

From idea to App Store

Take an idea for an iOS app, and turn it into reality

A screenshot of a cell phone

Description automatically generated

Table of Contents

[Overview 3](#_Toc42812820)

[We Provide 4](#_Toc42812821)

[We Cover 4](#_Toc42812822)

[Prerequisites (You Need) 4](#_Toc42812823)

[Development Phases 5](#_Toc42812824)

[High-Level View 7](#_Toc42812825)

[Development Environment Setup 9](#_Toc42812826)

[Get Things Working Quick ‘n’ Dirty 10](#_Toc42812827)

## Overview

This tutorial provides detailed a roadmap and step-by-step instructions on how to take an idea for an iOS app and turn it into a reality that is available on the Apple App Store.

A picture containing mirror

Description automatically generated

The **Placezz** app, together with its backend **Vapor**-based (Swift on the server) system provides the following functionality:

* Allows the user to **capture data related to their favourite places**
* The data includes **geographic location**, plus **text** and **media** (images, videos) and sharing **permissions** (see below)
* Users may **share places** with other users

A drawing of a face

Description automatically generated

The example chosen for this tutorial is representative of a typical modern, real-world iOS app project:

* Presents a beautiful, intuitive interface to the user in a **native iOS** app
* **Adapts** itself to work correctly on all devices from an iPhone SE to an iPhone 11 Pro Max and a 12.9-inch iPad Pro
* **Caches data** to minimize network traffic, copes with non-availability of the network and **syncs** to the backend when network connectivity is available
* Interfaces with a cloud-deployed backend system via a **REST Web API**
* Provides additional functionality to the user via **in-app purchases**
* Uses **Git** and **GitHub** to manage source code and associated resources
* Uses **Docker** to create and deploy test environments

### We Provide

* Full **step-by-step instructions**, guidance and discussions on technology choices
* All **resources** required in the creation of the app

### We Cover

* **High-level design** of the app
* Deciding on **the technology stack** from available choices
* Designing the **UI**
* Designing the **code** components and their **interfaces**
* Designing the **data model** and **persistence** layer
* **Iteratively** creating the app, phase-by-phase to add features and complexity
* Creating the **UI** **programmatically**
* Creating the **backend system** using Vapor (Swift)
* **Networking**
* How to create **In-App Purchases**
* Working with **databases**
* Working with **Git** and **GitHub**
* Working with tools like **Paw** and **Proxyman**
* Working with **Docker** (locally and remotely)
* **Testing** (UI and unit tests)
* Working with Apple’s **App Store Connect** to get your app into the **App Store**

### Prerequisites (You Need)

* A Mac running **macOS Catalina** with **Xcode 11** or higher installed
* A paid **Apple Developer account** if you want to distribute the app on the App Store
* Basic familiarity with **Swift** and **iOS development** using Xcode (i.e. you’re not a complete novice)

### Development Phases

The main phases of creating the app will be:

A picture containing drawing

Description automatically generated

* Setting up the Development Environment
  + How to set up everything you need to develop the app and backend system
  + This includes Xcode, Vapor, Git and a number of other tools such as Proxyman, PAW, etc.
* Quick ‘n’ Dirty Working Technical Proof
  + “Kick the tires” to prove to ourselves our technical solution will work
  + It’s always good to get something working immediately so we can proceed with the rest of the project knowing that the thing actually works!
* High-Level Design
  + What do we want it to do and roughly what will it look like
* The Data Model
  + What data will be working with on the front and backend
* First Pass Code Design
  + Sketch out what the main components of the front and backend code will look like
* First Pass App
  + Get a very rough version of the app working with mocked-up data
* First Pass Backend
  + Implement a quick ’n’ dirty version of the backend working
* First Pass App and Backend Integration
  + Get the front and backends talking to each other
* Complete Backend and Tests
  + Do a full implementation of the backend REST API, along with unit tests
* Complete App and Tests
  + Complete (in stages) the app and integration with the REST API backend, along with unit and UI tests
* Refactoring
  + Improve the app and backend and refactor where necessary
* Production Ready
  + Get the app and backend ready for production use
* The App Store
  + All the steps required to get your app into the App Store

## High-Level View

Before we go any further, let’s set the scene and take a quick look at what we’re going to be building.

Essentially, there are two pieces that work together to give the user a complete app experience:

* placezz-api
  + A **Vapor** (Swift on the server) **REST** (see below) **Web API** backend system that can store user data and respond to requests from the iOS app
  + Data is passed between the app and the backend as **JSON** (see below)
* Placezz app
  + An **iOS** app that enables the user to “bookmark” their favorite places, along with maps, text, images and videos
  + Works together with the **placezz-api** system to store and retrieve user place data

A close up of a device

Description automatically generated

…REST is …

… JSON (JavaScript Object Notation) is an open standard data exchange format

## Development Environment Setup

Dadsa

A screenshot of a cell phone

Description automatically generated

## Get Things Working Quick ‘n’ Dirty

sdasd