



Βιοϊατρική

Κουμάκης Λευτέρης,
Μαρία Χατζημηνά

Εργαστήριο 6

Use Mobile Camera

- Package:
 - Image_picker
 - url: https://pub.dev/packages/image_picker/install
- Το **imageSource** μπορεί να είναι είτε:
 - Gallery
 - Camera
- Στον emulator αν δεν λειτουργεί η κάμερα τρέχουμε από τον φάκελο emulator (στο Android SDK path) την εντολή
 - **emulator -camera-front webcam0 -avd το_ονομα_του_emulator**

```
//είναι ασυγχρονη συναρτηση που επιστρεφει ενα future γιατο δηλων  
Future<void> takePicture() async {  
  final picker = ImagePicker();  
  final imageFile = await picker.pickImage( //ImageSource.camera &  
    | source: ImageSource.camera, maxWidth: 600, maxHeight: 600);  
}
```

Use Mobile Camera

- Σε πακέτα που χρησιμοποιούν native στοιχεία του κινητού όπως την κάμερα, μικρόφωνο κτλ. Διαβάζουμε την ενότητα “Read me” του πακέτου.
- Παράδειγμα για την κάμερα:
 - Android:
 - No configuration required - the plugin should work out of the box.
 - iOS:
 - Add the following keys to your **Info.plist** file, located in `<project root>/ios/Runner/Info.plist`:
 - **NSPhotoLibraryUsageDescription** - describe why your app needs permission for the photo library. This is called Privacy - Photo Library Usage Description in the visual editor.
 - **NSCameraUsageDescription** - describe why your app needs access to the camera. This is called Privacy - Camera Usage Description in the visual editor.
 - **NSMicrophoneUsageDescription** - describe why your app needs access to the microphone, if you intend to record videos. This is called Privacy - Microphone Usage Description in the visual editor.

Save Image to Mobile

- Packages:
 - path_provider : χρησιμοποιείται για να βρούμε paths
 - https://pub.dev/packages/path_provider
 - path : χρησιμοποιείται για να δημιουργήσουμε paths
 - <https://pub.dev/packages/path>

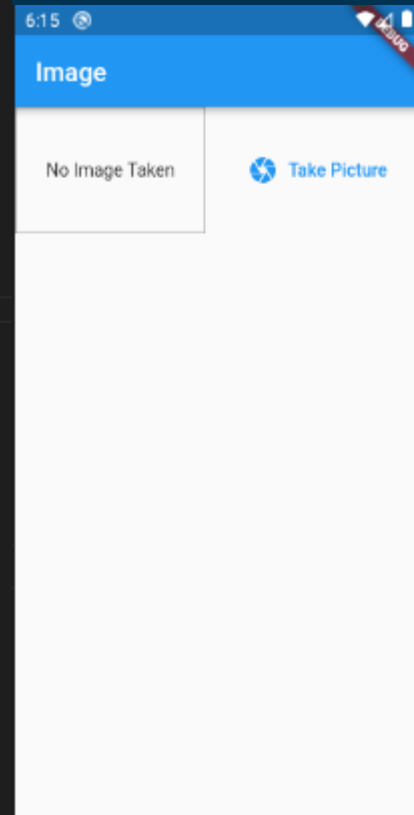
```
class _MyAppState extends State<MyApp> {
  late File _storedImage ;
  var image;
  bool imagefile = false;

  //είναι ασynchρονη συναρτηση που επιστρεφει ενα future γιαυτο δηλωνουμε οτι είναι και async
  Future<void> _takePicture() async {
    final picker = ImagePicker();
    final imageFile = await picker.pickImage( //ImageSource.camera δηλωνει οτι θελουμε να παρουμε φωτογραφια απο την καμερα
      source: ImageSource.camera, maxWidth: 600, maxHeight: 600);
    final appDir = await syspath.getApplicationDocumentsDirectory(); //ο φακελος που αποθηκευονται οι εικονες(οχι προσωρινα)
    if (imageFile != null){
      print("inside");
      final fileName = path.basename(imageFile.path); //το path που αποθηκευεται η εικονα προσωρινα
      File file = File(imageFile.path);
      final savedImage = await file.copy('${appDir.path}/${fileName}'); //ετσι περναμε σαν string τις τιμες των μεταβλητων ( με το $)
      print(savedImage.path); //τυπωνω το path που αποθηκευτηκε στην συσκευη
      setState() {
        _storedImage = File(imagefile.path);
        imagefile = true;
      });
    }
  }
}
```

Save Image to Mobile

- FlatButton.icon
 - Δημιουργούμε ένα κουμπί με εικονίδιο
- Expanded:
 - Η χρήση ενός expanded widget στο child μιας γραμμής (Row) ή μιας στήλης (Column) το αναγκάζει να επεκταθεί για να γεμίσει τον διαθέσιμο χώρο κατά μήκος του κύριου άξονα (π.χ. οριζόντια για το Row ή κάθετα για το Column).

```
return Scaffold(  
  appBar: AppBar(  
    title: Text("Image"),  
  ), // AppBar  
  body: Row( //παρομοιο με το column απλα η στοιχιση ειναι το ενα widget διπλα στο αλλο  
    children: <Widget>[  
      Container(  
        width: 150,  
        height: 100,  
        decoration: BoxDecoration(  
          border: Border.all(width: 1, color: Colors.grey),  
        ), // BoxDecoration  
        child: imagefile != false //αν η εικονα απο την καμερα εχει δημιουργηθει  
          ? Image.file(imagefile) //εμφανισε τη στο κουτι  
            : Text( //διαφορετικα εμφανισε ενα κειμενο  
              'No Image Taken',  
              textAlign: TextAlign.center,  
            ), // Text  
          alignment: Alignment.center,  
        ), // Container  
      SizedBox(  
        width: 10,  
      ), // SizedBox  
      Expanded(  
        child: FlatButton.icon( //χρησιμοποιουμε ενα κουμπι με εικονιδιο  
          icon: Icon(Icons.camera), //το εικονιδιο της καμερας  
          label: Text('Take Picture'), // το κειμενο που θα φαινεται στο κουμπι  
          onPressed: _takePicture, //η future συναρτηση _takePicture που δηλωσαμε παραπανω  
        ), // FlatButton.icon  
      ), // Expanded  
    ], // <Widget>[]  
  ), // Row  
); // Scaffold
```



Useful Links

- Icons list: <https://api.flutter.dev/flutter/material/Icons-class.html>
- Charts package: https://pub.dev/packages/charts_flutter#-readme-tab-
- Available charts: <https://google.github.io/charts/flutter/gallery.html>

UI Design

- Πως γνωρίζουμε ότι ένα UI Design είναι καλό;
 - **Learnability/Δυνατότητα εκμάθησης**: Μπορεί ο χρήστης να ολοκληρώσει τις εργασίες του με μικρή δυσκολία;
 - **Efficiency/Αποδοτικότητα**: Η διεπαφή επιτρέπει στον χρήστη να ολοκληρώσει τις εργασίες εγκαίρως;
 - **Memorability/Απομνημόνευση**: Μετά την αποχώρηση από έναν ιστότοπο, πόσο πιθανό είναι ο χρήστης να θυμάται πώς να τον χρησιμοποιήσει την επόμενη φορά που θα τον επισκεφτεί;
 - **Errors/Σφάλματα**: Ποια βήματα κάνει η διεπαφή για να μειώσει την πιθανότητα σφάλματος του χρήστη και πως επιτρέπει στους χρήστες να διορθώσουν ένα σφάλμα;
 - **Satisfaction/Ικανοποίηση**: Απολαμβάνει ο χρήστης την αλληλεπίδραση με τον ιστότοπο;

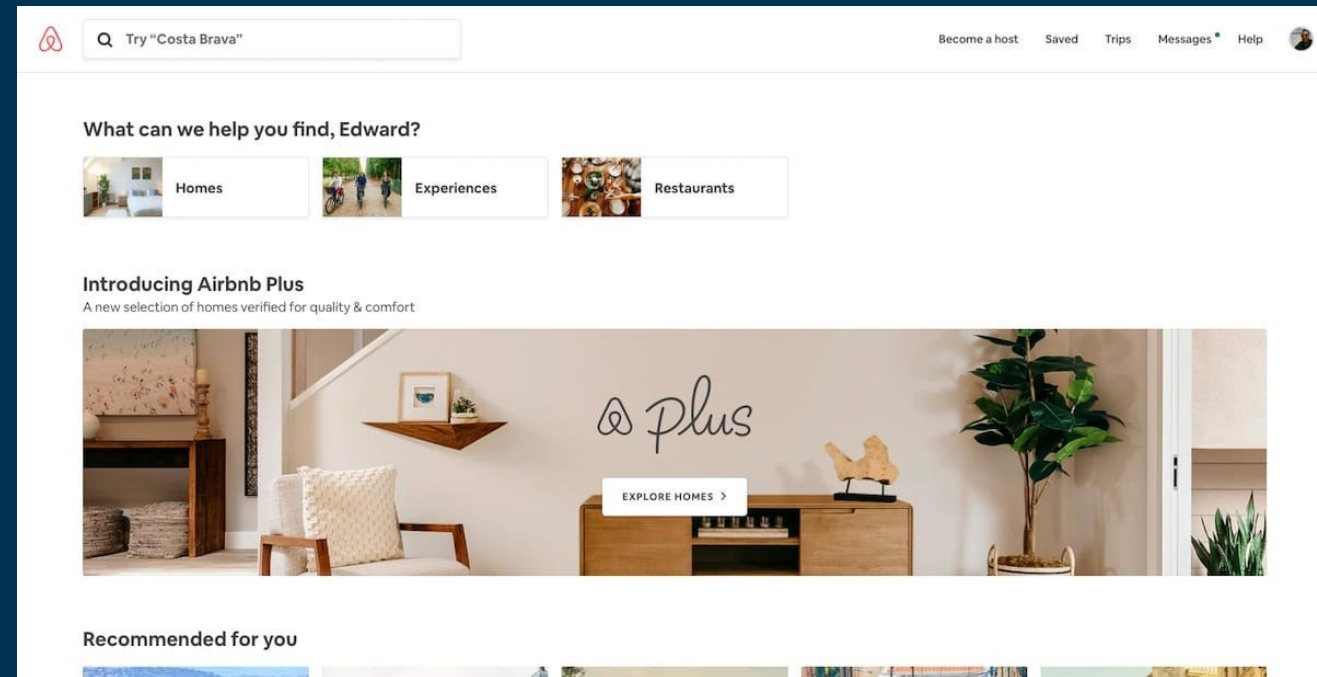
Επιτυχημένα Παραδείγματα: Medium

- Online reading and publishing platform → [Medium](#)
 - Ελάχιστη χρήση χρώματος
 - Γενναιόδωρη απόσταση μεταξύ των γραμμών
 - Καλά επιλεγμένος συνδυασμός τυπογραφίας
 - Τα άρθρα είναι σε μορφή μιας στήλης που το κάνει εύκολο στην ανάγνωση
 - Εκτιμώμενος χρόνος ανάγνωσης
 - Άμεσος σχολιασμός σε συγκεκριμένα κομμάτια των άρθρων



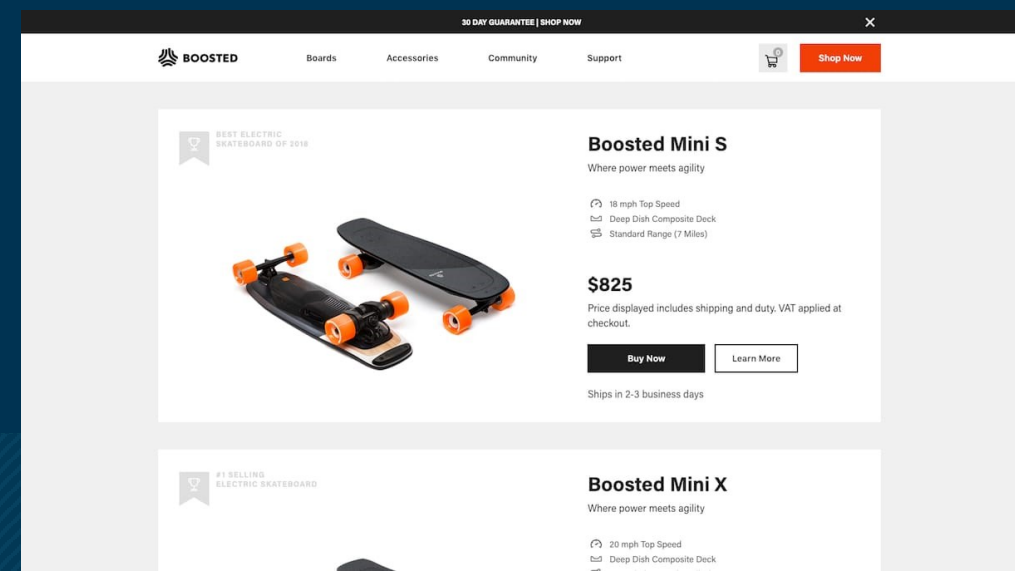
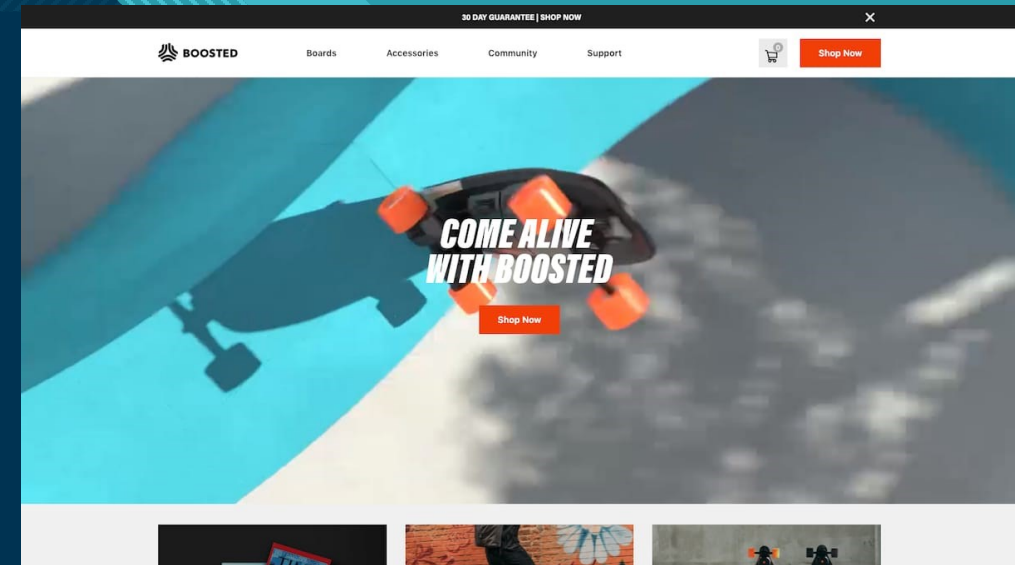
Επιτυχημένα Παραδείγματα: Airbnb

- Επιτυγχάνει δύο πράγματα
 - Να κάνετε κράτηση για ένα μέρος διαμονής
 - Δημιουργία εμπιστοσύνης ανάμεσα σε δύο εντελώς ξένους.
- Απλή διαδικασία εύρεσης διαμονής
- Clean design
- Χρήσιμο κείμενο υπόδειξης
 - “What can we help you find, Edward?”



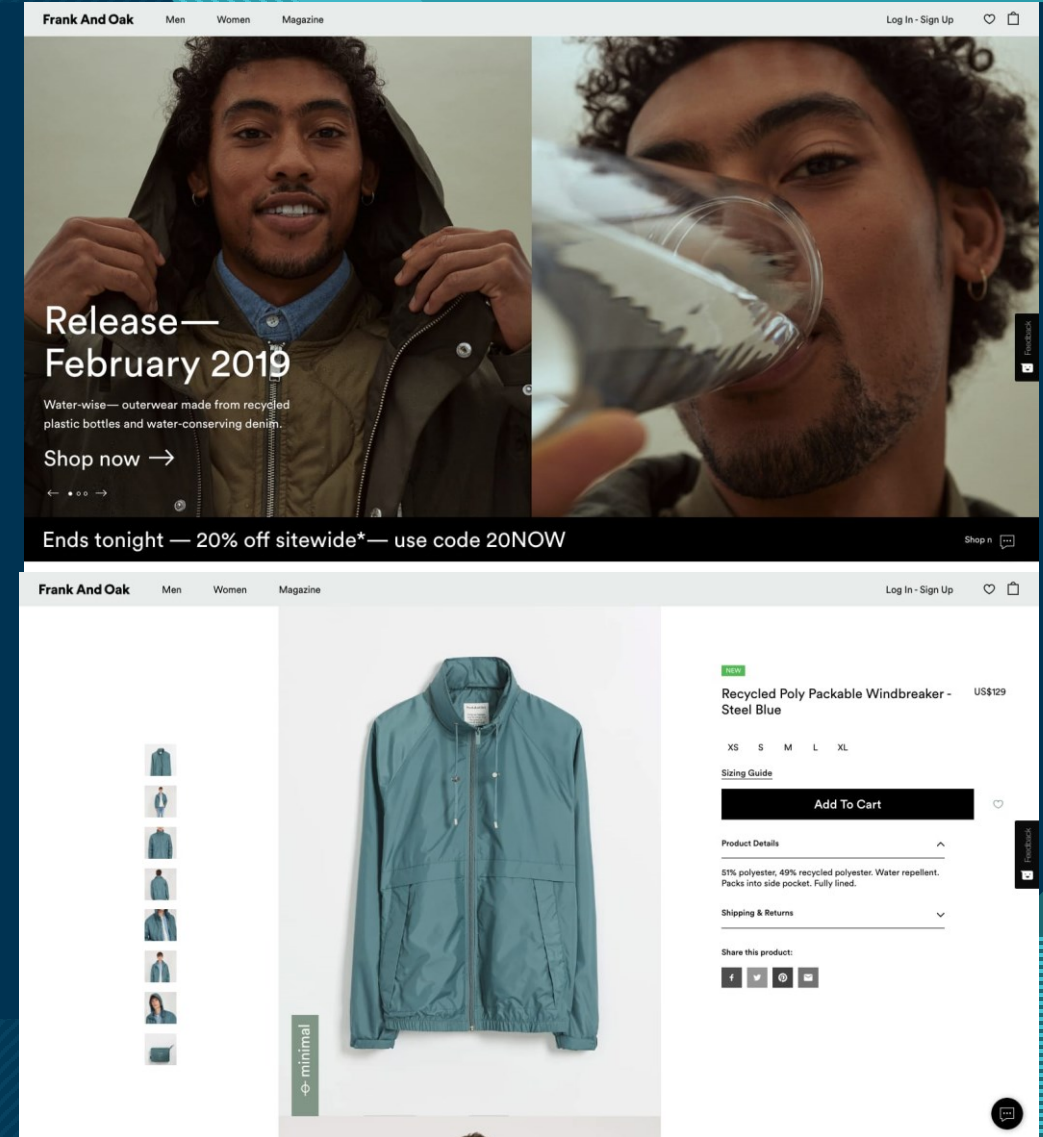
Επιτυχημένα Παραδείγματα: Boosted Boards

- Οι χρήστες καλωσορίζονται αμέσως με ένα δυναμικό βίντεο του προϊόντος.
- Η διεπαφή είναι απλασμένη από χρώμα
- Παρέχονται μόνο οι πιο σχετικές πληροφορίες: εικόνα, μέγιστη ταχύτητα, εύρος, τιμή, χρόνος αποστολής και, φυσικά, «Buy Now».



Επιτυχημένα Παραδείγματα: Frank and Oak

- Χρησιμοποιεί τη διεπαφή χρήστη για να τραβήξει την προσοχή των χρηστών
- Η εταιρεία χρησιμοποιεί τολμηρή φωτογραφία και καθαρή, sans-serif τυπογραφία
- Αφού κάνετε κλικ σε ένα αντικείμενο, η σελίδα του προϊόντος εστιάζει σε δύο πράγματα: τη φωτογραφία του προϊόντος και το κουμπί αγοράς, καθώς είναι ουσιαστικά τα μόνα χρωματιστά στοιχεία στη σελίδα.



Best Practices for Designing an Interface

- **Διατηρήστε τη διεπαφή απλή.** Οι καλύτερες διεπαφές είναι σχεδόν αόρατες στον χρήστη. Αποφεύγουν τα περιττά στοιχεία και είναι ξεκάθαρα στη γλώσσα που χρησιμοποιούν στις ετικέτες και στα μηνύματα.
- **Χρησιμοποιήστε στρατηγικά χρώμα και υφή.** Μπορείτε να κατευθύνετε την προσοχή ή να ανακατευθύνετε την προσοχή μακριά από αντικείμενα χρησιμοποιώντας χρώμα, φως, αντίθεση και υφή προς όφελός σας.
- **Χρησιμοποιήστε την τυπογραφία για να δημιουργήσετε ιεραρχία και σαφήνεια.** Εξετάστε προσεκτικά πώς χρησιμοποιείτε τη γραμματοσειρά. Διαφορετικά μεγέθη, γραμματοσειρές και διάταξη του κειμένου που βοηθούν στην αύξηση της ευανάγνωσης και της αναγνωσιμότητας.

Best Practices for Designing an Interface

- **Να είστε σκόπιμοι στη διάταξη της σελίδας.** Λάβετε υπόψη τις χωρικές σχέσεις μεταξύ των στοιχείων στη σελίδα και δομήστε τη σελίδα με βάση τη σπουδαιότητά της. Η προσεκτική τοποθέτηση των αντικειμένων μπορεί να βοηθήσει να επιστήσει την προσοχή στις πιο σημαντικές πληροφορίες και μπορεί να βοηθήσει στην αναγνωσιμότητα.
- **Βεβαιωθείτε ότι το σύστημα επικοινωνεί αυτό που συμβαίνει.** Πάντα να ενημερώνετε τους χρήστες σας για τις ενέργειες, τις αλλαγές στην κατάσταση ή τα σφάλματα. Η χρήση διαφόρων στοιχείων διεπαφής χρήστη για την επικοινωνία της κατάστασης και, εάν είναι απαραίτητο, τα επόμενα βήματα μπορεί να μειώσει την απογοήτευση του χρήστη.

Best Practices for Medical and Healthcare Apps

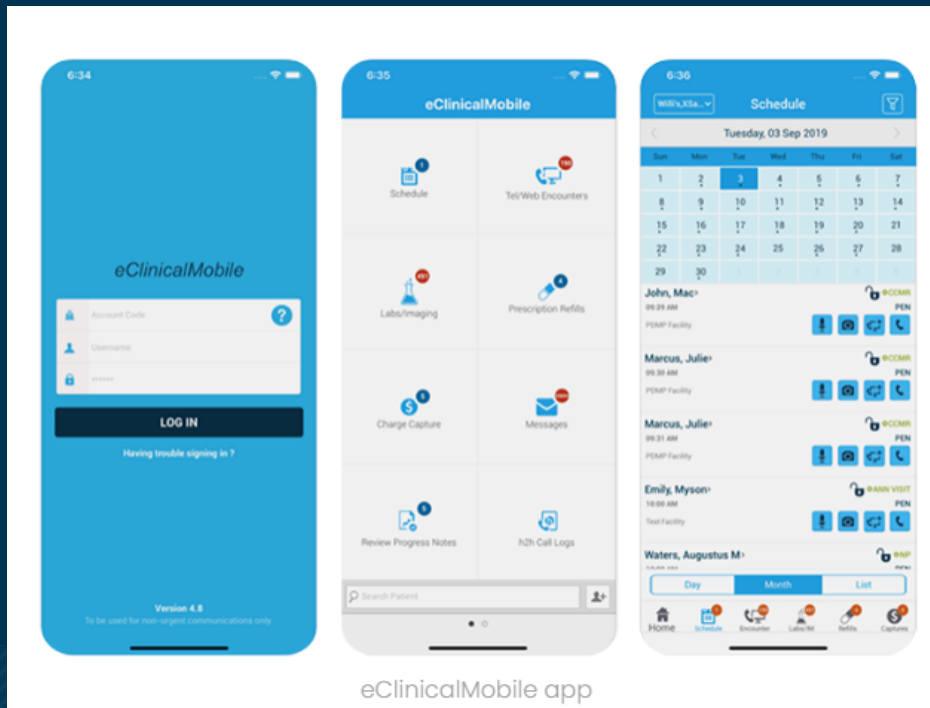
- Εξετάστε τη σημασία των συνδυασμών χρωμάτων
- Η ιατρική χρωματική παλέτα είναι αρκετά περιορισμένη. Είναι δύσκολο να φανταστεί κανείς έναν οργανισμό υγειονομικής περίθαλψης όπου το κίτρινο νέον είναι το κορυφαίο χρώμα. Αυτό οφείλεται στην επίδραση του χρώματος στη διάθεση και την ευεξία των ασθενών.
- Τα φανταχτερά και ζωντανά χρώματα μπορούν να οδηγήσουν σε αισθήματα επιθετικότητας. Μια ιατρική εφαρμογή θα πρέπει να εκφράζει μια φιλική στάση και να είναι θεραπευτική και ήρεμη.



Χρώματα

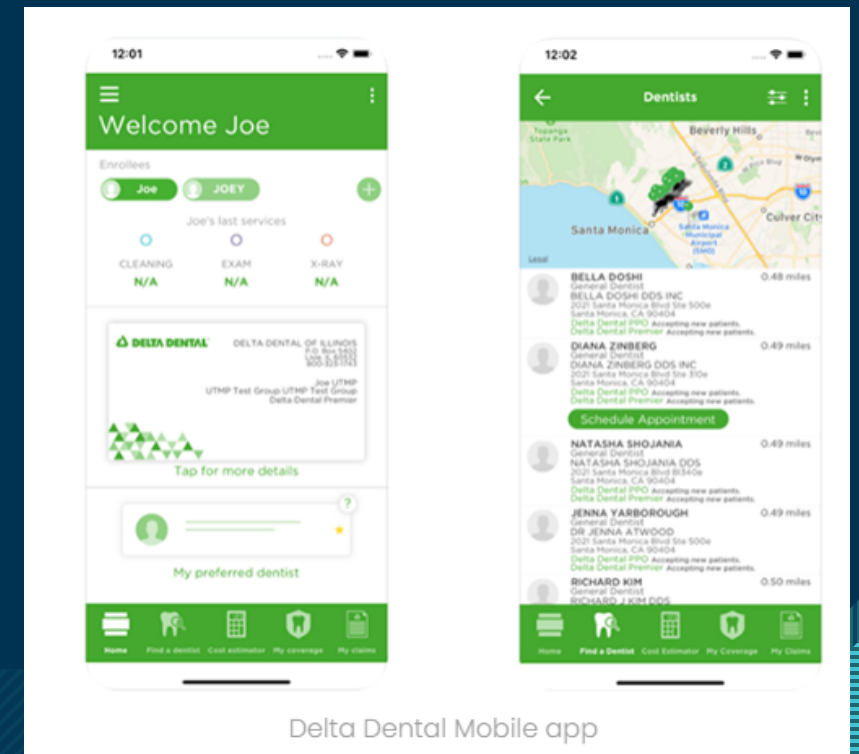
Μπλε

- Το μπλε είναι ένα φυσικό χρώμα και μπορείς να το βρεις συχνά στον ουρανό και στο νερό. Ίσως γι' αυτό το μπλε συνδέεται με την ηρεμία και τη γαλήνη.



Πράσινο

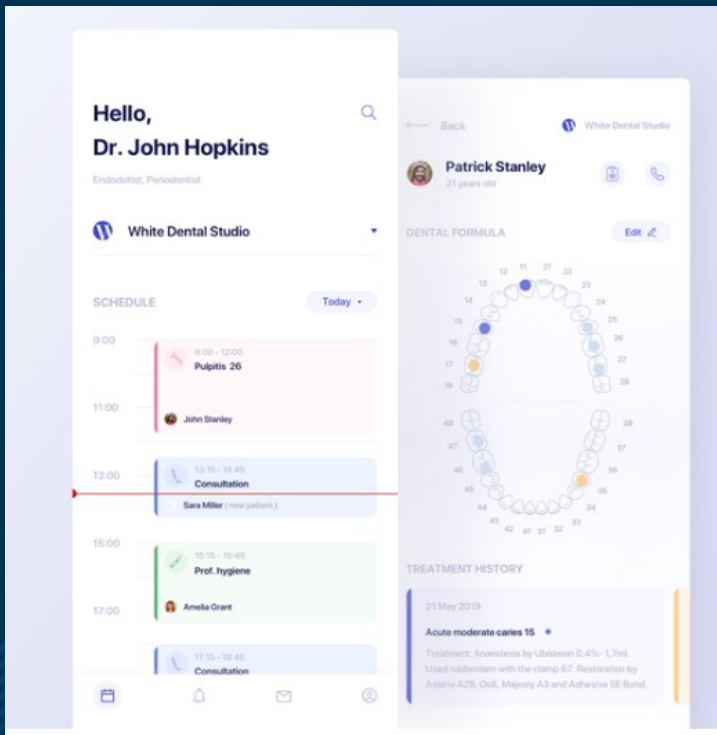
- Το πράσινο συμβολίζει τον κόσμο της φύσης. Πιστεύεται ότι το πράσινο απομακρύνει το άγχος και βελτιώνει την ευημερία.



Χρώματα

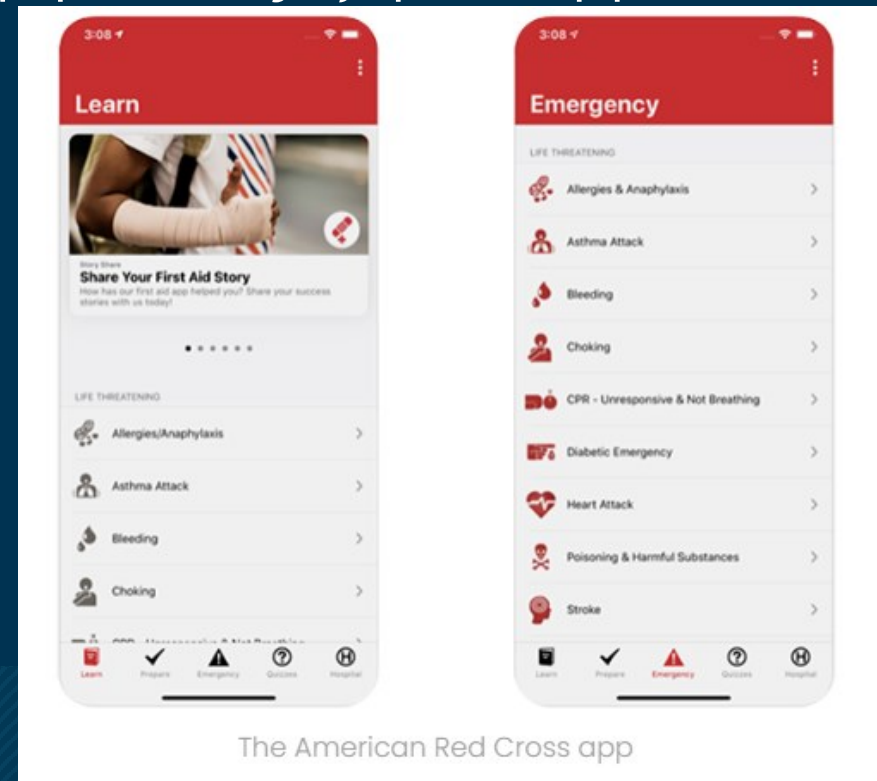
Λευκό

- Το λευκό χρησιμοποιείται συχνά για να μεταφέρει μια αίσθηση αγνότητας και ειρήνης. Προκαλεί θετικά συναισθήματα και ηρεμεί τους χρήστες.



Κόκκινο

- Το κόκκινο είναι επικίνδυνο γιατί αυτό το χρώμα μπορεί να προκαλέσει μια επιθετική διάθεση. Αλλά σε ορισμένες περιπτώσεις, το κόκκινο μπορεί να χρησιμεύσει ως εξαιρετικό εργαλείο.

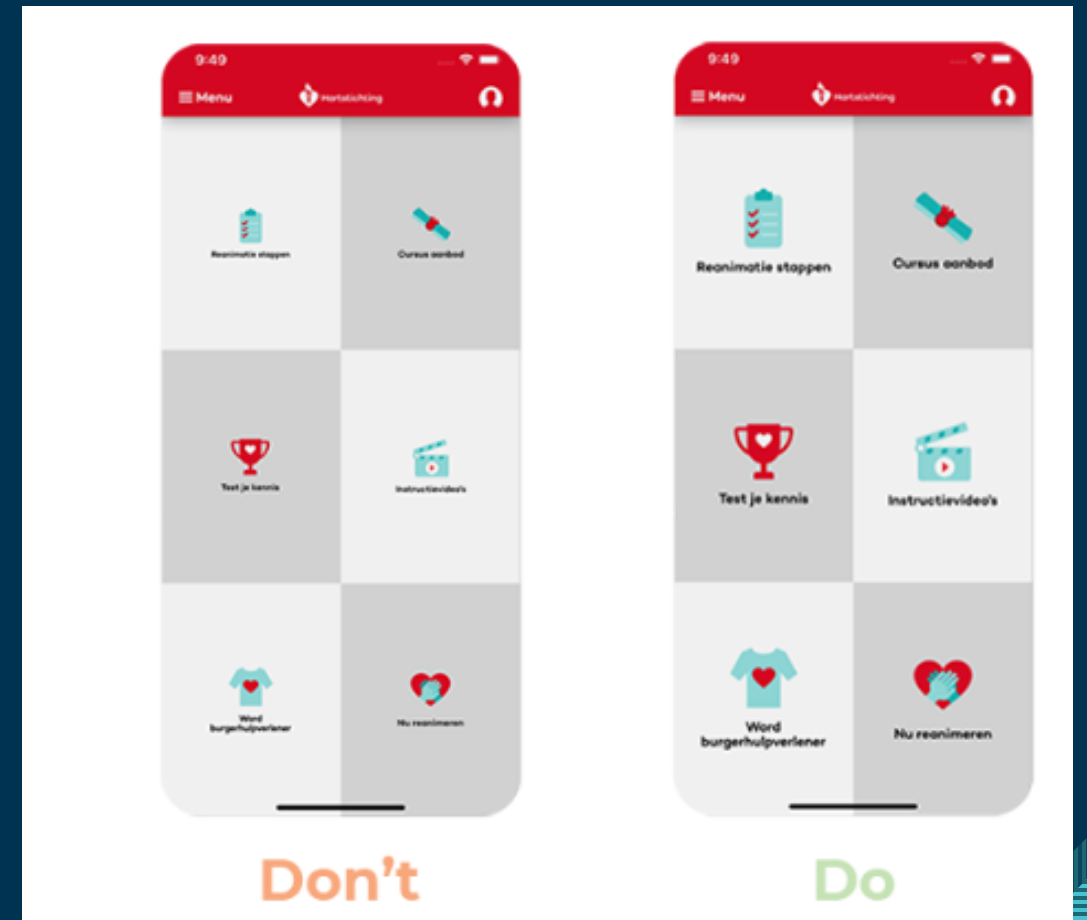


The American Red Cross app

Κουμπιά

Να είστε προσεκτικοί στα κουμπιά

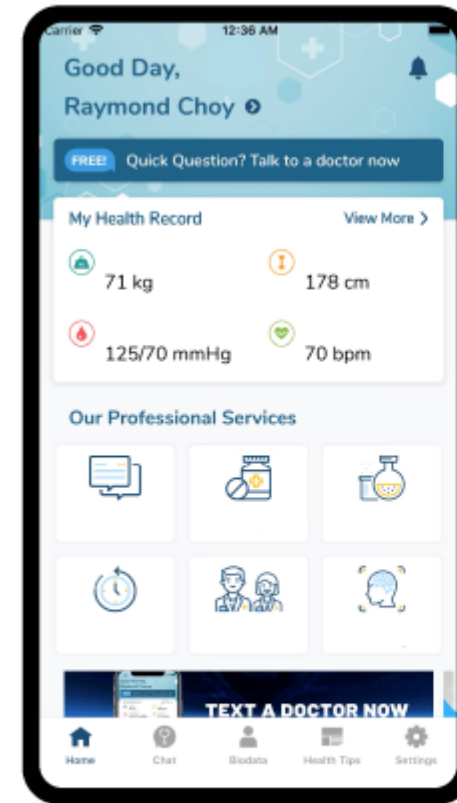
- Το μέγεθος και η τοποθέτηση των κουμπιών είναι ένα σημαντικό μέρος του σχεδιασμού διεπαφής χρήστη μιας ιατρικής εφαρμογής, καθώς βοηθά τους χρήστες να πλοηγηθούν και να εκτελέσουν τις επιθυμητές ενέργειες. Όλα τα κουμπιά πρέπει να είναι αρκετά μεγάλα και να βρίσκονται σε τέτοια απόσταση ώστε οι χρήστες να μπορούν να αλληλεπιδρούν μαζί τους χωρίς τυχαία χτυπήματα. Αλλά αποφύγετε να κάνετε τα κουμπιά τεράστια, καθώς αυτό θα παραβιάσει την ακεραιότητα της διάταξης.



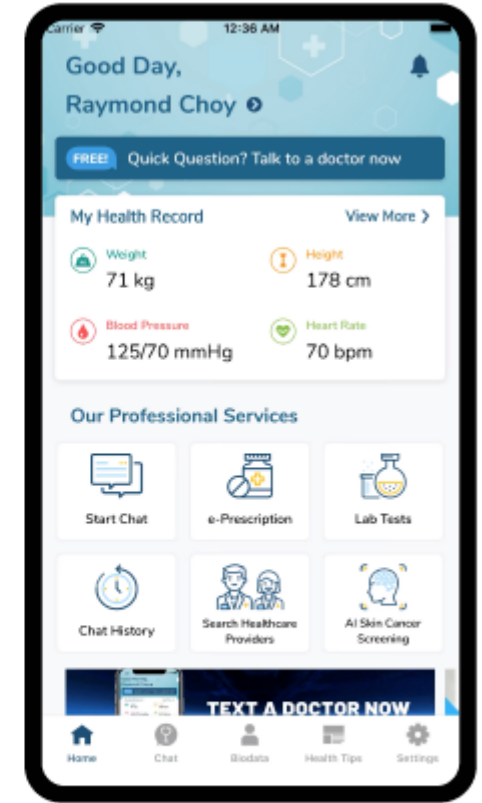
Εικονίδια

Χρησιμοποιήστε responsive icons

- Τα εικονίδια πρέπει να είναι διαισθητικά, έτσι ώστε οι χρήστες να μπορούν να καταλάβουν τι βρίσκεται πίσω από αυτά. Παρόλο που τα εικονίδια είναι ένα εξαιρετικό γραφικό στοιχείο, δεν πρέπει να τα χρησιμοποιείτε σε κάθε οθόνη, ωστόσο. Μην παραπλανείτε τους χρήστες και συνοδεύετε κάθε εικονίδιο με κείμενο.



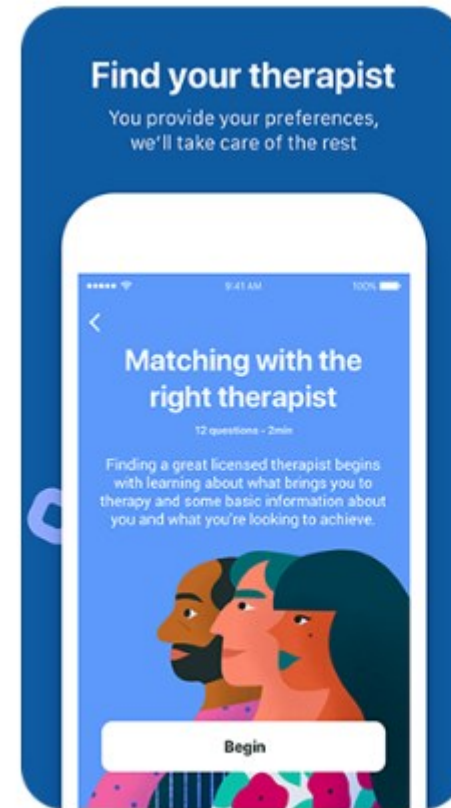
Don't



Do

Εικόνες και φωτογραφίες

- Οι εικόνες και οι φωτογραφίες μπορούν επίσης να δημιουργήσουν μια ευχάριστη εντύπωση όταν χρησιμοποιείτε μια ιατρική εφαρμογή. Χρησιμοποιήστε εικόνες υψηλής ανάλυσης για να είναι οπτικά ευχάριστες. Επίπεδες εικόνες και φωτογραφίες ανθρώπων μπορούν επίσης να δημιουργήσουν μια θετική ατμόσφαιρα.

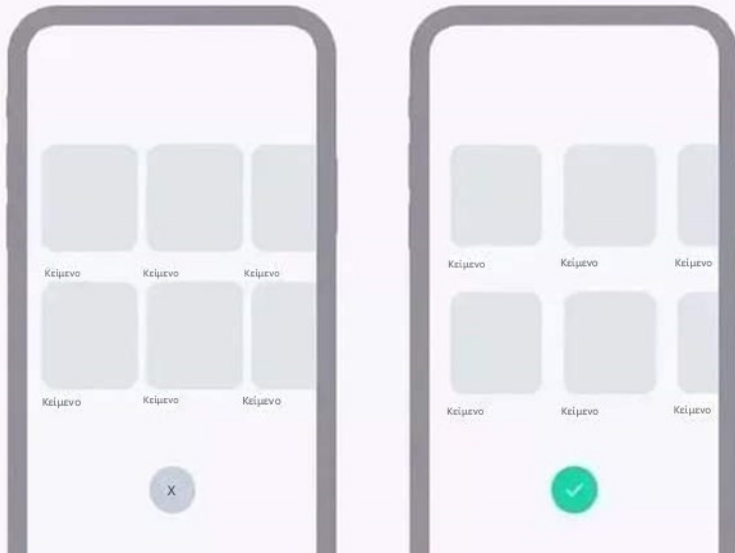


Talkspace Online Therapy app and Delta Dental Mobile app

User Interface Design

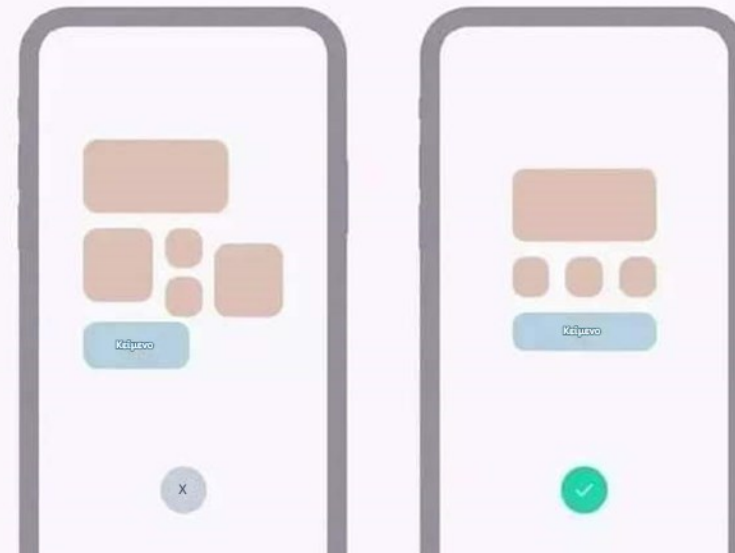
Κενό διάστημα

Το να δώσετε στον σχεδιασμό σας χώρο για να αναπνεύσει είναι ζωτικής σημασίας για ευκολότερη πλοήγηση.



Περιορισμός επιλογής

Μία από τις ισχυρές πλευρές του μινιμαλισμού στις διεπαφές είναι η βελτιωμένη συγκέντρωση του χρήστη.



User Interface Design

Απλές γραμματοσειρές

Η χρήση απλών και minimal γραμματοσειρών θα κάνει τα κείμενά σας και ολόκληρο το σχέδιό σας πιο ευανάγνωστα.

Oxygen Mono
ALGERIAN
Gilla



Product Sans
SF Pro Display
Poppins



Οι μικρές λεπτομέρειες έχουν σημασία

Οι σχεδιαστές πιστεύουν ότι χρειάζεται μόνο να μάθουν πώς να χρησιμοποιούν χρώματα και να τοποθετούν στοιχεία, αλλά όλα πηγαίνουν πιο βαθιά από αυτό.

Operating system
Description
Posts Classwork



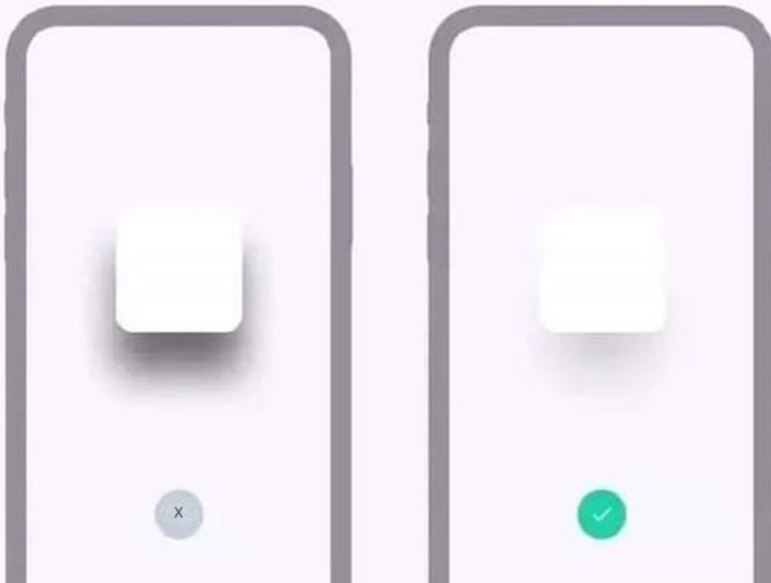
Operating System
Description
Posts Classwork



User Interface Design

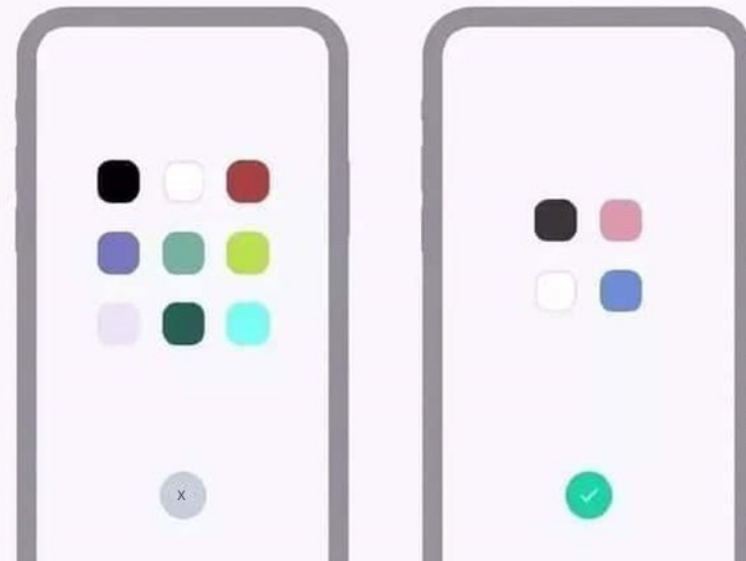
Βαθιές σκιές

Αυτό είναι ένα τεράστιο λάθος στον κόσμο του σχεδιασμού UI, αλλά συχνά μπορεί να φαίνεται καλύτερο εάν χρησιμοποιούσατε ελαφρύτερα μαύρα για να μειώσετε την καταπόνηση των ματιών και την τεράστια αντίθεση.



Περιορισμένη χρωματική παλέτα

Αυτό ενισχύει τα επιλεγμένα χρώματα και δεν αποσπά την προσοχή των χρηστών με υπερβολική ποικιλία.



Publish app to Android και iOS store

- Android:
 - Φάκελος android → app → src → main → AndroidManifest.xml
- Android:label : το όνομα της εφαρμογής
- Να είστε σίγουροι ότι δηλώνετε τα σωστά permissions
- Υπάρχει και AndroidManifest.xml στον φάκελο debug αλλά αυτός ο φάκελος είναι για όταν γράφουμε κώδικα

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2     package="com.example.publish_example">
3     <!-- συμπληρωσα εδω το permission να εχει προσβαση η εφαρμογη στο ιντερνετ-->
4     <uses-permission android:name="android.permission.INTERNET"/>
5
6     <application
7         android:label="Publish Example"
8         android:icon="@mipmap/ic_launcher">
9         <activity
10             android:name=".MainActivity"
11             android:launchMode="singleTop"
12             android:theme="@style/LaunchTheme"
13             android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale|fontScale|fontWeightAdjustment"
14             android:hardwareAccelerated="true"
15             android:windowSoftInputMode="adjustResize">
16             <!-- Specifies an Android theme to apply to this Activity as soon as
17                  the Android process has started. This theme is visible to the user
18                  while the Flutter UI initializes. After that, this theme continues
19                  to determine the Window background behind the Flutter UI. -->
```

Android

- Package identifier: πρέπει να είναι μοναδικό στο App Store
- Αν το αλλάξετε στο **AndroidManifest.xml** θα πρέπει να το αλλάξετε όπου αναφέρετε όπως:
 - android → src → debug → AndroidManifest.xml
 - android → app → main → kotlin → com → example → publish_example → MainActivity.kt
 - android → app → build.gradle

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.publish_example">
    <!-- συμπληρωσα εδω το permission να εχει προσβαση η εφαρμογη στο ι
    <uses-permission android:name="android.permission.INTERNET"/>
```

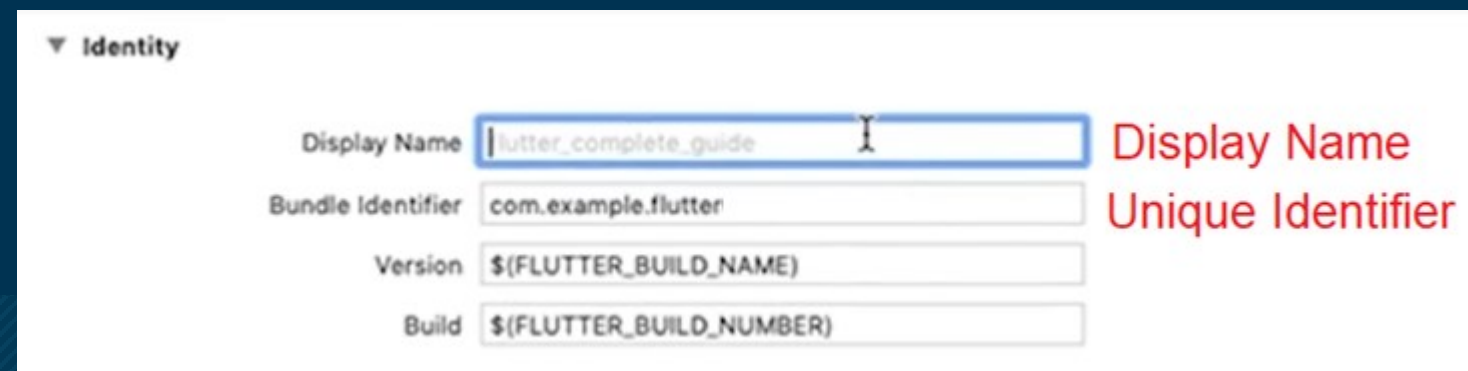
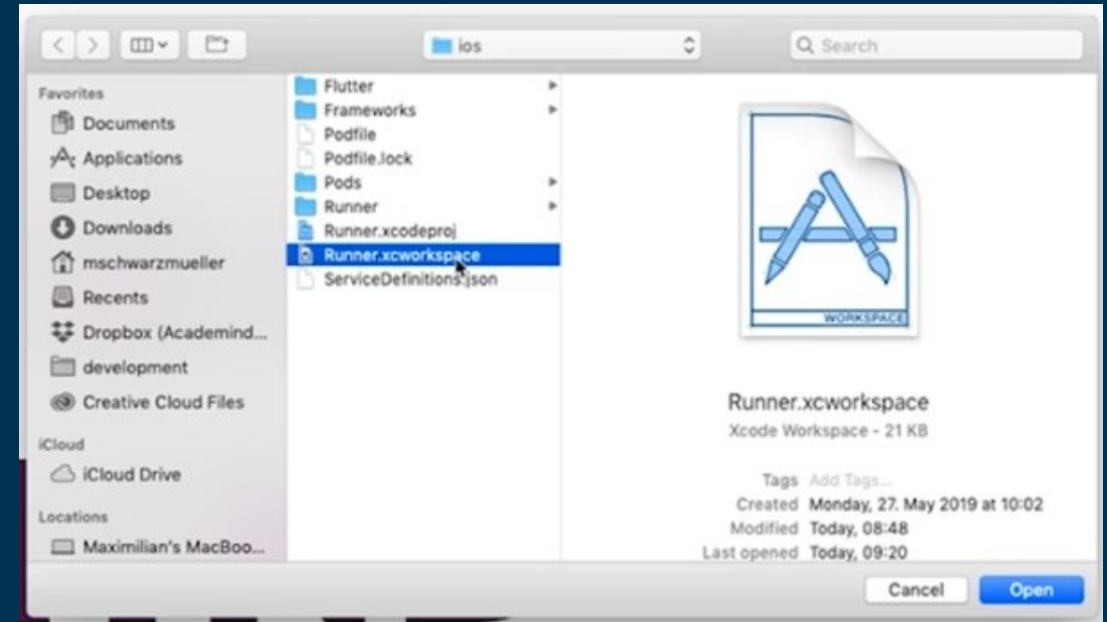
```
> .dart_tool          1  package com.example.publish_example|
> .idea              2
  < android           3  import io.flutter.embedding.android.FlutterActivity
  < app               4
    < src             5  class MainActivity: FlutterActivity() {
      < debug         6  }
      < main          7
        < java
          < kotlin\com\example\...
            MainActivity.kt
```

```
  < android           38
  < app               39
    < src             40
      < debug         41
      < main          42
      < profile       43
        build.gradle  44
      < gradle        45
        < gradle     46
          < gradle   47
            < gradle
```

```
    }
  sourceSets {
    main.java.srcDirs += 'src/main/kotlin'
  }
  defaultConfig {
    // TODO: Specify your own unique Application
    applicationId "com.example.publish_example"
    minSdkVersion 16
```

iOS

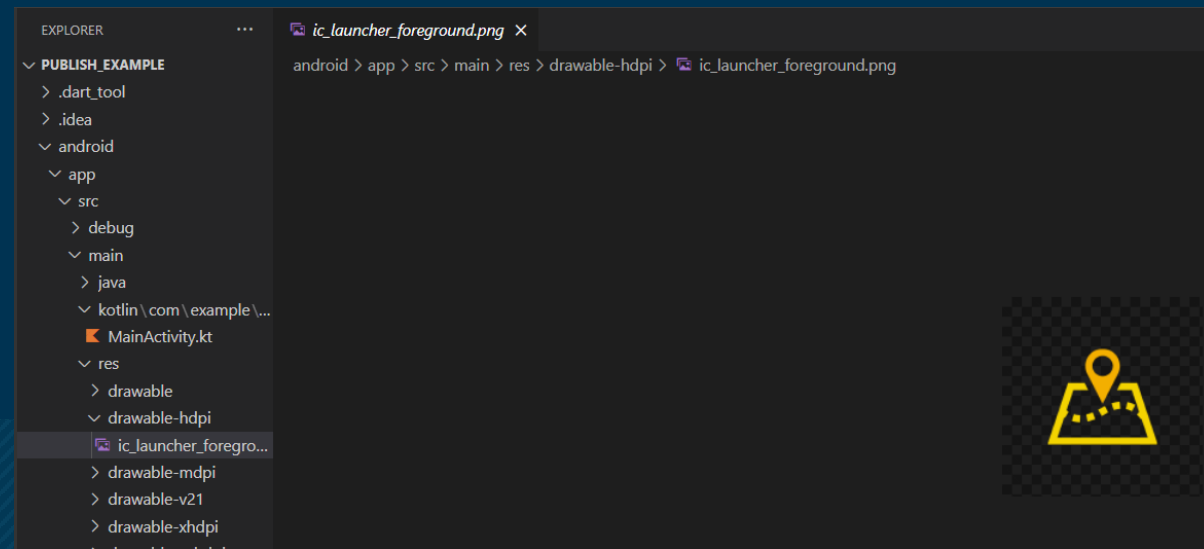
- ios → Runner → Info.plist :
ελέγχουμε αν είναι σωστά όλα τα permissions που χρειάζεται το app
- Το package identifier και το όνομα της εφαρμογής τα περνάμε από το XCode
 - ανοίγουμε τον φάκελο ios → Runner.xcworkspace



Adding Icons

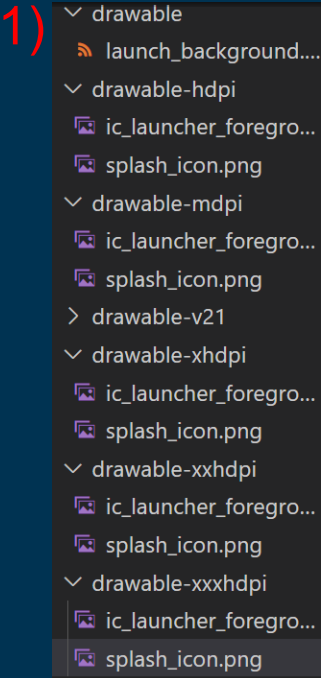
- flutter_launcher_icons
 - Χρησιμοποιούμε αυτό το πακέτο για να μας δημιουργήσει αυτόματα όλες τις εικόνες που χρειάζονται
 - Το βάζουμε στο pubspec.yaml γιατί δεν θέλουμε να είναι μέρος της τελικής εφαρμογής
- Image_path: εικονίδιο εφαρμογής
- Adaptive icon : τα στρογγυλά εικονίδια που χρησιμοποιούνται
 - Adaptive background : χρώμα
 - Adaptive foreground : εικόνα
- Θα δημιουργήσει εικονίδια για κάθε διαφορετική έκδοση που χρησιμοποιεί ο χρήστης και για Android και για iOS
- Τρέχουμε την εντολή που έχει το πακέτο στο Readme : `flutter pub run flutter_launcher_icons:main`

```
> android 37
> build 38
dev_dependencies: 38
  flutter_test: 39
    sdk: flutter 40
  flutter_launcher_icons: ^0.11.0 41
  # The "flutter_lints" package below contains a set of recommended lints to 42
  # encourage good coding practices. The lint set provided by the package is 43
  # activated in the `analysis_options.yaml` file located at the root of your 44
  # package. See that file for information about deactivating specific lint 45
  # rules and activating additional ones. 46
  flutter_lints: ^2.0.0 47
flutter_icons: 49
  android: "launcher_icon" 50
  image_path: "dev_assets/places.png" 51
  adaptive_icon_background: "#191919" 52
  adaptive_icon_foreground: "dev_assets/places-adaptive.png" 53
  min_sdk_android: 21 # android min sdk min:16, default 21 54
! analysis_options.yaml 55
publishicons.iml 56
pubspec.lock 57
pubspec.yaml 58
# For information on the generic Dart part of this file, see the 57
# following page: https://dart.dev/tools/pub/pubspec 58
```



Adding Splash Screen - Android

- Στο Android → app → src → main → res → launch_background.xml: ορίζουμε τι βλέπει ο χρήστης όταν φορτώνει η εφαρμογή
- 1) Βάζουμε σε όλους τους φακέλους drawable_hdpi κτλ. τις εικόνες για το splash screen δηλαδή το **splash_icon.png**
- 2) Στο values → colors.xml προσθέτουμε το χρώμα για το splash
- 3) Στο **drawable → launch_background.xml** και στο **drawable-v21 → launch_background.xml** προσθέτω το χρώμα από το colors και την εικόνα



2)

```
android > app > src > main > res > values > colors.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3  |   <color name="ic_launcher_background">#191919</color>
4  |   <color name="splash">#191919</color>
5  </resources>
```

3)

```
android > app > src > main > res > drawable-v21 > launch_background.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <!-- Modify this file to customize your launch splash screen -->
3  <layer-list xmlns:android="http://schemas.android.com/apk/res/android">
4  |   <!--item android:drawable="?android:colorBackground" /-->
5  |   <item>
6  |       <color android:color="@color/splash" />
7  |   </item>
8  |   <!-- You can insert your own image assets here -->
9  |   <item>
10 |       <bitmap
11 |           android:gravity="center"
12 |           android:src="@drawable/splash_icon" />
13 |   </item>
14 </layer-list>
15
```

Adding SplashScreen - iOS

