



uOttawa

**University of Ottawa
Faculty of Engineering**

GNG 1103: Engineering Design

Project Deliverable K: User Guide

Name :	Student number :
Aidan Billou	(300199731)
Peter Lee	(300173633)
Noor Madhoun	(300162420)
Magtireh Mahad	(8866131)
Sarah Qadri	(300117201)

December 7, 2020

Abstract

A current delay exists between technological tools available to the construction industry. As a result of this obstruction, construction work becomes more tedious as workers need to keep up the tradition of using pen and paper to draw, modify and redraw their designs. This issue is increasingly affecting the market for products to aid workers in their everyday jobs. Although many products do exist, they either require purchasing helmets or the applications don't include multiple features. Therefore a disconnect exists between these three aspects and no application is available to present a proper solution to this issue.

Table of Contents

Abstract	i
Table of Contents	ii
List Of Figures	ii
List Of Tables	iv
Introduction	1
Needs identification & product specifications process (benchmarking tables)	1
Benchmarking	3
Definitions	6
Target Specifications	7
Conceptual design	9
Global Concept #1	9
Project plan, execution, tracking, billing	18
Risks and Contingency Plans	21
Bill of Materials	23
Prototyping (3 prototypes)	24
Prototype I	24
Main menu	25
Video tutorial	26
Annotate	26
Language settings	29
AR camera	31
Open project & Start new project	32
Layers	33
Tools	33
Insert	34
Final solution	36
Prototype III	36
FAQ	36
Tutorial	37
Menu Pop-Up	37
Select Pop-Up	38
New Modifications	39
Analysis	43
Conclusion and recommendations and Future Plan	45

List Of Figures

- [Figure 1. Registration Page](#)
- [Figure 2. Welcome and Help Page](#)
- [Figure 3. Menu Pop-Up](#)
- [Figure 4. Tools Pop-Up](#)
- [Figure 5. Insert Feature Pop-Up](#)
- [Figure 6. Annotate Pop-Up](#)
- [Figure 7. Select Pop-Up](#)
- [Figure 8. Layers Pop-Up](#)
- [Figure 9. Layers Available](#)
- [Figure 10. Price Breakdown and Changes Pop-Up](#)
- [Figure 11. Global Concept #1 Flowchart](#)
- [Figure 12. Gantt Chart 1](#)
- [Figure 13. Gantt Chart 2](#)
- [Figure 14. Gantt Chart 3](#)
- [Figure 15. Gantt Chart 4](#)
- [Figure 16. English Welcome Screen as seen in Unity game view](#)
- [Figure 17. French Welcome Screen as seen in Unity game view](#)
- [Figure 18. English Tutorial Video Player as seen in Unity game view](#)
- [Figure 19. English Annotate Screen as seen in Unity game view](#)
- [Figure 20. English Annotate Help Pop-Up as seen in Unity game view](#)
- [Figure 21. French Annotate Help Pop-Up as seen in Unity game view](#)
- [Figure 22. English Language Selection Screen as seen in Unity game view](#)
- [Figure 24. Language Selection Screen as seen in Unity game view when user hovers over buttons](#)
- [Figure 24. Language Selection Screen as seen in Unity game view when user hovers over buttons](#)
- [Figure 25. Design Progress](#)
- [Figure 27. AR Camera Output](#)
- [Figure 27. AR Camera Output](#)
- [Figure 29. Script for ARTap, allowing placemat](#)
- [Figure 29. Script for ARTap, allowing placemat](#)
- [Figure 31. Open Project and Start New Project Scene View in Unity](#)
- [Figure 31. Open Project and Start New Project Scene View in Unity](#)
- [Figure 32. Layers Game View in Unity](#)
- [Figure 34. Tools, English and French, in Game View in Unity](#)
- [Figure 34. Tools, English and French, in Game View in Unity](#)
- [Figure 35. Inserted Model Chair using Plane Detection](#)
- [Figure 36. Design Flowchart with updated tasks](#)
- [Figure 37. FAQ in game view on Unity](#)
- [Figure 38. Tutorial pop-up in game view on Unity](#)
- [Figure 39. Menu pop-up in game view on Unity](#)
- [Figure 40. Select menu in game view on Unity](#)
- [Figure 43. English annotate page in game view on Unity](#)
- [Figure 43. English annotate page in game view on Unity](#)
- [Figure 43. English annotate page in game view on Unity](#)
- [Figure 44. French annotate page in game view on Unity](#)
- [Figure 46. English welcome page on game view in Unity](#)

- [Figure 46. English welcome page on game view in Unity](#)
- [Figure 48. French welcome page on game view in Unity](#)
- [Figure 48. French welcome page on game view in Unity](#)
- [Figure 49. Main interface with new appearance on game view in Unity](#)
- [Figure 50. Design Flowchart with updated tasks](#)

List Of Tables

- [Table 1. Customer Needs and its interpreted Importance](#)
- [Table 2. Design Criterion and Functional/Non-functional Criteria](#)
- [Table 3. Metrics and Benchmarking Properties](#)
- [Table 4. Comparison of Importance Given by each Plug-in](#)
- [Table 5. Functional Requirements](#)
- [Table 6. Constraints](#)
- [Table 7. Non-Functional Requirements](#)
- [Table 8. Project Plan](#)
- [Table 9. Schedule Risks and Contingency Plans](#)
- [Table 10. Bill of Possible Materials](#)
- [Table 11. List of Folders and Subfolders in the Final Product File](#)

1. Introduction

In response to the need of requiring a properly functioning well rounded AR application, we have created an application incorporating not only what our user wants, but also what they require. Focusing on user experience and client needs, we created an application called Pellucid. This deliverable encapsulates our step by step procedure going from problem statement to our final prototype and finished product.

2. Needs identification & product specifications process (benchmarking tables)

Due to the current state of the construction industry, change is required to attract more workers as well as help improve the work process. It is our goal to devise a solution that will blend the innovation of technology and the creativity of construction. Our mission is to develop a VAR (virtual altered reality) technology that will make lifelike models accessible with the touch of a fingerprint.

Table 1. Customer Needs and its interpreted Importance

Importance (5>1)	Need
5	Able to see BIM in either VR or AR
2	Tech and environment friendly
3	Software application must be open source or free of use
5	The product can see every single component of the building in 3D (wires, pipes, walls...)

4	The product is compatible with android and ios
4	The product is easy to understand and accessible
3	The product is accessible off-site and on-site
3	The product graphics are accurate
3	The product is easy to use

Based on our needs, we are then able to differentiate them by which is functional and which is not functional. The purpose of doing so is to classify our different aspects of the product to understand what requires the most importance in terms of functionality.

Table 2. Design Criterion and Functional/Non-functional Criteria

Design Criteria	Functional or non functional
Switch between operating modes	Functional
- Takes little memory - Simplified commands to reduce energy consumption	Non Functional
Accessible to everyone Available for free	Non Functional
Accuracy and Specificity of displayed information	Functional
Compatible with the two main operating systems: Android and IOS	Functional

-Bilingual user guide - Available tutorials	Functional
Accessible from anywhere	Functional
Accuracy and aesthetics	Non Functional
User friendliness	Functional

3. Benchmarking

After having categorized our aspects based off of functionality, we can compare different pre-existing tools that incorporate the same ideas. Benchmarking takes different applications and compares their individual prioritization of features we are looking to create. This step helps us better understand what is already on the market so that we can create something up to its standard. In the first table, each product describes whether it possesses a certain feature or not, whereas the second table shows their prioritization based on a grading scale out of 5 points.

Table 3. Metrics and Benchmarking Properties

#	Metric	AR foundations	Vuforia	ManoMotion
1	Switch between operating mode	Yes	Yes	Yes
2	Open Source	Not open source. [1]	Vuforia is not open source, but it has a free version and the price of the full version is reasonable. [2]	Contains open source software subject to license terms. [3]

3	Compatible with different operating systems	Yes	Yes	Yes
4	Accuracy and aesthetics	Tolerable	Tolerable	Tolerable but depends on the surroundings(e.g light)
5	Cost	Free with Unity. For unity plans, see below. [4]	Develop for free, purchase deploying. Basic - \$42/mo, Basic + Cloud - \$99/mo and more.	The SDK is free of charge for both private and business users up to 10 000 end users for each application.
6	Motion Tracking	No	No	Yes
7	Limited Area	No	Yes (you need to select a target area)	No
8	2D Tracking	Yes	Not clear	Not clear
9	Device Requirement	Android 7.0(minimum) Iphone 6S	Android 4.1.x+ ios 9+	Any smartphone with camera feature
10	Device tracking	Yes, included in ARCore, ARKit, Magic Leap, HoloLens. [5]	Yes included, but must enable Vuforia Fusion. [6]	Only hand and skeletal tracking. [12]
11	Plane detection	Yes, included in ARCore, ARKit,	No, so far Vuforia doesn't provide plane tracking of the	The developers have integrated ARkit thus allowing the device to detect

		Magic Leap. [5]	real world [7]	planes.
12	Occlusion/depth masking	Yes, included in ARKit. [5]	Yes, use the occlusion model in Unity.	Yes, included in SDK 2.0
13	Measures Distance	Yes. [8]	Yes. [9]	Yes
14	Anchors	Yes, included in ARCore, ARKit, Magic Leap, HoloLens. [5]	Yes, using World Anchor. [10]	Yes, using Google ARCore

Table 4. Comparison of Importance Given by each Plug-in

#	Metric	Importance (5>1)	AR Foundations (5>1)	Vuforia (5>1)	ManoMotion (5>1)
1	Switch between operating mode	5	5	5	5
2	Open Source	3	0	2	4
3	Compatible with different operating systems	5	5	5	5
4	Accuracy and aesthetics	3	4	3	2
5	Cost	3	5	3	1
6	Motion Tracking	2	0	0	5
7	Limited Area	4	5	3	5

8	2D Tracking	4	4	1	1
9	Device Requirement	3	3	4	5
10	Device tracking	5	5	4	3
11	Plane Detection	5	5	0	5
12	Occlusion/depth masking	3	4	3	3
13	Measures Distance	4	5	5	5
14	Anchors	4	4	4	4
Total		53	54	42	53

Therefore, the plugins with the most similar total importance to the desired design criteria and metrics are Manomotion, and AR Foundations.

3.1. Definitions

Operating mode: In our case, the viewing mode. Either AR or VR.

Open Source: The source code of an open source program is available for everyone to use, modify and distribute for free.

Compatible with different operating systems: Ability to function with ease across different devices.

Accuracy and aesthetics: Allows users to precisely navigate through a structure that while being pleasing to the eye.

Cost: Money required to buy and use the product.

Motion Tracking : It is an ability to respond when physical motion is detected.

Limited Area : It defines the maximum area in real life on which the device can detect as editable.

2D Tracking: 2D tracking is when you only use two coordinates to represent an object.

Device Requirement:: Defines the minimum or ideal configuration.

Device Tracking: track the device's position and orientation in physical space.

Tracking: The ability of the AR device to find its relative position and orientation in the physical world.

If the environment is too dark, for example, the device might "lose tracking", which means it can no longer accurately report its position.

Plane Detection: Detect horizontal and vertical surfaces.

Occlusion/Depth masking: Apply distance to objects in the physical world to rendered 3D content, which achieves a realistic blending of physical and virtual objects. Allows a more seamless view of 3D.

Measures Distance: Measures the distance of real life objects.

Anchor: An anchor is a particular point in space that you want the device to track. The device typically performs additional work to update the position and orientation of the anchor throughout its lifetime.

Anchors are generally resource-intensive objects.

4. Target Specifications

Once we have completed our needs identification and benchmarking, we need to do the target specification to clearly define what we must incorporate into our product. This step separates our features in three different categories: functional requirements, non functional requirements and constraints that limit the progress of our work.

Table 5. Functional Requirements

#	Design Specifications	Relation	Value	Units	Importance (5>1)	Verification Method
Functional Requirements						
1	Switch between operating mode	-	Yes	-	5	Testing

2	Accuracy and Specificity of displayed information	>	-	-	5	Testing
3	Compatible with different operating systems	=	Yes	IOS & Android	4	Testing
4	Bilingual user guide	=	Yes	French and English	4	Testing
5	Accessible from anywhere	=	Yes	Cloud	3	Testing
6	User friendliness	=	Yes	-	4	Testing

Table 6. Constraints

#	Design Specifications	Relation	Value	Units	Importance (5>1)	Verification Method
Constraints						
8	Time	>	>dec 1	days	4	\
9	Cost	=	50	\$	3	Estimate
10	Computer system RAM	<	8	GB	3	testing
11	System requirements	<=	Phone with a camera	N/A	3	Testing

Table 7. Non-Functional Requirements

#	Design Specifications	Relation	Value	Units	Importance (5>1)	Verification Method
Non-Functional Requirements						
11	Sustainability	=	Yes	N/A	2	Testing
12	Accessible to anyone	=	Yes	N/A	3	Testing
13	Accuracy and Aesthetics	<	9	%	3	Testing

5. Conceptual design

5.1. Global Concept #1

After having completed all the essential steps to design thinking, we can finally put together our design concept, including features influenced by the previous steps.

At initial contact, the user will be prompted to either register a new account with the app using their emails, Google, Facebook, or a company login using their work email addresses as shown in Figure 1. Then they will be asked for their language preferences, and the main two languages offered are English and French. Video youtube tutorials will be accessible in French and English. However, written tutorials will be offered in other language options such as Arabic, Urdu, Korean, and more.

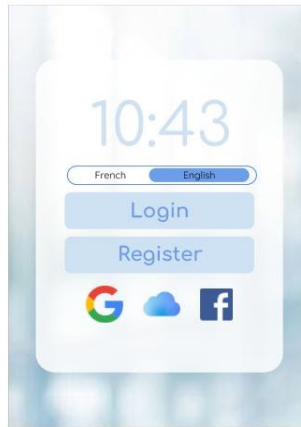


Figure 1. Registration Page

As seen in Figure 2 below, the welcome page will ask for the user to either start a new project, open an existing one, or go through freelance mode. A new project means the BIM needs to be uploaded and no previous changes will be present. The help function will allow users to read through frequently asked questions, access our contact information and it will also contain a textbox for users to submit any questions they want to ask. The tutorial will be mandatory for the user's first run through and will create pop up boxes pointing towards a specific feature and explaining exactly what their function is and how to use it.

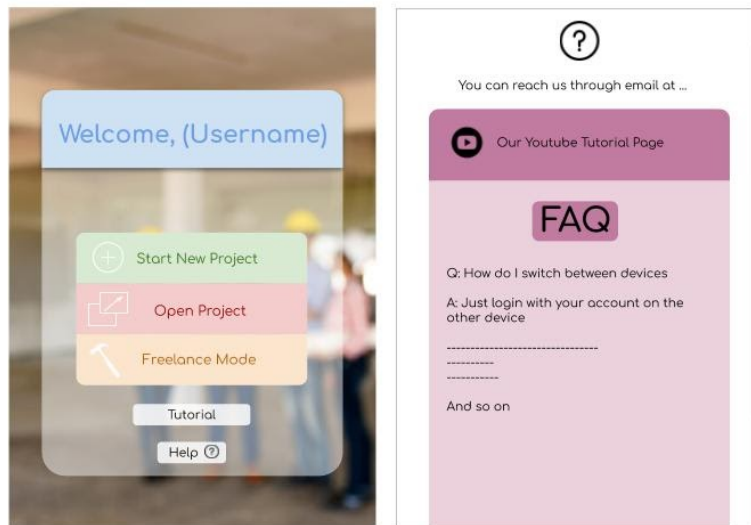


Figure 2. Welcome and Help Page

As you get forwarded into the app and the camera is open, the user will be able to see 6 clickable options, and the project title. The title will be named after the BMI file uploaded or whatever the user chose if they were in freelance mode. Three horizontal lines can be seen on the top left corner indicating the menu, which when clicked, contains the option to logout/sign-out, return to the main menu, or ask for help.

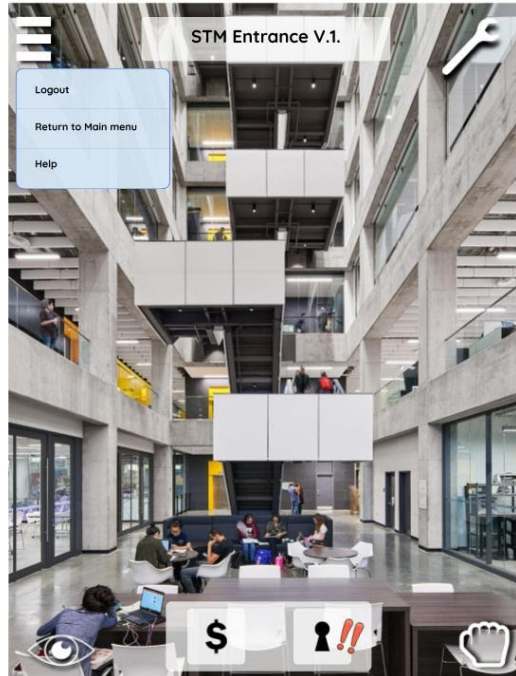


Figure 3. Menu Pop-Up

On the top right corner is a wrench icon named “tools” which contains the options to delete, insert, or annotate a feature. The delete will allow the user to delete an individual feature and everything around, under, and over it will still be visible.

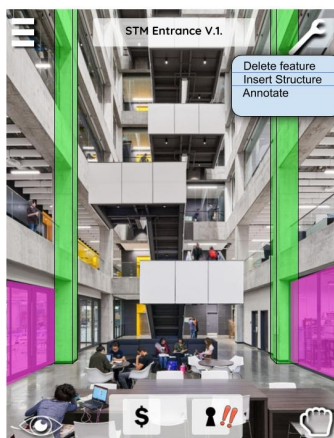


Figure 4. Tools Pop-Up

When Insert Structure is selected, a library of readily available features/structures will pop up, allowing the user to filter through the specific kind of structure they want to insert (doors, walls, hinges, beams etc), the material they need, and the size they require. Everything made of one material will be color coded the same for ease of view. When the item is being placed, the user will be informed if either the placement or the size is not in compliance with safety regulations.

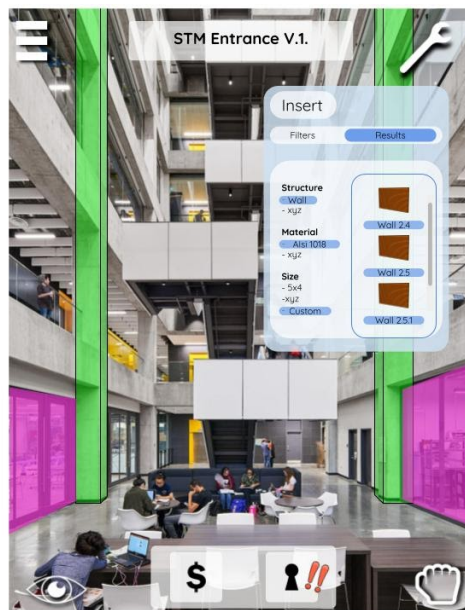


Figure 5. Insert Feature Pop-Up

The annotate feature will allow the user to select the item they want to write a note on, and then a pop up will allow them to write an unlimited amount of characters (text, numbers, and symbols only, no emojis).

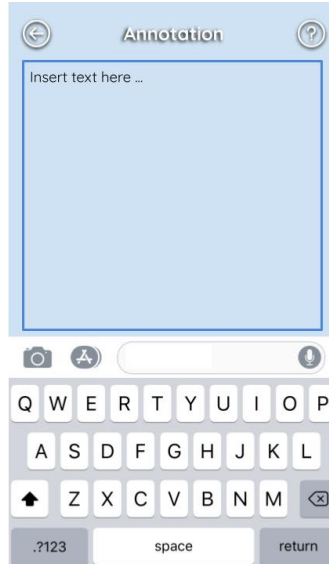


Figure 6. Annotate Pop-Up

The hand icon also known as the “select” feature will be prompted when clicked on a 3D object and the user may Annotate, Hide, Measure, or to display all dimensions/information available. There is some overlap of options available in tools and select because the user can click tools and then find an object to work on, or click select and find an object to work on before deciding what to do for it. This is left here to appeal to different working styles of users. The annotate feature works the same as it does in tools. The hide feature works almost exactly like the delete feature, the only difference is that the item will not be permanently deleted, only made invisible and the user is able to reverse this. This would be useful when visualizing and figuring out order. For example when a wall is selected to hide, only that wall will be invisible, everything around, below, and above it will be shown all the same. When display all dimensions is selected, the user will be shown a pop up window with all the details of the item they selected. Measure would allow the user to select point A and point B and after this, they can measure length in a straight line of whichever area they wanted in the real world. They can also measure

things like radius, area, volume, using the values they input or the ones provided with the BMI. The units are all convertible between imperial and metric standards.

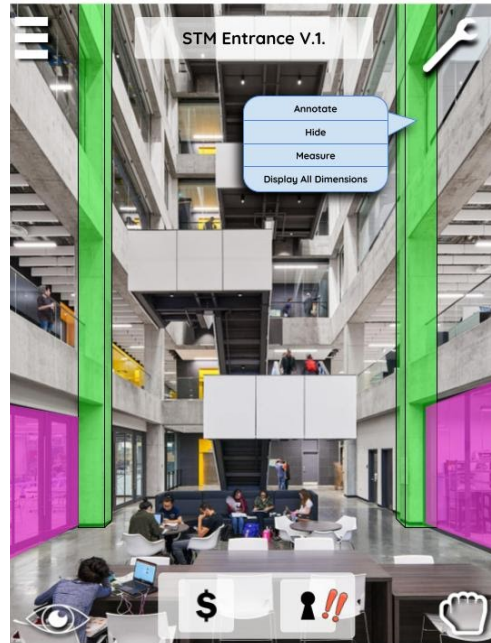


Figure 7. Select Pop-Up

The eye icon displays the Layers feature, when selected, the user will see a pop up containing an eye next to the name of each layer. If the eye is selected, the selected layer becomes invisible. The kinds of layers we would have are structural, mechanical, electrical etc. as much as the BMI allows. You can click multiple eyes closed at the same time and vice versa.

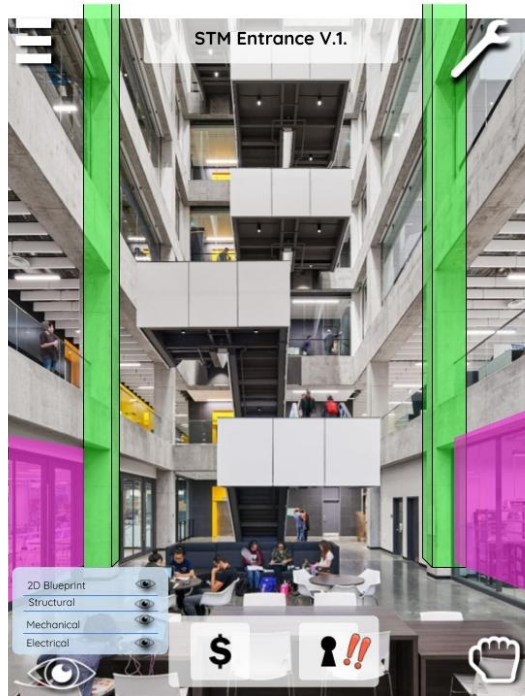


Figure 8. Layers Pop-Up

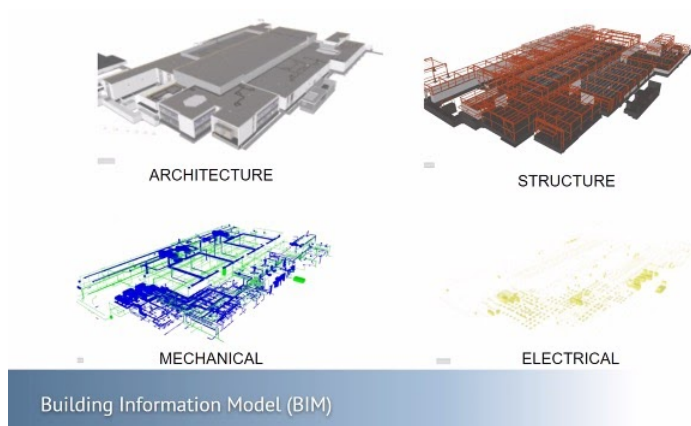


Figure 9. Layers Available

In the bottom center, there are two features, price breakdown, and changes. Price breakdown allows the user to view the total cost of all added features from tools and see an approximate range. “Changes made” contains two tabs, personal and everyone. When personal is selected, the user can see exactly what changes they made to the model and in what order. In the

“everyone” tab, the user can see what changes everyone else working on the same model and connected to the same cloud software (as they would at a private organization), have done. In addition to this, two red exclamation marks will appear once other people start making changes.

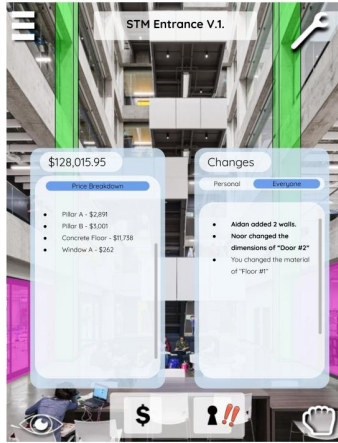


Figure 10. Price Breakdown and Changes Pop-Up

Below demonstrates a flowchart for the logical run through of the app through the lens of a user.

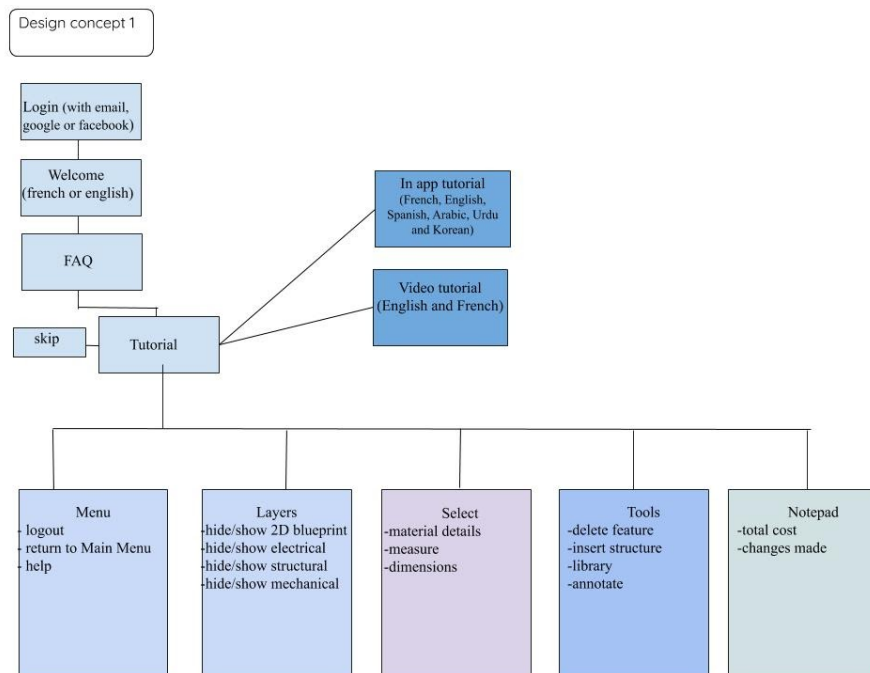


Figure 11. Global Concept #1 Flowchart

6. Project plan, execution, tracking, billing

Once the conceptual design has been finalised and chosen, we must create an organised plan to assure the execution of our project does not get delayed or hindered by avoidable issues. To do this, we must create a detailed set of tasks, specify the owner, the due date as well as the amount of time it will take to complete each singular job. When a specific plan is set, it is then possible to create a Gantt chart, a visual tool that may facilitate time management.

Table 8. Project Plan

Task	Owner	Duration	Due date
Project schedule and Cost (deliverable E)	Everyone	7 days	October 22, 2020
General idea for game is complete (packages and assets considered, research is done)	Everyone	7 days	Reading week
Prepare procurement documents	Everyone	4 days	October 27, 2020
Prototype I and customer feedback (deliverable F)	Everyone	7 days	November 5, 2020

Prototype II and Customer Feedback (Deliverable G)	Everyone	7 days	November 12, 2020
Revise Risk Assessment	Everyone	5 days	November 26, 2020
Prototype III and Customer Feedback (Deliverable H)	Everyone	14 days	November 26, 2020
Design Showcase Presentation(Deliverable I)	Everyone	14 days	November 26, 2020
Communication Plans	Everyone	4 days	December 1, 2020
Final Presentation (Deliverable J)	Everyone	4 days	December 1-3, 2020
User Guide (Deliverable K)	Everyone	7 days	December 3, 2020

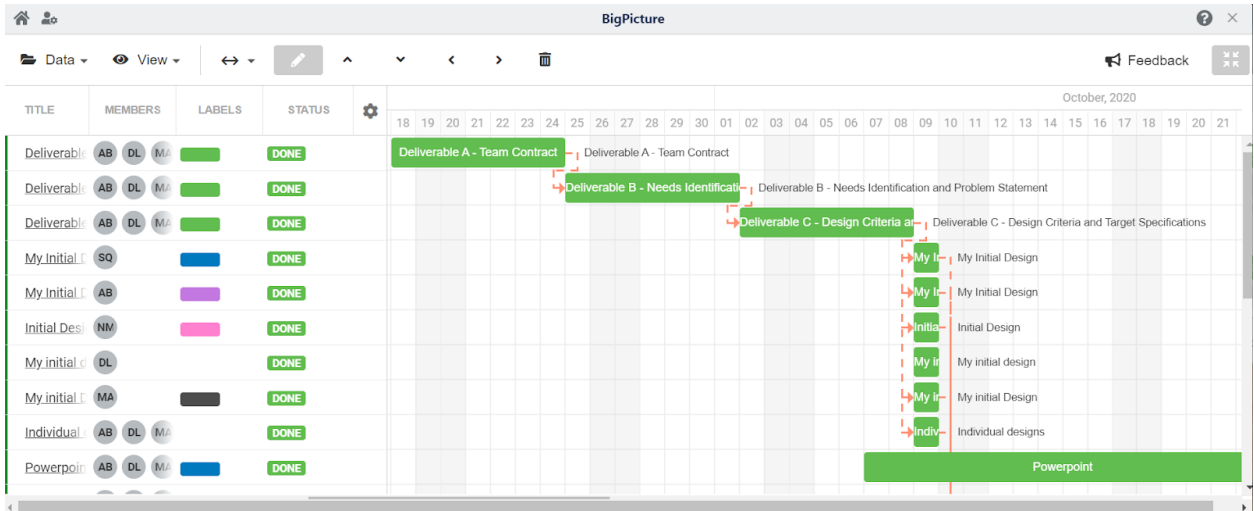


Figure 12. Gantt Chart 1

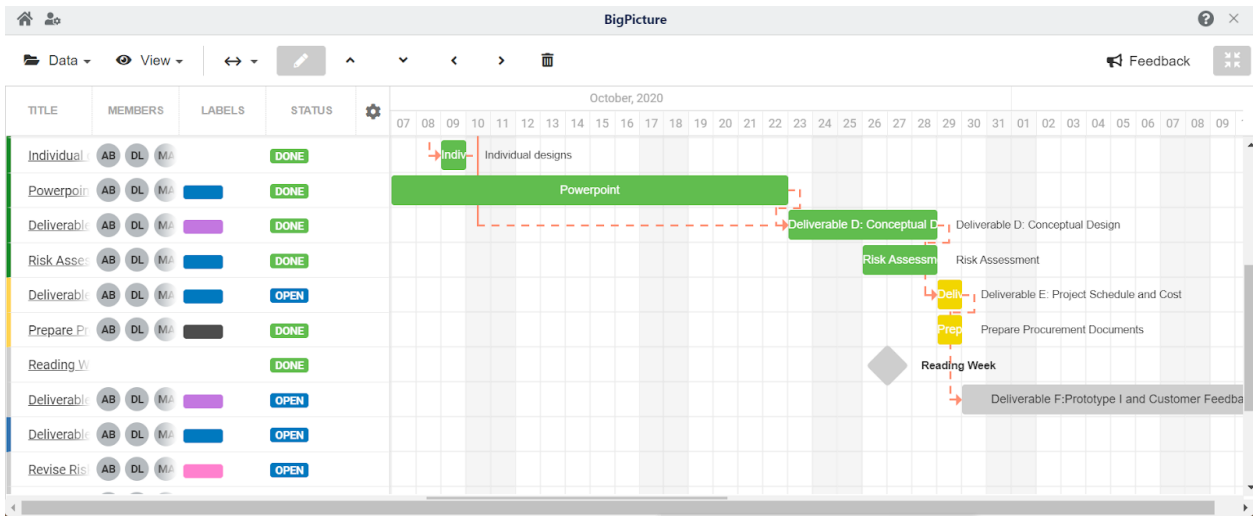


Figure 13. Gantt Chart 2

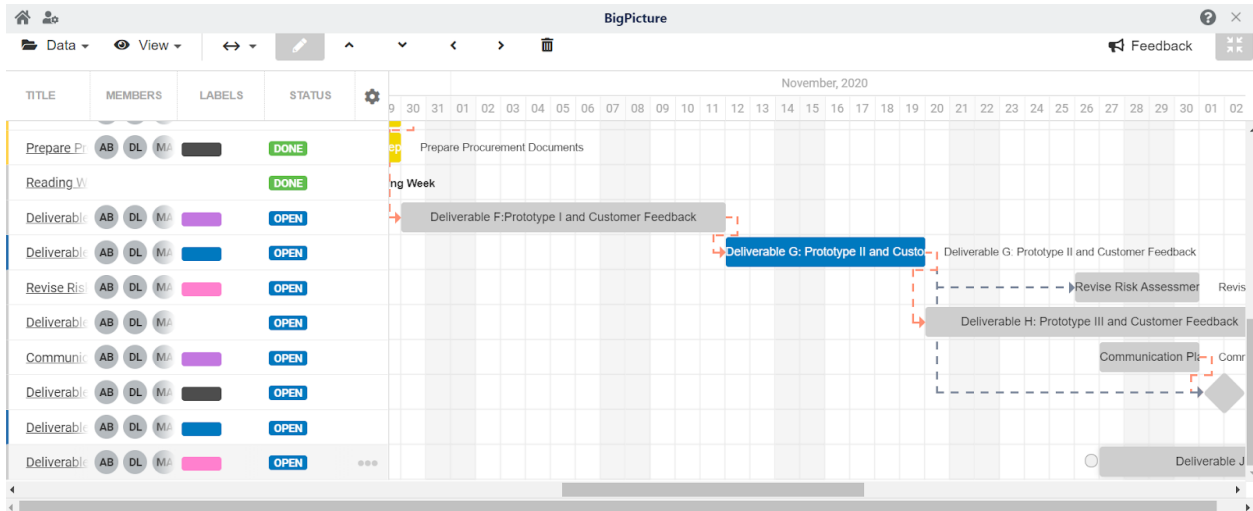


Figure 14. Gantt Chart 3

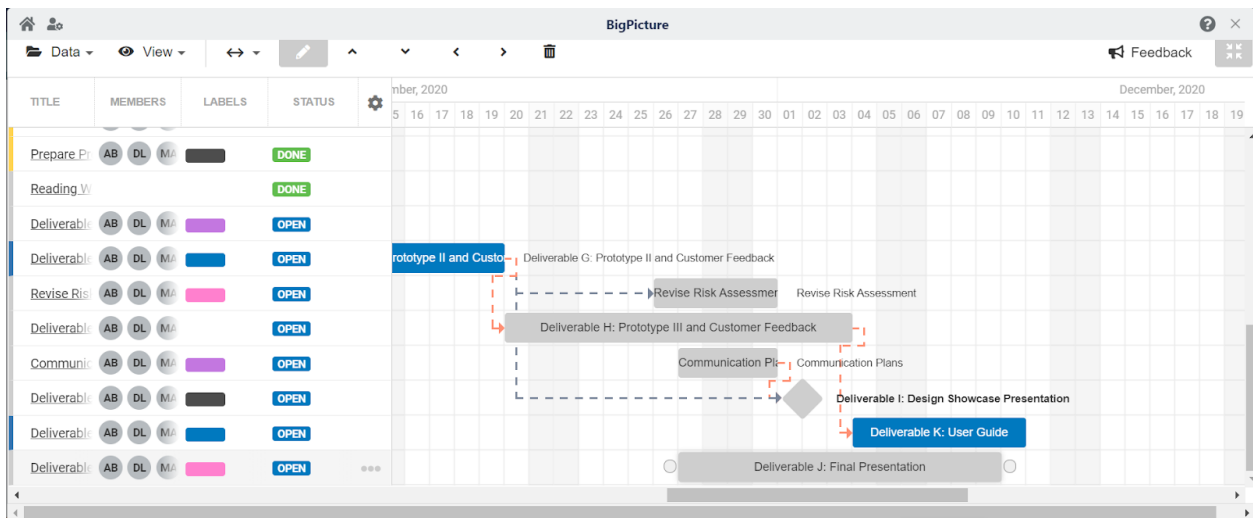


Figure 15. Gantt Chart 4

7. Risks and Contingency Plans

Potential risks are important to include in our planning. In fact, these may lead to not only slowing down the project progress, but the team dynamics as well which then directly affects the way the group works and advances together. Creating contingency plans to help have a sort of safety net around certain issues helps us be prepared for any possible situation and outcome our work as a team may

encounter. Many things can go wrong, from a team issue to a technical malfunction from the system we are working with, facing, and solving the issue at hand are of utmost priority. According to our team contract, we have set certain rules and protocols to follow in certain cases such as a member of the team failing to deliver their work on time. In this case, the work would be completed as soon as possible to avoid getting sanctions and the team member would get a single warning. If the events reoccur, the TAs would be informed. We may also very probably face issues with the systems we are using. Sometimes, these systems do not function well on everyone’s computers, or they do not cooperate with us, or different versions of the system are offered and not all of them work the same... In this very likely case, it is therefore important to have set guidelines amongst the team members to know how to handle the situation properly and advance without getting off track set by our project plan. Therefore, it is essential to plan for potential risks because some of them are more than likely to occur.

Table 9. Schedule Risks and Contingency Plans

Risks	Chance	Impact	Contingency
Issues with Unity	HIGH	HIGH	Try to debug as soon as possible, look up in forums for people with similar issues
One team member can't complete a task	LOW	HIGH	The rest of the team comes up with a plan to split the task and minimize delays
A planned feature of the app is not feasible	Moderate	Moderate	Have a meeting and choose a more feasible option
More than planned time is needed	Low	High	Estimate the appropriate extra amount of hours needed and add it

Issues with Unity:

- It is possible that using different unity versions might create problems for the group since each version has different features. Unity itself might have issues in terms of specific features not properly working or not performing the way we wanted it to. One of the biggest issues we might face using Unity is the possibility that the application itself crashes and gives us a hard time to use.

Our team member can't complete a task:

- It is likely that a team member is sick or overwhelmed with work or personal issues which may cause them not to be able to complete tasks on time.

A planned feature of the app is not feasible:

- Using unity and multiple plugins, it is very likely that either one of them have features that end up not being what we need for our project. This means that what we had planned to use needs to be modified and we would need to search for an alternative.
- It is difficult to figure out the right code for a specific feature we want.

More than planned time is needed:

- In case of potential delay in our project advancement, it is possible that we need more time to complete the project since the tasks did not take the same amount of time to complete as we had initially planned.

8. Bill of Materials

With a 50\$ budget, it is important to detail potential costs for systems and subsystems we may need. With an amount precisely small, we need to make sure that it is spent in an efficient and cost effective manner. Therefore, the following table presents potential costs.

Table 10. Bill of Possible Materials

Part #	Part name	Description	Cost(\$)
1	AR Foundations	Unity Plugin to help aid in	0\$
2	Unity 2019.3.15	Game Development Software used as a base for the entire project	0\$
3	SRDebugger [1]	Plugin that lets users fix potential bugs in the game itself	30\$
4	uIntelliSense — Unity API Assistant [2]	Plugin that translates code into words to explain their meaning	12.95\$
Total			42.95\$

9. Prototyping (3 prototypes)

Prototype I

Our first prototype is an extremely simple version of our final product. This version acts like a tester, giving us a slight idea of how much work and time it will take to make the whole product. This is also our first opportunity to learn how to create and work with specific functionalities from Unity.

9.1. Main menu

The main menu has the most features all grouped in one spot for this prototype. There are 3 features linked to starting a new project as well as two additional buttons that lead the user to either the tutorial or an FAQ.

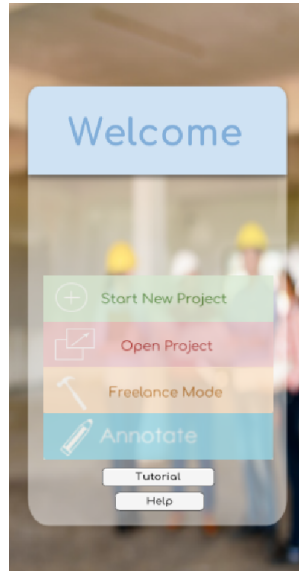


Figure 16. English Welcome Screen as seen in Unity game view



Figure 17. French Welcome Screen as seen in Unity game view

9.2. Video tutorial

The video tutorial button brings the user to a video that explains the basic features of the application. For this prototype, we do not yet have a tutorial video, however the tutorial button brings the user to a dummy video.

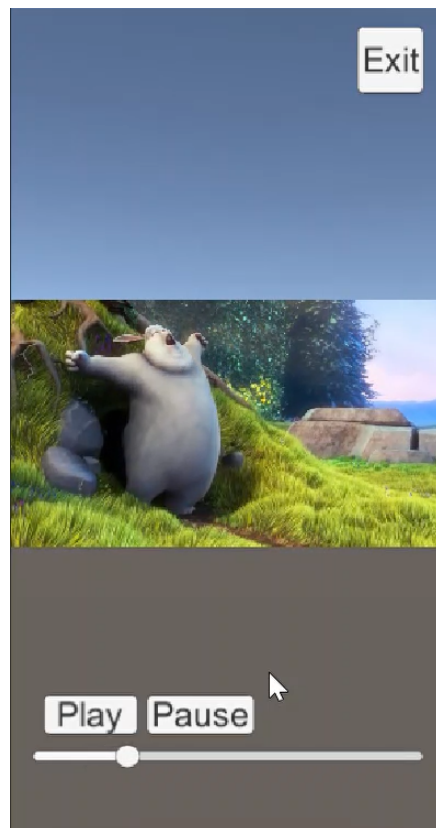


Figure 18. English Tutorial Video Player as seen in Unity game view

9.3. Annotate

In the annotate scene, the textbox prompts the user to type unlimited characters. Users may also scroll up and down by clicking and dragging the scroll bar as shown in the screenshot.

When the user clicks done after finishing their annotation, the information is saved into the app data (functions like a cloud). There is also the return button here which will be accessible everywhere throughout the app to let the user switch in between scenes.

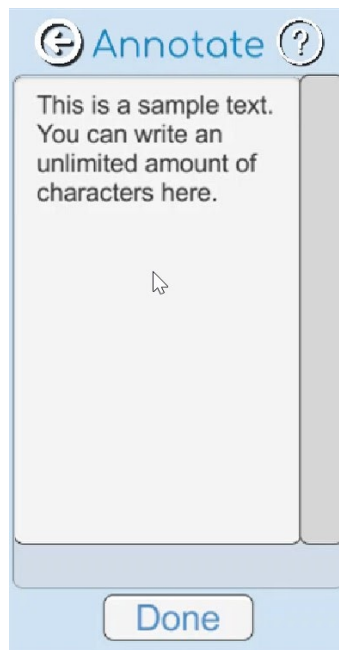


Figure 19. English Annotate Screen as seen in Unity game view

The help button gives the user access to a pop up that shows how to use the annotate feature.



Figure 20. English Annotate Help Pop-Up as seen in Unity game view

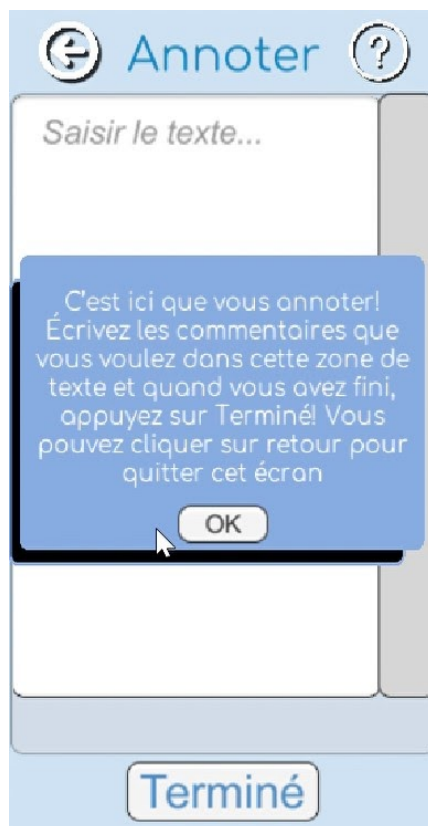


Figure 21. French Annotate Help Pop-Up as seen in Unity game view

9.4. Language settings

The language setting is a very simple feature. By choosing either English or French, it interchanges between the two languages depending on what the user chose.



Figure 22. English Language Selection Screen as seen in Unity game view

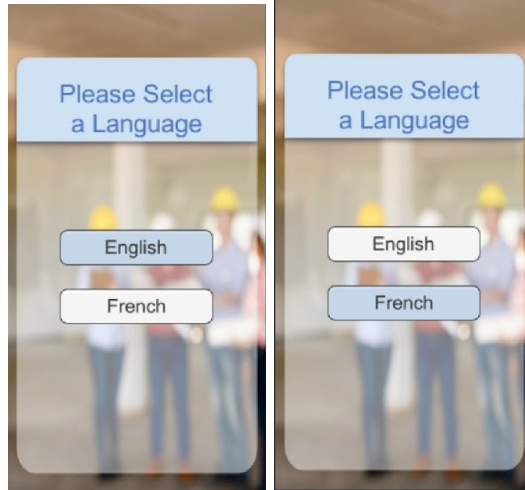


Figure 24. Language Selection Screen as seen in Unity game view when user hovers over buttons

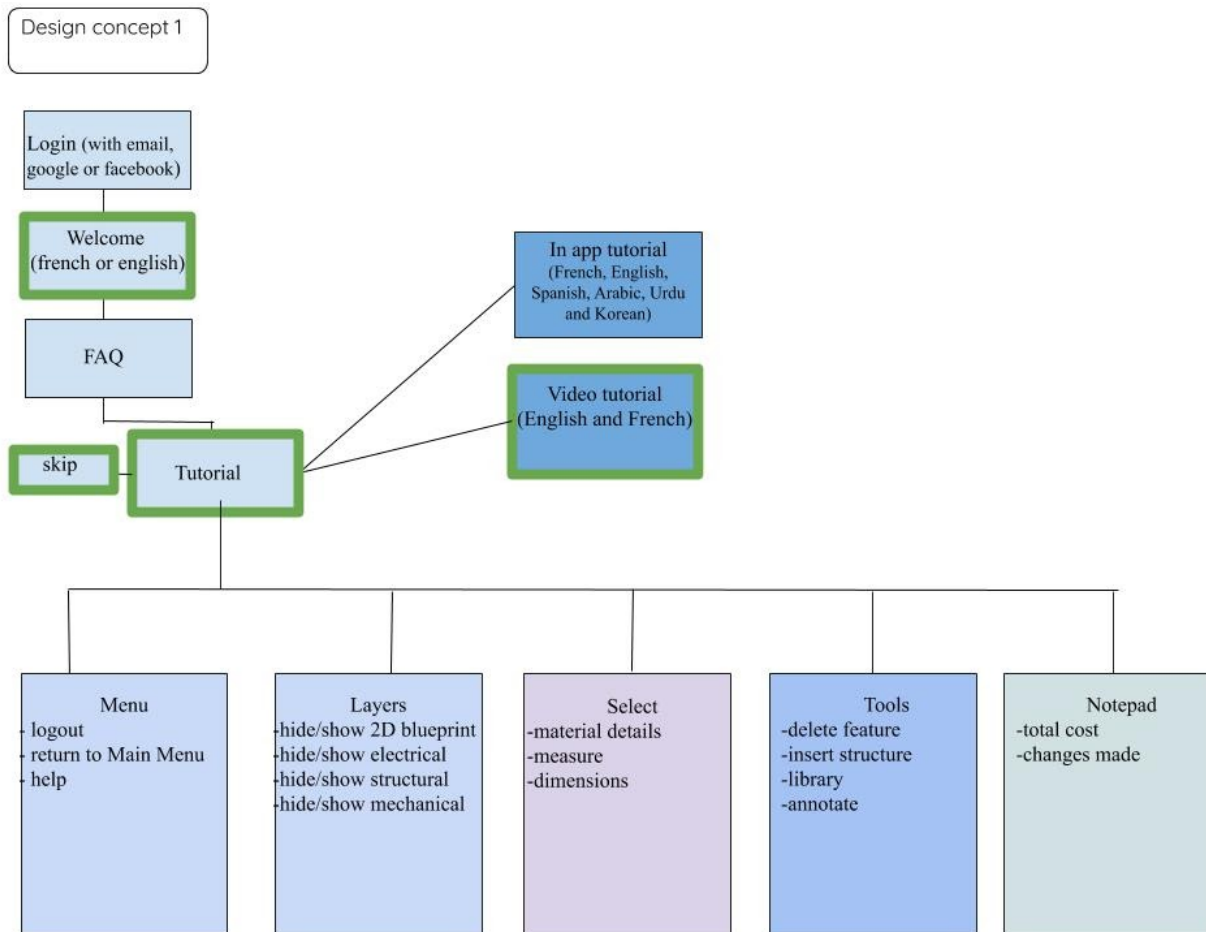


Figure 25. Design Progress

Prototype 2

Based on prototype I, we have managed to create simple functionalities and features for our application, such as a tutorial video, a welcome page, language selection... Since this prototype is supposed to include the majority of our functionalities, we have indeed added the following features.

9.5. AR camera

As mentioned in the previous deliverable, the AR camera has been a feature we have been focusing on from our early stages of conceptual design. It is important that we perfect this feature from the first step since the whole application is built around its ability to utilise AR technology. Previously, we had set up different strategies for this functionality, and in this prototype, after spending time working through Unity, with help from the TA's, we have created a functional AR camera within our application.

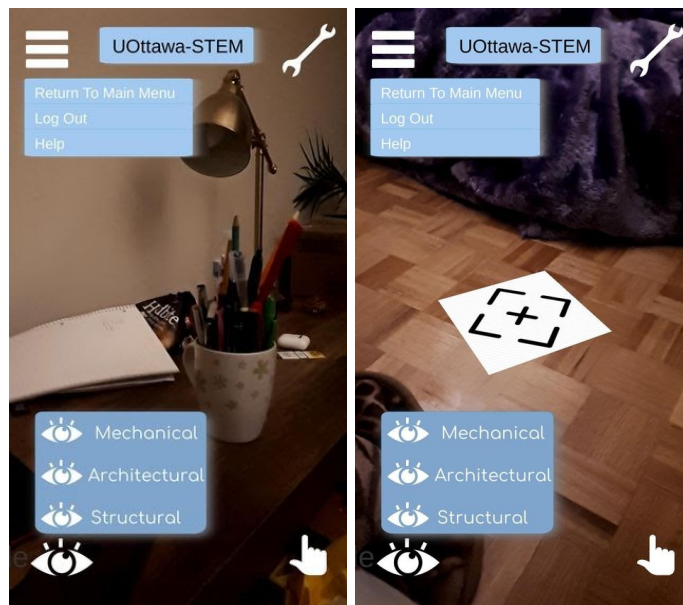


Figure 27. AR Camera Output

Assembly Information	
Filename	Assembly-CSharp.dll
using System.Collections;	
using System.Collections.Generic;	
using UnityEngine;	
using UnityEngine.XR.ARFoundation;	
//using UnityEngine.Experimental.XR;	
using UnityEngine.XR.ARSubsystems;	
using System;	
public class ARTap : MonoBehaviour	
{	
public GameObject objectToPlace;	
public GameObject placementIndicator;	
//private ARSessionOrigin arOrigin;	
private ARRaycastManager arRaycastManager;	
private Pose placementPose;	
private bool placementPosesValid = false;	
void Start()	
{	
//arOrigin = FindObjectOfType<ARSessionOrigin>();	
arRaycastManager =	
FindsObjectOfType<ARRaycastManager>();	
}	
void Update()	
{	
UpdatePlacementPose();	
UpdatePlacementIndicator();	
if (placementPosesValid && Input.touchCount > 0 &&	
Input.GetTouch(0).phase == TouchPhase.Began)	
{	
PlaceObject();	
}	
}	
private void PlaceObject()	
{	
Instantiate(objectToPlace, placementPose.position,	
placementPose.rotation);	
}	
private void UpdatePlacementIndicator()	
{	
if (placementPosesValid)	
{	
placementIndicator.SetActive(true);	
}	
placementIndicator.transform.SetPositionAndRotation(placement	
Pose.position, placementPose.rotation);	
}	
else	
{	
placementIndicator.SetActive(false);	
}	
private void UpdatePlacementPose()	
{	
var screenCenter =	
Camera.current.ViewportToWorldPoint(new Vector3(0.5f, 0.5f));	
var hits = new List<ARRaycastHit>();	
//arOrigin.Raycast(screenCenter, hits,	
TrackableType.Planes);	
arRaycastManager.Raycast(screenCenter, hits,	
TrackableType.Planes);	
placementPosesValid = hits.Count > 0;	
if (placementPosesValid)	
{	
placementPose = hits[0].pose;	
var cameraForward = Camera.current.transform.forward;	
var cameraBearing = new Vector3(cameraForward.x, 0,	
cameraForward.z).normalized;	
placementPose.rotation =	
Quaternion.LookRotation(cameraBearing);	
}	
}	

Figure 29. Script for ARTap, allowing placemat

9.6. Open project & Start new project

We have also built the scenes that appear when the welcome page buttons are selected. In our first prototype, we had made them as buttons that did not lead us to anywhere specific. For this prototype, the buttons lead us to two different scenes corresponding to either starting a new project or opening another. For each of those new scenes, we also find their own new individual features that let the user either share their work for collaboration, access a list of projects they may already have, edit the project name, add blueprints.

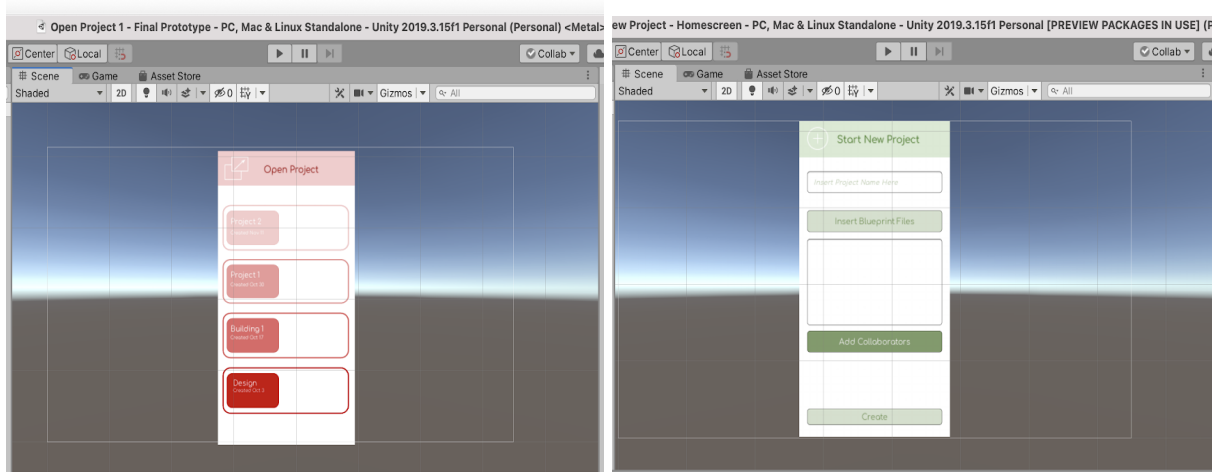


Figure 31. Open Project and Start New Project Scene View in Unity

9.7. Layers

Another important feature we developed was the layering aspect. This feature allows users to view their project from different perspectives based on a specific structure they may choose to hide or display. Similarly to the AR camera, this feature is one of the aspects our client insisted on developing. Therefore, a particular amount of attention was accorded to developing this system. As shown in the screenshots below, this feature includes showing or hiding the 2D blueprint, the electrical layer, the structural layer and the mechanical structures. The layering feature, just like the others, involves the incorporation of button structures as well as a toggle system incorporated from Unity.

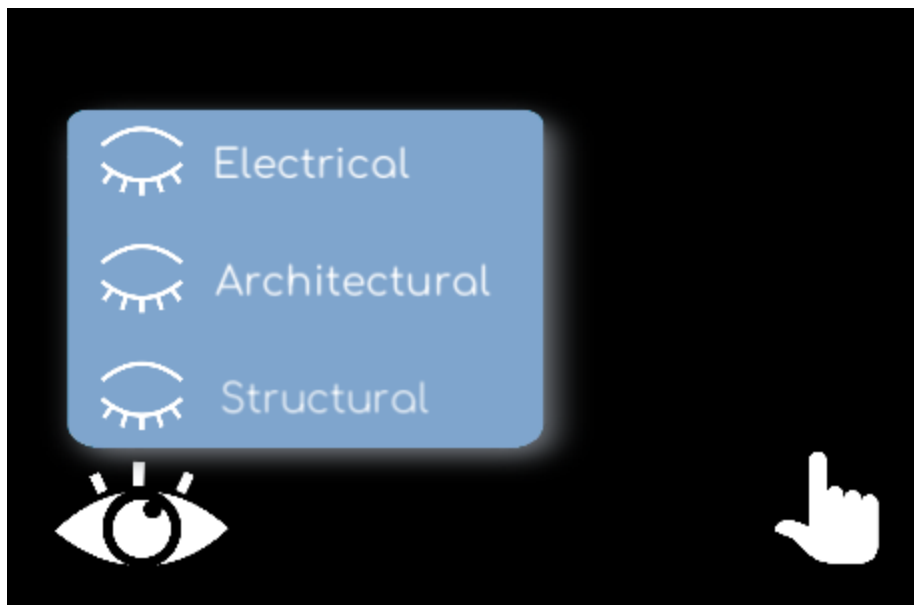


Figure 32. Layers Game View in Unity

9.8. Tools

We have also built our scenes for our tools pop-up menu. This includes as shown in the screenshots below, the delete feature, a library and our annotate portion.

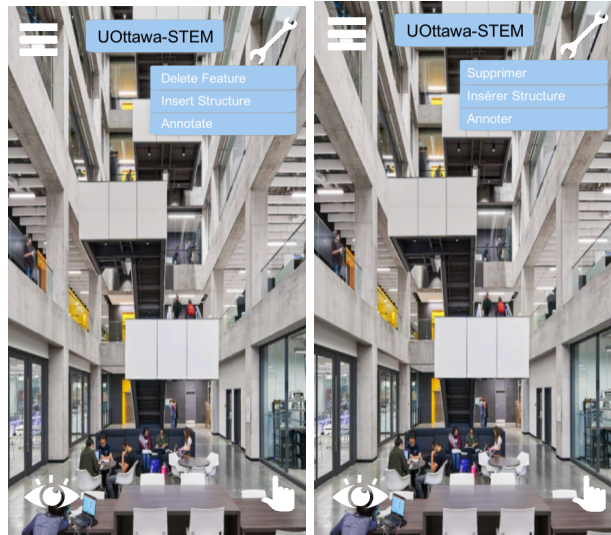


Figure 34. Tools, English and French, in Game View in Unity

9.9. Insert

The last feature we have developed for this prototype is our insert structure as shown below. The AR plane detector will scan the flat plane surface of the area, indicating it with a chunk of yellow plane. Once it is done, the user can place an object by clicking on any space that is scanned. The object can be moved around by clicking with a finger and holding it. Otherwise, it will stay in one place.



Figure 35. Inserted Model Chair using Plane Detection

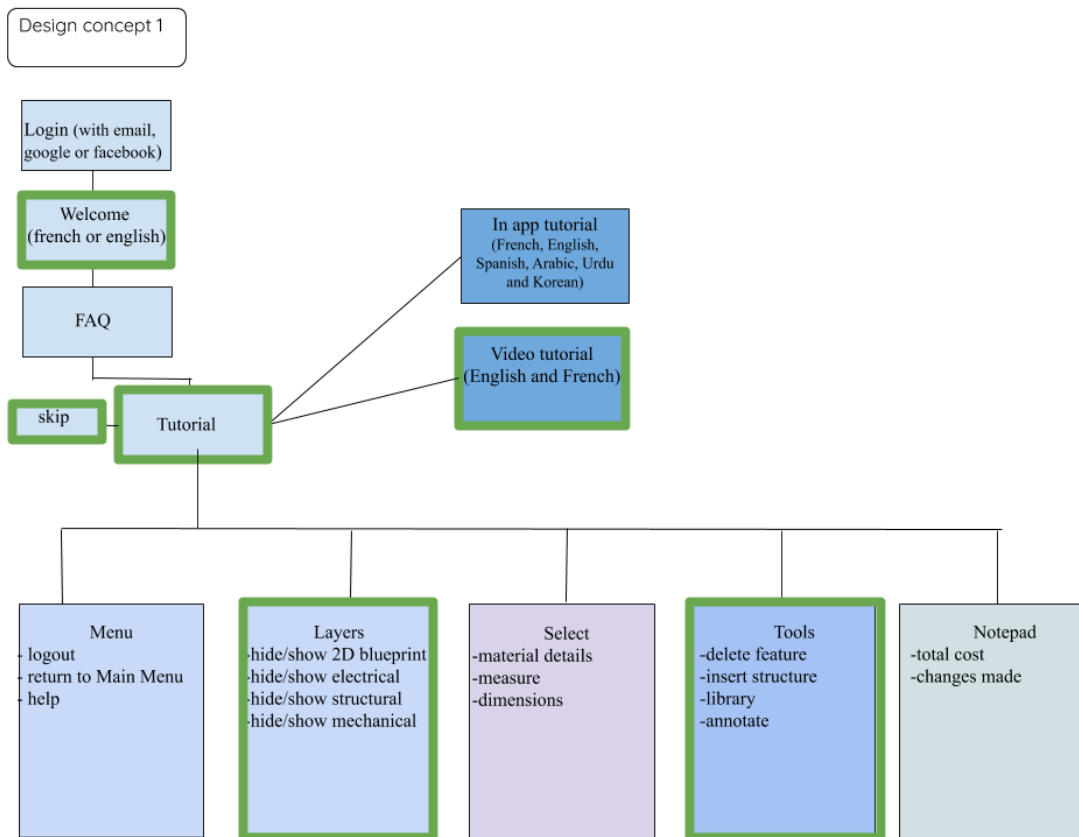


Figure 36. Design Flowchart with updated tasks

10. Final solution

Prototype III

Since this is our final prototype, we do not have many more features to add. This version only has 4 new features: the FAQ, the menu, the select option and the tutorial.

10.1. FAQ

As mentioned in our initial conceptual designs, as well as based off of client needs, we have now included our FAQ page. This would obviously improve our user experience since it helps guide any confused users to better understand how to use the application. Our FAQ is a simple page with available answered questions as shown in the images below.

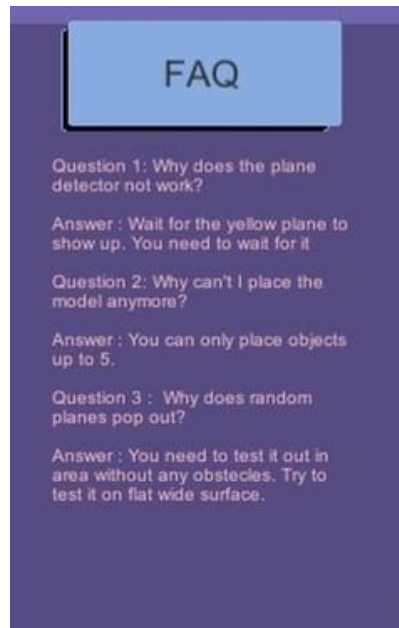


Figure 37. FAQ in game view on Unity

10.2. Tutorial

After being led to the FAQ page, it is possible for the user to come across a quick tutorial that swiftly explains basic functionalities of the application. This feature can be revisited at any point during the use of the application. However, it is mandatory to go through it initially. The user may choose to go through a video tutorial in English or follow a step by step tutorial offered in multiple languages. The tutorial would appear as follows:



Figure 38. Tutorial pop-up in game view on Unity

10.3. Menu Pop-Up

As planned in our conceptual design, we have incorporated our menu pop-up, an essential feature to our prototype. As shown below, this step involved creating a toggle menu using assets from Unity. When the toggle menu is opened, we find two buttons: return to the main menu and help. Both act as functional buttons programmed through Unity. The first one brings the user back to the first scene, where they may either start a new project or open a pre-existing one (as shown in deliverable G). Meanwhile, the second button let's the user go back to the FAQ where they encounter the FAQ questions or can also even go through the tutorial once more. These features are shown in the following figure:



Figure 39. Menu pop-up in game view on Unity

10.4. Select Pop-Up

Our final new addition to our prototype is the select pop-up. Our select menu consists of a toggle type menu including two buttons for measuring and dimensions. Keeping the same configuration as the other pop up menus, this one includes two new features that are essential to making construction decisions. The buttons are each respectively in charge of measuring specific details such as length of a beam while dimensions give specifications about certain parts (ex: volume).



Figure 40. Select menu in game view on Unity

10.5. New Modifications

Aside from what we already had created and added, for our final prototype, we have included the use of a UI pack to finalize the visual aspect of our application. Differences in appearance may appear as shown below:

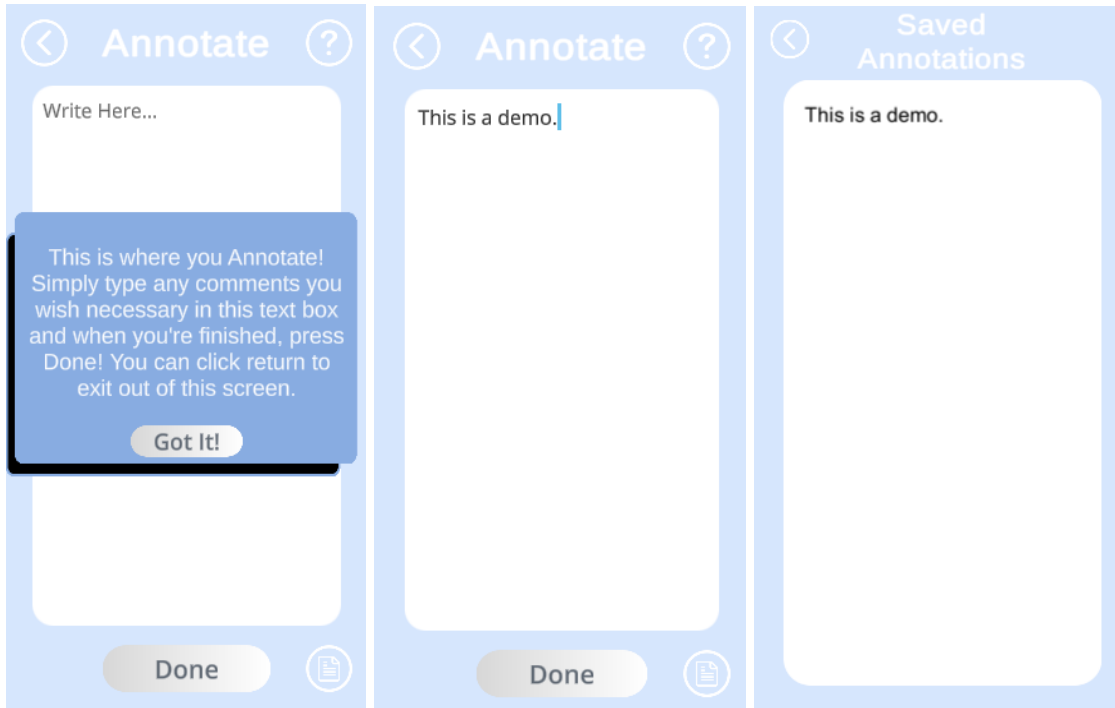


Figure 43. English annotate page in game view on Unity



Figure 44. French annotate page in game view on Unity



Figure 46. English welcome page on game view in Unity



Figure 48. French welcome page on game view in Unity

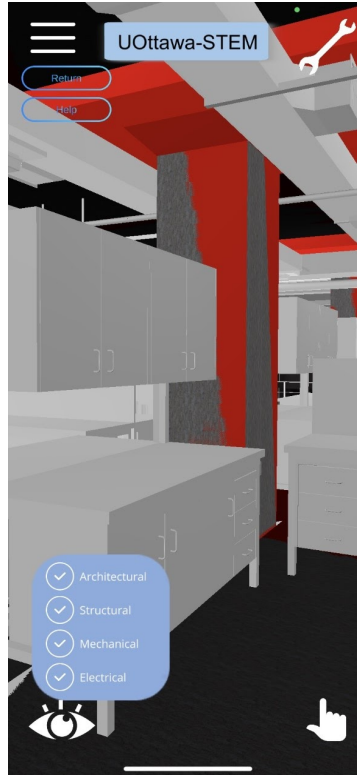


Figure 49. Main interface with new appearance on game view in Unity

With everything listed above, we have our final product, ready for design day. The following flow chart shows our final product and all its components:

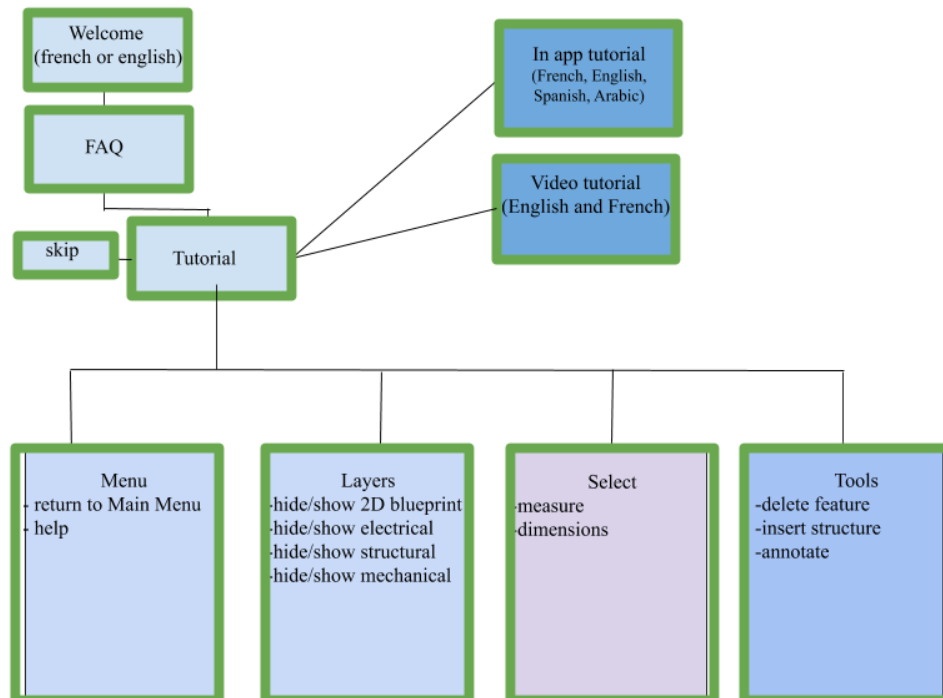


Figure 50. Design Flowchart with updated tasks

11. Analysis

In our first prototype, we were able to create very basic characteristics for our application. These initial steps are very important to ensure that we can later on add to what we already have instead of needing to restart from scratch for every single characteristic. Out of all of the work we have done on unity, the simplest thing to do was create buttons for sections like annotate, language settings and even the video tutorial. However, what has been more complex to figure out is how to utilise the AR camera and make it functional with the rest of application. Although we did not incorporate this into this first prototype, planning it for the next one was involved in the process for the first one. We need to therefore focus more attention on that aspect of the application. Seeing that creating buttons is very simple, we now know what will require more time to build and what will require less attention. With this knowledge, we may schedule

our work in a way that will account for what needs more time. Coding on Unity let us as a team plan how to build the rest of the application by utilising what we have learned. This first prototype also helped us see if there were any ideas we needed to simplify even more to incorporate into our built application.

In the second prototype, we faced expected issues with Unity for the first time while using the application. This was not a surprise as we had planned it in an earlier deliverable when assessing risks and managing them. Although we had known this issue would arise, it still negatively impacted our progress as we had to find an alternative solution to what we had initially planned (in this case using a 3D model other than the one provided by our client). Despite these hurdles, we were able to accomplish one of the biggest challenges in our work: setting up our functional AR camera. However, it is important to note that this aspect requires more attention and eventually reworking depending on how we see fit while preparing our final prototype and Design Day presentation. For this prototype we had also made our first purchases for Unity, which in itself took time as we had to decide how to spend our money strategically to stay under budget. This prototype helped us advance in our project progress, but in particular, helped us understand what exactly is essential and what is not for our final presentation in the coming couple of weeks.

The final deliverable and prototype seemed to be the one that required the most attention and organisation. In fact, the final prototype, our deliverable and a final presentation as well as the plan for said presentation were all due at the same time to be completed in a period of only a week. Therefore, our group needed exemplary organisation as well as communication throughout this period of time. Aside from this work, this period of time, approaching multiple exams created a culmination of work and stress which our group needed to manage quickly and

efficiently. Although this seemed like a heavy burden to carry during an entire week, we managed to organise ourtime, giving ourselves deadlines to be able to divide the week for Unity work and final presentation preparation work. In theory, this seems like an easy task, but we needed to work together to advance fast all while avoiding needing to restart if one aspect of our work did not advance properly. This prototype really fully utilized our Unity store asset, a UI pack that would help us create shortcuts for buttons without creating unnecessary clutter on screen and preserving an attractive aesthetic front for our application. This prototype was our conclusion to our Unity work and the final necessary tool for our presentations.

12. Conclusion and recommendations and Future Plan

Although our final product is complete and ready for presentation, we cannot ignore the many other features we eventually would plan on having incorporated into our product. These features are mentioned in our previous deliverable with our conceptual designs as well as our product diagram. We would like to add the following features:

- Library: This feature would be included inside the tools pop-up menu. This feature would allow users to go through a list of different pre-existing elements they can import into their project. This would facilitate the task of incorporating specific structures into a project. An example of this would be inserting a door that would be available inside the library(different versions of a door would exist so the user may use whichever one they prefer). The library would also preferably have prices displayed for different structures so the user can see if their modifications stay under their budget.

- Material detail: This feature would be found in the select pop-up menu. This feature allows users to display their materials on surfaces. This would work as a colour code system, so that each material is associated to a colour. Doing so avoids creating clutter on surfaces by displaying names of different materials. It also facilitates understanding projects since it is more visually appealing.
- Price Breakdown and Changes Pop-Up: These two features go hand in hand and would be found in the notepad menu. Price Breakdown would create a sort of total receipt so the user may see how much his total project costs. Unlike the library which only displays the price of individual items, this feature gives the total price breakdown of the project. The changes pop-up let's users see the "history" of what modifications were made on the project.
- Login and logout: In our future endeavours, we would like to add a login/logout feature that allows users to go in and out of their respective accounts. This would allow users to maintain their own personal accounts versus their commercial/job accounts. Having an account system also assures that there is a minimum amount of security surrounding peoples' work. For our login, we would like to not limit users to having to login on an account they make for the application, but let them link their Pellucid accounts to Google or Facebook. This allows users to have a faster sign-in experience but also makes it easier for their work to be saved onto a cloud system.

Table 11. List of Folders and Subfolders in the Final Product File

Name of Folder	Brief List of Contents
Assets	<p>Contains:</p> <ul style="list-style-type: none"> ★ All 3D Model (STEM) .fbx files ★ All Images and Videos ★ All Materials ★ All Scenes ★ All Scripts ★ All Fonts ★ All Prefabs ★ XR files ★ Purchased UI Pack ★ Imported free assets from Asset Store
Library	<p>Contains cached files. Library caching is activated by default to improve the build speed after the first checkout. You can disable Library caching in the Unity Cloud Build options if it causes problems.</p>
Logs	<p>All messages, warnings, and errors written to the Console window in the Editor is written to the log files</p>
Packages	<p>Collection of downloaded Free Packages from the Asset Store</p>

Project Settings	Collection of all project settings determined by creator
obj/Debug	The obj directory contains intermediate files generated from your source code. They are building blocks that are used in order to produce the final executable. This directory also acts as a cache between builds and allows you to do incremental compilation.

You can find all our technical project files related to Unity work on the following public github repository, accessible via this link:

<https://github.com/SarahQadri/FinalPellucid>