## CHAPTER1: INTRODUCTION

* 1. **Background and Context**

Nowadays, the more technology is developed, the more comfortable the human life is. To increase the quality of life, home decoration, the interior is very important. But with the busy life as today, the choice of furniture suitable for the room, the house is not easy. To buy things that people like, people have to calculate, measure the area of the room, home and think this place should put something appropriately. And then, they have to go to the furniture store to pick out furniture, color. It also takes a lot of time if the selected furniture does not fit the house.

So, I apply an achievement of technology to build a comfortable and easy life for everyone. One of these things is a smartphone. I use the AR technology combine with the smartphone to help you to design and decorate your house. Here is a new way to resolve your headache.

For that reason, I decided to "***Building an application for Home Furnishing using Augmented Reality***".

* 1. **Purpose**

This application is not just a simple home decoration application. It helps people can consider what type of furniture which is suitable for their room, their house without going to the furniture store directly. The application will show many types of furniture with many different colors, dimensions.

The furniture is the virtual object such as tables, lamps, chairs, beds, pictures, painting, etc. but they have dimension is like the real furniture.

The people can use the camera of the phone to put the objects in the 3D space (we can see the virtual object in the real world) and it helps people can see the clear view: pictures, tables, chairs, ... how they look like in your room. If you don’t like them, you can remove them and choose another furniture and you can interact with them such as drag to change the position of them to make sure where is the good place to put them and also, you can rotate them to see how they look better.

This application can associate with third parties (furniture stores). After the people choose items, they can go to the store and buy them. It can help people save a lot of time. The 3D models are built the same with the real furniture.

After decorating people can take a picture or record the video and share to media social network to get ideas from the others for your room.

* 1. **Scope**

This project I and my partner (Dinh Huu Quan - class: 13T1, is instructed by Mr. Truong Ngoc Chau) did together. So, here is the list of features and assignment of each members:

|  |  |
| --- | --- |
| **Dinh Huu Quan – Class: 13T1** | **Nguyen Thi Tinh – Class: 13T3** |
| * Build 3D models (the furniture such as tables, beds, lamps, pictures...). * Create UI for the whole system. * Scan the plane such as the floor, the table, the wall, etc. And after that, create the visual plane by his own to be easier to see. * Control gestures when interacting with 3D models: use two fingers to rotate them. * Change the gravity of the 3D models when it put on the horizontal plane and the vertical plane (Ex. Pictures). * Create the local database for the whole system and insert data into the database. | * Build 3D models (the furniture such as tables, beds, lamps, pictures...). * Put 3D models after scan the plane and create an anchor to put 3D models to make sure it will be stable on the plane. * Control gestures when interacting with 3D models: double tap an object to remove, keep and drag an object to another position using one finger. And using two fingers to scale an object. * Take a picture/ a video and save to local database or share to social networks. * Make an audio when interacting with the 3D objects. |

* 1. **Theories and Technologies**

As a developer in the future, I want to improve my skill at coding by learning new technologies. As you all know, the technology changes every day, if we want to keep up, we need to never stop learning the new technology.

This application will bring the AR experience to the user, it means this application combines the real world with virtual objects. We considered some AR frameworks, such as Vuforia, Kudan, SimpleCV, ARKit, ArCore… After researching and comparing all of them, we decided to develop this application on iOS platform, using Swift language and ARKit framework.

To have the 3D models look like the real furniture and import them to our project, we had to build them by ourselves, but the time is limited so I have just built some 3D models, and another were downloaded from the Internet. We did use Blender software to do that.

This application requires an iOS device with an A9 or later processor and the iOS version of 11.3 or later.

## iPhone Operating System

## *Introduction*

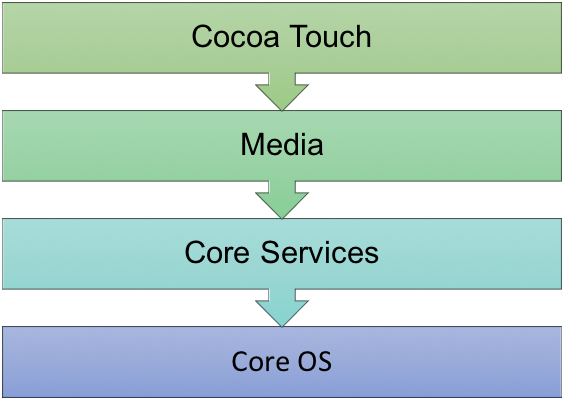
iOS (formerly iPhone OS) is a mobile operating system created and developed by Apple Inc. exclusively for its hardware. It is the operating system that presently powers many of the company's mobile devices, including the iPhone, iPad, and iPod Touch. It is the second most popular mobile operating system globally after Android.

The iOS user interface is based upon direct manipulation, using multi-touch gestures. Interface control elements consist of sliders, switches, and buttons. Interaction with the OS includes gestures such as swipe, tap, pinch, and reverse pinch, all of which have specific definitions within the context of the iOS operating system and its multi-touch interface. Internal accelerometers are used by some applications to respond to shaking the device (one common result is the undo command) or rotating it in three dimensions (one common result is switching between portrait and landscape mode). Apple has been significantly praised for incorporating thorough accessibility functions into iOS, enabling users with vision and hearing disabilities to properly use its products.

Major versions of iOS are released annually. The current version, iOS 11, was released on September 19, 2017. It is available for all iOS devices with 64-bit processors; the iPhone 5S and later iPhone models, the iPad (2017), the iPad Air and later iPad Air models, all iPad Pro models, the iPad Mini 2 and later iPad Mini models, and the sixth-generation iPod Touch.

### iOS layered architecture

At the highest level, iOS acts as an intermediary between the underlying hardware and the apps you create. Apps do not talk to the underlying hardware directly. Instead, they communicate with the hardware through a set of well-defined system interfaces. These interfaces make it easy to write apps that work consistently on devices having different hardware capabilities.



*Figure 1.1: Layered architecture*

The implementation of iOS technologies can be viewed as a set of layers. Lower layers contain fundamental services and technologies. Higher-level layers build upon the lower layers and provide more sophisticated services and technologies.

Apple delivers most of its system interfaces in special packages called *frameworks*. A *framework* is a directory that contains a dynamic shared library and the resources (such as header files, images, and helper apps) needed to support that library. To use frameworks, you add them to your app project from Xcode.

1.4.1.3 Features

The home screen, rendered by SpringBoard, displays application icons and a dock at the bottom where users can pin their most frequently used apps. The home screen appears whenever the user unlocks the device or presses the physical "Home" button whilst in another app. Before iOS 4 on the iPhone 3GS (or later), the screen's background could be customized only through jailbreaking, but can now be changed out-of-the-box. The screen has a status bar across the top to display data, such as time, battery level, and signal strength. The rest of the screen is devoted to the current application. When a passcode is set and a user switches on the device, the passcode must be entered at the Lock Screen before access to the Home screen is granted.

iOS offers various accessibility features to help users with vision and hearing disabilities. One major feature, VoiceOver, provides a voice reading information on the screen, including contextual buttons, icons, links and other user interface elements, and allows the user to navigate the operating system through gestures. Any apps with default controls and developed with a UIKit framework gets VoiceOver functionality built in. One example includes holding up the iPhone to take a photo, with VoiceOver describing the photo scenery.

Multitasking for iOS was first released in June 2010 along with the release of iOS 4. Only certain devices—iPhone 4, iPhone 3GS, and iPod Touch 3rd generation—were able to multitask. The iPad did not get multitasking until iOS 4.2.1 in November. Currently, multitasking is supported on iPhone 3GS+, iPod Touch 3rd generation+, and all iPad models. Implementation of multitasking in iOS has been criticized for its approach, which limits the work that applications in the background can perform to a limited function set and requires application developers to add explicit support for it.

* + 1. **Swift programming language**
       1. ***Introduction***

Swift is a general-purpose programming language built using a modern approach to safety, performance, and software design patterns. Swift is designed to work with Apple's Cocoa and Cocoa Touch frameworks and the large body of existing Objective-C (ObjC) code written for Apple products.

The goal of the Swift project is to create the best available language for uses ranging from systems programming, to mobile and desktop apps, scaling up to cloud services. Most importantly, Swift is designed to make writing and maintaining correct programs easier for the developer. To achieve this goal, we believe that the most obvious way to write Swift code must also be: *safe*, *fast*, and *expressive*.

On December 3, 2015, the Swift language, supporting libraries, debugger, and package manager were published under *the Apache 2.0 license* with a Runtime Library Exception. Swift is now free to be ported across a wide range of platforms, devices, and use cases.

* + - 1. ***Architectural overview***

The features of Swift are designed to work together to create a language that is powerful, yet fun to use. Some additional features of Swift include:

* + - * Closures unified with function pointers.
      * Tuples and multiple return values.
      * Generics.
      * Fast and concise iteration over a range or collection.
      * Structs that support methods, extensions, and protocols.
      * Functional programming patterns, e.g., map and filter.
      * Powerful error handling built-in.
      * Advanced control flow with *do*, *guard*, *defer*, and *repeat* keywords.

The Swift language is managed as a collection of projects, each with its own repositories. The current list of projects includes:

* + - * The Swift compiler command line tool.
      * The standard library bundled as part of the language.
      * Core libraries that provide higher-level functionality.
      * The LLDB debugger which includes the Swift REPL.
      * The Swift package manager for distributing and building Swift source code.

Xcode playground support to enable playgrounds in Xcode.

1.4.2.3 Features

* Syntactic sugar

Under the Cocoa and Cocoa Touch environments, many common classes were part of the Foundation Kit library. This included the NSString string library (using Unicode), the NSArray and NSDictionary collection classes, and others. Objective-C provided various bits of syntactic sugar to allow some of these objects to be created on-the-fly within the language, but once created, the objects were manipulated with object calls.

* Access control

Swift supports five access control levels for symbols: open, public, internal, fileprivate, and private. Unlike many object-oriented languages, these access controls ignore inheritance hierarchies: private indicates that a symbol is accessible only in the immediate scope, fileprivate indicates it is accessible only from within the file, internal indicates it is accessible within the containing module, public indicates it is accessible from any module, and open (only for classes and their methods) indicates that the class may be subclassed outside of the module.

* Optionals and chaining

Swift supports five access control levels for symbols: open, public, internal, fileprivate, and private. Unlike many object-oriented languages, these access controls ignore inheritance hierarchies: private indicates that a symbol is accessible only in the immediate scope, fileprivate indicates it is accessible only from within the file, internal indicates it is accessible within the containing module, public indicates it is accessible from any module, and open (only for classes and their methods) indicates that the class may be subclassed outside of the module.

* Value types

In many object-oriented languages, objects are represented internally in two parts. The object is stored as a block of data placed on the heap, while the name (or "handle") to that object is represented by a pointer. Objects are passed between methods by copying the value of the pointer, allowing the same underlying data on the heap to be accessed by anyone with a copy. In contrast, basic types like integers and floating point values are represented directly; the handle contains the data, not a pointer to it, and that data is passed directly to methods by copying. These styles of access are termed pass-by-reference in the case of objects, and pass-by-value for basic types.

* Protocol-oriented programming

A key feature of ObjC is its support for categories, methods that can be added to extend classes at runtime. Categories allow extending classes in-place to add new functions with no need to subclass or even have access to the original source code. An example might be to add spell checker support to the base NSString class, which means all instances of NSString in the application gain spell checking. The system is also widely used as an organizational technique, allowing related code to be gathered into library-like extensions. Swift continues to support this concept, although they are now termed extensions, and declared with the keyword extension.

* + 1. **Augmented Reality**

Augmented reality, or AR for short, is a human computer interface paradigm within the general computer graphics discipline. AR aims at moving digital information into the physical world, thereby blurring the border between the physical and the virtual in a way that appears natural to the user. It enables a more intuitive, yet complex interface between man and machine. There is, however, no one widely accepted definition of what AR really is. AR was started by Sutherland with his seminal work on head mounted displays (Sutherland 1968). That work presented the first AR system (see Figure 1). But it was not until the 1990s that an attempt at clearly defining AR was given. Caudell and Mizell coined the term “augmented reality” in 1992 (Caudell and Mizell 1992) and two years later, a first attempt at defining AR was made. There exist two widely known definitions of AR today. The first, by Milgram and Kishino (see Figure 2), defines AR within the “Reality–Virtuality” continuum (Milgram and Kishino 1994). The second, by Azuma, gives more detailed criteria on what the prerequisites for AR are (Azuma 1997). Azuma’s original definition requires:

* + A combination of physical and virtual data
  + Registration in the physical world in 3D
  + Computed interactively/on-the-fly.

Combining physical and virtual information is what most people understand for AR. Adding the constraint of three-dimensional registration intentionally rules out all applications that merely display information over a video feed with a disregard of the underlying data, such as the news ticker on news shows. Requesting that the system should be computed on-the-fly differentiates AR from offline computer augmented movies.

To do AR project, nowadays Apple and Google provide for us two libraries to support us do that: ARKit ( from Apples and the projects will be run on the iOS device with iOS 11.3 and A9 processor or later) and ARCore ( from Google, the projects will be run on the devices which are produced by Google such as Google Pixel and Samsung Galaxy S7 or later).

* + 1. **ARKit Framework**

****

* **ARKit concept**

ARKit (Apple ARKit) is Apple’s augmented reality ([AR](https://whatis.techtarget.com/definition/augmented-reality-AR)) development platform for [iOS](https://searchmobilecomputing.techtarget.com/definition/iOS) [mobile devices](https://whatis.techtarget.com/definition/mobile-device).

ARKit allows developers to build high-detail AR experiences for [iPad](https://searchmobilecomputing.techtarget.com/definition/iPad) and [iPhone](https://searchmobilecomputing.techtarget.com/definition/iPhone). Environments captured through the device can have animated [3D](https://whatis.techtarget.com/definition/3-D-three-dimensions-or-three-dimensional) virtual text, objects and characters added to them. AR scenes made by one individual are persistent and can be seen by others visiting the location later.

ARKit was introduced along with iOS 11. As ARKit is specified to run on Core A9 and higher iOS devices, the AR experiences can have more detailed content and maintain better environmental awareness. With iPhone X, ARKit can perform real-time face scanning and use this data to drive facial expressions of 3D characters.

Using the iOS device’s camera, [accelerometers](https://whatis.techtarget.com/definition/accelerometer), [gyroscope](https://whatis.techtarget.com/definition/gyroscope) and [context awareness](https://whatis.techtarget.com/definition/context-awareness), ARKit performs environment mapping as the device is moved. [Sensor fusion](https://whatis.techtarget.com/definition/sensor-fusion) of the inertial sensor data with the data from the camera allows for highly accurate location awareness and mapping. The software picks out visual features in the environment such as planes and tracks motion in conjunction with information from the inertial sensors. The camera is also used to determine light sources by which AR objects are lit. Apple’s solution to the increased detail and therefore memory usage is a sliding map where old data disappears for new. Users can place anchors to mark creations they want to save.

* **ARKit 1.5**

The latest update of ARKit in iOS 11.3 delivers new features that let you create an even more realistic user experience. With improved scene understanding, your app can see and place virtual objects on vertical surfaces, and more accurately map irregularly shaped surfaces. Real world images, such as signs, posters, and artwork can be integrated into the AR experience, so your app can fill a museum with interactive content or bring a movie poster to life. And now, the pass-through camera view of the real world is higher resolution and supports auto-focus for a sharper view in more situations.

* **Hardware and Software Integration**
* TrueDepth Camera: iPhone X and ARKit enable a revolutionary capability for robust face tracking in augmented reality apps. Using the TrueDepth camera, your app can detect the position, topology, and expression of the user’s face, all with high accuracy and in real time, making it easy to apply live selfie effects or use facial expressions to drive a 3D character.
* Scene Understanding and Lighting Estimation: With ARKit, iPhone and iPad can analyze the scene presented by the camera view and find horizontal and vertical planes in the room and can track and place objects on smaller feature points as well. ARKit also makes use of the camera sensor to estimate the total amount of light available in a scene and applies the correct amount of lighting to virtual objects.
* Visual Inertial Odometry: ARKit uses Visual Inertial Odometry (VIO) to accurately track the world around it. VIO fuses camera sensor data with Core Motion data. These two inputs allow the device to sense how it moves within a room with a high degree of accuracy, and without any additional calibration.
* High Performance Hardware and Rendering Optimizations: ARKit runs on the Apple A9, A10, and A11 processors. These processors deliver breakthrough performance that enables fast scene understanding and lets you build detailed and compelling virtual content on top of real-world scenes. You can take advantage of the optimizations for ARKit in Metal, SceneKit, and third-party tools like Unity and Unreal Engine.
  + 1. **Realm Database**



**Realm** is an open-source object database management system, initially for mobile (Android/iOS), also available for platforms such as Xamarin or React Native, and others, including desktop applications (Windows), and is licensed under the Apache license.

In 2016 September, the **Realm Mobile Platform** was announced, followed by the first stable release in January 2017. It allows two-way synchronization between the Realm Object Server, and the client side databases that belong to the given logged-in user. Both a developer, and a commercial edition was released, along with a business license for integrating with other database management systems such as PostgreSQL.

The most notable features of Realm are the following:

* As Realm is an object store, its typed language-specific APIs map typed objects directly into the Realm file - therefore classes are used as the schema definition.
* Relationships between objects are allowed via "links". Each "link" creates a "backlink" as an inverse relationship to whichever objects are linking to the current object.
* The query results returned by Realm are thread-local views to the current "database version" (as Realm handles concurrency with [MVCC architecture](https://en.wikipedia.org/wiki/Multiversion_concurrency_control)), and these views "automatically update" when a transaction is committed from *any thread*, as long as Realm is able to update its instance version (which is possible on threads that are able to receive change notifications). When this happens, Realm calls change listeners that are added to its query results (if they've changed).
* Each thread-local view returns proxy objects that only read from/write to the database when an accessor method is called, meaning all database access is lazy-loaded. Please note that writes are allowed only while in a write transaction.
* As each query result and each proxy object is a view to the underlying data, any change made to the database is reflected in all objects that point to the same data. Realm generally calls this behavior "zero-copy architecture" (along with the previously mentioned lazy-loaded data access).

Realm Database is an alternative to SQLite and Core Data. Thanks to its zero-copy design, Realm Database is much faster than an ORM, and often faster than raw SQLite.

Realm Database

F:\Users\QuanDinh\Desktop\Capture.PNG

* + 1. **Blender**



Blender is a professional, free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games. Blender's features include 3D modeling, UV unwrapping, texturing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animating, match moving, camera tracking, rendering, motion graphics, video editing and compositing. It also features an integrated game engine.

## CHAPTER 2: ANALYSIS AND DESIGN

* 1. **Analysis**

Follow the project’s name, the application must interact with the real world through using the camera. Then, we can put 3D models on the screen and interact with them. It also needs some miscellaneous features to make the app work smoother and more accurate.

This application had these features:

**2.1.1. Interact with 3D models**

This feature is the main feature of this application. This feature will include three sub-features:

+ Put 3D models on the screen: When the user wants to decorate their room. They just open the application choose Decoration Screen button, give this application the permission to access the Camera and the Microphone and then detect the plane. After having the plane[[1]](#endnote-1), the user can choose any items on the list and the items will be put at the center of the camera. So, they can see that furniture in the 3D space through the camera of the phone.

+ Control gesture when interacting with the 3D models: When the user wants to change the place of that item, they just need to use their finger to keep and drag it to another position. They also can rotate that item. If the 3D model is too small or too big, they can scale that item.

+ Remove 3D models: When the user doesn't want to use that item, they can double tap it and the application will show Delete button, the system will show the confirm message, so they can click Yes button to delete that item out of the screen so that they can choose another item.

**2.1.2. Take a picture and share**

After user decorates the room, the user can take a picture of the screen (this picture will show the virtual models in the real world) and they automatically save in gallery database. After that, we can click the Share button to save that picture into the phone or share the picture/video to media social network such as Google plus, Facebook, … or personal message, email, ….

**2.1.3. Manage Settings mode**

This feature is like utilities for the user.

The user can click the Options button on the screen. It will show the list of option, such as:

* Reset the place: This feature means the system will remove all the objects on the screen and the user can start to create a new one.
* Use Audio: If the user chooses this feature so when the user interacts with the objects on the screen, the system will have the sound.
* Record video: If taking a picture is not enough, the user can click to record the video after decorating, and the also can save or share to media network. This function is the same take a picture, it is automatically saved in gallery database after stopping record a video. They can also click to watch the video which has just recorded.

**2.1.4. Manage Gallery**

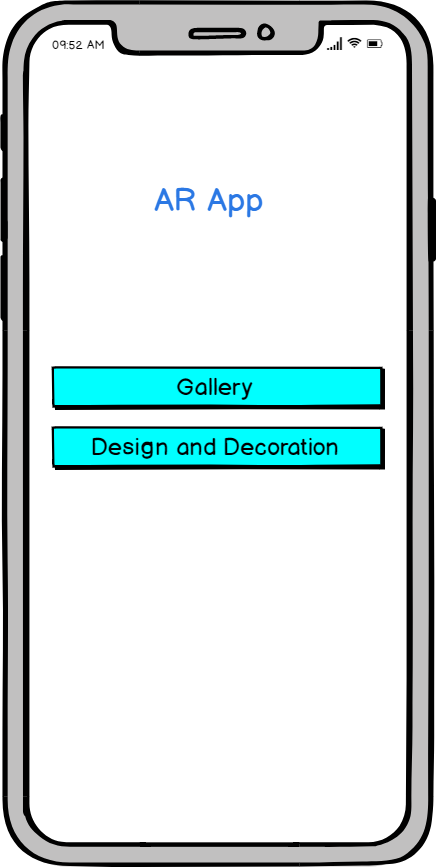
After the user can take a picture or record a video, they will be saved in the gallery database. In the Home Screen, the user clicks Gallery button, it will show all of the images, videos of which are taken, recorded by the user before.

The user can click to see the detail of them or delete them.

## 2.2. Wire framing

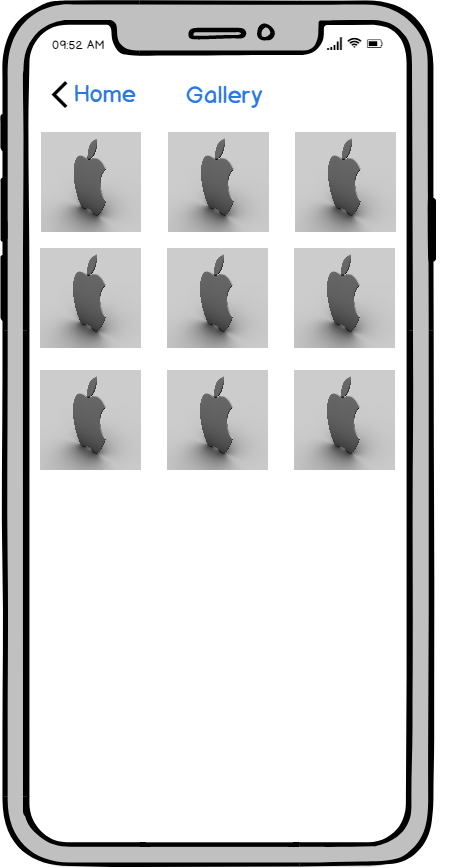
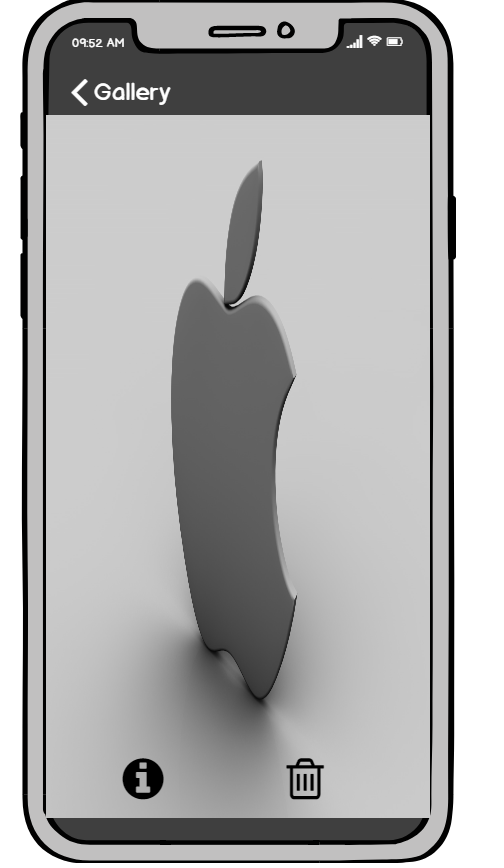
**2.2.1. The Home Screen**

This screen is the first screen will be shown when the user opened the application. It includes two buttons: Gallery button and Design and Decoration button.



**2.2.2. The Gallery Screen**

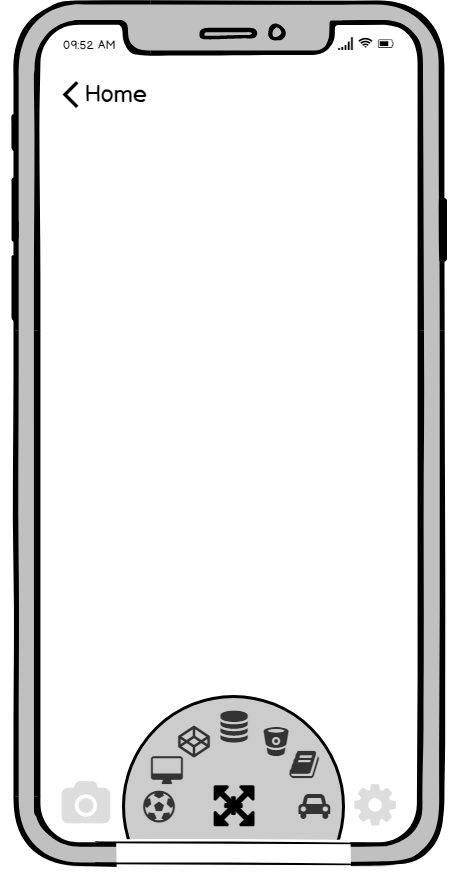
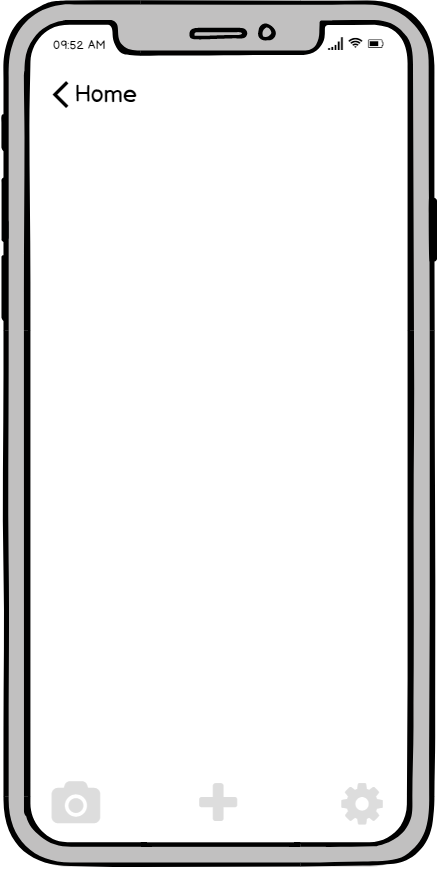
When the user clicks Gallery button, it will open the Manage Gallery Screen. And when the user clicks one of that pictures/videos, it will be shown detail and the user can play the video and user can choose Delete button to delete them.

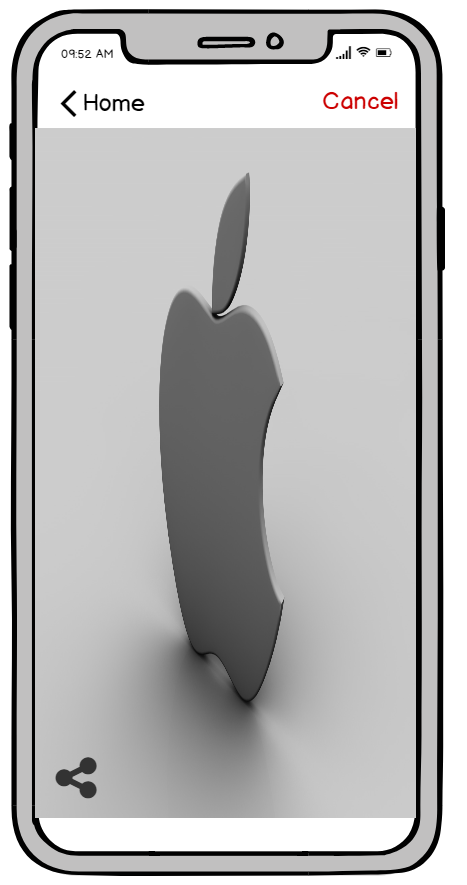
****

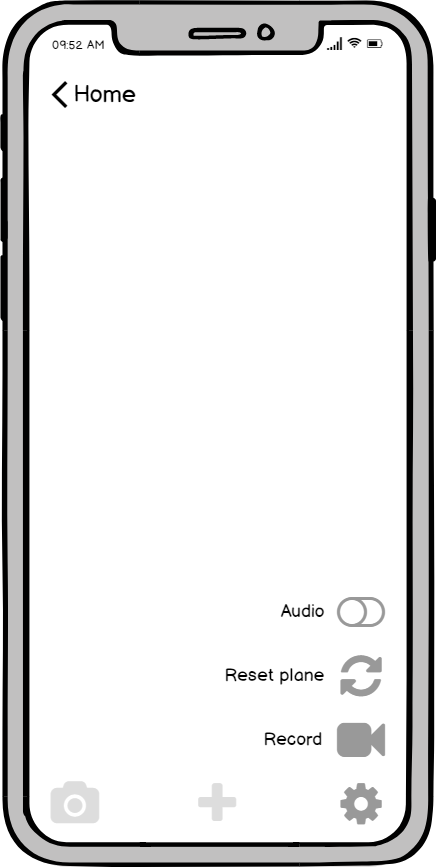
**2.2.3 The Design and Decoration Screen**

When the user clicks Design and Decoration button at the Home screen, it will open the camera and inform to the user to detect the plane.

* After detecting the plane. the user can click Add (**+**) button to choose the item**.**

****

* When the user clicks Camera (F:\Users\QuanDinh\Desktop\camera.png) button, it will take a picture and show Share button and Cancel button. Users can click to share or cancel that action.
* When the user click Settings ( ) button, it will show the list of settings mode such as reset place, use audio, record video.



**2.3. Use-case diagram**

**2.3.1. Overall diagram**

The application includes these features:

- Detect Plane.

- Choose 3D models.

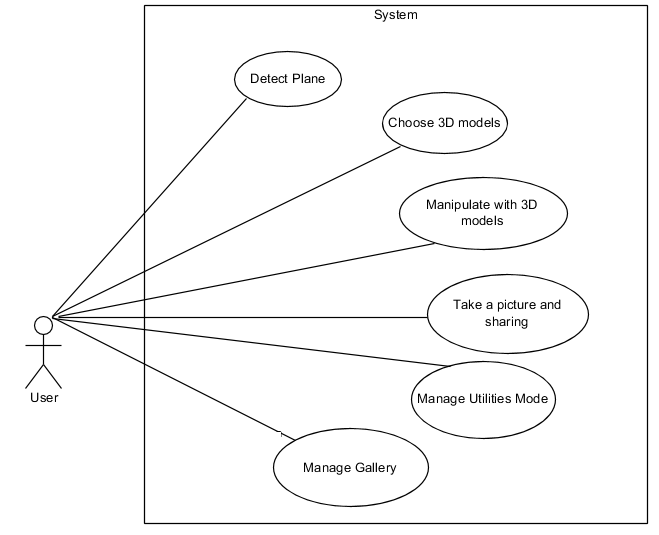
- Manipulate with 3D models on the screen.

- Take a picture and share to social network.

- Manage Utilities mode for the user.

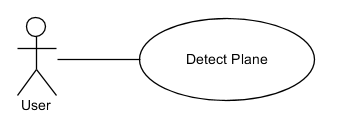
- Manage Gallery.

With these features above (the details of the features will be explained below), the application has an actor (the users) interact with the system. Here is overall use-case.



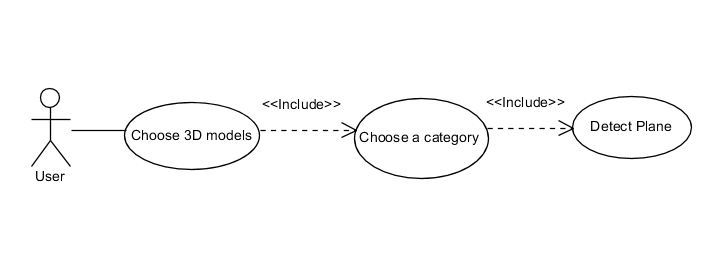
2.3.2. Detect Plane

When the user opens the application and chooses Start Decoration button at Home Screen. The system will show the text "Detect Plane", the user must let the phone scan around the floor, wall to find the surfaces (it's called Plane). When they have detected Plane, they can put 3D models on the screen.



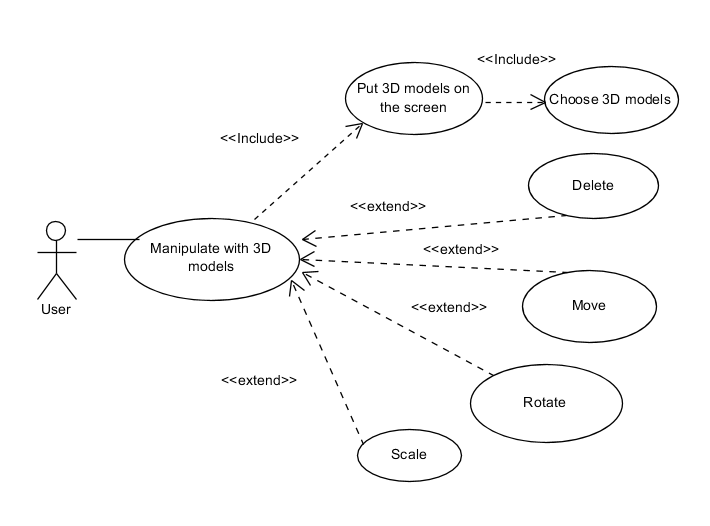
2.3.3. Choose 3D models.

After the users have detected Plane, the user can choose 3D models from the system. But before that, they must choose a category to correspond with 3D models they want to choose.



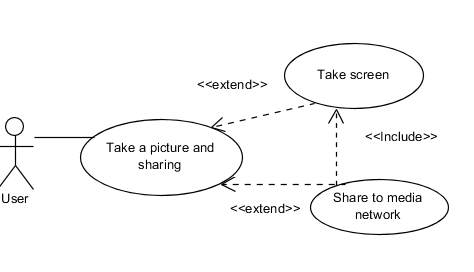
**2.3.4. Manipulate with 3D models diagram**

When the users choose one 3D model from the system. It will be put at the center of the screen (and that position is a detected plane) so they can interact with items directly such as keep the item to move to another position, rotate the item, scale the item and tap to delete the item.



**2.3.5. Take a picture diagram**

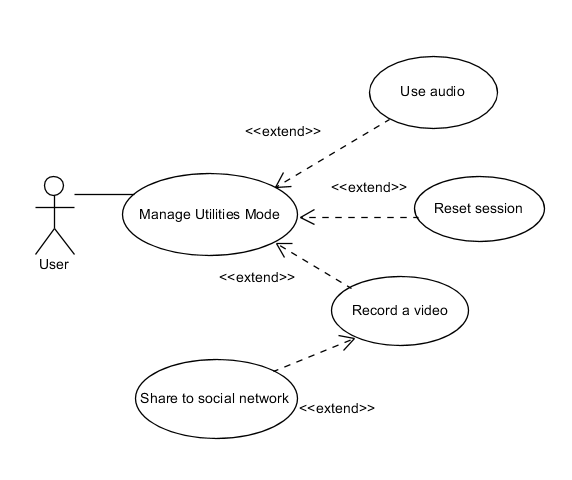
To share social network, the user must take screen by clicking camera button.



**2.3.6. Utilities mode diagram**

When the user clicks the Setting button, it will have three features use audio, reset session and record a video.

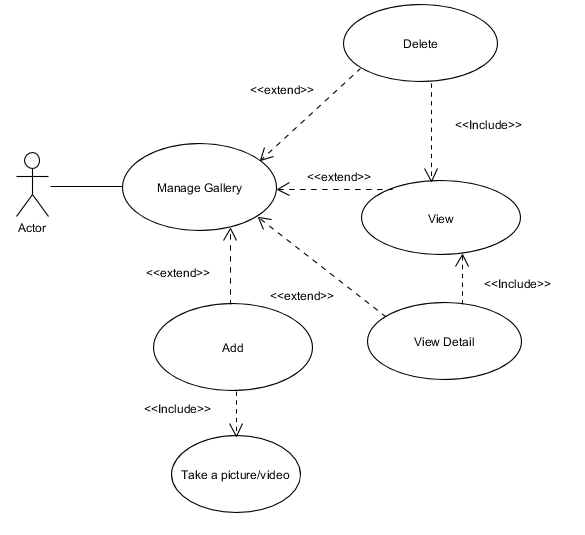
When the user records a video, they can continue choose to share to social network.



**2.3.7. Manage Gallery diagram**

Manage Gallery includes these features: Add, View, View Detail and Delete.

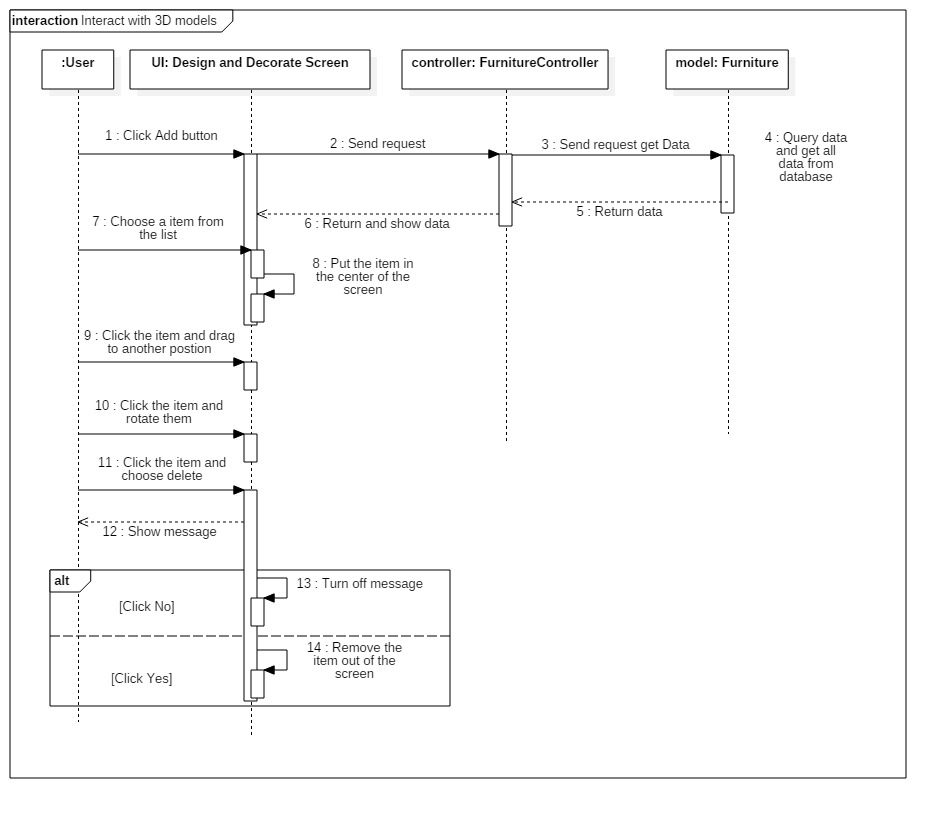
After the user take a picture/video, it can save in the database automatically. And the user can click to see detail or delete them.



**2.4. Sequence diagram**

With these use-case diagrams, here are these sequence diagram of those features above.

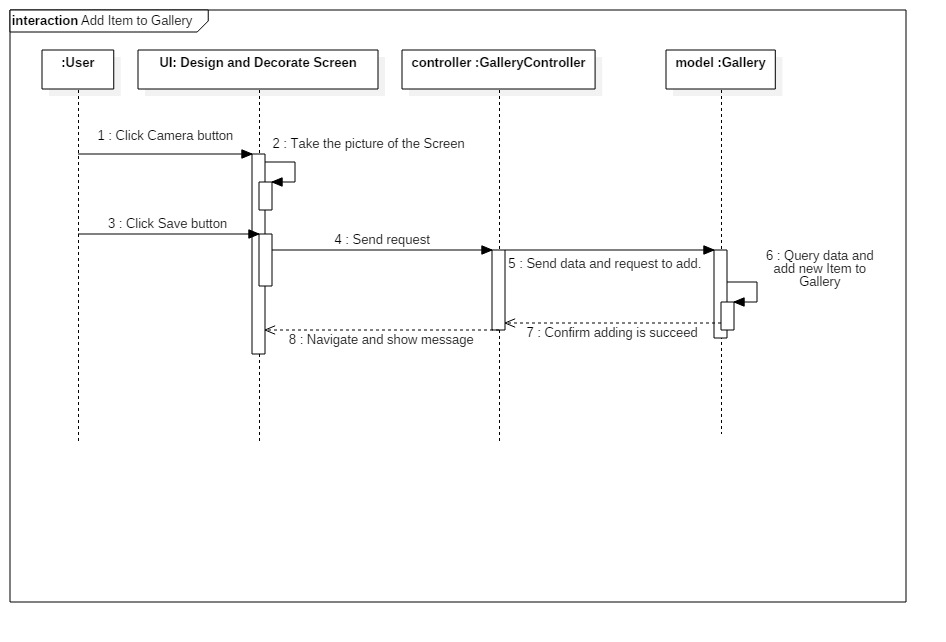
**2.4.1. Interact with 3D models sequence diagram**

****

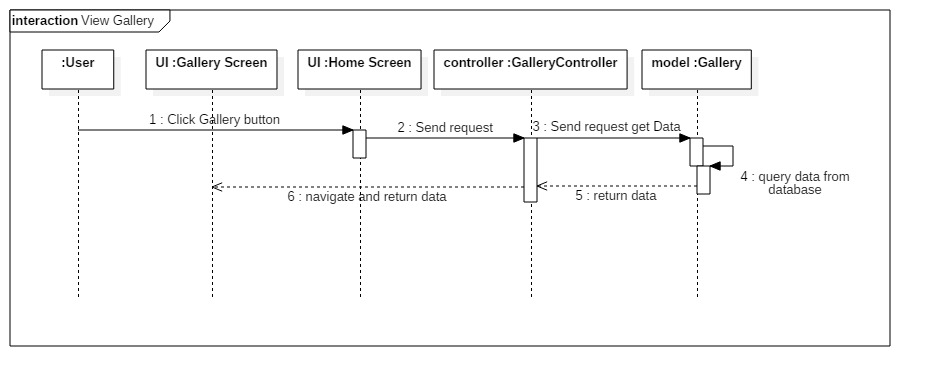
**2.4.2. Manage Gallery sequence diagram**

Manage Gallery includes three sub features: Add, View, Delete

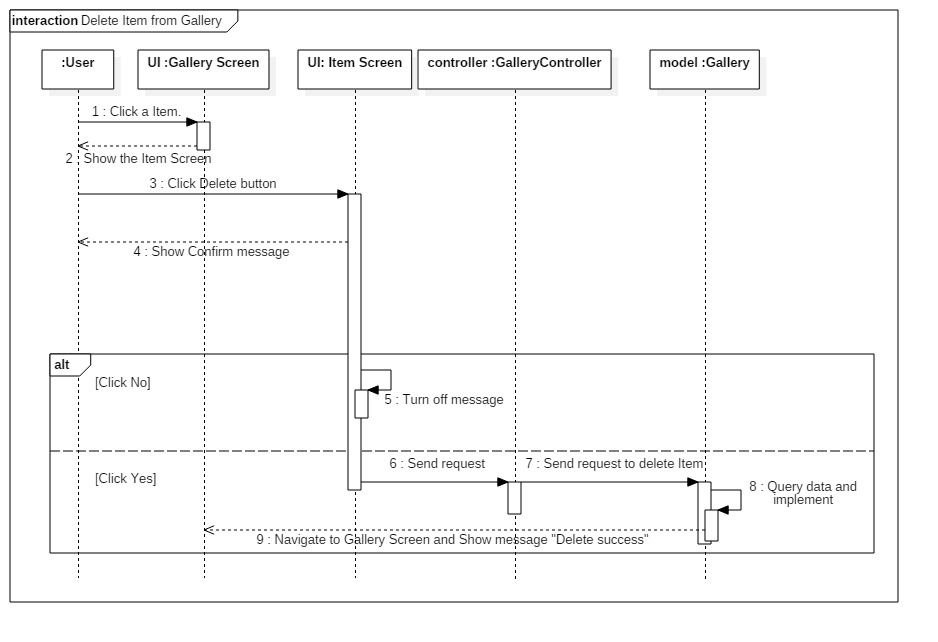
* Add the picture into the gallery.

****

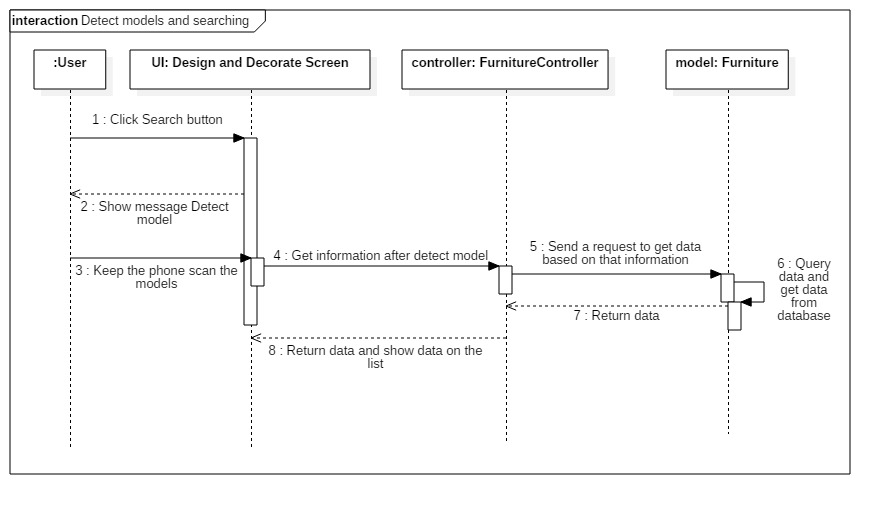
* View Gallery



* Delete an item of the gallery



**2.4.3. Detect and searching sequence diagram**

****

1. There are two types of Plane Detection: The Plane on horizontal surfaces and The Plane on vertical surfaces. The plane is not a transport. It is a kind of surfaces such as walls, tables, floors, etc. When the user wants to put 3D models on the screen, the application will check that plane on the center of the camera. If the plane is vertical, they just only can put the 3D models such as pictures, wall clock, … If they try to put tables, sofa, ... the system will show the warning message. [↑](#endnote-ref-1)