



MedTech
Mediterranean
Institute of Technology



ISS 496- Senior Project Final Report

FabLab Machines Discovery

In collaboration with: Orange Digital Center

Elaborated by:

Aymen HAMMAMI, Aziz MAAZOUZ, Mahdi KLOUZ, Youssef AYDI

Supervised by:

Mrs. Mariem TURKI – Orange Digital Center

Mr. Mohamed Chedli BEN YAGHLANE – Orange Digital Center

Dr. Asma AMDOUNI – SMU (South Mediterranean University), MedTech

Tunis, 2022-2023

Table of Contents

1- Project Overview	4
1.1- Introduction.....	4
1.2- Project Description	4
1.3- Problem Definition.....	4
1.4- Vulnerabilities and Existing Solutions.....	5
1.5- Proposed Solution	5
1.6- Literature Review	6
1.7- Conclusion	6
2- Product Specifications	7
2.1- Introduction.....	7
2.2- Project Management	7
2.3- Product Backlog.....	8
2.4- Sprint Backlog.....	10
2.5- Requirements Specification	12
2.6- Project Considerations	16
2.7- Conclusion	17
3- Product Design.....	18
3.1- Introduction.....	18
3.2- High-Level Design.....	18
3.2-1. Architecture Overview	18
3.2-2. Component Diagram	19
3.3- Detailed Design.....	20
3.3-1. Class Diagram	20
3.3-2. Sequence Diagram	21
3.3-3. Package Diagram.....	22
3.3-4. Deployment Diagram.....	22
3.4- UI Design	23
3.5- Conclusion	29
4- Development and Prototype Solution	30
4.2- Development Details.....	30
4.3- Development Support.....	31
4.4- Development Results	32
4.5- Conclusion.....	36
5- Tests, Results, and discussions	37

5.1 Introduction.....	37
5.2 Interactive Testing.....	37
5.3 Unit Testing.....	37
5.4 Performance Testing	38
5.5 Conclusion	39
6- General Conclusion & Future work.....	40
References.....	41

List of Figures

Figure 1: List of features included in the product backlog (from Microsoft Azure)	8
Figure 2: List of product backlog items that were completed in Sprint 1	10
Figure 3: List of product backlog items that were completed in Sprint 2	10
Figure 4: List of product backlog items that were completed in Sprint 3	11
Figure 5: List of product backlog items that were completed in Sprint 4	11
Figure 6: List of product backlog items that were completed in Sprint 5	11
Figure 7: List of product backlog items that were completed in Sprint 6	12
Figure 8: Quality Scenario for Performance	14
Figure 9: Quality Scenario for Accuracy.....	14
Figure 10: Quality Scenario for Flexibility	15
Figure 11: Use Case Diagram	15
Figure 12: Layered diagram describing the architecture of the project.....	18
Figure 13: Component diagram displaying layers of the project in more details	19
Figure 14: Class Diagram.....	20
Figure 15: Sequence Diagram of the use case "Scan Machine"	21
Figure 16: Package Diagram	22
Figure 17: Deployment Diagram.....	22
Figure 18: Design of the loading page, and the start page that follows it	23
Figure 19: Design of the first page of the onboarding pages	24
Figure 20: Design of the second page of the onboarding pages	25
Figure 21: Design of the third page of the onboarding pages.....	26
Figure 22: Design of the last page of the onboarding pages.....	27
Figure 23: Popup Types.....	28
Figure 24: Machine tags used for recognition	29
Figure 25: Augmented Reality View - 3D Printer	32
Figure 26: Augmented Reality View - Use Instruction popup was clicked	33
Figure 27: Augmented Reality View - 3D Simulation is playing.....	34
Figure 28: Augmented Reality View - CNC Router 3 Axes	35
Figure 29: Augmented Reality View - Laser Cutting Machine	36

List of Tables

Table 1: Scrum ceremonies timeline	8
Table 2: List of product backlog items	10
Table 3: Functional requirements.....	13
Table 4: Non-functional requirements	13
Table 5: List of devices used during development and testing	31
Table 6: List of unit tests	38

1- Project Overview

1.1- Introduction

This document discusses the development process of the project by going through all its important aspects: methodology, requirements, specifications, design, implementation, testing and results.

In this section, the document describes the project and defines the problem the project is trying to eradicate.

1.2- Project Description

The project, called “FabLab Machines Discovery” is an Augmented Reality (AR) application that runs on mobile devices.

The goal of the project is to allow any visitor or member of the “FabLab Space” to display essential information about machines upon capturing them using the mobile device camera.

These machines allow the users to create prototypes of their products and are critical to the success of their project ideas.

Our project offers them an opportunity to get familiar with the machines specifications and use cases before transitioning from theory to practice.

1.3- Problem Definition

Whether you are a startup or a small team with a project idea, you must begin your product development journey with a prototype that would allow you to get funding from investors, feedback from customers and a proper visualization of your idea.

This is the reason “FabLab Space” exists in the first place, giving these teams free access to machines capable of producing prototypes that would be too expensive to buy.

However, most of the time, these users are unfamiliar with the machines and lack training and expertise. They are also unsure about which machine to use, and how to abide by its specifications.

This leads them to incorrectly design their product’s 3D models, to make a wrong choice of base materials, and to end up with a low-quality prototype.

1.4- Vulnerabilities and Existing Solutions

Currently, the staff at “FabLab Space” are the only ones that can assist these users and provide them with information regarding the available machines.

Introducing all the machines and discussing their specifications each time a new member joins the “FabLab Space” is a tedious process and distracts the staff from more important tasks such as maintaining the machines, making sure the users are following security guidelines and assisting teams with practical knowledge when running the machines.

1.5- Proposed Solution

This is where our project comes in.

With “FabLab Machines Discovery,” any user can receive a virtual tour covering rich information about the important machines and tools of “FabLab Space.”

The information is displayed in the form of popups that are positioned around the recognized machine.

When clicked, these popups would display multiple kinds of information depending on the type of the popup:

- General Information
- Technical Information
- Safety Risks
- Use Instructions

Moreover, there is a special popup that would play the machine’s simulation when clicked. This simulation would visually describe the way the machine works using 3D animations.

To conclude, this helps the users become familiar with the machines beforehand and improves their choice of materials and specifications.

1.6- Literature Review

Augmented Reality (AR) is a technology that blends the virtual and real world to create an interactive experience.

According to (Carmigniani & Furht, 2011) in “Handbook Of Augmented Reality”, it integrates computer-generated information into the physical environment to enhance the user's perception of reality: Users can view virtual objects and cues in real-time, in any 3D space, and in any format.

Also, AR technology is not limited to any specific display technology and can enhance all senses, including touch, smell, and hearing. The primary goal of AR is to simplify the user's life by bringing virtual information to the immediate surroundings, resulting in an augmented perception of reality.

In summary, AR technology aims to improve the user's interaction with the real world by providing an enhanced perception of the environment.

Our team concluded that implementing Unity and Vuforia would be excellent options for developing a robust AR mobile application.

Unity is a powerful game engine that provides developers with a flexible platform for creating 3D & AR experiences and supports deploying to multiple platforms. Moreover, Unity's extensive library of 3D assets and tools makes it easy for developers to create and implement virtual objects that would be projected in the real-world environment.

Vuforia is an innovative AR software development kit (SDK) that provides developers with the necessary tools to create AR applications by providing already implemented computer vision technology. This allows applications to recognize real-world objects and track them in real-time, creating a more immersive and fluid experience for the user.

When combined, Unity and Vuforia provide developers with a powerful and flexible platform to create AR applications with ease. The integration between the two technologies is seamless, making it easy to implement Vuforia's computer vision capabilities into Unity projects.

1.7- Conclusion

In this section of the document, we looked at the project's description, its goals, and its problem-solution relationship.

2- Product Specifications

2.1-Introduction

In this section, the document will present the specifications of the project by depicting the adopted methodology, the development timeline & tasks, considerations and both functional and non-functional requirements.

2.2- Project Management

There is no doubt that any successful project will necessitate good management of time and costs and continuous teamwork & collaboration.

For the development of the project, we opted for agile approaches.

“Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches ... An agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.” (Atlassian, 2023)

Using the points below, we demonstrate that in more detail.

2.2-1. Methodology

We decided to adopt Scrum as an agile framework for the project’s development.

Scrum is widely used in the software industry and is a “lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.” (Schwaber & Sutherland, 2020)

It made sense to adopt Scrum since the project needed to change and improve continuously while considering the stakeholders’ feedback.

2.2-2. Timeline

We have divided the project into features, where each feature is composed of a set of backlog items.

We did this to have a logical representation of the project and to define its priorities.

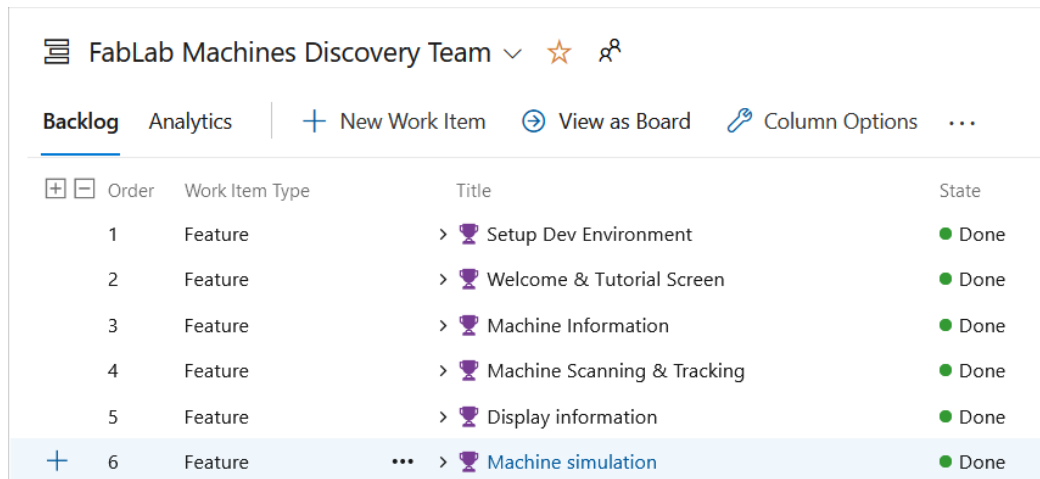
We split the development period into 6 2-week sprints.

In this subsection, we present how we conducted the Scrum ceremonies.

Sprints	From	To
Sprint 1	6 th Feb 2023	19 th Feb 2023
Sprint 2	20 th Feb 2023	5 th March 2023
Sprint 3	6 th March 2023	19 th March 2023
Sprint 4	20 th March 2023	2 nd April 2023
Sprint 5	3 rd April 2023	16 th April 2023
Sprint 6	17 th April 2023	30 th April 2023

Table 1: Scrum ceremonies timeline

2.3-Product Backlog



FabLab Machines Discovery Team				
Backlog	Analytics	+ New Work Item	View as Board	Column Options
Order	Work Item Type	Title	State	
1	Feature	> 🏆 Setup Dev Environment	● Done	
2	Feature	> 🏆 Welcome & Tutorial Screen	● Done	
3	Feature	> 🏆 Machine Information	● Done	
4	Feature	> 🏆 Machine Scanning & Tracking	● Done	
5	Feature	> 🏆 Display information	● Done	
+ 6	Feature	... > 🏆 Machine simulation	● Done	

Figure 1: List of features included in the product backlog (from Microsoft Azure)

In the figure above, we can see the list of features that our team implemented during the development of this project.

Each feature is parent to multiple tasks that were assigned to team members during the sprints.

Item Number	Backlog Item	Backlog Feature
1	Create GitHub Repository	Setup Dev Environment
2	Setup Unity 2021.3.18f1	
3	Integrate Vuforia within Unity	
4	Configure Vuforia Target Manager	
5	Add initial Unity & Vuforia initialization code	
6	Import TMP to the project's packages	
7	Import DOTween to the project's packages	
8	Add popup information of 3D Printer in English	Machine Information
9	Add popup information of 3D Printer in French	
10	Add popup information of Laser Cutter in English	
11	Add popup information of Laser Cutter in French	
12	Add popup information of CNC Router in English	
13	Add popup information of CNC Router in French	
14	Configure Unity's localization package and database	
15	Gather machine information and divide them into popup categories	
16	Design machine tags and upload them in Vuforia Target Manager	Machine Scanning & Tracking
17	Test tracking with Image Targets and Multi-Image Targets	
18	Measure machines' dimensions and position the tags	
19	Implement script that handles missing camera permission	
20	Implement link between the Onboarding scene and the Augmented Reality scene	Welcome & Tutorial Screen
21	Design the Onboarding panels/views	
22	Add French localization for the Onboarding panels	
23	Implement Translation animations (Vector3)	
24	Implement the UI of panel #1 in Unity	
25	Implement the UI of panel #2 in Unity	
26	Implement the UI of panel #3 in Unity	
27	Implement the UI of panel #4 in Unity	
28	Implement the UI of panel #5 in Unity	
29	Implement the UI of panel #6 in Unity	
30	Implement swiping between the panels	
31	Implement script for fade animations (In & Out)	
32	Implement script for wipe animations (In & Out)	
33	Design UI that displays content of a popup when it's clicked	Display Information
34	Implement the Popup component	
35	Implement the PopupInfoUI component	
36	Design UI for the camera scene	
37	Implement Pop Animation (Vector3)	
38	Test and adjust the positions of R3N1 popups	
39	Test and adjust the positions of G1S6 popups	
40	Test and adjust the positions of C6N9 popups	
41	Implement the simulation of 3D Printer	Machine Simulation
42	Implement the simulation of Laser Cutter	

43	Implement the simulation of CNC Router	Machine Simulation
44	Implement script for simulation popup that plays the 3D animation when clicked	
45	Add simulation popup type to original popup script	

Table 2: List of product backlog items

2.4-Sprint Backlog

Below are the respective sprint boards for each sprint the team had.

The figure below belongs to the summary of the tasks done during Sprint 1, which started on February 6th and ended on February 19th.

FabLab Machines Discovery Team					February 6 - February 19 14 work days	
Taskboard	Backlog	Capacity	Analytics	+ New Work Item	Column Options	...
Sprint 1						
Order	Title	State	Assigned To	Rema...		
1	Create Github repository	Done				
2	Install Unity 2021.3.18f1	Done				
3	Design welcome screen view	Done				
4	Setup Unity to work with Git	Done				
5	Design the tutorial steps / onboarding views	Done				
6	Implement the UI of view #3 in Unity	Done	Mahdi Klouz			
7	Setup Vuforia within Unity	Done				
8	Implement the UI of view #4 in Unity	Done	Youssef Aydi			
9	Implement the UI of view #1 & #2 in Unity	Done	Aziz Maazouz			
10	Implement navigation between onboarding views	Done	Aymen Hamm...			
11	Implement base UI animation script	Done	Aymen Hamm...			
12	Implement script for fade animations (In & Out) (Image & Te...	Done	Aymen Hamm...			
13	Implement script for wipe animations (In & Out) (Image & T...	Done	Aymen Hamm...			
14	Setup Vuforia Target Manager	Done				
15	Add initial Unity & Vuforia code	Done				
16	Add TMP and DOTween to the project packages	Done				
17	Implement the UI of view #5 in Unity	Done	Aymen Hamm...			

Figure 2: List of product backlog items that were completed in Sprint 1

The figure below belongs to the summary of the tasks done during Sprint 2, which started on February 20th and ended on March 5th.

FabLab Machines Discovery Team					February 20 - March 5 14 work days	
Taskboard	Backlog	Capacity	Analytics	+ New Work Item	Column Options	...
Sprint 2						
Order	Title	State	Assigned To	Rema...		
1	Implement link between onboarding scene and vuforia ...	Done	Aymen Hamm...			
2	Configure localization in the project (EN/FR)	Done	Aymen Hamm...			
3	Add French localization for onboarding scene	Done	Youssef Aydi			
4	Gather machine information and sepearte them by popup t...	Done	Aziz Maazouz			
5	Test tracking with image target and multi-image target	Done				

Figure 3: List of product backlog items that were completed in Sprint 2

The figure below belongs to the summary of the tasks done during Sprint 3, which started on March 6th and ended on March 19th.

FabLab Machines Discovery Team ☆ 🔍 March 6 - March 19
14 work days

Taskboard **Backlog** Capacity Analytics + New Work Item 🔗 Column Options ... 🔍 Sprint 3

+ 📄 Order	Title	State	Assigned To	Rema...
+ 1	Add popups information of Raise3D N1 in English to lo...	Done	Aymen Hamm...	
2	Add popup information of Raise3D N1 in French to localizati...	Done	Aymen Hamm...	
3	Add popups information of Laser Cutter in English to localiz...	Done	Youssef Aydi	
4	Add popups information of Laser Cutter in French to localiza...	Done	Youssef Aydi	
5	Measure dimensions of machines	Done		
6	Design machine tags for vuforia target manager	Done	Aymen Hamm...	
7	Design UI that displays content of popup when it's clicked	Done	Aziz Maazouz	
8	Implement the UI of component that displays the popup's c...	Done	Aziz Maazouz	
9	Implement Popup component	Done	Aymen Hamm...	
10	Implement PopupInfoUI component	Done	Aymen Hamm...	
11	Design UI for the camera scene	Done	Mahdi Klouz	
12	Add popups information of CNC Router in English to localiz...	Done	Aziz Maazouz	
13	Add popups information of CNC Router in French to localiza...	Done	Aziz Maazouz	
14	Implement Translation animations (C#, Vector3)	Done	Youssef Aydi	
15	Implement Scale animations (C#, Vector3)	Done	Youssef Aydi	
16	Implement Pop animations (C#, Vector3)	Done	Youssef Aydi	

Figure 4: List of product backlog items that were completed in Sprint 3

The figure below belongs to the summary of the tasks done during Sprint 4, which started on March 20th and ended on April 2nd.

FabLab Machines Discovery Team ☆ 🔍 March 20 - April 2
14 work days

Taskboard **Backlog** Capacity Analytics + New Work Item 🔗 Column Options ... 🔍 Sprint 4

+ 📄 Order	Title	State	Assigned To	Rema...
+ 1	Implement the 3D printer simulation	Done	Mahdi Klouz	
2	Design simulation popup icon and colors	Done	Aymen Hamm...	
3	Implement script for simulation popup that plays the animat...	Done	Youssef Aydi	
4	Add simulation popup type to scripts	Done	Aymen Hamm...	
5	Implement script that handles missing permission error	Done	Aymen Hamm...	

Figure 5: List of product backlog items that were completed in Sprint 4

The figure below belongs to the summary of the tasks done during Sprint 1, which started on April 3rd and ended on April 16th.

FabLab Machines Discovery Team ☆ 🔍 April 3 - April 16
14 work days

Taskboard **Backlog** Capacity Analytics + New Work Item 🔗 Column Options ... 🔍 Sprint 5

+ 📄 Order	Title	State	Assigned To	Rema...
+ 1	Test R3N1 popups and adjust positions accordingly	Done		
2	Test G1S6 popups and adjust positions accordingly	Done		
3	Test C6N9 popups and adjust positions accordingly	Done		

Figure 6: List of product backlog items that were completed in Sprint 5

The figure below belongs to the summary of the tasks done during Sprint 1, which started on April 17th and ended on April 30th.

Order	Title	State	Assigned To	Rema...
1	Implement CNC Router 3 Axis simulation	Done	Ayman Hamm...	
2	Implement the laser cutter simulation animation	Done	Ayman Hamm...	

Figure 7: List of product backlog items that were completed in Sprint 6

2.5-Requirements Specification

2.5-1. Functional Requirements

Requirement	Description	Significance
Machine scanning	Whenever the user is using the application's camera, the system should be able to capture and recognize the machines in real-time.	<p>The project will use the Vuforia Engine within Unity3D to add advanced computer vision functionality to the application.</p> <p>This will allow it to recognize real-world images and objects.</p> <p>The project will use the Vuforia Target Manager as a database for the machine images, which will be checked each time an object.</p>
Machine tracking	The system will be able to track a recognized machine in real-time and adapt the position of the projected information.	<p>The project will use the Vuforia Engine within Unity3D to add advanced computer vision functionality to the application.</p> <p>This will allow it to be aware of real-world spaces and react to the movement of the camera and/or the tracked image.</p>
Extraction of relevant information	The system will find the relevant information to be displayed upon recognizing a machine.	<p>The project will use independent data containers internally implemented within Unity called Scriptable Objects.</p> <p>The Scriptable Objects will store relevant information about the machines and will be extracted using an id or</p>

		keyword unique to the recognized machine.
Projection of information	Upon recognizing a machine, the system will display popups that can be clicked. Once clicked, the popup would display relevant information about the machine.	<p>The project will use the Vuforia Engine within Unity3D to add advanced computer vision functionality to the application.</p> <p>The User Interface of the popup will be implemented as a child component to Vuforia's Image Target feature, so that the content is only shown when a valid target is tracked.</p>
Display of machine's simulation	One of the popups that could be clicked would play an animation instead. This animation would visually simulate how the machine works.	The project will use Unity's built-in Animator component and 3D models that may be available on the Unity's Asset Store to create robust animations for each machine.
User clicks handling	The system can react to clicks on buttons and execute related actions properly.	The project will use Unity's built-in Button component and click events within the Vuforia Engine to react when a press on the button occurs.

Table 3: Functional requirements

2.5-2. Non-functional Requirements

Quality	Description
Performance	The system will process instructions quickly without using many resources.
Usability	The system will make it easy for the user to perform any specific task with minimal complications.
Accuracy	The system will not mistake a scanned machine for another one, always displaying the correct information.
Flexibility	The system will run smoothly on mobile devices regardless of screen resolution and will be open to the addition of new features.

Table 4: Non-functional requirements

To better describe the desired non-functional requirements above, we designed a few quality scenarios:

This scenario focuses on the performance aspect of the project, and describes how the application should load fast, regardless of the device used.

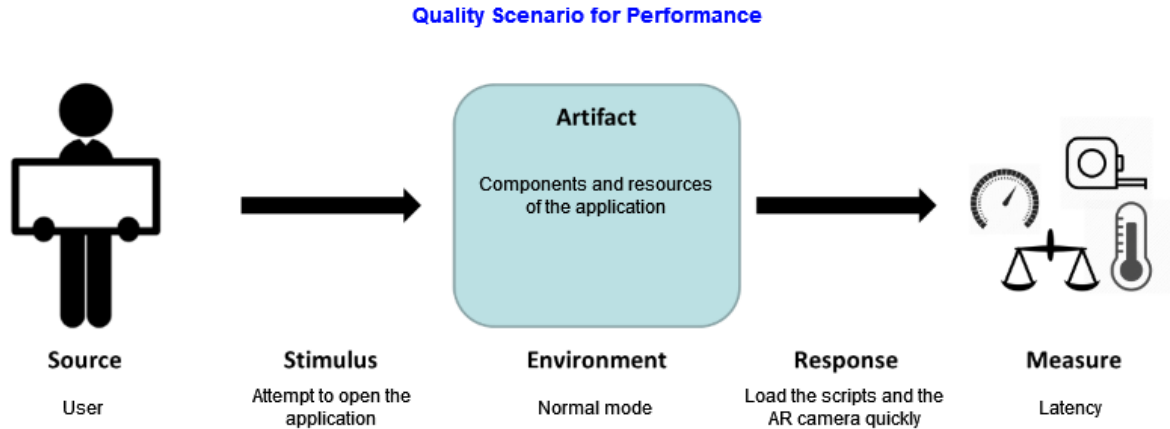


Figure 8: Quality Scenario for Performance

This scenario focuses on the accuracy aspect of the project, and describes how the application should never take a machine for another one. Recognizing the tags should be accurate and precise.

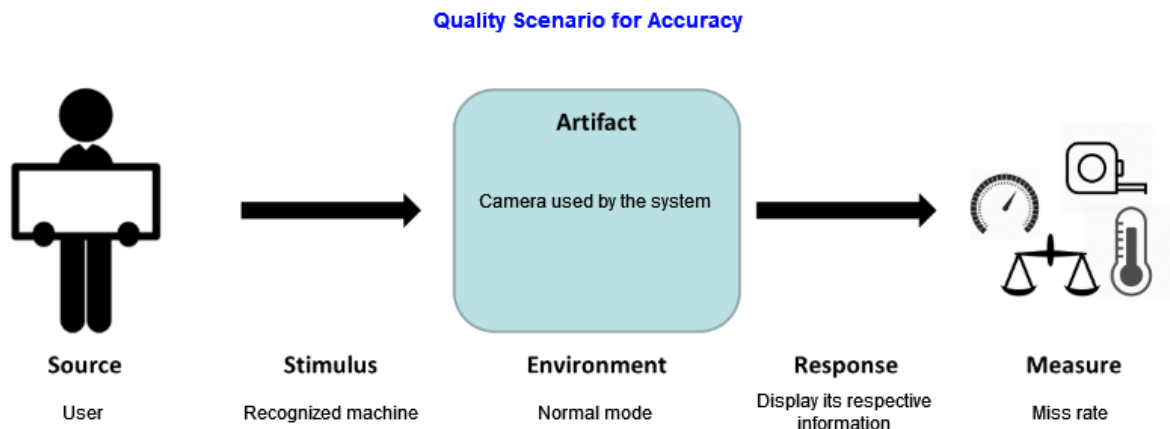


Figure 9: Quality Scenario for Accuracy

This scenario focuses on the flexibility aspect of the project, and describes how the user interface of the application should adapt to the different screen resolutions.

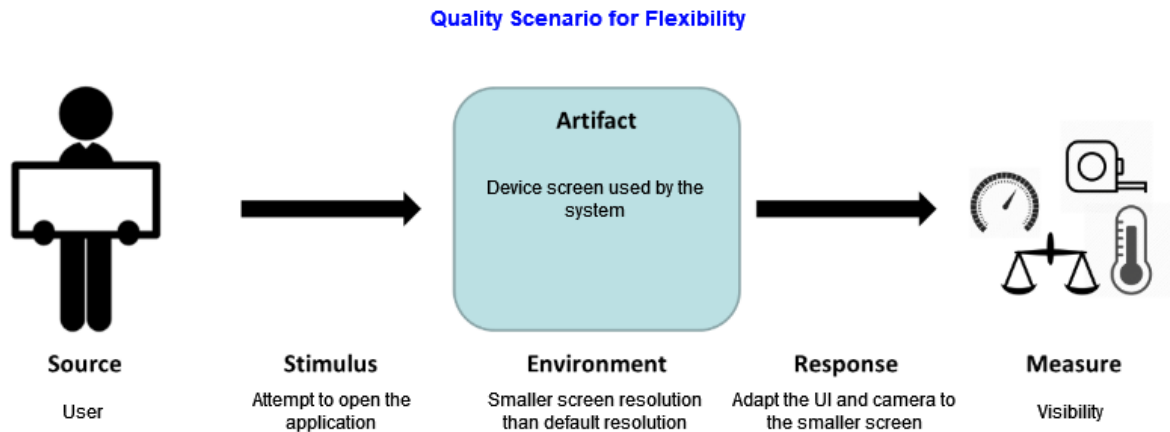


Figure 10: Quality Scenario for Flexibility

2.5-3. Use Case Diagram

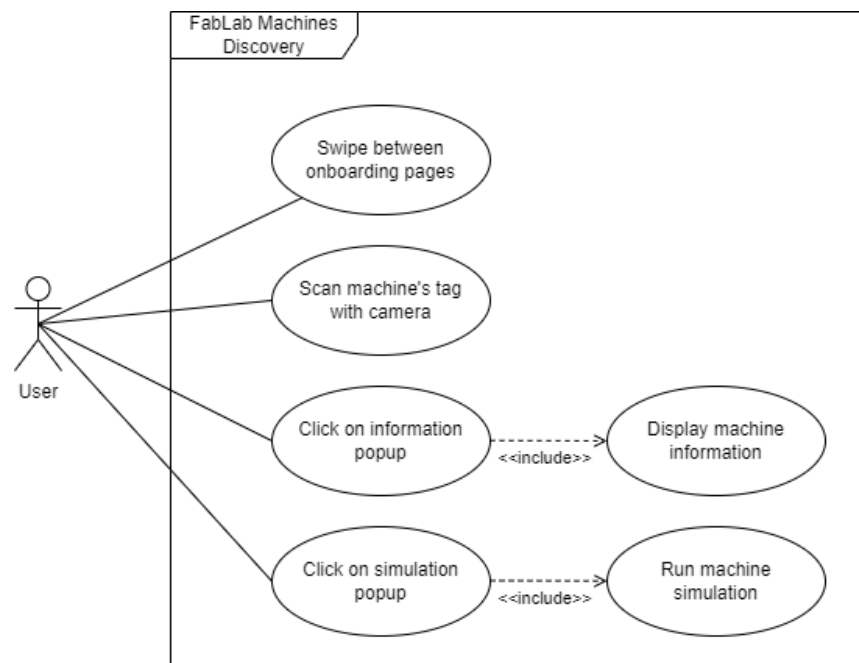


Figure 11: Use Case Diagram

This use case diagram describes the most important tasks that the user of the application can make:

- **Swipe between onboarding pages:** this is the first scene that the user encounters, where they can go through pages that work as a tutorial for the application.
- **Scan machine's tag with camera:** the tag is used to recognize the machine. Using the camera, the user can scan it inside the application.
- **Click on information popup:** when the machine is recognized, popups will appear around the machine. When clicked, these popups display their information.
- **Click on simulation popup:** a special kind of popup. When clicked, it will play a 3D animation that describes how the machine works

2.6-Project Considerations

2.6-1. Project Constraints

- **Portability:** The project will be delivered to operate on mobile devices.
- **Time:** The project will be developed in 12 weeks (about 3 months).
- **Experience:** Some team members working on the project are not familiar with Unity3D, C# or Vuforia Engine and will have to spend time learning these technical tools.

2.6-2. Project Limitations

The application will be implemented using the following technologies:

- **Unity 3D:** popular engine with many built-in tools capable of delivering high quality mobile applications.
- **Vuforia SDK (Software Development Kit):** software development kit that adds advanced computer vision functionality to any augmented reality application and is usable within Unity 3D.
- **C#:** the programming language used to implement scripts in Unity 3D.
- **Git & GitHub:** source control tools that allow collaboration during development.

2.6-3. Project Delimitations

- During the development of the project, we did not have easy access to the “FabLab Space” machines. This was due to two reasons. The first one is that we had to find means of transportation to the location of the machines since we had to be present there to take pictures of them and to assess our application. The second reason is that we had to schedule meetings with the space’s manager, hence we could not access the space at any time.
- During the development of the project, the stakeholders were usually unavailable, and so we only had one meeting with them every two weeks.

2.6-4. Project Standards

Project standards describe conventions and rules the team should abide by when developing the project. These standards are important since they try to improve aspects of project management & design, and enhance the quality of the final product.

We decided to consider the following standards:

ISO/IEC/IEEE 42010:2011 - Systems and software engineering - Architecture description

“ISO/IEC/IEEE 42010:2011 addresses the creation, analysis and sustainment of architectures of systems through the use of architecture descriptions. A conceptual model of architecture description is established. The required contents of an architecture description are specified. Architecture viewpoints, architecture frameworks and architecture description languages are introduced for codifying conventions and common practices of architecture description. The required content of architecture viewpoints, architecture frameworks and architecture description languages is specified. Annexes provide the motivation and background for key concepts and terminology and examples of applying ISO/IEC/IEEE 42010:2011.” (ISO, 2011)

ISO/IEC/IEEE 15288:2015 - Systems and software engineering - System life cycle processes

“ISO/IEC/IEEE 15288:2015 establishes a common framework of process descriptions for describing the life cycle of systems created by humans. It defines a set of processes and associated terminology from an engineering viewpoint. These processes can be applied at any level in the hierarchy of a system's structure. Selected sets of these processes can be applied throughout the life cycle for managing and performing the stages of a system's life cycle. This is accomplished through the involvement of all stakeholders, with the goal of achieving customer satisfaction.” (ISO/IEC/IEEE, 2015)

IEEE 1008-1987 IEEE Standard for Software Unit Testing

“An integrated approach to systematic and documented unit testing is defined. It uses unit design and implementation information, in addition to unit requirements, to determine the testing's completeness. The testing process described is composed of a hierarchy of phases, activities, and tasks and defines a minimum set of tasks for each activity. The standard can be applied to the unit testing of any digital computer software or firmware and to the testing of both newly developed and modified units.” (IEEE, 2009)

2.7-Conclusion

In this section of the document, we looked at aspects of the project management, requirements, and a set of considerations.

3- Product Design

3.1- Introduction

In this section, the document will describe the project's architecture and design starting with the high-level design, followed by more detailed architectural aspects and user interface design.

3.2- High-Level Design

3.2-1. Architecture Overview

The architecture style that the project adopted is the layered style, more specifically:

- **Presentation Layer:** has the user interface, camera preview and augmented reality graphics. Unity components are used to display the camera view and the user interface.
- **Business Layer:** responsible for processing images provided by the camera to try and recognize machines and rendering their respective information and popups in augmented reality. Vuforia Engine provides these advanced vision and image features.
- **Data Layer:** stores the images of machines that should be recognized and their respective information. Vuforia Target Manager facilitates storing the data of these machines.

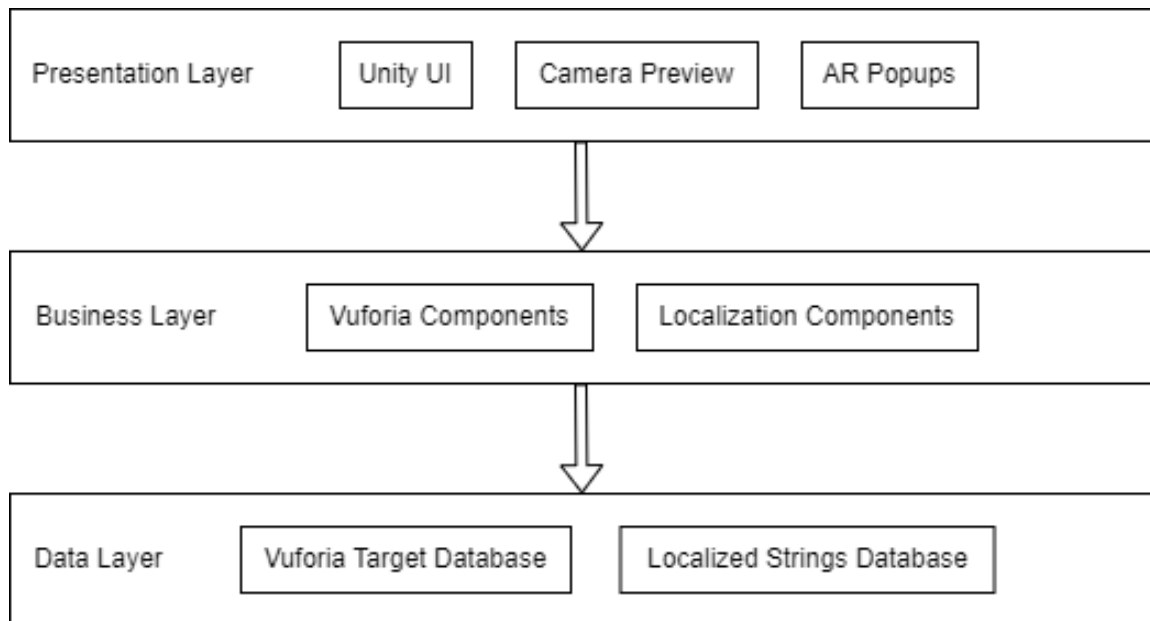


Figure 12: Layered diagram describing the architecture of the project

3.2-2. Component Diagram

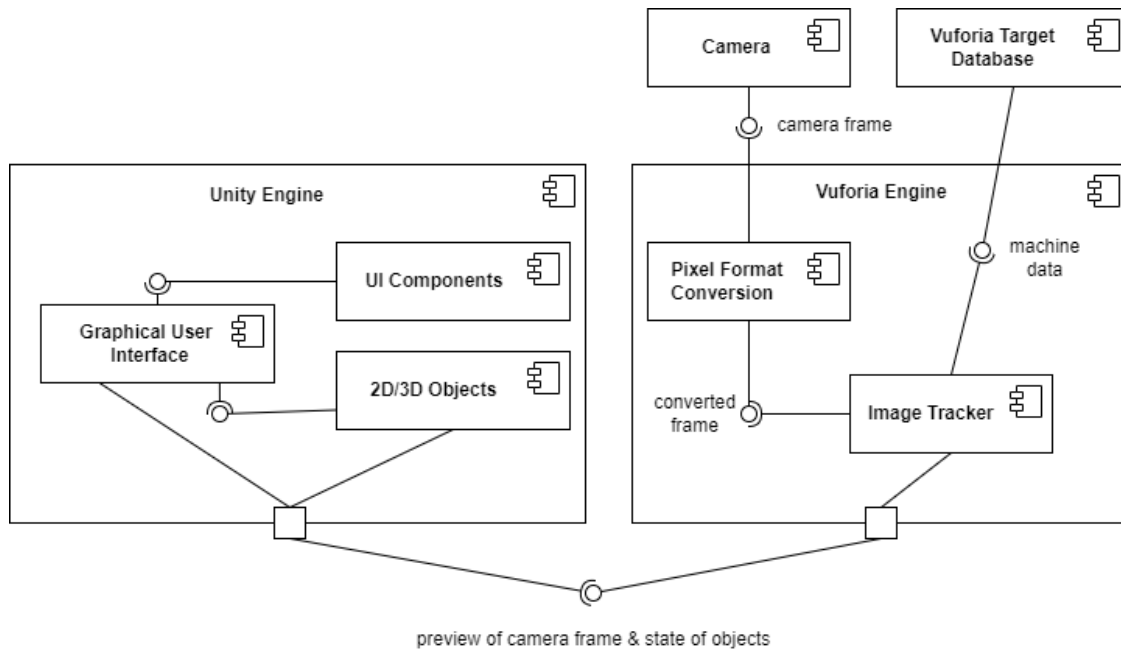


Figure 13: Component diagram displaying layers of the project in more details

The purpose of a component diagram is to show the relationship between different components in an application.

In this case, we used the component diagram to better explain how Vuforia's augmented reality features work hand in hand with Unity's 3D and UI capabilities to achieve a robust AR experience.

As you can see on the top right of the diagram, the camera provides Vuforia with the current camera frame. This frame will then be converted to pixel format, and compared with the existing tags in Vuforia Target Database. Vuforia will then communicate its findings to Unity.

If a machine was recognized, Unity would display its respective popups, UI, and 3D animations.

3.3- Detailed Design

3.3-1. Class Diagram

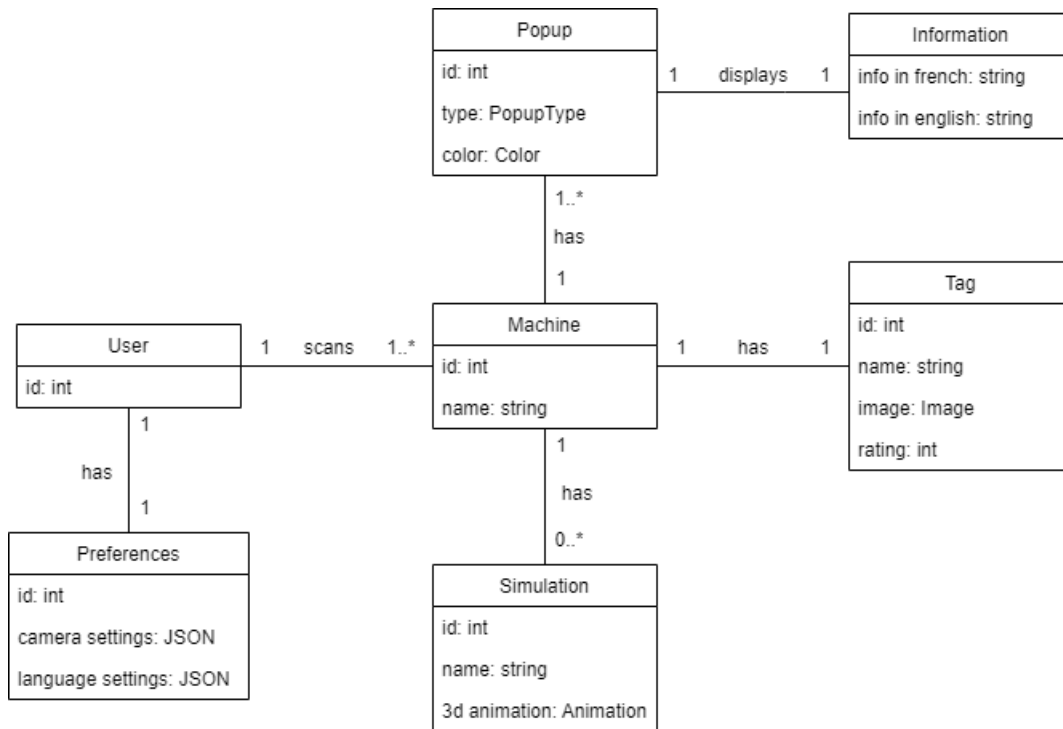


Figure 14: Class Diagram

This class diagram is used for general conceptual modeling of the structure of the application.

The most important part of the diagram is the ‘Machine’ class, as the main functionality of the application revolves around recognizing machines.

Every machine has a unique tag that is used for its recognition. This is displayed through the one-to-one relationship between ‘Machine’ and ‘Tag’.

When a machine is recognized, its popups would appear around it in the augmented reality space. This is displayed through the one-to-many relationship between ‘Machine’ and ‘Popup’.

Some machines may also have a simulation that describes how they work.

3.3-2. Sequence Diagram

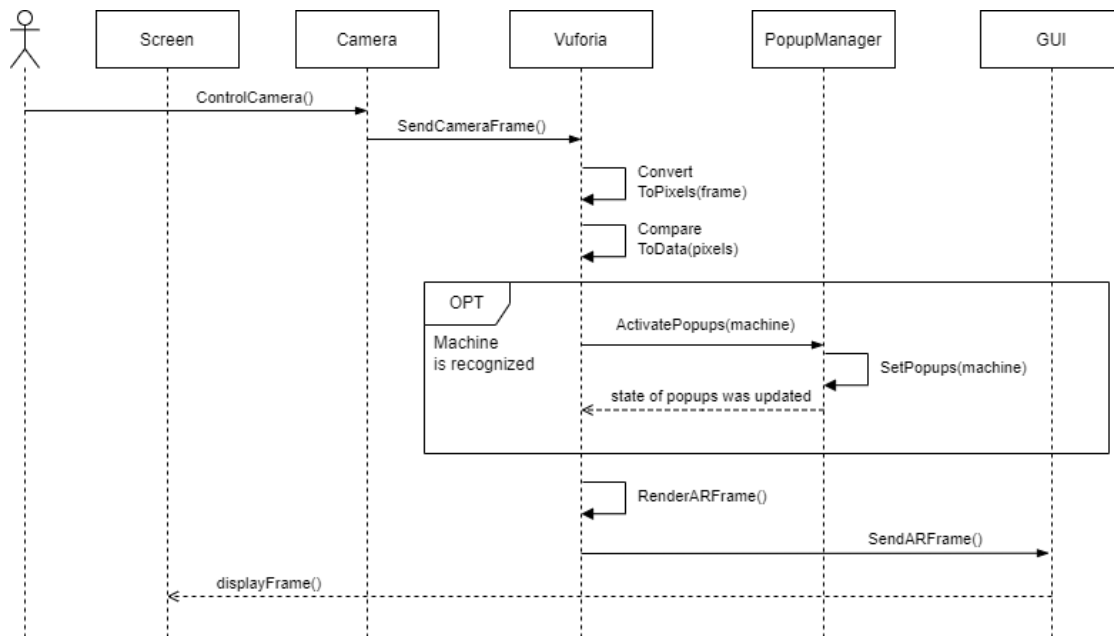


Figure 15: Sequence Diagram of the use case "Scan Machine"

This sequence diagram describes how, and in what order, scanning and recognizing a machine works.

It starts with the user controlling the camera through the application. Vuforia will retrieve the current camera frame and convert it into pixel format. The result will then be compared to the existing targets in the database.

If a machine is present in the database, in other words, if a machine is recognized, the popup manager will display the machine's popups around the machine in the augmented reality space.

Unity's Graphical User Interface (GUI) will then render both Vuforia's generated camera frame and the activated popups, if any, and display the frame on the screen.

3.3-3. Package Diagram

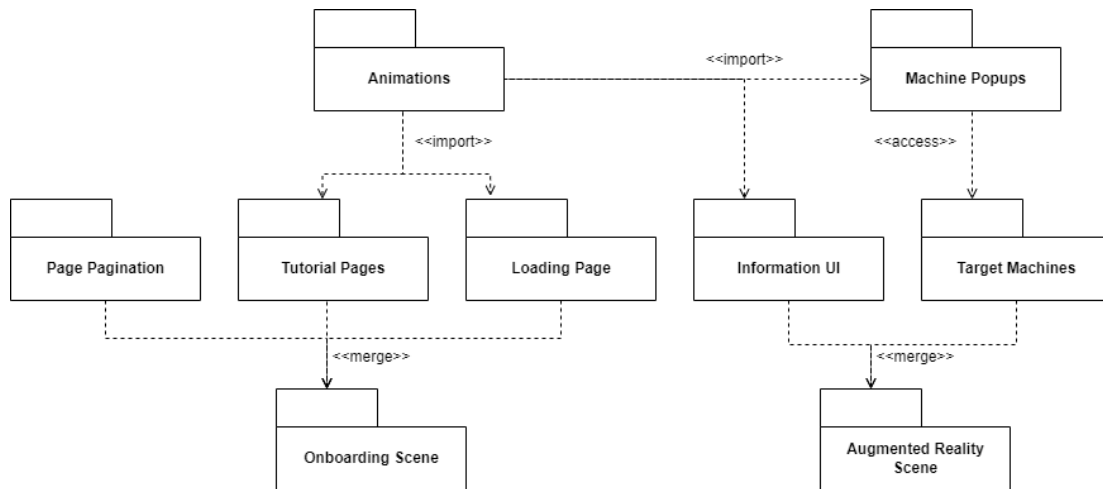


Figure 16: Package Diagram

“Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages. A package is a grouping of related UML elements, such as diagrams, documents, classes, or even other packages.” (LucidChart)

3.3-4. Deployment Diagram

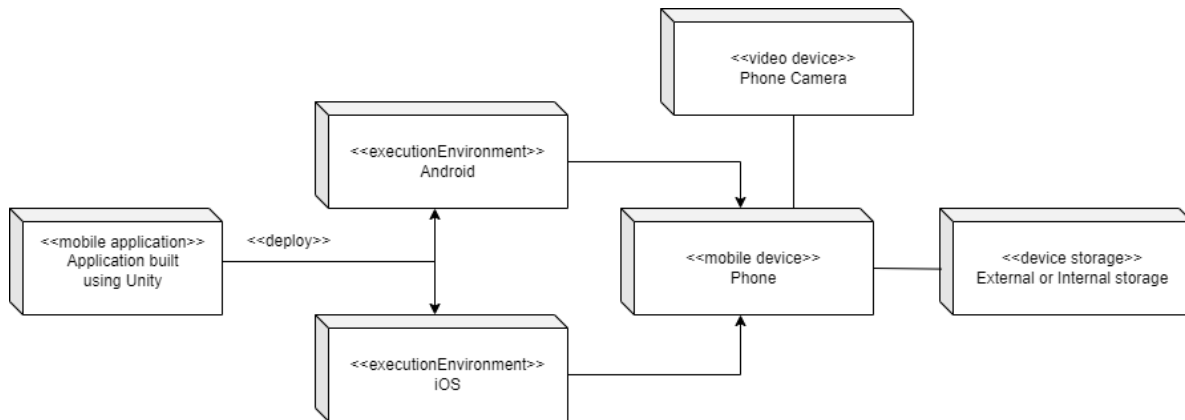


Figure 17: Deployment Diagram

As you can see in this deployment diagram, the application would be built using Unity’s own tool.

The application would be built for both Android and iOS, since one of your quality attributes is to make the application as flexible as possible.

The project can then be installed on a mobile device, where it will require access to the device’s camera and storage space.

3.4- UI Design

In this section, we will showcase the user interface of the project.

We will start with the onboarding pages that the user must go through the first time they open the application.



Figure 18: Design of the loading page, and the start page that follows it

The designs above are the first pages the user would see when opening the application for the first time.

On the left, is the loading page that appears initially when the application is still loading the scene.

Once the loading finishes, the user is greeted with the page on the right.

Clicking on the “El Tutorial” on the bottom of the page will make the user transition to the page below.



Figure 19: Design of the first page of the onboarding pages

The design above is the first page of the onboarding pages. The onboarding pages serve to inform the user about the purpose of the application and give useful hints as to how to use the application.

This page describes the purpose of the project briefly.

On the bottom, a simple pagination component is displayed which tracks the currently visible page to help the user swipe between them.

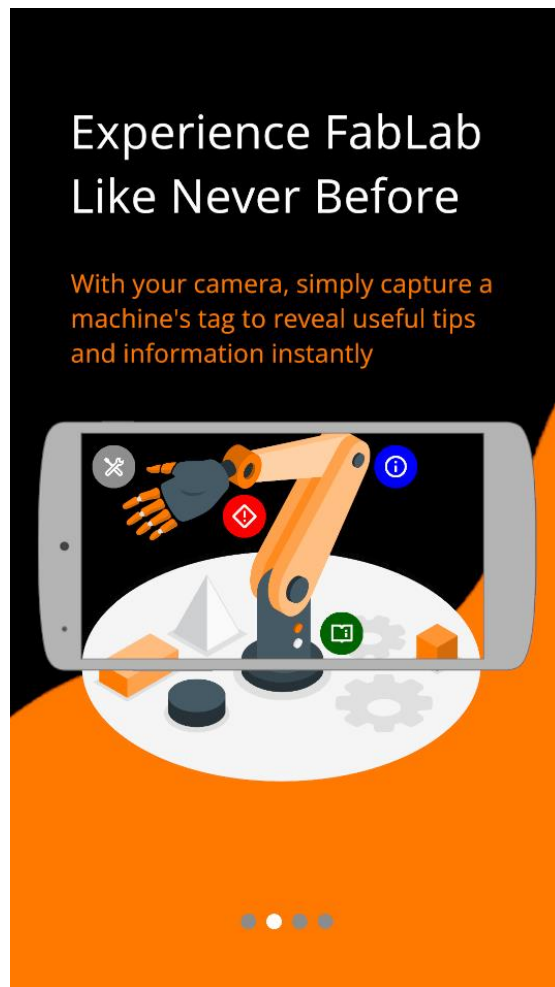


Figure 20: Design of the second page of the onboarding pages

The design above is the second page of the onboarding pages. When the user swipes to the right, they will transition to this page from the first page.

This page describes the way the application projects popups on the machine.

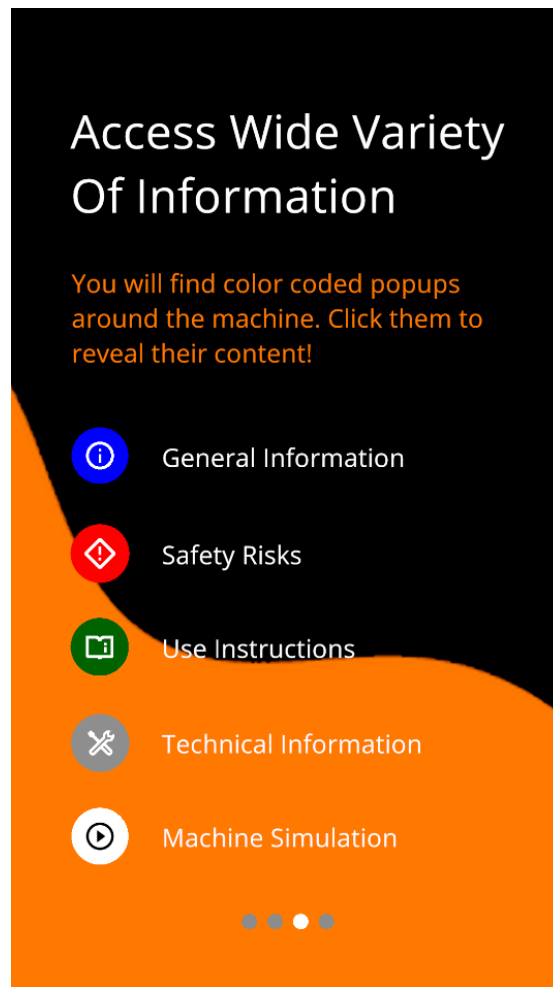


Figure 21: Design of the third page of the onboarding pages

The design above is the third page of the onboarding pages. When the user swipes to the right, they will transition to this page from the second page.

This page offers information about the color-coded popups and the topic they reference.

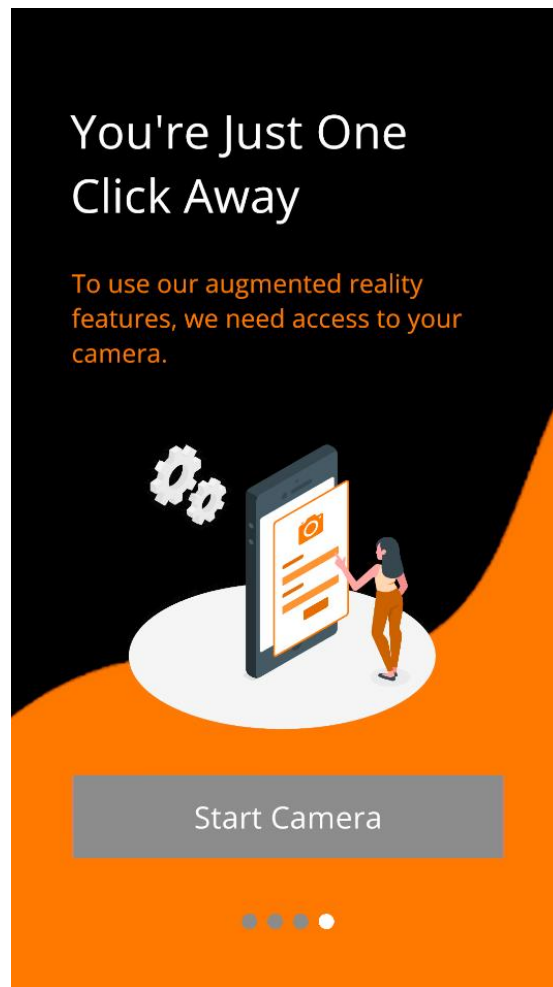


Figure 22: Design of the last page of the onboarding pages

The design above is the last page of the onboarding pages. When the user swipes to the right, they will transition to this page from the third page.

This page informs the user that the application requires permission to access the camera and includes a 'Start Camera' button on the bottom of the page, which would load the camera and start the augmented reality experience.

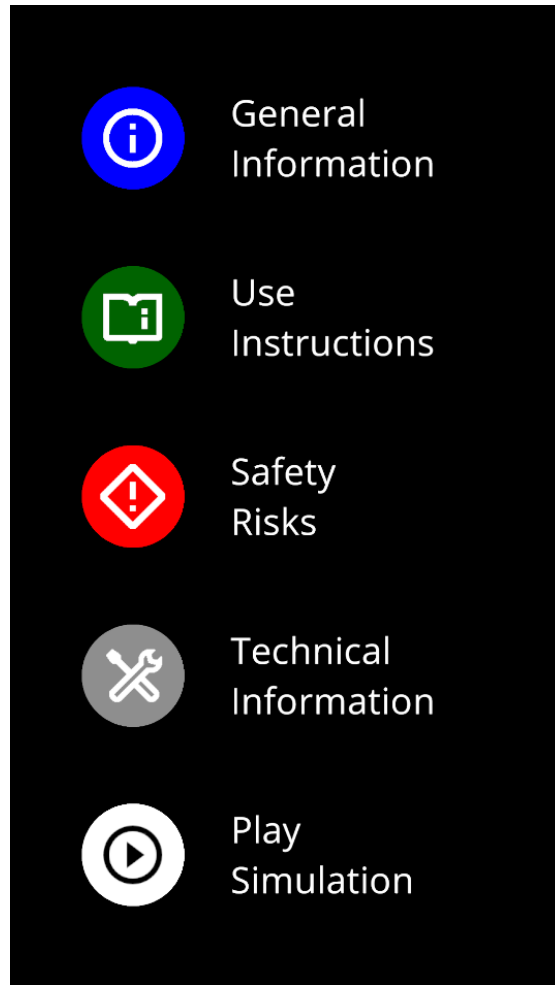


Figure 23: Popup Types

One more important part of the project is the set of popups that are going to be displayed around the recognized machines.

The figure above displays the several types of popups that are implemented within the application.



Figure 24: Machine tags used for recognition

Finally, this application would not work without the tags that make recognizing the different machines possible.

The design of the tags is minimalistic: It uses the initials of the machine and a randomly assigned color.

However, this is extremely beneficial for the computer vision functionality and for creating a good recognizable target.

3.5- Conclusion

In this section, the document depicted the project's architecture and design using multiple UML diagrams and screenshots of the UI design.

4- Development and Prototype Solution

4.1- Introduction

In this section, the document will describe the project's development by depicting the technical details and support behind the development process.

4.2- Development Details

To abide by the requirements of the project, we needed to define a set of technologies that would help us deliver an Augmented Reality experience, with a good-looking user interface in accordance with the project deadline.

Although developing the application is possible without relying on an engine, we chose to work with an engine called **Unity** for the following reasons:

- Unity offers cross-platform development tools and support, which would allow us to deploy the application on both Android and iOS.
- Unity has rich documentation and multiple tutorials available online, which makes it easier to implement features.
- Unity includes multiple tools, packages and assets that help develop high quality 3D, UI, and AR content at a faster rate.

We also decided to use **Vuforia**, an augmented reality framework, instead of the other available options such as Unity's AR Foundation, Pikkard AR because of the following reasons:

- Vuforia can be integrated as a package within Unity and is quite easy to configure.
- Vuforia Target Manager is a great tool to create a database for the machine tags, and to evaluate the usability of each tag individually.
- Vuforia offers essential components for augmented reality that are already implemented and can be customized through inheritance.

4.3- Development Support

Since the application is to be deployed to mobile devices, we needed to occasionally run and test the application of multiple mobile devices (both iOS and Android) to make sure that the user interface was flexible and adjusted properly to the screen resolution and to the device's selected language (The application includes French localization, and is set to English as default).

With that in mind, we used the following phones to proceed with the development of the application:

Phone Model	Operating System	Screen Resolution	Camera Quality	Memory Specs	CPU Specs
Samsung Galaxy A8	Android 9	1080 x 2220 pixels	16 MP 1080px	4GB RAM	Octa-core (2x2.2 GHz & 6x1.6 GHz)
Apple iPhone 13	iOS 16	1170 x 2532 pixels	12 MP 2016px	4GB RAM	Hexa-core (2x3.23 GHz & 4x1.82 GHz)
Samsung Galaxy M11	Android 12	720 x 1560 pixels	13 MP 1080px	3GB RAM	Octa-core 1.8 GHz

Table 5: List of devices used during development and testing

4.4- Development Results

Other than the onboarding pages that were displayed in Chapter 3.4 (UI Design), the application also includes the main feature: a robust augmented reality experience that helps the users become more familiar with the machines.

In the figure below, we look at the example of the **3D Printer** being recognized through its tag 'R3N1' colored in Orange.



Figure 25: Augmented Reality View - 3D Printer

Around the machine, several color-coded popups are positioned waiting to be clicked by the user. **This concept applies to all the machines that are recognizable by the application.**

When a popup is clicked, the UI at the bottom of the screen displays the information stored within it, and updates its theme to match the popup's color.



Figure 26: Augmented Reality View - Use Instruction popup was clicked

The white popup, however, is special. When clicked, it plays a 3D simulation of the machine that visually describes the way machine works.



Figure 27: Augmented Reality View - 3D Simulation is playing

Other machines that the application can recognize are the **Laser Cutting Machine** and the **CNC Router 3-Axes Machine**.

Below, you can see how the popups are positioned around these machines (to the left), as well as their unique 3D simulation (to the right).

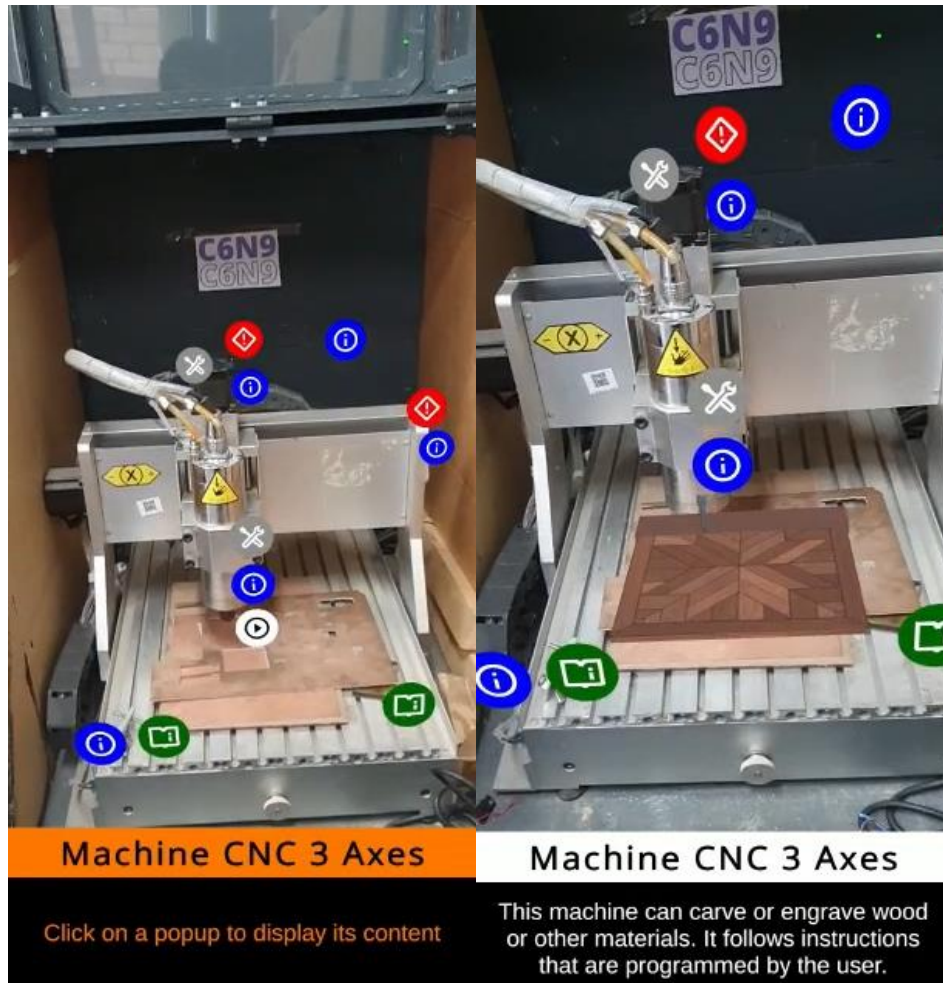


Figure 28: Augmented Reality View - CNC Router 3 Axes



Figure 29: Augmented Reality View - Laser Cutting Machine

4.5- Conclusion

In this section, the document discussed the project's development details and support, and the reasons for choosing the used tech stack.

5- Tests, Results, and discussions

5.1 Introduction

In this chapter, the document will discuss the testing phase of the project, which includes interactive testing, unit testing and performance testing.

5.2 Interactive Testing

During the project's development, we needed to ensure that the user interface was working properly on mobile devices independently of the screen resolution.

To do so, we used a device simulator to evaluate the flexibility of the user interface.

“The Device Simulator is a Unity Editor feature that simulates how certain aspects of your application will appear and behave on a mobile device.” (Unity, Device Simulator, 2021).

We also had to continuously build and deploy the application to our own mobile devices for further testing, since some errors would not occur inside the device simulator. One of these errors is lacking the permission to access the camera, which our augmented reality features rely on.

Testing during development allowed us to quickly oversee these problems and to make sure the user interface was set up correctly.

5.3 Unit Testing

The ‘Unity Test Framework’ package was used to efficiently assess the application.

“The Unity Test Framework package (formerly the “Unity Test Runner”) is a tool that allows you to test your code in both Edit mode and Play mode, and on target platforms such as Standalone, Android, or iOS.” (Unity, Unity - Manual: Unit Testing, 2023)

Below are some of the unit tests that we executed while developing the application:

Functionality Being Tested	Expected Outcome	Actual Outcome	Solution
Swipe right when on the last onboarding page	No swipe should happen since it is the last page	Swipe occurred and pagination failed	Add boundary check when swiping right on last page
Swipe left when on the first onboarding page	No swipe should happen since it is the first page	Swipe occurred and pagination failed	Add boundary check when swiping left on last page

Pagination: update pages' dots colors	The current page will have its dot colored in white, while the other inactive pages' dots would be grey	Once a dot becomes white, it does not change back to grey when its page is no longer the current one	Add a loop that changes all inactive pages' dots back to grey after a swipe happens
Load Augmented Reality scene when 'Start Camera' button is clicked	Load the AR scene while displaying a simple loading screen	The loading screen does not always appear	Change the loading function from synchronous to asynchronous
Text fade in animation	The opacity of the text changes from 0 to 1 over time	The opacity does not change	Correct value of initial opacity
Display of information when a popup is clicked	All popups should display their respective information properly	Some popups threw an error when clicked	Some popups were not assigned information text, and this needed to be corrected
Recognize the machine tag of Laser Cutter	Recognize it correctly and display the relevant popups	Wrong popups of another machine were displayed	Correct the reference of the machine that the tag was pointing towards

Table 6: List of unit tests

5.4 Performance Testing

Since we wanted to ensure that the application is flexible and usable on multiple mobile devices regardless of the operating system or the platform used, we had to optimize the source code and execute performance tests to be able to check if a component is using resources heavily or slowing the whole application down.

This was achieved using Unity's built-in Profiler tool.

"The Unity Profiler is a tool you can use to get performance information about your application. You can connect it to devices on your network or devices connected to your machine to test how your application runs on your intended release platform. You can also run it in the Editor to get an overview of resource allocation while you are developing your application." (Unity, Unity - Manual: Profiler overview, 2023)

One performance issue that we faced was the visible stutter on the second page of the onboarding pages, where the animations would not play smoothly, and the application would freeze for up to one second.

This was not noticeable inside Unity's engine since computers have more processing power than mobile devices, but after some testing, we noticed most of the mobile devices would suffer a loss in performance on that specific page.

At first, we thought this was due to the number of animations that we implemented on that page, however the code for the animations was optimized and should not have been causing any performance problems.

With the help of Unity's Profiler, we could identify the real source of the problem: The application used to render the second, third, and last page simultaneously once the 'El Tutorial' button was clicked.

Enabling all the components and objects at the same time resulted in an overhead which impacted performance negatively.

This was the reason we added the loading page, where we could enable all these components beforehand, which completely removed the visible stutter from the onboarding pages and improved performance by a large margin: from 15 frames per second to over a 100 on the second page.

5.5 Conclusion

In this section, the document highlighted the interactive and unit tests that our team executed to make sure the application works properly regardless of the mobile device.

6- General Conclusion & Future work

“FabLab Machines Discovery,” is an Augmented Reality (AR) application that runs on mobile devices regardless of platform and screen resolution.

The goal of the project is to allow any visitor or member of the “FabLab Space” to display essential information about machines upon capturing them using the mobile device camera.

The application implements the following:

- Recognize machines using their tags accurately.
- Display rich and essential information about a machine by clicking on the different popups.
- Play a 3D simulation of the machine to visually describe the way it works.
- Offer a unique augmented reality experience that helps the user become familiar with the machines.
- Support both French and English localization.
- Run on both iOS and Android.

Even though we are proud of the work done on “FabLab Machines Discovery” and the opportunity it offers to newcomers to the “FabLab Space”, we believe multiple improvements could be implemented in the feature to further boost the augmented reality experience.

Future work could be one or many of the following:

- Add more machines to the list of recognizable ones.
- Add links to machine references and videos for users who seek more information.
- Add hints that help the user better scan the machine tags.
- Improve the user interface which displays information and tips in the augmented reality view.
- Reduce jittering of the popups on some mobile devices.
- Support more languages like Arabic and German.

References

- Atlassian. (2023). *What is Agile?* Retrieved from Atlassian: <https://www.atlassian.com/agile>
- Carmigniani, J., & Furht, B. (2011). *Handbook Of Augmented Reality*. Springer.
- IEEE. (2009, December). *IEEE 1008-1987*. Retrieved from IEEE.org: <https://standards.ieee.org/ieee/1008/1462/>
- ISO. (2011, 12). *ISO/IEC/IEEE 42010:2011 Systems and software engineering — Architecture description*. Retrieved from ISO.org: <https://www.iso.org/standard/50508.html>
- ISO/IEC/IEEE. (2015, May). *ISO/IEC/IEEE 15288:2015*. Retrieved from ISO.org: <https://www.iso.org/standard/63711.html>
- LucidChart. *Package Diagram Tutorial | LucidChart*. Retrieved from LucidChart: <https://www.lucidchart.com/pages/uml-package-diagram>
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*.
- Unity. (2021, April 12). *Device Simulator*. Retrieved from Unity Documentation: <https://docs.unity3d.com/Packages/com.unity.device-simulator@3.1/manual/index.html>
- Unity. (2023, April 14). *Unity - Manual: Profiler overview*. Retrieved from Unity Documentation: <https://docs.unity3d.com/Manual/Profiler.html>
- Unity. (2023, April 14). *Unity - Manual: Unit Testing*. Retrieved from Unity Documentation: <https://docs.unity3d.com/Manual/testing-editortestrunner.html>

Abstract

The "FabLab Machines Discovery" is an Augmented Reality (AR) application that runs on iOS and Android devices and provides a unique and fun experience for visitors and members of the FabLab Space.

The application enables the users to scan machines using their mobile devices and to display essential information about them in real-time. It accurately recognizes machines using their tags and displays various types of information through clickable popups, including general information, technical information, safety risks, and use instructions. It also offers a 3D simulation of the machines, which visually describes how they work. With support for French & English localization and augmented reality functionalities, this project helps users become familiar with the machines and improves their choice of materials and specifications.

Keywords: Augmented Reality (AR), Mobile Development, Tag Recognition, Software Architecture
