

//TODO: Team Name

1812 Sir Isaac Brock Way
St. Catharines, ON, L2S 3A1
(905) 688 - 5550

xLights Companion Pro

January 16th, 2021

Contents

1. Introduction
 - a. Purpose
 - b. Scope
 - c. Intended Audience
 - d. Intended Use
 - e. Definitions, Acronyms, and Abbreviations
 - f. Overview
2. Overall Description
 - a. User Needs
 - b. Assumptions and Dependencies
3. System Features and Requirements
 - a. Functional Requirements
 - b. System Features
 - c. Non-functional Requirements
 - d. Other Specifications

Introduction

xLights is an open source program that allows people to design light sequences timed to music. This allows easy tuning of lights through effects and animations to create unique light displays. xLights Companion Pro is an Android and iOS application that is designed to improve productivity and reduce the learning curve for all xLights users.

There are a set of questions that get asked on a frequent basis by people new to the xLights community. The learning curve to this product can sometimes be steep and there is no good collective resource for information to help these people out. They need help with things like searching for sequenced songs, prop parts, and specialized calculators.

As a team, we conducted a poll on Facebook in the Official xLights Support Group with over 14,000 members to determine if there is a need for an app to help with the xLights product. We also gathered responses to what features the community would like to see in it. The poll had 268 votes as “interested” and 4 votes as “not interested” within 24 hours. This shows us that there is a purpose for our app with a 98.5% positive ratio of those who saw the poll.

Our goal is to cut down on redundant questions from new users of the xLights community. The main goal for this app is an all in one resource that the xLights user would need. The features we are planning to implement will allow for a simple one stop shop for all things xLights. The importance of this app will be to solve the frustration people may feel trying to set up their lights and create an experience that is easy and fun to use.

This document elaborates the software requirements document for the system “xLights Companion Pro”. The system requirements are abstracted into many views and components which are explained in detail. The document follows the 4+1 view model as the reference model for this document.

Purpose

This document gives an outline of the project we will be working on over the course of the semester. The document will be used to keep track of our vision for the app, and will allow us to measure our progress based on the requirements we defined here.

Scope

This requirements analysis document covers both the static and dynamic aspects of our xLights application. We are using the 4+1 Model to get as many views of the system as possible and with this collection of views we get the full scope of the system. Static behaviour is represented using class diagrams and other views while dynamic behaviour is represented by case diagrams as well as sequence diagrams.

Intended Audience

People who want to, or actively use the xLights light sequence application. Beginners to the xLights app who are overwhelmed and not sure where to start, or seasoned xLights users who want a simple and easy way to accomplish many of their xLights-relevant tasks. This majority of users will be middle-aged homeowners.

Intended Use

Become a part of the xLights community, and accomplish tasks related to using xLights in a quicker way. This includes links to xLights forums and community websites, a calculator to help people choose the right parts to work together for their light display, and easy shop links to speed up the buying process.

Overview

This report will describe an analysis of the architecture of the xLights Companion Pro app. It will outline the representation, goals, constraints, and use cases. We will also cover and outline the logical, process, deployment, and implementation views of the system. Finally, we cover the user interface design that persists across platforms.

Overall Description

User Needs

The users of the app will consist of users of the xLights application, and testers. The primary user will be the xLights users. For these users the most important goal is ease of use and easily being able to find the information that they would like to use. If they cannot find the tool, or find it too difficult to access the tools they need, they may decide to stop using it. The secondary user of the app is the testers. For the testers they need to be able to look at and test all individual parts of the application in a quick and easy manner. This means being able to quickly get to any page from any page is ideal, and minimize the amount of time needed to use any given page.

Assumptions and Dependencies

We will assume that all users of our application will have at least a general idea about the xLights product. Due to this application being used with the xLights product, we will also assume that if not right away, eventually users will buy programmable lights and a controller (if they don't already own them). This application will **not** depend on users already owning the xLights products. The only dependency that exists is access to a smart mobile device to download our application.

System Features and Requirements

Functional Requirements

- Users should be able to find links to various xLight community web pages and resources.
- Users should be able to place AR props in the application.

-
- Users should be able to calculate whether certain physical components will work together for their project.

System Features

- Toolbar/ Navbar
- Search Engine
 - Search basic user-friendly terms (Users who don't have expert knowledge)
 - Prop Vendor Search
 - Song Search
 - Lyric Search
- AR prop placement
 - AR photo tool
- Links
 - xLights app
 - xLights social media
 - Zoom
 - Forums
 - Video resources
 - Remote falcon (local)
 - Part Supplier Links (Only a link if external)
 - Build plans
- Power Consumption/Cost Calculator
- Voltage divider
- Organize props into power slots
- Tree Calculator (how many lights per tree)
- AC/DC Calculator

Non-functional Requirements

Persistence:

- Information that a user saves will be saved locally on their device in the form of JSON data.

Reliability/Availability:

- Offline aspects will be available at all times.
- Calculator functionality should be available offline.
- Performance for both the android and iOS products will have the same timeout times of 30sec when accessing resources from api calls or other microservices.

Recoverability:

- If user data errors occur, an error message will be displayed to the user.

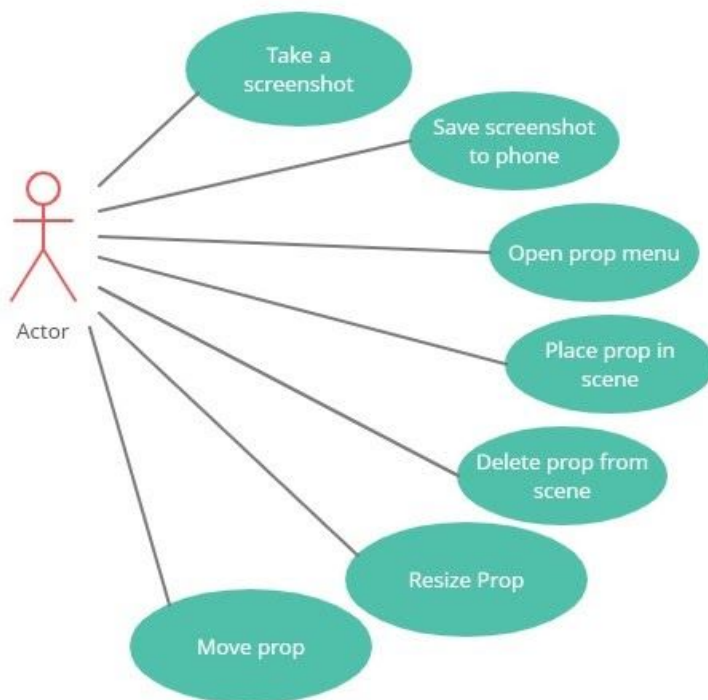
Usability:

- Android apps should feel native to Android users, iOS apps should feel native to iOS users.
- Both Android and ios versions of the application layouts should feel very similar to each other when switching between devices.
- The app should load in less than 500 milliseconds.
- Users should not have to create an account to use the app.
- Android and iOS should have a similar/familiar functionality and feel to each other.

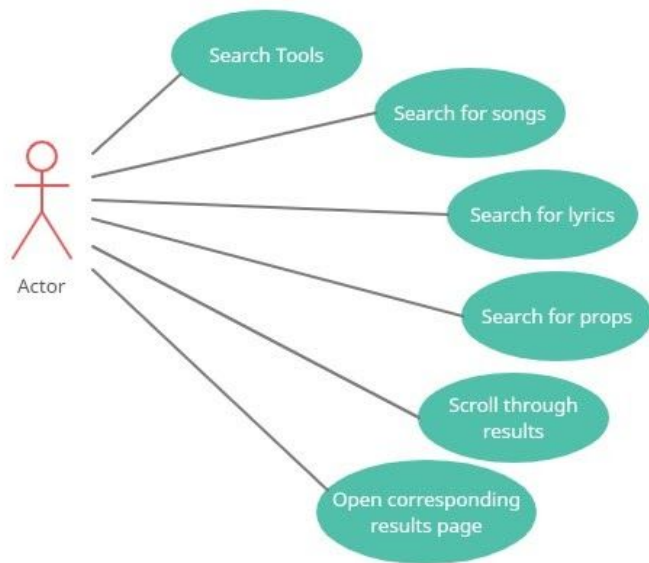
Other Specifications

Use case and sequence diagrams

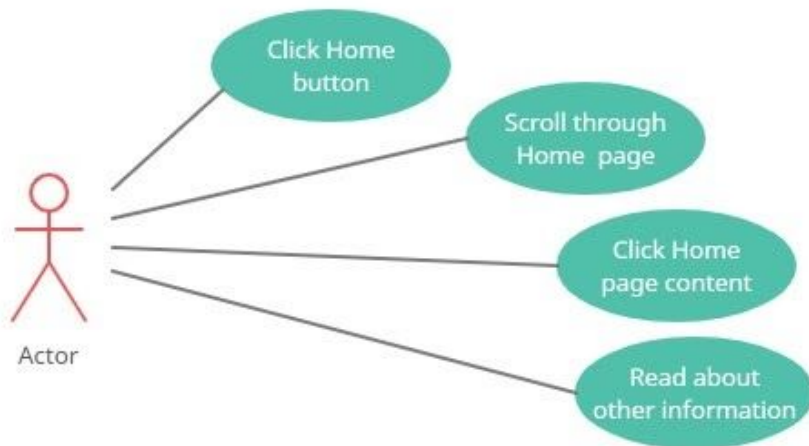
Use Cases: (User, System Feature)



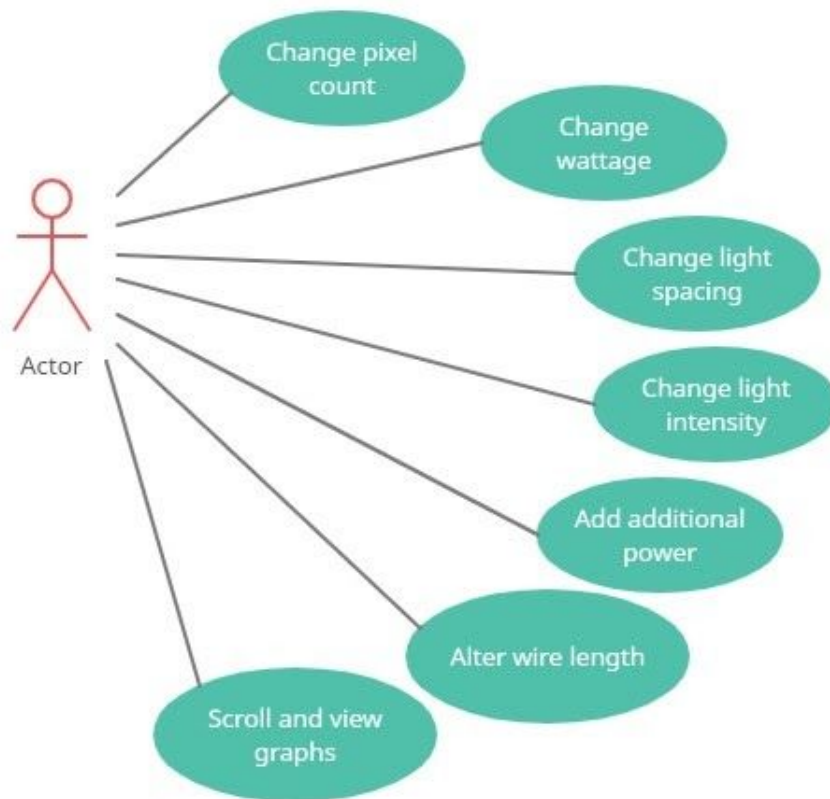
Use-case Diagram #1 - AR Props



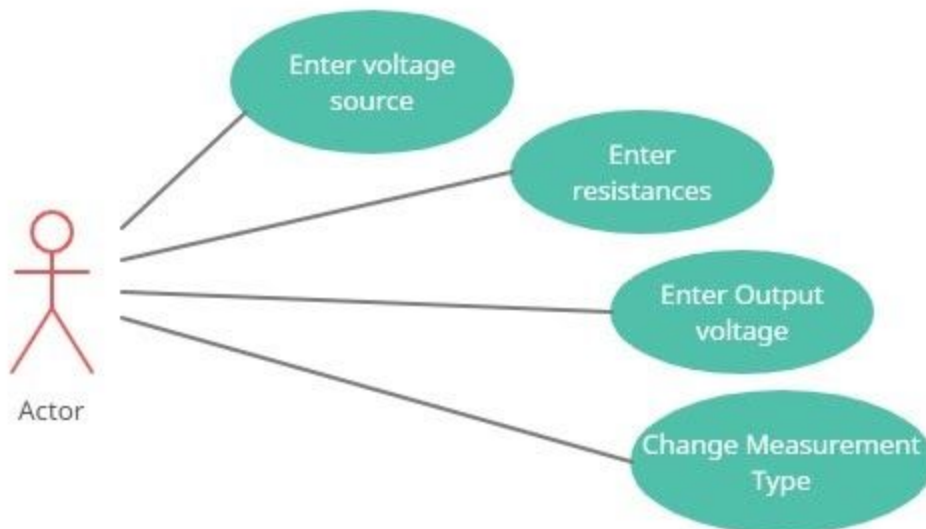
Use-case Diagram #2 - Search Engine



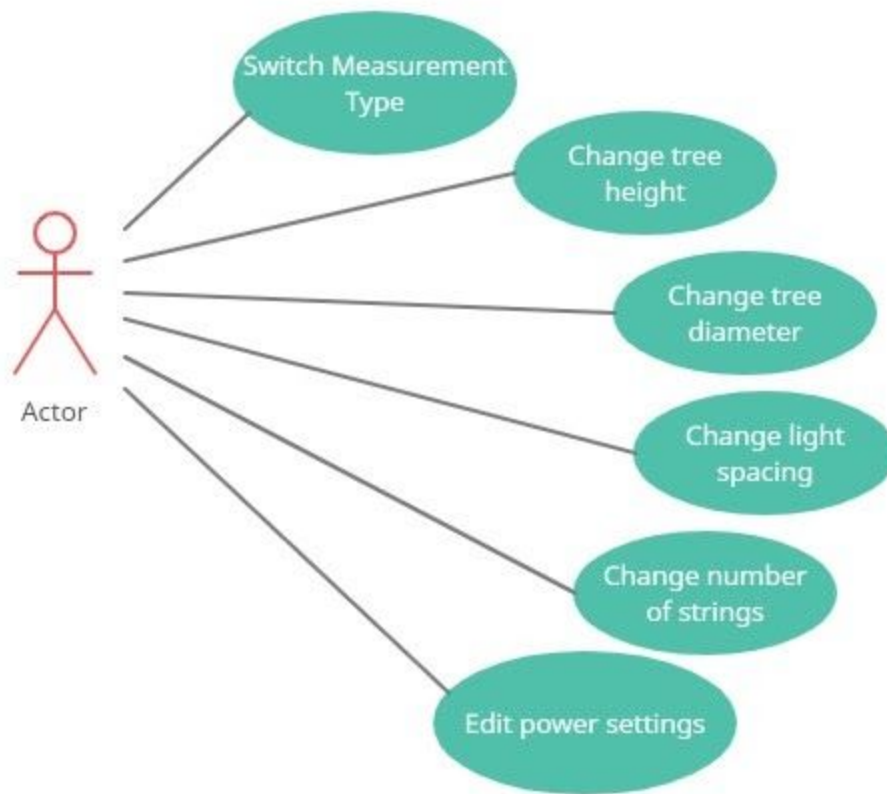
Use-case Diagram #3 - Home Page



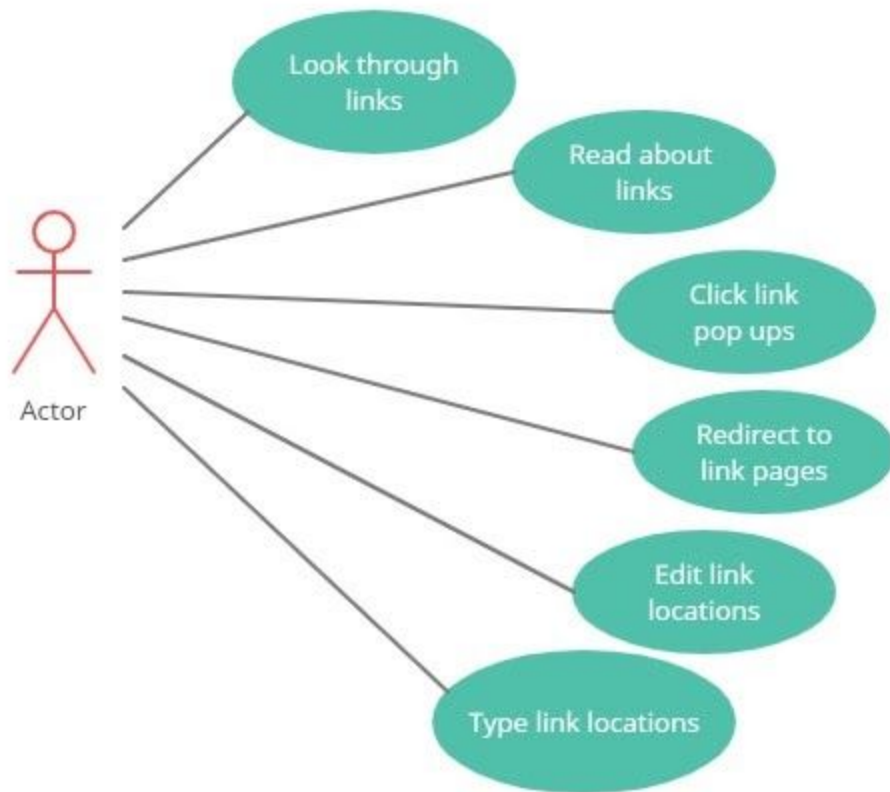
Use-case Diagram #4 - Electricity Calculator



Use-case Diagram #5 - Voltage Divider



Use-case Diagram #6 - Mega Tree Calculator



Use-case Diagram #7 - Links Page

Use case name	Take a screenshot.
Scenario	Take a screenshot of the current AR image on their device.
Triggering event	User taps the screenshot button or uses the built-in screenshot feature on Android devices. i.e. Hold the lock button + volume up/down.
Brief description	When the user takes a screenshot, the app will prompt them to save the image to their device so they can view it later.

Actors	User.
Related use cases	Save screenshot to phone.
Pre-conditions	The user should be on the AR screen.
Post-conditions	If the user takes a screenshot, allow them to save the screenshot to their device.
Flow of events	<p>User launches the app.</p> <p>User taps the AR menu.</p> <p>User goes outside and places props around their house using the built-in AR menu.</p> <p>User is satisfied with their placement of AR props.</p> <p>User taps the screenshot button.</p>
Exception conditions	Null.

Use case name	Save screenshot to device.
Scenario	Save screenshot of current screen to device after tapping the take screenshot button.
Triggering event	User taps the take screenshot button.
Brief description	When the user takes a screenshot, the current screen is saved as an image and they can decide if they would like to save it to their device or discard it.

Actors	User.
Related use cases	Take a screenshot.
Pre-conditions	<p>The user should be on the AR screen.</p> <p>The user should tap the take a screenshot button.</p>
Post-conditions	User can save or discard image.
Flow of events	<p>User launches the app.</p> <p>User taps the AR menu.</p> <p>User goes outside and places props around their house using the built-in AR menu.</p> <p>User is satisfied with their placement of AR props.</p> <p>User taps the screenshot button.</p>
Exception conditions	User discards image. Image will not be saved to device.

Use case name	Open prop menu.
Scenario	Opens the prop menu to allow placement of props in AR image.
Triggering event	User clicks the open prop menu button.

Brief description	When the user opens the AR menu, they will want to place props around their home to see what they look like. By opening up the AR menu, they will have a better understanding of what props they can place.
Actors	User.
Related use cases	Null.
Pre-conditions	The user should be on the AR screen. The user should tap the open prop menu button.
Post-conditions	The user can place the props around their home.
Flow of events	User launches the app. User taps the AR menu. User opens the prop menu by tapping the open prop menu button.
Exception conditions	Null.

Use case name	Place prop in scene.
Scenario	Places a prop from the prop menu onto the AR canvas around the user's house.
Triggering event	User taps on a prop and drags it onto the canvas.

Brief description	When the user wants to see what a prop looks like, they can open the prop menu and drag props onto the AR canvas. This will give a real-time view of what the prop would look like before being purchased.
Actors	User.
Related use cases	Delete prop from scene. Resize prop. Move prop.
Pre-conditions	The user should be on the AR screen. The user should have the prop menu open.
Post-conditions	User can delete prop from scene. User can move prop around scene. User can resize prop in scene.
Flow of events	User launches the app. User taps the AR menu. User opens the prop menu by tapping the open prop menu button. User places a prop on the AR canvas by tapping and dragging it into the scene.
Exception conditions	Null.

Use case name	App Search.
---------------	-------------

Scenario	User wishes to use the search bar to find a specific tool or item.
Triggering event	User clicks the search button.
Brief description	User can look up keywords and find related items or tools throughout the rest of the app.
Actors	App user.
Related use cases	Null.
Preconditions	User clicked search bar / button.
Post conditions	List of relevant items appears.
Flow of events	User clicks the search bar. User types in their search. User is redirected to list based on input.
Exception conditions	Null.

Use case name	Delete prop from scene.
Scenario	Deletes a prop from the current scene.
Triggering event	User taps the prop they wish to delete then they tap the delete button or drag it back into the prop selection menu.

Brief description	If a user does not wish to see a certain prop anymore, they can remove it, so it is no longer visible to them. They can click the delete button when the prop is selected or drag it into the menu to remove it.
Actors	User.
Related use cases	Place prop in scene.
Pre-conditions	The user should be in the AR screen. There must be a prop in the scene.
Post-conditions	Null.
Flow of events	User launches the app. User taps the AR menu. User opens the prop menu by tapping the open prop menu button. User places a prop on the AR canvas by tapping and dragging it into the scene. User deletes the prop by tapping it and then immediately tapping the delete prop button or dragging it back into the prop menu.
Exception conditions	Null.

Use case name	Resize prop.
Scenario	Resizes a prop in the current scene.

Triggering event	User taps the prop they wish to resize.
Brief description	Once a prop has been tapped, user can then pinch their fingers on the screen to increase the size or decrease the size of the prop.
Actors	User.
Related use cases	Move prop.
Pre-conditions	The user should be in the AR screen. There must be a prop in the scene.
Post-conditions	Null.
Flow of events	User launches the app. User taps the AR menu. User opens the prop menu by tapping the open prop menu button. User places a prop on the AR canvas by tapping and dragging it onto the scene. User resizes the prop by tapping it and then pinching their fingers on the screen.
Exception conditions	Null.

Use case name	Move prop.
Scenario	Moves a prop in the current scene.

Triggering event	User taps a prop they wish to move.
Brief description	Once a prop has been tapped, user can then drag the selected prop around the scene.
Actors	User.
Related use cases	Resize prop.
Pre-conditions	The user should be in the AR screen. There must be a prop in the scene.
Post-conditions	Null.
Flow of events	User launches the app. User taps the AR menu. User opens the prop menu by tapping the open prop menu button. User places a prop on the AR canvas by tapping and dragging it onto the scene. User selects the prop and can move it by dragging their finger across the screen.
Exception conditions	Null.

Use case name	Home Tab.
Scenario	View main tab of application.

Triggering event	Open app or click “Home” tab.
Brief description	This is the first thing a user sees upon starting the application. It can be considered the hub of the app where main content and other tabs can be accessed.
Actors	User.
Related use cases	AR Tab, Tools Tab, Search Tab.
Preconditions	Wi-Fi connection (for some but not all content).
Post conditions	Null.
Flow of events	<p>1.</p> <p>User opens the application, the home tab is automatically open.</p> <p>or</p> <p>User is on another tab and clicks the home tab. The view is then switched.</p> <p>2.</p> <p>One of the content items is clicked and the tab is switched, or a webpage is opened to access that content.</p>
Exception conditions	User is not connected to Wi-Fi so some content on the home tab may be inaccessible.

Use case name	AC/DC Calculator.
---------------	-------------------

Scenario	Calculate resistance, current, voltage, power, and impedance.
Triggering event	User opens the AC/DC calculator from the tools view.
Brief description	There are two different equations for calculating the variables of AC and DC electricity. The user can enter the variables they know to get the variables they are looking for.
Actors	User.
Related use cases	Voltage Divider, Power Consumption Calculator, Tree Calculator.
Preconditions	AC: 2 out of 4 variables [resistance, current, voltage, power] DC: 2 magnitudes out of 4 and 2 phase angles out of 4 for the 4 variables [impedance, current, voltage, power].
Post conditions	Missing variables are calculated.
Flow of events	User clicks the tools tab and finds the AC/DC calculator. Upon clicking the tool, the two sections for AC/DC are outlined with the corresponding variables. User enters the necessary number of variables for either equation and clicks calculate. The user saves/clears the values.
Exception conditions	Invalid values are entered, the user is notified and cannot calculate until appropriate changes are made.

Use case name	Voltage Divider.
Scenario	Calculate the voltage split with specified resistances.
Triggering event	User opens the voltage divider from the tools view.
Brief description	Voltage drops over different resistances and number of splits in the wire, so this calculator takes the input voltage and resistances to calculate the voltage to each line.
Actors	User.
Related use cases	AC/DC Calculator, Power Consumption Calculator, Tree Calculator.
Preconditions	Input voltage and resistance on each line.
Post conditions	Voltage split is calculated.
Flow of events	User clicks the tools tab and finds the voltage divider. Upon clicking the tool, the input variables are outlined. User enters necessary variables and clicks calculate. The user saves/clears the values.
Exception conditions	Invalid values are entered, the user is notified and cannot calculate until appropriate changes are made.

Use case name	Power Consumption Calculator.
---------------	-------------------------------

Scenario	Calculate the cost/consumption of power for a given wattage and time frame.
Triggering event	User opens the power consumption calculator from the tools.
Brief description	Using the wattage and time that the devices are on, the power consumption can be calculated and if the price of power is given, the cost to run the devices over a period can be calculated.
Actors	User.
Related use cases	AC/DC Calculator, Voltage Divider, Tree Calculator.
Preconditions	Wattage and time (optionally electricity price).
Post conditions	Consumption/cost is calculated.
Flow of events	User clicks the tools tab and finds the power consumption calculator. Upon clicking the tool, the variables needed are outlined. User enters necessary variables and clicks calculate. The user saves/clears the values.
Exception conditions	Invalid values are entered, the user is notified and cannot calculate until appropriate changes are made.

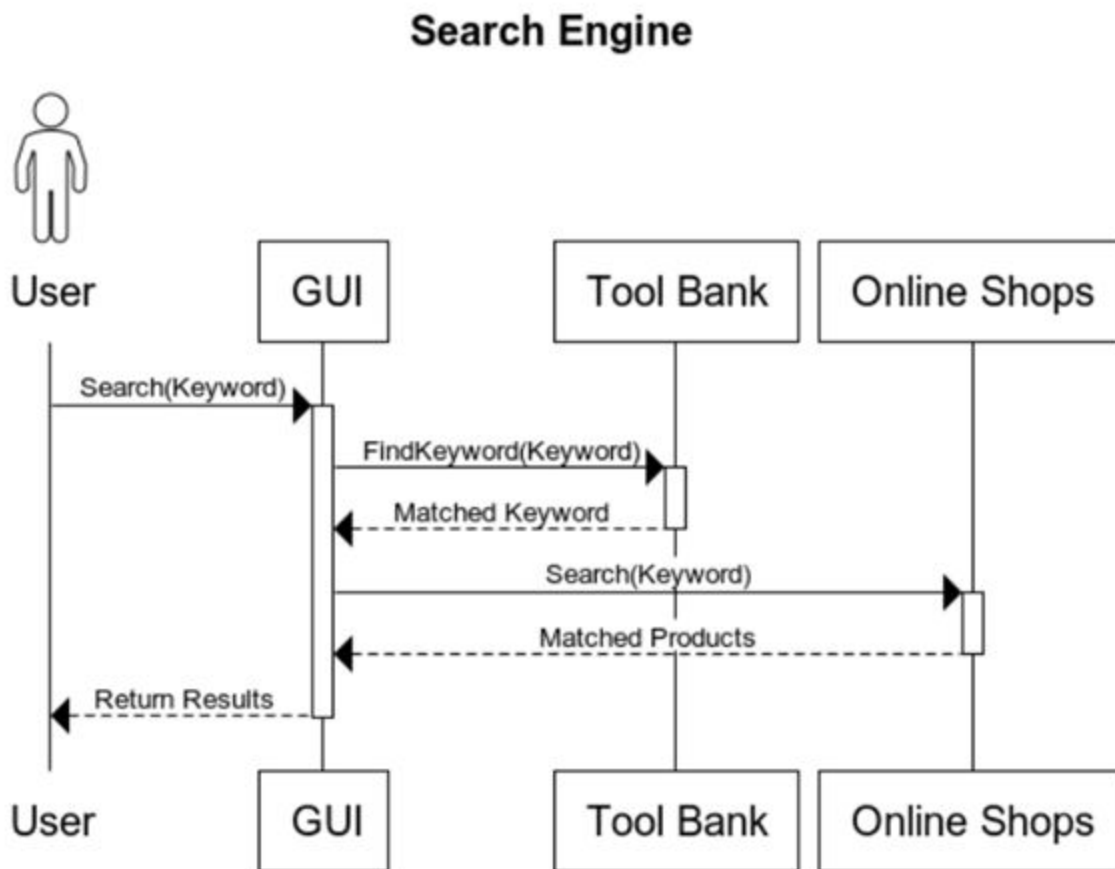
Use case name	Tree Calculator.
---------------	------------------

Scenario	Calculate the number of lights needed for specified dimensions of trees.
Triggering event	User opens tree calculator from tools.
Brief description	Using dimensions of tree and light spacing the number of lights and distance needed, for a certain tree can be calculated or vice versa.
Actors	User.
Related use cases	AC/DC Calculator, Voltage Divider, Power consumption calculator.
Preconditions	Tree height, bottom diameter, light spacing, rotations around tree.
Post conditions	Missing variables are calculated.
Flow of events	User clicks the tools tab and finds the tree calculator. Upon clicking the tool, the variables are outlined. User enters necessary variables and clicks calculate. The user saves/clears the values.
Exception conditions	Invalid values are entered, the user is notified and cannot calculate until appropriate changes are made.

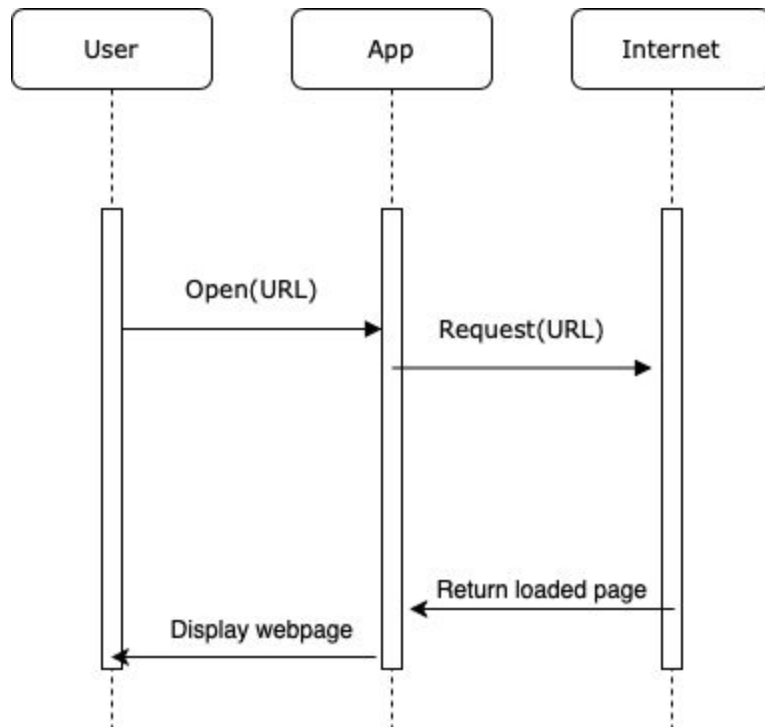
Use case name	Links.
Scenario	Links for different resources are provided.
Triggering event	User clicks one of the links provided.

Brief description	There are different resources available to the user on applications or web pages outside the application.
Actors	User.
Related use cases	Null.
Preconditions	Wi-Fi connection.
Post conditions	Webpage/application is opened.
Flow of events	User clicks one of the links and is either redirected to an external webpage or is brought to another application.
Exception conditions	User is not connected to Wi-Fi, so links do not redirect.

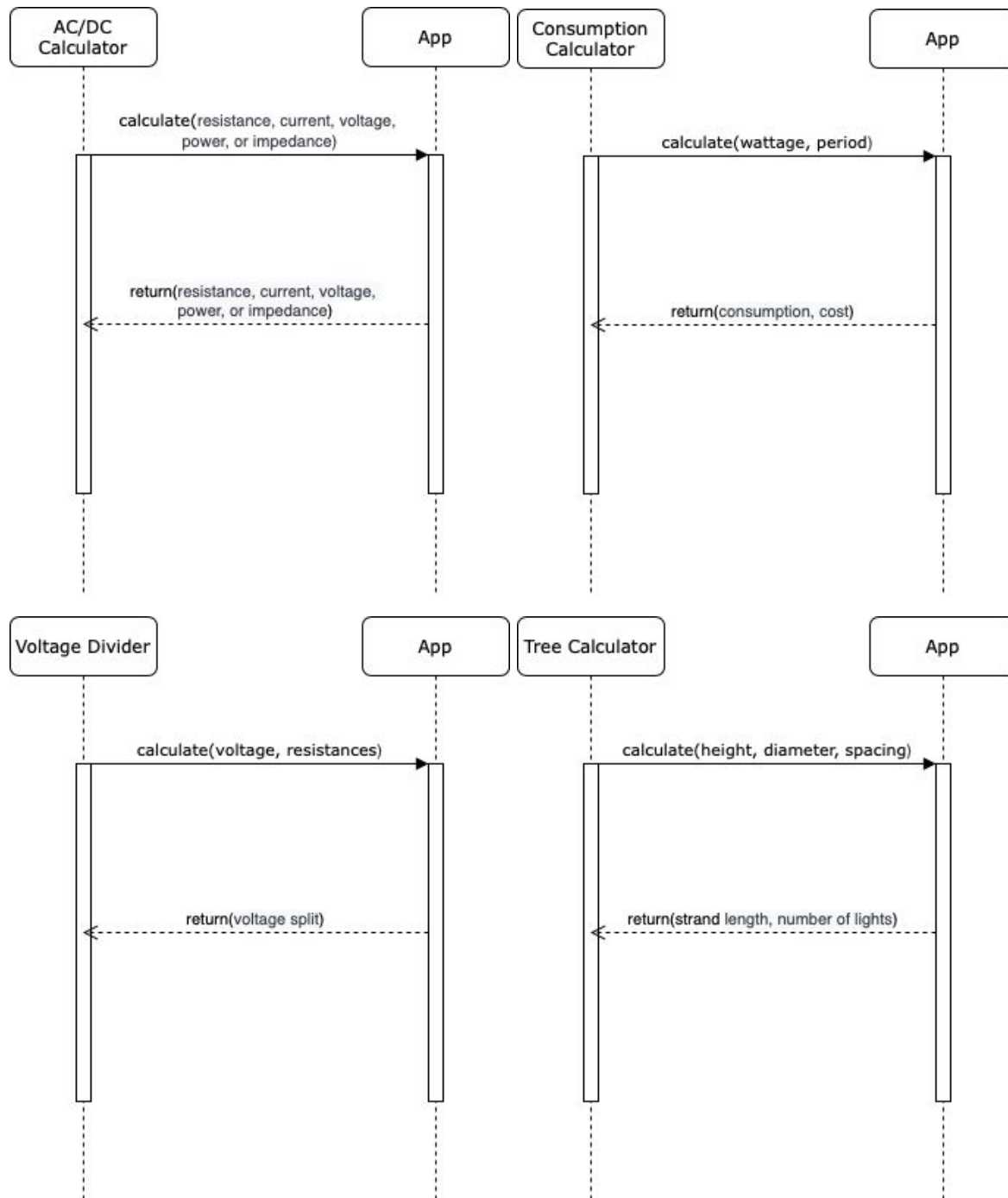
Sequence Diagrams:



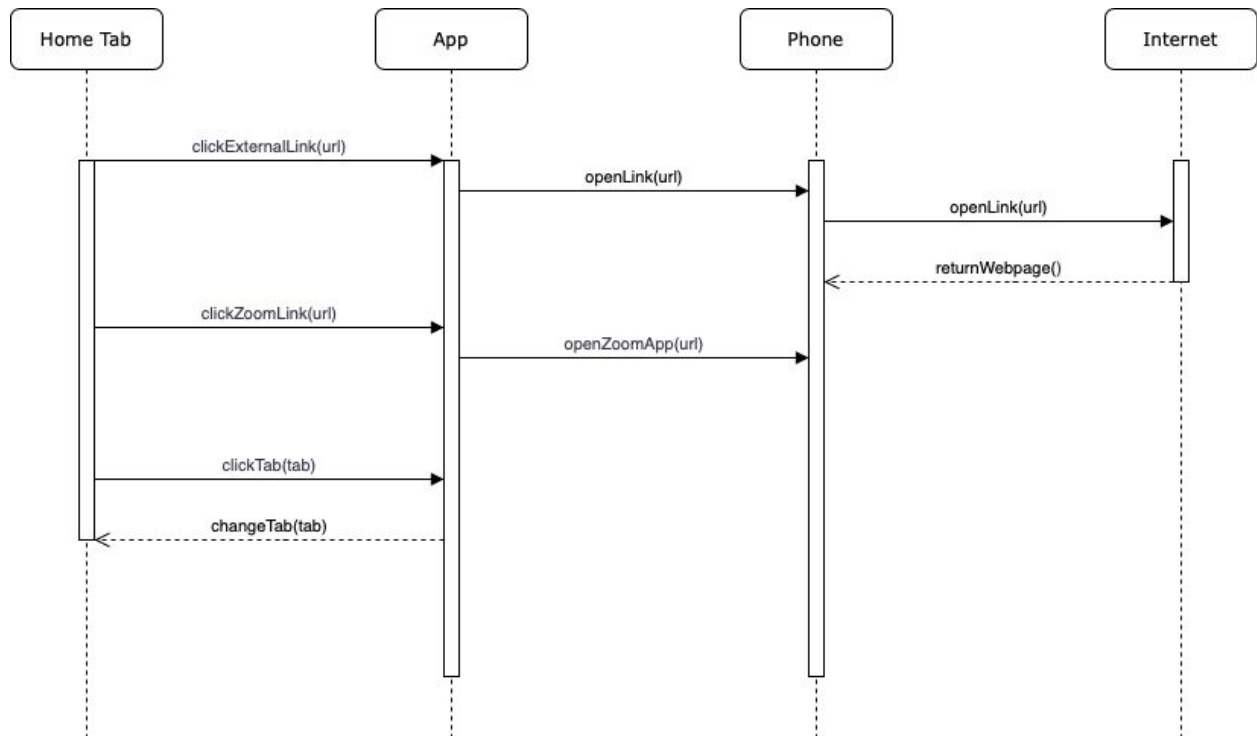
When a user wishes to make a search, They type the search into the GUI. The GUI first checks the tool word bank to see if the item that they searched is a part of one of the tools within the app itself. If the search contains a keyword within the bank it will return the associated tool to the user within the results. Next the search will be run across the online shops to see if they contain any products related to the users search. Again any related items will be returned. Finally all related items are displayed as a list to the user to pick form.



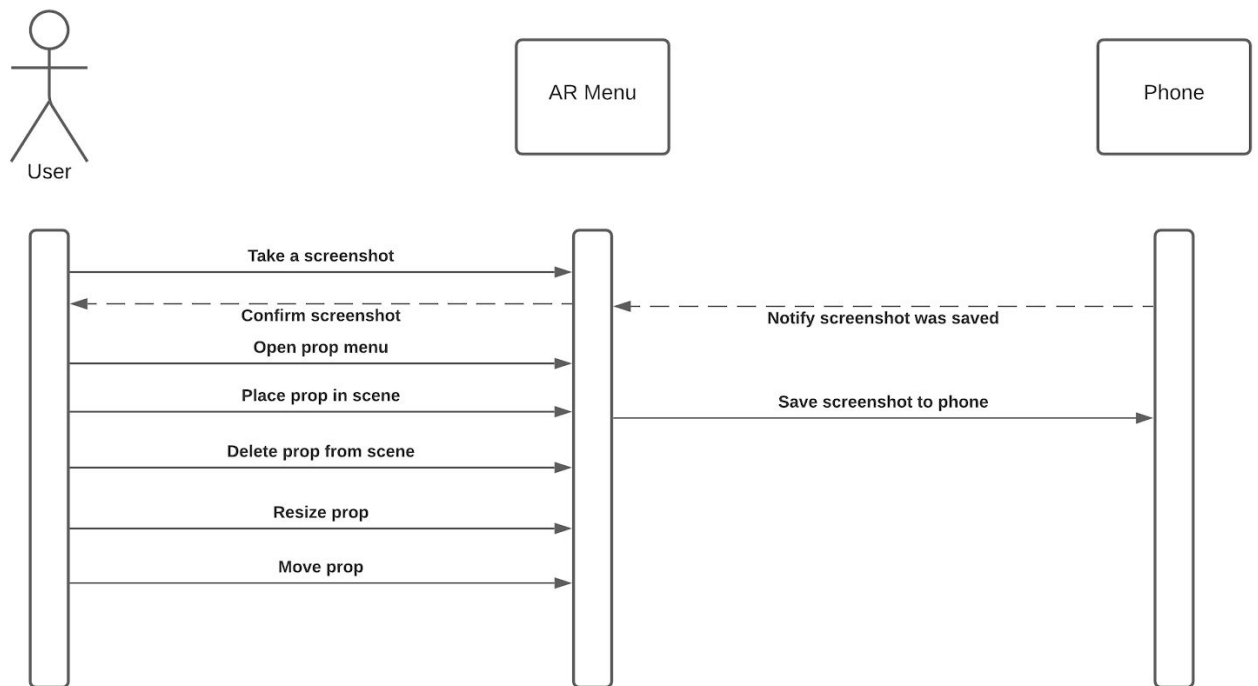
The diagram above represents the link page, which includes links to relevant xLights pages and communities on the internet. When a user clicks a link, the app will load the URL and display the page to the user.



All these calculators follow the same sequence. The user goes to the tool and enters some number of variables and gets a result from the calculator depending on the information they entered.

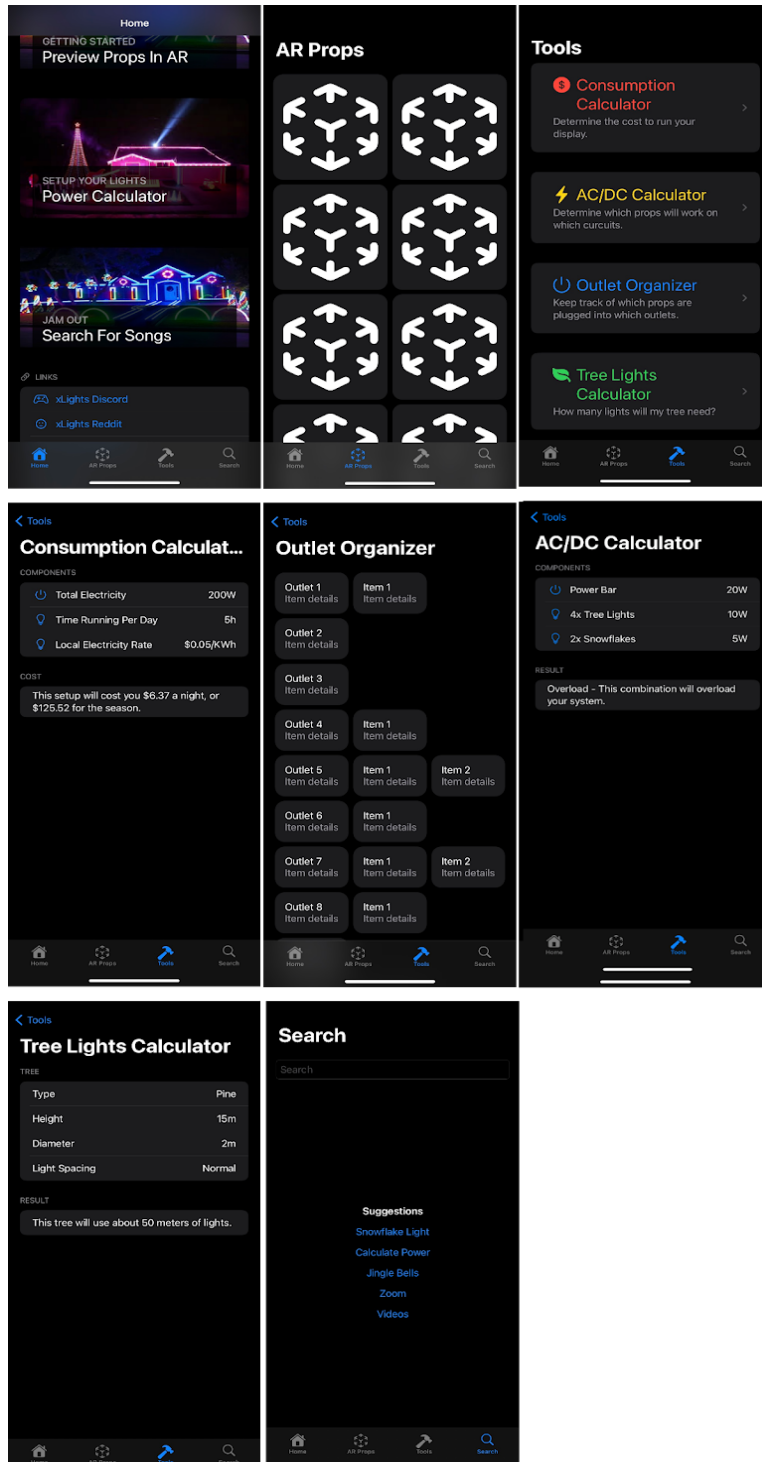


The home tab is the main tab of the application where most sequences start. The user is presented with this page and has a few options. There may be some quick links or they can change tabs to explore the features of the application. Some features reach external applications and web pages from this tab.



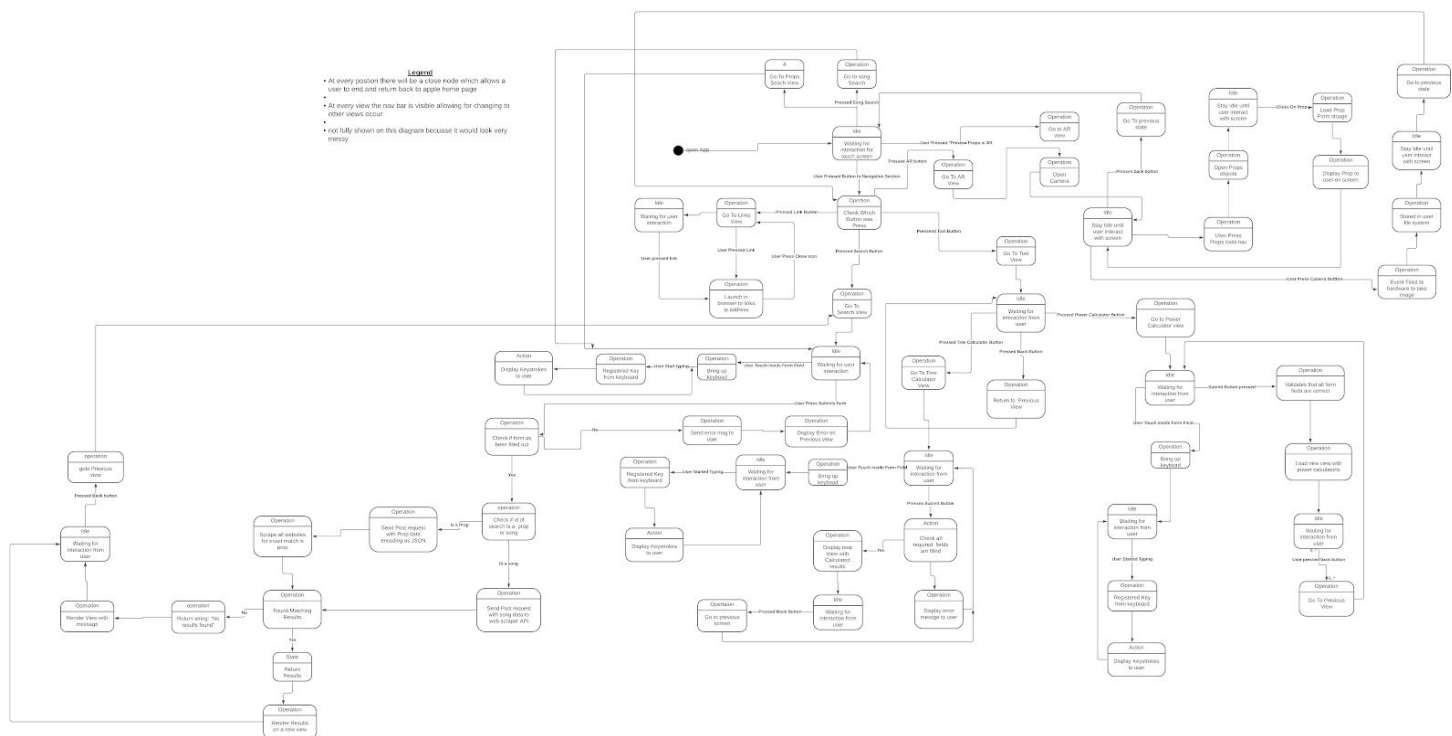
The AR prop page allows users to preview christmas props on their lawn. They are able to tap to add different props to the scene, remove them, edit them, and they can take a screenshot of props they are happy with and maybe want to buy.

UI Design



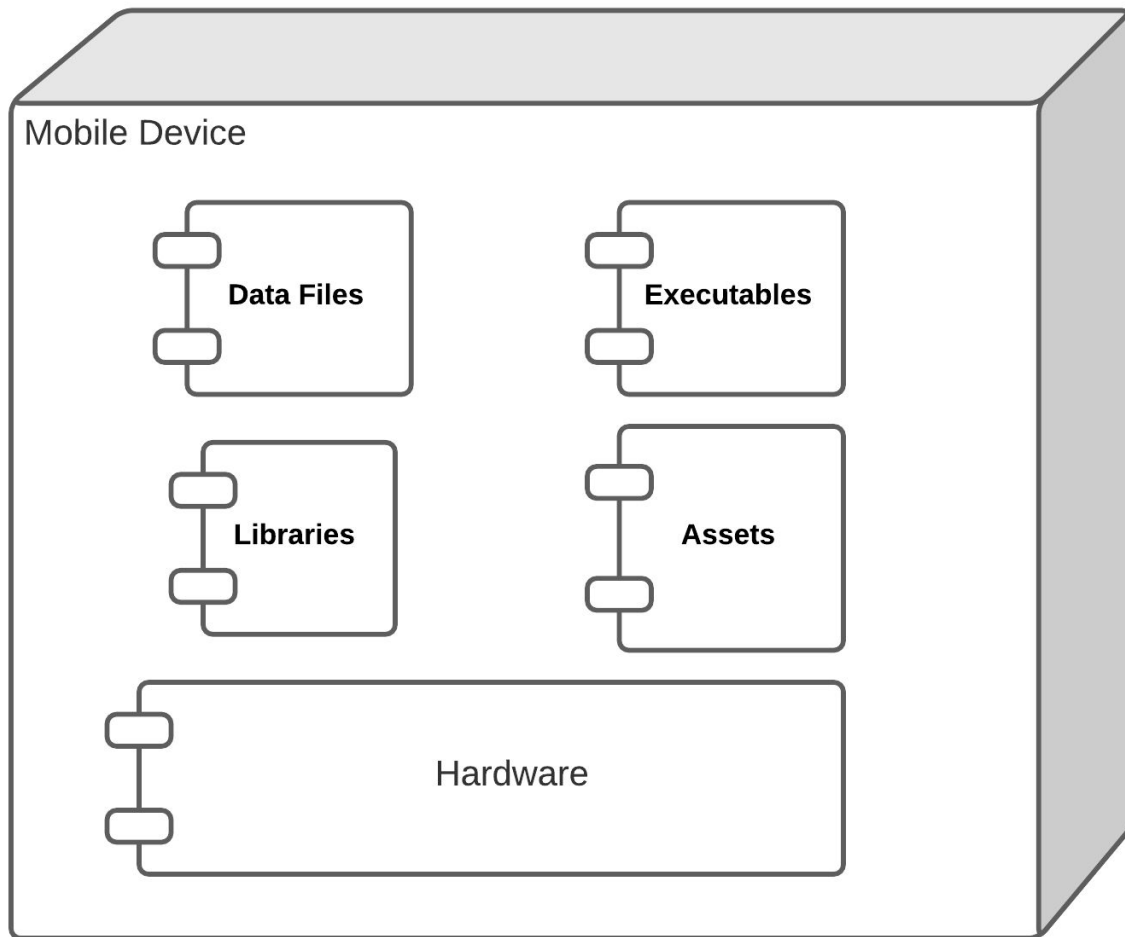
App

State Diagram



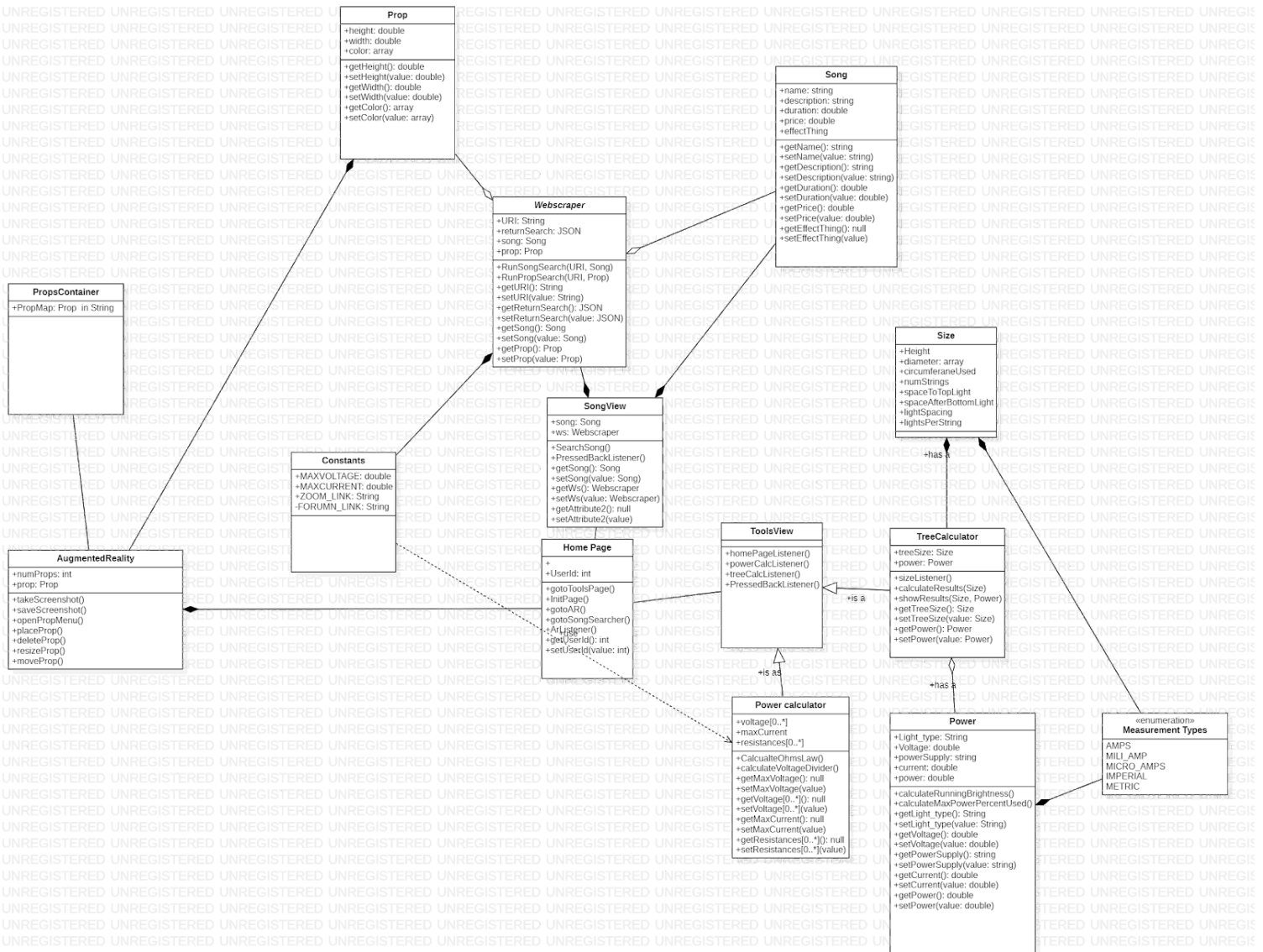
Attached is an image of the state diagram.

Deployment Diagram



For the deployment diagram above the application will use both xcode and Android studio for building the application. Android studio will add most of the executable files as well as the libraries in the application. The assets will be stored in a folder which will reside on the mobile device of the user. All files relating to the application will be stored locally on the clients mobile device. The main components of hardware that the application will interact with will be the Camera, filesystem and lastly the photo gallery.

Class Diagram



Attached as a png file is a blown up image of the class diagram.