Dorsch Lab

# Consent App

*2020.11.24.ET*

## Introduction

The consent app is a multi-platform single-codebase application written in Google's Flutter/Dart programming language. This app provides a template for presenting Internal Review Board-approved (IRB) consent documents in a digital format appropriate for web and mobile device applications.
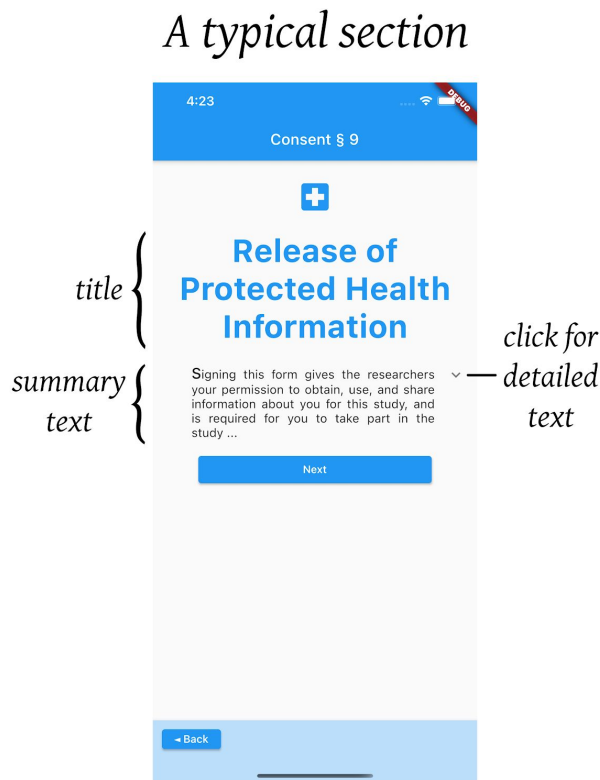
To achieve this goal, the consent document itself is written in *Markdown* format ([www.markdownguide.org](www.markdownguide.org)). Markdown is essentially a "*plain text*" file with very simple formatting conventions that can then be translated by software into standard typographic items such as paragraphs, bulleted lists, indented blocks, headings, bold or italicized words and phrases, *et cetera*.

The Markdown format was chosen because it is simple for software to read but also very easy for humans to write and read. Once the language in the consent document has been finalized, the file `consent.md` becomes an *asset* of the Flutter app that is loaded at run time.

A few simple conventions are used to demarcate section headings and summary and detailed text within each heading. These conventions are unique to the app and assist the app in understanding how and where to break up the single consent document into individual sections that can be displayed in web and mobile apps.

## Section Layout

The layout of a typical section of a consent document is shown in fig. 1:
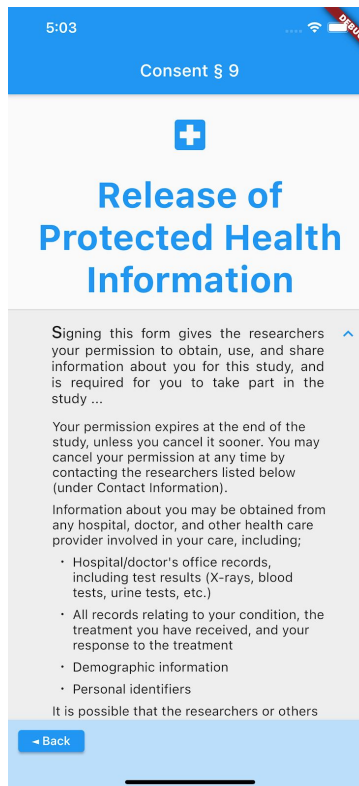


*A typical section*

The section consists of the following parts:

1. *section title*
2. *summary text*
3. *detailed text*

In many cases, the *summary text* is sufficient and no *detailed text* is necessary. When no *detailed text* is present, the dropdown arrow will not appear.

When the detailed text is present, tapping or clicking on the summary text or on the down arrow will expand the tile and reveal the detailed text paragraphs (fig. 2):
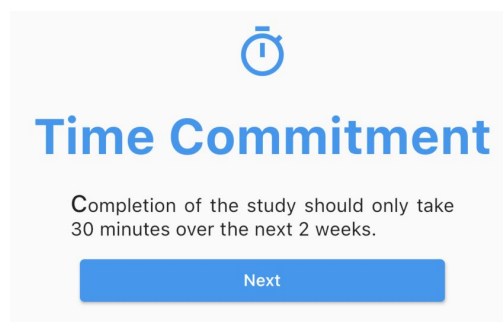
The user can scroll down to see any hidden text.

## Formatting sections in the `consent.md` file

For a *simple section* that consists of just a title and some summary text, all you need to do is place a *colon* between the title and summary text, like this:

```
Time Commitment: Completion of the study should only take 30 minutes
over the next 2 weeks.
```

In the app, it will look like this (fig. 3):

For a *full section* consisting of both summary and details, place the title on a line by itself followed by a colon, and then use the key words `summary` and `detail` followed by a colon respectively to begin the texts, like this:

```
Data Gathering:
summary: This website will collect some personal information about
you, as well as certain information from your medical and prescription
medication history ...

detail: The information that will be collected includes:

 - first name
 - last name
 - email address
 - age
```
*... etc. ...*

## FORMATTING LIMITATIONS IN THE SUMMARY SECTION

The summary section will be displayed as a single paragraph. There are no options available to make any of the text bold or italicized or any other specialized formatting.

## FORMATTING OPTIONS IN THE DETAIL SECTION

In the detail section, you can make use of common Markdown formatting conventions to highlight text in bold, italics, or to do other things like create a bulleted list. At the time of this writing, there are still some limitations mainly due to the way that the app automatically parses the consent document into individual sections.

## LOCATION OF THE `consent.md` FILE

The `consent.md` file must be present in the `assets` directory. Be sure to edit the `pubspec.yaml` file to declare the `consent.md` file as an asset.

## ICONS FOR EACH SECTION

The icons that will be associated with each section are defined in the `lib/iconMap.dart` file (fig. 4):

```dart
import 'package:flutter/material.dart';

class IconMap{

  static Map<String,IconData> _iconMap = {
    'default':Icons.waves,
    'Introduction':Icons.waves,
    'Time Commitment':Icons.timer,
    'Time':Icons.timer,
    'Data Gathering':Icons.tune,
```

As you can see, the Material UI icons are associated with specific *section titles.* If you change or add any sections in the `consent.md` file, I recommend you also edit `iconMap.dart` accordingly.

Note however that the app will resort to using the *default* `Icons.waves` icon (*or any other icon that you specify as a default*) for any cases where a section title in the `consent.md` file does not have a match in `iconMap.dart`. So if you are in a hurry, don't worry. The app will still work just fine with just about any variation of a consent document that you give to it.

Note also that the naming of icons in the Material Design UI is not very intuitive. The best course of action is to simply scroll through the visual representations of the icons in the Material icon documentation at https://material.io/resources/icons/ until you find something that visually matches the main theme of the section you are working on.

## The Authorization section

The last section of your consent.md file should be the **Authorization** section and should look something like this:

```
Authorization: By clicking or tapping on the "agree" button below,
you confirm that you have read and understood the information in this
consent disclosure and that you agree to participate in this study and
allow this website or mobile app to access your health data.
```

When the app sees the title "Authorization", it knows that it needs to provide *agree* and *decline* buttons in place of the usual *next* button (fig. 5):

## SETTING UP DESTINATIONS FOR THE AGREE AND DECLINE BUTTONS

As currently written, the app provides a `config.json` in the `assets` directory where you can specify web destinations for the agree and decline buttons. By default, *agree* points to the Dorsch lab web page and *decline* will take you to nutrition.gov's healthy heart page:

```
{
  "consent_url":"https://pharmacy.umich.edu/dorsch-lab",
  "decline_url":"https://www.nutrition.gov/topics/diet-and-health-conditions/heart-health"
}
```

Obviously you can change these to be whatever is appropriate.

*If you want the agree button to continue on to another route in a flutter app, then some additional coding will be required.*

## BUILDING THE APP FOR DIFFERENT PLATFORMS

Thanks to Flutter, building the app for different platforms is easy.

For iOS, the usual `flutter run` command suffices.

For a web version for debugging, try `flutter run -d chrome`.

For a final web build, use `flutter build web` as documented on https://flutter.dev/docs/get-started/web.

*** end of document ***