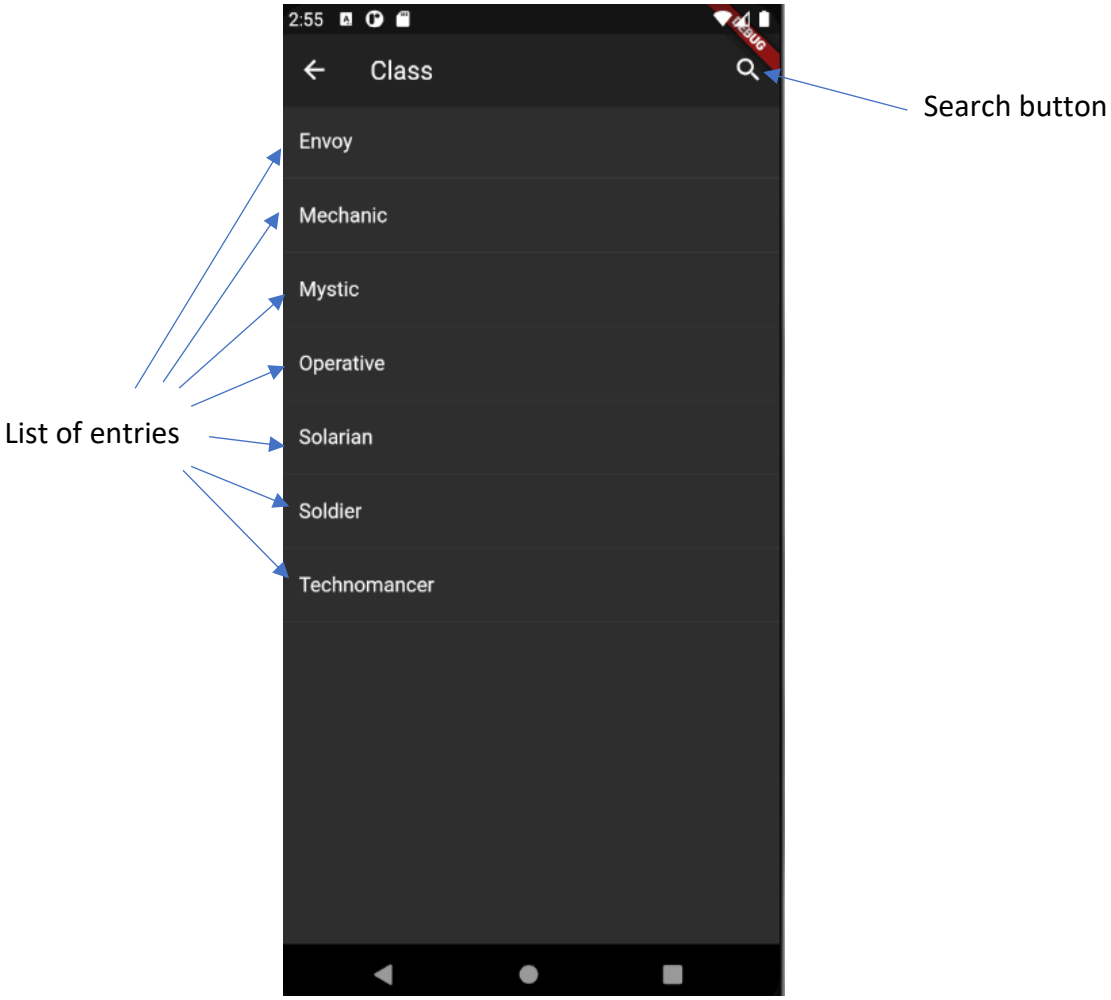
- The Backwards Arrow button will return you to the main screen. These are found standardised across all screens.
- Each of the buttons are detailed here:
    - The spells button will bring you to an implemented index, useful for both players and game masters.
    - The Equipment button will bring you to a new screen.
    - All other buttons lead to unimplemented indexes.

2.1.8 Equipment Screen



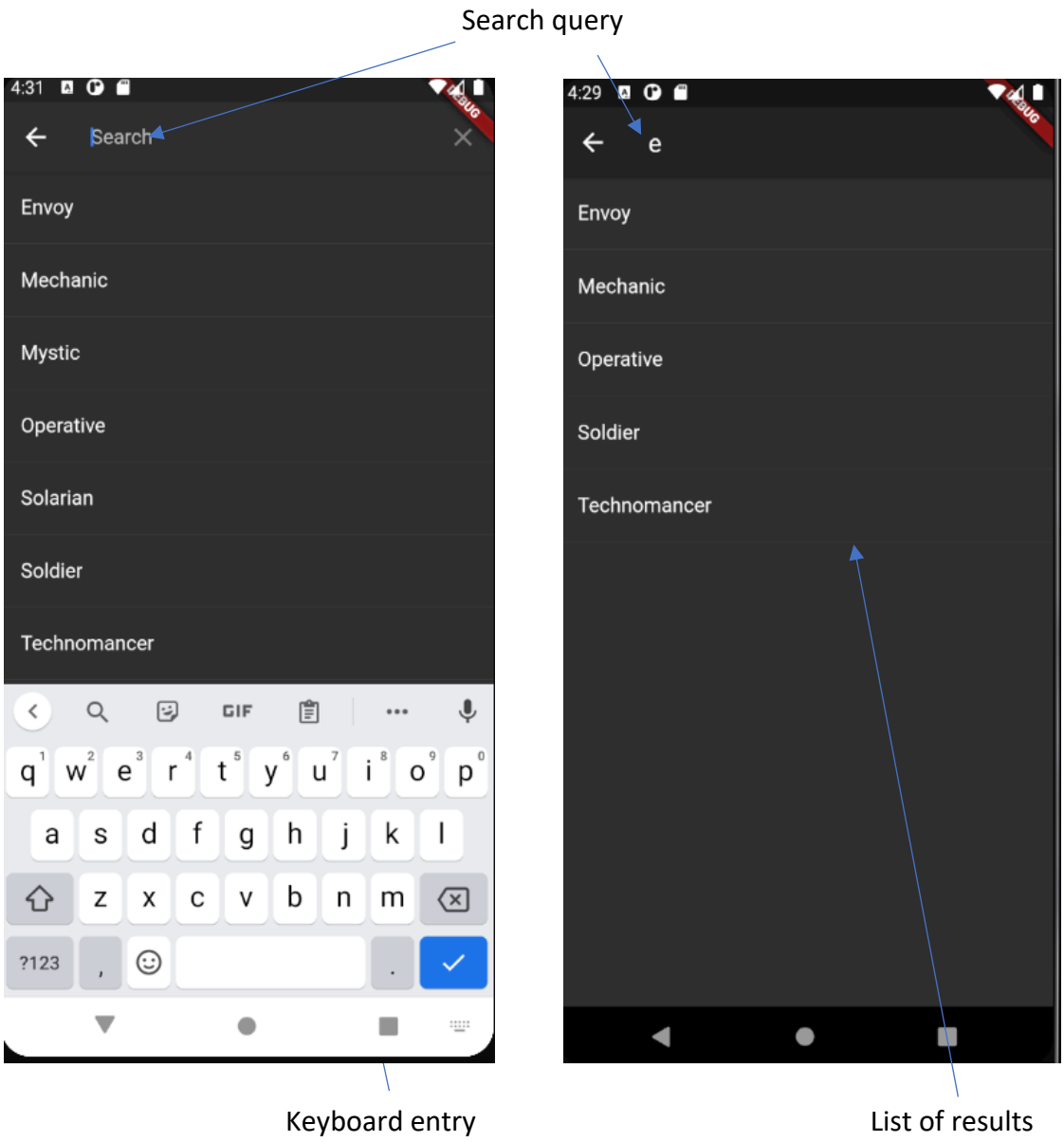
- The Backwards Arrow button will return you to the main screen. These are found standardised across all screens.
- Each of the buttons are detailed here:
    - The weapons button will bring you to an implemented index, useful for both players and game masters.
    - All other buttons lead to unimplemented indexes.

2.1.9 Index Screens



- The Backwards Arrow button will return you to the main screen. These are found standardised across all screens.
- Each entry, when tapped or clicked, will open details pertaining to that entry.
- The search button, when pressed, will present a search bar which will accept a search entry you may input, which will parse upper-case and lower-case letters for you. If no results are found, you will get a message saying, “No results found!”.

March 2021


Search query
4:31
Search
Envoy
Mechanic
Mystic
Operative
Solarian
Soldier
Technomancer
4:29
e
Envoy
Mechanic
Operative
Soldier
Technomancer
Keyboard entry
List of results

March 2021

3. Developer Considerations

The project from the ground up has been designed for expandability, and the following considerations should be taken when expanding the project:

- Any new ‘index’ objects should inherit from the index object.
- Any new JSON files should be structured in a key-pair format.
- Error handling should be integrated into each object.
- JSON files should be named sfrpg_INDEX where INDEX is the new index to be added.
- Each index should follow the same template as laid down by the implemented indexes.