Sorghum App Developer Documentation

Kansas State Extension Platform

Kansas State University | Spring 2020

Table of Contents

Program Structure Overview

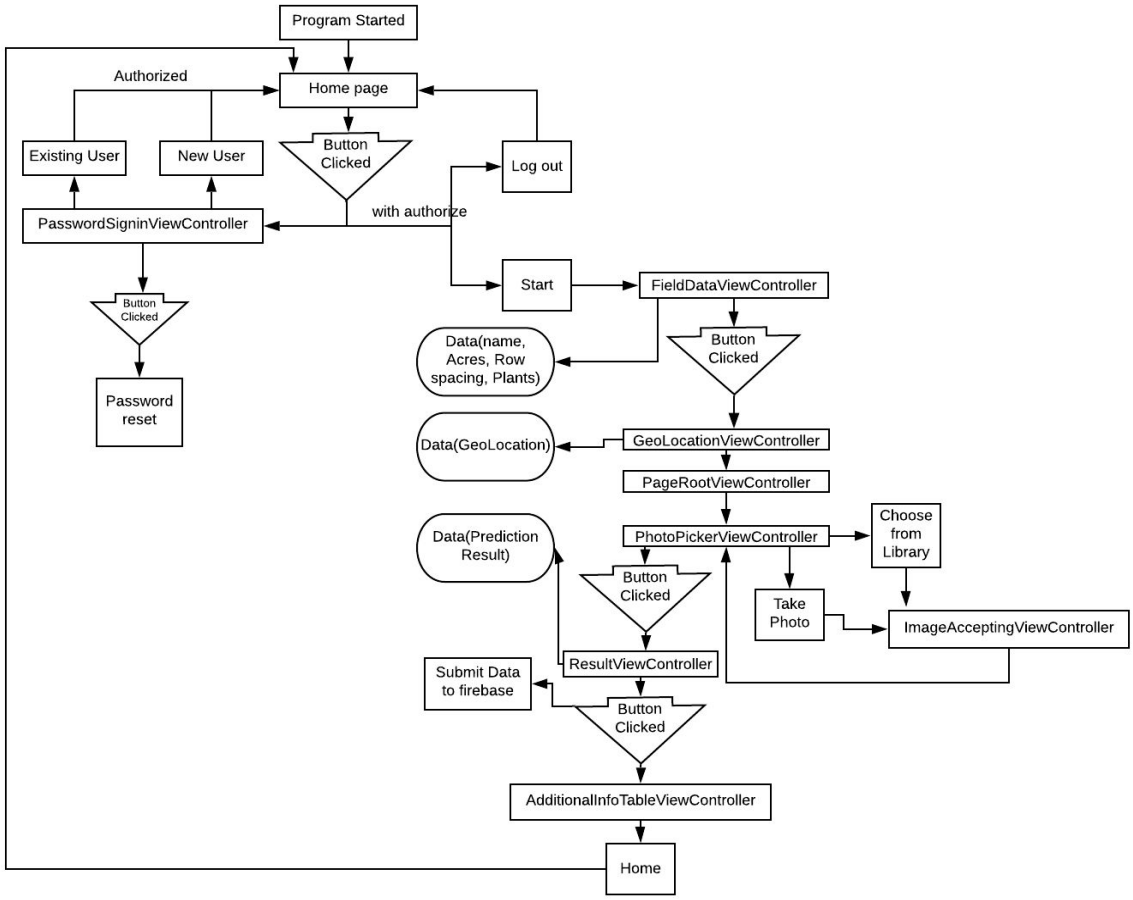


Figure 1.0

Figure 1.0 illustrates how this Sorghum App program Structure works. How it collects data to fill out the needed information for the research table and yield prediction by interacting with the user command.

Class Responsibilities:

- **Home page**
    - Pulls home page UI to current screen. It checks to see if the user is signed in. If not, it displays a button called sign in. This button will call the PasswordSigninViewController. If the user is signed in, then the UI will have two available buttons. One called “Start” and the other one called “Log Out”. If the user presses “Start”, it will call the FieldDataViewController. If “LogOut” is

pushed, the UI will be reset back to the original screen which only allows “Log in” to be pressed.

- **New User**
    - Creates a new user with imputed data and stores it in the database. Then loops back into the home page. Users are now authorized if valid information is inputted.
- **Existing user**
    - Creates a sign in UI with a user ID field and a password field. If the user enters a valid user name and password, then the user will be authorized and put back to the home screen with “Start” and “Log Out” activated. If the information is incorrect, it will show the user that their inputted information was invalid and prompt them to try again.
- **PasswordSigninViewController**
    - Checks if inputted email is linked to an account. If not, will send it to New User. If so, will send it to Existing User
- **Password reset**
    - Controls asking the user what they would like to change their password to. It will then change that password in the database.
- **Logout**
    - Signs the current user out and throws them back to the original sign in page.
- **Start**
    - Starts the main functionality of the app by first calling FieldDataViewController.
- **FieldDataViewController**
    - Creates a UI for the user to input various data inputs for the sorghum field in question. Will take that data and store it for future use. Will also be stored in the database. They will be used to help calculate the final yield of the field.
- **GeoLocationViewController**
    - Stores the geographical location which is taken from the users cellphone. Sends this data to the database.
- **PageRootViewController**
    - In charge of displaying visual information to the user on how to correctly take a picture of a sorghum head for the app. Allows the user to skip this information if desired.
- **PhotoPickerViewController**
    - Displays UI information on currently accepted pictures and allows the user to pick up to 10 pictures. The user can either click “Choose from Library” or “Take photo”. “Choose from Library will have the user pick a photo from their phone's photo library. It will then call the ImageAccpetingViewController to check if the

photo is readable by the app or not. If it is not, will kick the user back to the original UI of the PhotoPickerViewController and tell them it was an invalid picture. The “Take photo” button is similar, but just pulls up the phone's camera app instead of looking at its photo library.

- **ImageAcceptingViewController**
    - Looks at the photo given to it and checks to see if it is a sorghum head. It also checks to see if it can identify a measuring source and if it is formatted correctly.
- **ResultViewController**
    - Calculates and displays the information of the field. The UI includes a slider at the bottom which allows the user to change the amount of Seeds per pound. It will update the visualized data according to what the user has selected. Once the submit button at the bottom is pressed, it will ask the user for permission to store all of the data which it has collected. If the user agrees, it will send the data to the database and will then call the AdditionalInfoTableViewcontroller. If the user disagrees, then it will just call the AdditionalInfoTableViewcontroller.
- **AdditionalInfoViewController**
    - Show the user the total yield prediction of their field in bushels. It also displays links to “Sorghum research publications” and the “Sorghum production handbook”. Additionally, it provides credits to the “Noun Project”. The user can then press the “Home” button at the bottom of the screen to take them back to the home page. The user will stay signed in.

Development Environment Setup

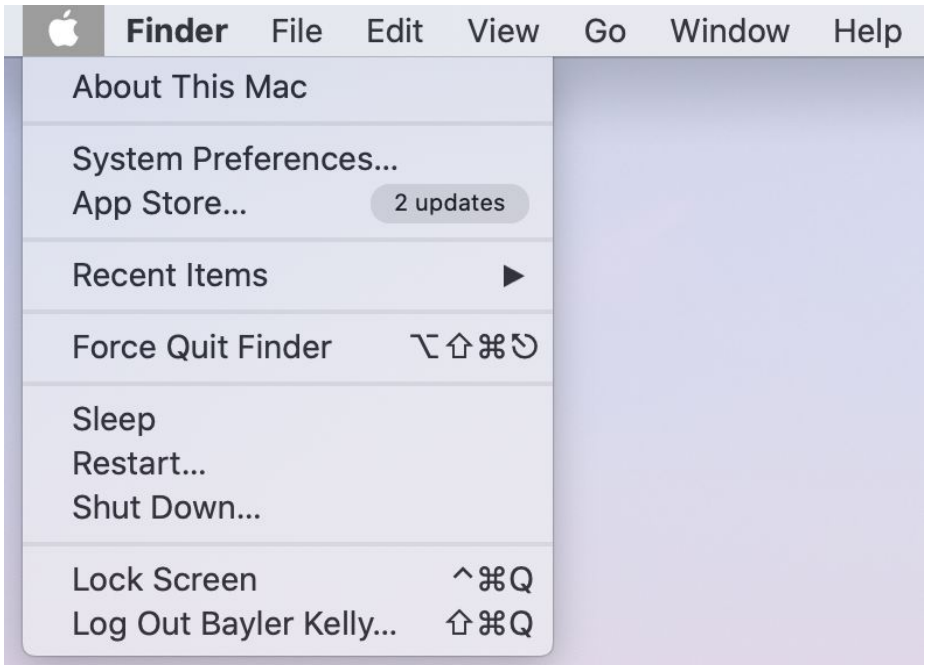
This section provides step-by-step instructions on how to install the proper IDE for this project, clone/download the current version of the sorghum application from GitHub and prepare the system to run locally. These instructions are written for a Mac machine or a system virtually running this operating system.

XCode 11 Setup

Check current OS version

XCode 11 currently requires a Mac running macOS Mojave 10.14.4 or later. To check your system’s current operating system, do the following:

- In the top-left corner of your screen, click the Apple icon. From this dropdown menu, select “About This Mac” to find information about your current machine.

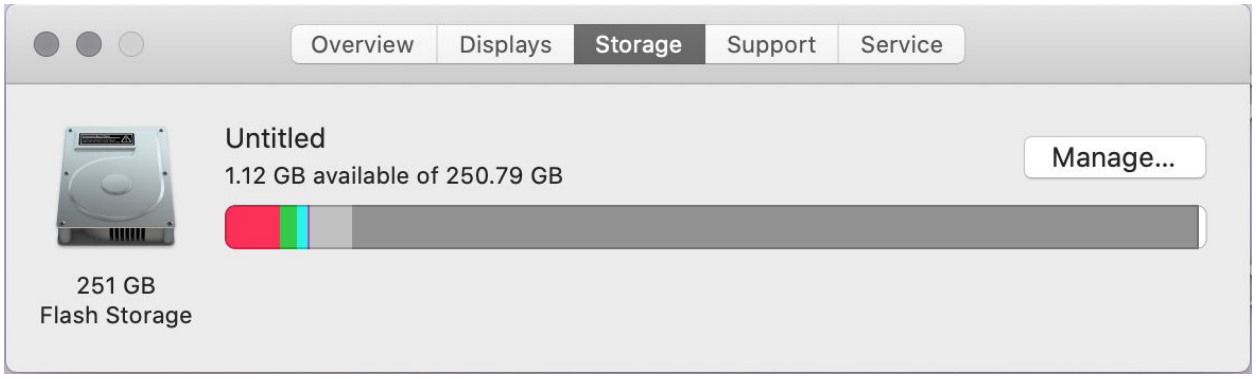


- From this page, you can find your current device, various component information, and your current Mac operating system. As long as your machine is running Mojave 10.14.4 and beyond or any version of Catalina beyond 10.15, it will be compatible with XCode 11.

Check current available storage

XCode 11 currently requires at least 8.1 GB of free storage on your hard drive. To check your system's current available storage, do the following:

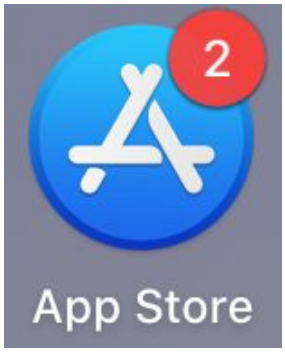
- Return to the "About This Mac" page from the above-noted instructions. From here, click the "Storage" tab along the navigation bar. It may take a moment to calculate your current storage usage. Your storage statistics will be displayed as done below.



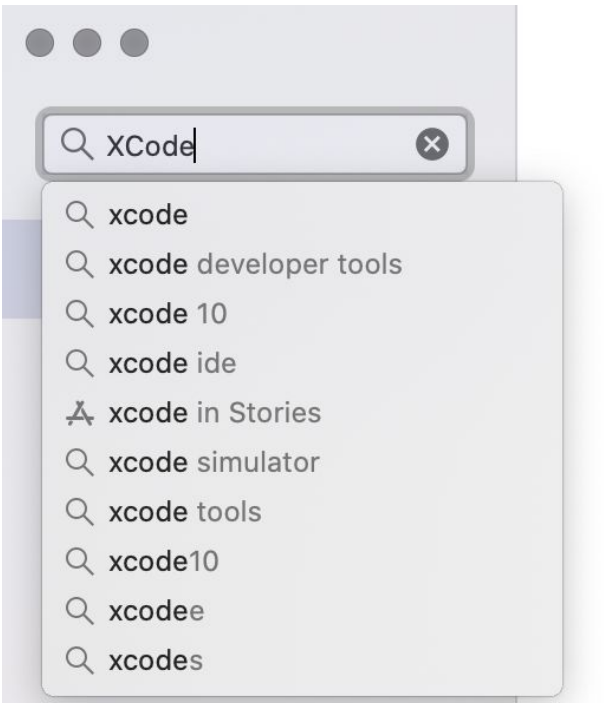
Install XCode 11

Once you've met the above requirements, you are reading to install XCode 11 on your machine. To begin this process, do the following:

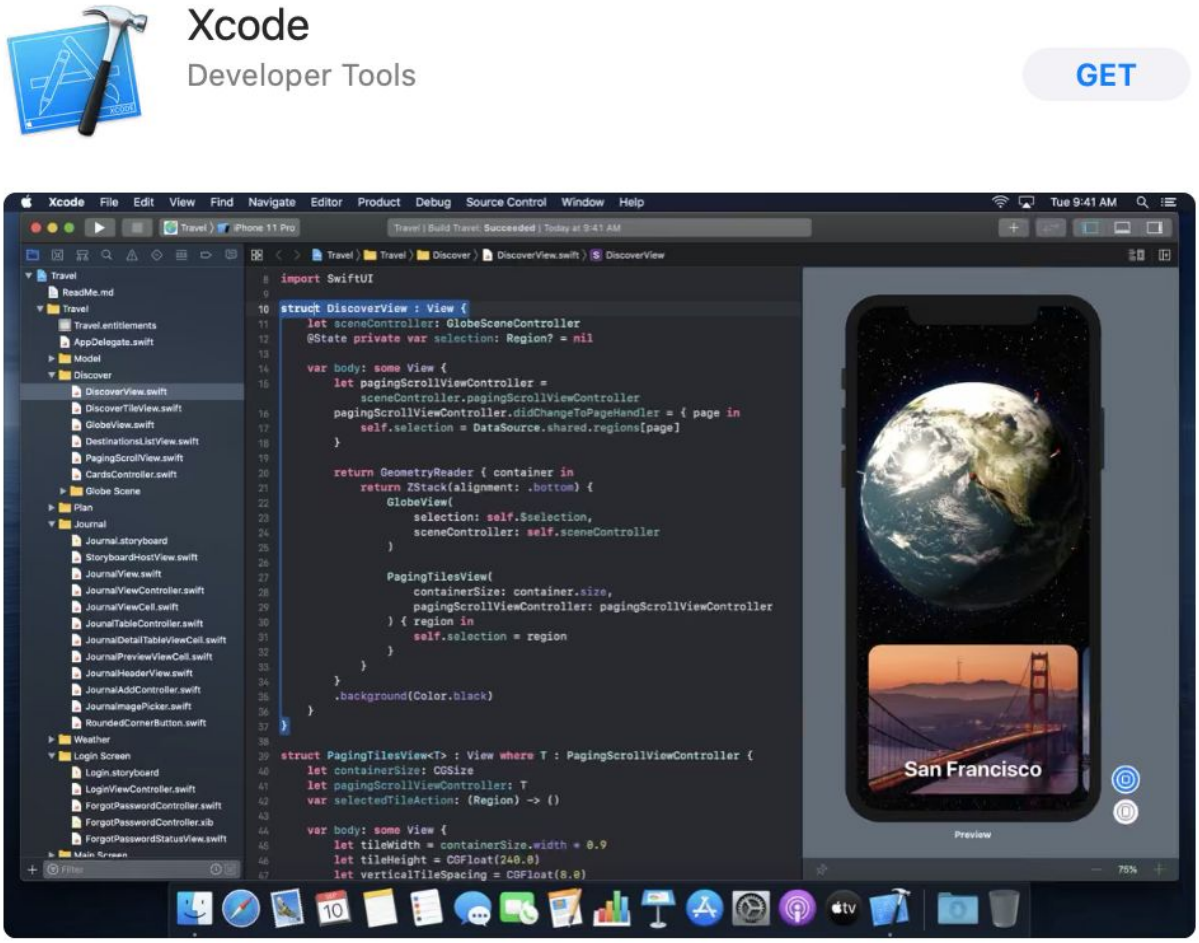
- Begin by navigating to the App Store. This application can, by default, be found on the Dock, but is also available using the Launchpad. Once it has been located, click the icon to open the App Store



- Once the App Store has loaded properly, find the search bar in the top-left corner of the application. Type “XCode” into the entry box and click on the result labeled “XCode”.



- As soon as your entries have loaded properly, the first result should look like the following image. This is the IDE that we want to install to our machine. Click “GET” to begin this installation.
    - After doing so, you’ll be prompted to input your Apple ID and password to complete this transaction, depending on your account settings. At this point, any prompts are necessary to complete the XCode installation.



- Once the download and installation has completed, XCode will be available for use in the Launchpad. The application looks like the image below.

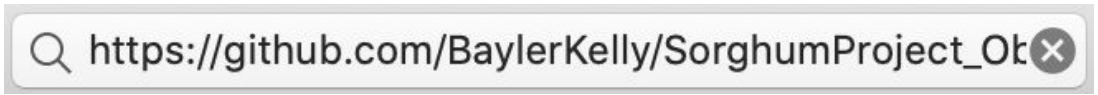
Running the Sorghum app locally

Cloning the current GitHub repository

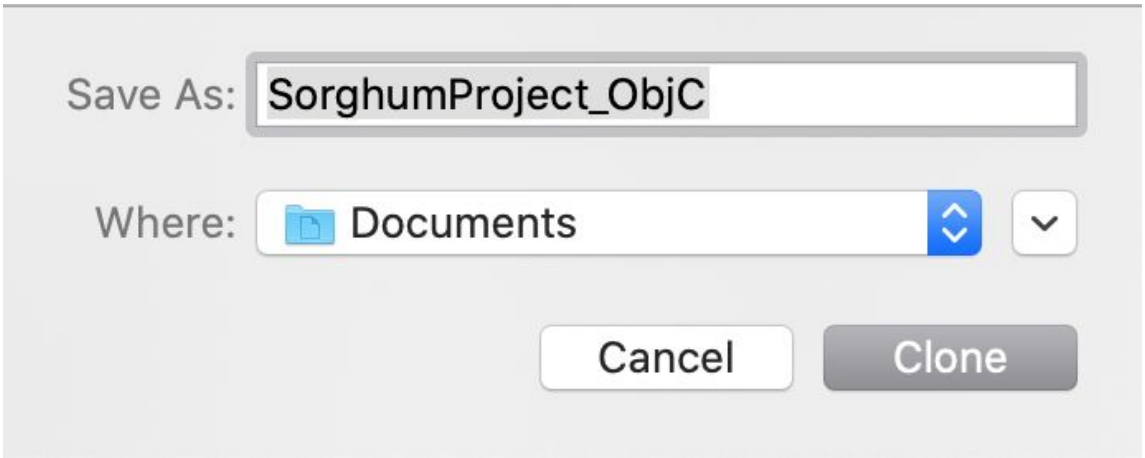
The first necessary step now that we have XCode 11 properly installed is to run the application and clone the current Sorghum app repository so that it can be run locally. To begin this process, do the following:

- Once it launches, you will be prompted with the window below. From this page, click "Clone an existing project" in the bottom-left area.

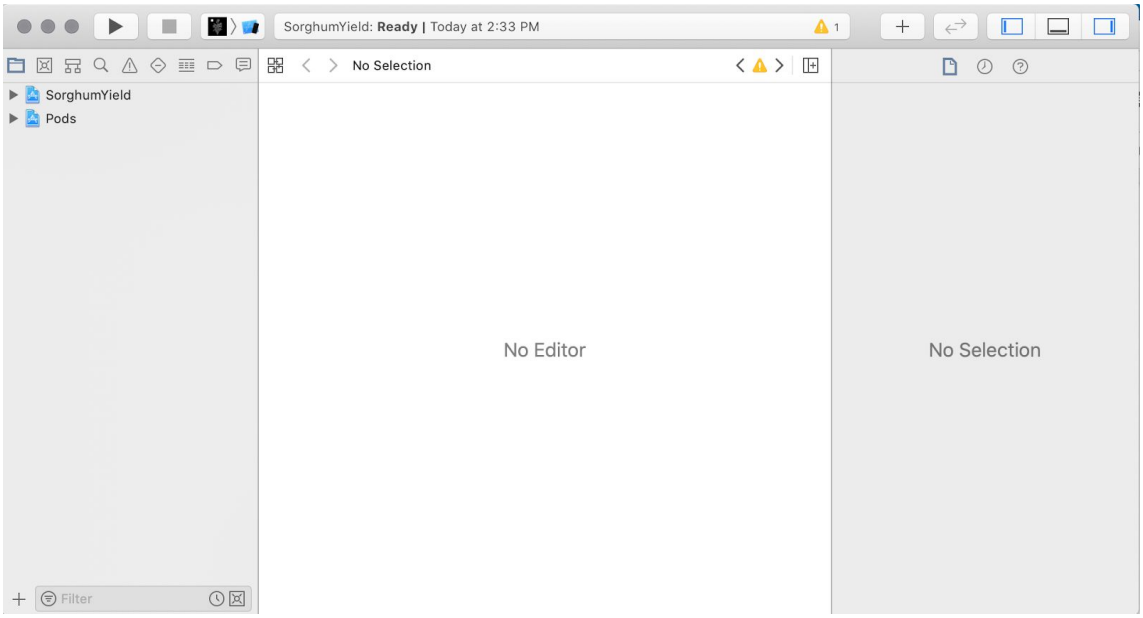
- After this, a new window will open asking for a link to a current project's repository. In the entry box, input the current GitHub cloning link for the Sorghum app. Proceed to the next step by clicking the "Clone" button in the bottom-right corner of the window.
    - Current clone link: https://github.com/BaylerKelly/SorghumProject_ObjC.git



- Following this, you will be prompted to indicate a location to save the cloned repository to. Determine where you would like to save this project and click the “Clone” button to continue.
    - After clicking the “Clone” button, XCode will begin cloning the project from the GitHub repository. This may take a few minutes, depending on your internet connection.



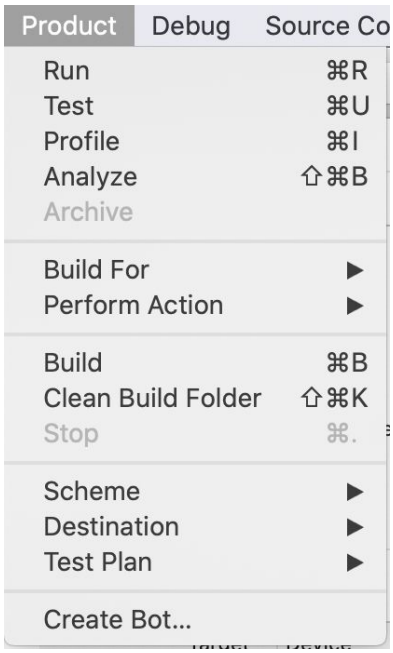
- Once the cloning process has completed, XCode will open the cloned project. It will open the project to the main editor screen like the following:



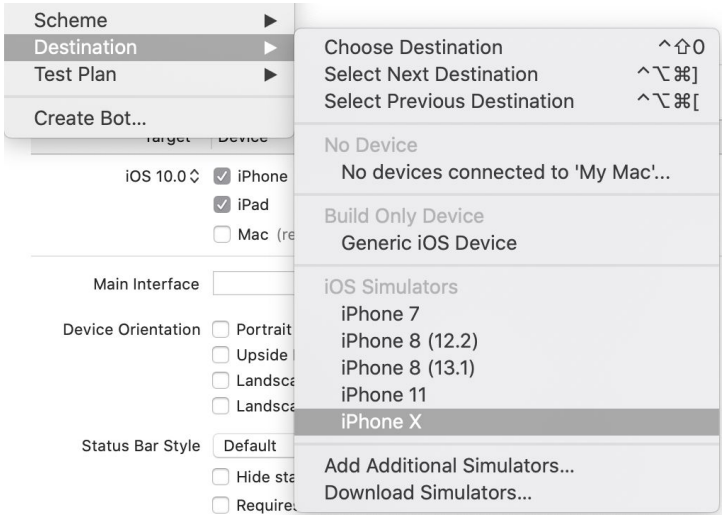
Running the project on the computer

The final step is to run the project locally on an XCode simulator. To setup a proper simulator and run the application, do the following:

- Along the top navigation bar for XCode, click “Product” to display the drop-down menu



- Once the drop-down menu is enabled, highlight “Destination” with your cursor. This will display a list of possible destinations to deploy/run the project on. Click any of the valid iOS simulators listed to set on which iOS device the project will run locally.



- The final step of this process is to the run project. In the top-left region of the XCode window, click the grey play button icon to run the current build.

Downloading the project on an iPhone

If you would like to download the app onto an iPhone, you must do the following.

- Add your AppleID to the XCode Accounts
  - From the top toolbar, click XCode > Preferences
  - Navigate to the Accounts tab
  - Click + under the Apple IDs section
  - Select AppleID and click “Continue”
  - Enter your AppleID credentials
  - Return to the main XCode project
- Configure the project for your device
  - Plug the iOS device into your computer
  - From the top toolbar, click Product > Destination > “Your iOS Device”
- Configure the project
  - In the left column, click Sorghum Yield
  - While Sorghum Yield is selected on the left side under “Targets”, insert a unique name under “Bundle Identifier” (Example: com.yourname.SorghumYield)
  - From the project toolbar, select “Signing & Capabilities”
  - Under the signing tab, select Your Team under the Team dropdown menu
  - Select “SorghumYieldUITests” on the left side under “Targets”
  - Under the signing tab, select Your Team under the Team dropdown menu
- Build the app
  - In the top left, click the play button to build the current project
  - If/when prompted, insert the password for the Apple keychain
  - If/when prompted, unlock the iOS device
- Approve yourself on your iOS device

- On your iOS device, navigate to Settings > General > Device Management
- Select your entry under “Developer App”
- Click “Trust “Apple Development: youraccount”

- Launch the app on your iOS device

App Certificate Renewal

This portion of the installation guide details the necessary process to renew the Sorghum app certificate when the app no longer opens properly. This procedure will need to occur nearly every 3-4 days.

- Pre-Installation Requirements:
  - Mac with XCode 11
  - iPhone with fairly current iOS version (Below 13.0)
  - Access to the Sorghum Yield repository

*NOTE: If the app has never been previously installed using your current computer or on your phone then follow the instruction process detailed in Sorghum App Installation.

1. Certificate Renewal:
   a. Basic Setup
      i. Open XCode on the computer you will be using for certificate renewal
      ii. Plug the iPhone into the USB port
   b. Configure the project for your device
      i. From the top toolbar, click Product > Destination > “Your iOS Device”
   c. Verify that the project is still signed by your account
      i. In the left column, click Sorghum Yield
      ii. While Sorghum Yield is selected on the left side under “Targets”, verify that there is a unique name under “Bundle Identifier” (Example: com.yourname.SorghumYield)
      iii. From the project toolbar, select “Signing & Capabilities”
      iv. Under the signing tab, select Your Team under the Team dropdown menu
      v. Select “SorghumYieldUITests” on the left side under “Targets”
      vi. Under the signing tab, select Your Team under the Team dropdown menu *NOTE: If you can’t find Your Team under the dropdown menu, navigate to the Sorghum App Installation process above. You will need to add your account to XCode. (Instruction 3d)

d. Build the app
    i. In the top left, click the play button to build the current project
    ii. If/when prompted, insert the password for the Apple keychain
    iii. If/when prompted, unlock the iOS device
e. Verify that you're approved on your iOS device
    i. On your iOS device, navigate to Settings > General > Device Management
    ii. Verify that your entry is under "Developer App"
    iii. If not, select your entry under "Developer App"
        1. Click "Trust "Apple Development: youraccount"
f. Launch the app on your iOS device