XR4DRAMA

Extended Reality for Disaster Management and Media Planning

H2020-952133

D4.7

AR Interactive Environment and Applications v2

Dissemination level:	Public
Contractual date of delivery:	Month 27, 31 January 2023
Actual date of delivery:	Month 28, 17 February 2023
Work package:	WP4 Interactive AR/VR solutions for media production and hazard management
Task:	T4.1: Visualization of DSS results in AR T4.2: Visual and GIS-assisted uninterrupted navigation for AR T4.3: Scene understanding for AR interaction to assist actors in the field
Type:	Demonstrator
Approval Status:	Final
Version:	0.9
Number of pages:	61
Filename:	D4.7_xR4Drama_ARInteractiveEnvironmentApplicationsV2_20230217_v0.9.pdf

Abstract

This deliverable documents the final version of the XR4DRAMA AR app, which was developed as the interaction interface for users in the field, i.e. 1st responders (PUC1) and location scouts and journalists (PUC2). The AR app is the result of three tasks: T4.1, which was about the

visualization of platform results in the mobile application, T4.2, which was about AR navigation outdoors and T4.3, which was about scene understanding algorithms running at the edge in order to support the AR navigation and the 1st responders.
The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	20/12/2022	Initialization of the document	Nikos Karantakis
0.2	03/01/2023	Added AR Use Cases	Minas Katsiokalis
0.3	10/01/2023	Revision of AR/MR Frameworks & HMDs	Minas Katsiokalis
0.4	18/01/2023	Revision of Implementation section	Nikos Karantakis, Minas Katsiokalis
0.5	31/01/2023	Object Detection: Use Cases & Implementation	Alexandros Nikopoulos
0.6	02/02/2023	Revision of the document	Minas Katsiokalis
0.7	09/02/2023	Final version	Ilias Kalisperakis
0.8	15/02/2023	Final version (reviewed, edited)	Alexander Plaum
0.9	17/02/2023	Final version for submission	Ilias Kalisperakis

Author list

Organization	Name	Contact Information
up2metric	Christos Stentoumis	christos@up2metric.com
up2metric	Panagiotis Bikiris	panagiotis.bikiris@up2metric.com
up2metric	Theodore Mandilaras	theodore.mandilaras@up2metric.com
up2metric	Nikos Karantakis	nikos.karantakis@up2metric.com
up2metric	Minas Katsiokalis	minas.katsiokalis@up2metric.com
up2metric	Alexandros Nikopoulos	alexandros.nikopoulos@up2metric.com
up2metric	Ilias Kalisperakis	ilias@up2metric.com

Executive Summary

This deliverable D4.7 supports the demonstration of the final version of the XR4DRAMA AR app and presents the implementation approach. The AR app serves as the interaction interface provided to the users in the field for the two Pilot Use Cases, 1st responders in the *Disaster Management* scenario (PUC1) and location scouters and journalists in the *Media Planning* scenario (PUC2). The AR app was developed as a mobile phone application, supporting both major ecosystems Android and iOS, with features that follow the user requirements. The AR app communicates bilaterally with the platform so the users get the latest information from the platform and at the same time the control room receives updates from the field. In this way, both actors in-situ and away have improved situation awareness.

This deliverable supplies demos for using the AR app, as well as basic implementation details. A complete list of the user requirements fulfilled by the AR app is also given. The final version of the AR app is connected and integrated with the backend and the GIS Service and has fulfilled the requirements with specific features and processes within the app. The 2D screens of the user interface and the 2D map view of the app were developed with fully operating features, such as receiving points of interest (POIs) from the GIS, editing, updating, uploading multimedia at POIs, and getting navigation routes that consider danger zones. Furthermore, the features related to POIs have been ported to the AR view. The accuracy of the outdoor navigation and the pose estimation of the user have been improved in this final version using a hybrid method of SLAM implementation and GPS localization. The final version also includes an extensive and thorough experimentation on object detection using ML, the outcome of which is fully integrated into the app as an advanced feature for scene understanding purposes.

Abbreviations and Acronyms

API	Application Programming Interface
AR	Augmented Reality
AT	Authoring tool
AWAA	Alto Adriatico Water Authority
DSS	Decision Support System
GNSS	Global Navigation Satellite System
GIS	Geographic Information System
GPS	Global Positioning System
HCI	Human-Computer Interaction
HMD	Head-Mounted Displays
ML	Machine Learning
MR	Mixed reality
OS	Operating System
OSM	OpenStreetMap
POI	Point of Interest
PUC	Pilot Use Case
ROI	Region of Interest
SGD	Stochastic Gradient Descent
SLAM	Simultaneous Localization And Mapping
TA	Technological Activity
TO	Technological Objective
UX	User experience
VOC	Virtual object classes
VPS	Visual Positioning System
VR	Virtual Reality
WP	Work Package
XR	Extended Reality

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The XR4DRAMA application is a location-aware application that extracts data points from OpenStreetMap and other internet sources of geodata and aggregates them into various Points-Of-Interest (POIs) and Regions-Of-Interest (ROIs). These POIs are organized into thematic categories and hold relevant information about the geographical landmarks they represent. The application user can interact with these points in the map view, or the AR view and alter them to organize one's tasks and keep the control room updated. Object detection using ML adds to the functionalities of the application providing useful assistance to actors in the field.

The deliverable starts with an introduction (Section 1) where the position of the AR app in the XR4DRAMA platform is described. Section 2 provides a brief overview of different frameworks available for developing MR application. Section 3 is devoted to the use cases that describe the current AR app capabilities. Section 4 has details on the implementation and Section 5 contains demo videos. The deliverable closes with a brief section of conclusions.

1.1 Tasks & Objectives

The AR app was developed in the scope of tasks T4.1, T4.2, as well as T4.3 (details to follow), and the developed features were based on the user requirements initially defined in deliverable D5.2 and finalized in D6.2.

Task **T4.1** ("Visualization of DSS results in AR") regards the development of an AR app capable of projecting and managing the information available in the XR4DRAMA platform, which helps the situation awareness of the field users, either the 1st responder (PUC1) or the location scout and the filming crew (PUC2). In order to support the demanding XR capabilities, the application was developed to support high-end mobile phones that run on both major operating systems (i.e. Android and iOS).

The scope of task **T4.2** ("Visual and GIS-assisted uninterrupted navigation for AR") was to position a field user in the GIS of the platform via the XR device and position the georeferenced info correctly in the world space in AR. This is achieved by exploiting the measurements from the mobile phone sensors (GNSS, IMU, and compass sensors), along with object detection and scene segmentation and SLAM, SfM algorithms, existing in commercial APIs, for pose estimation in an outdoor environment. The localization algorithm exploits the information available in the GIS.

The scope of task **T4.3** ("Scene understanding for AR interaction to assist actors in the field") was to develop the mobile phone algorithms that – based on computer vision and object detection – aid user navigation and automatic identifications of multiple objects, such as potential threats and dangers for the 1st responders.

The work described in D4.7 corresponds to "**IO3** - Develop enhanced interactive AR applications for outdoor media production and disaster management": This objective pertains to delivering an integrated platform for remote outdoor observation and planning, designed to cater to the needs of both media production companies and disaster relief organizations that require the visualization of an outdoor area and use this as a canvas for strategic planning.

1.2 User requirements fulfilled

The following tables document the user requirement supported by the AR app. Table 1 refers to the system requirements addressed; Table 2 shows the general information requirements; Table 3 and Table 4 refer to the PUC specific requirements. Most requirements of the AR app are fulfilled through the GIS Service, the Backend, and the Decision Support System (DSS).

Req ID	Name	Description
SYS-2	End-user interface	An HCI that allows end users to easily communicate with the system
SYS-3	Location ingests	The possibility to define a specific location
SYS-4	Location-query	A functionality of the system that allows end users to initiate a query regarding a specific location in web and cloud services
SYS-5	Aggregation of query status and results	The capacity of the system to observe the query and aggregate the identified content (e.g. videos, images, text) in an organised manner (categories, clusters, order)
SYS-6	Immersive visual representation	A functionality that visualises the location and additional information to enhance situation awareness (e.g., VR, AR)
SYS-10	Add own data	Control room staff can add images, videos, models, or scans to improve data, or change specific data points that might not have been available, e.g., availability of public parking
SYS-12	Mobile application	An application that allows for operating the system in and from the field
SYS-14	Remote access to Level 1 situation awareness	The system's capacity to grant remote users (e.g. location scout) access to a Level 1 situation awareness representation (partly or in total) via the mobile application. Citizens (PUC 1 only) should receive valuable information such as alerts, warnings about areas at risk, the position of the safe areas, sand-bag distribution, shelters.
SYS-16	System updates	The system processes new input (e.g. from location scout/first responders) and updates previous results and representations
SYS-17	Enhanced (Level 2) situation awareness	The updated data is used to create an enhanced version of the scene graph containing all available relevant content and information

Table 1: System-related requirements addressed by the GIS Service

ID	Category	Info ID	Subcategory	Level Information	Source of Information
		PUC2-09	Toilets	Level 1: Information via GIS data	OpenStreetMap
		PUC2-08	Water Supply	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		PUC2-08	Power Supply	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		PUC2-08	Internet Access	Level 1: Information via GIS data	OpenStreetMap
		PUC2-08	Gas Stations	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
GC-01	Facilities	PUC2-08	Charging Stations	Level 1: Information via GIS data	OpenStreetMap
		PUC2-08	Tourist Offices	Level 1: Information via GIS data	OpenStreetMap
		PUC2-08	Police Stations	Level 1: Information via GIS data	OpenStreetMap
		PUC2-08	Fire Departments	Level 1: Information via GIS data	OpenStreetMap
		G-02, PUC2-06	Public Buildings	Level 1: Information via GIS data Level 2: Research by location scout	Vicenza.gov.it
GC-02	Public Services	-	Cemeteries	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		PUC2-08	Pharmacies	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		PUC2-08	Doctors	Level 1: Information via GIS data	OpenStreetMap
GC-03	Medical	PUC2-08	Hospitals	Level 1: Information via GIS data	OpenStreetMap

		G-01, PUC2-03	Parking	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		G-01	Taxi	Level 1: Information via GIS data	OpenStreetMap
		G-01	Public Transport	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		G-01	Car Rentals	Level 1: Information via GIS data	OpenStreetMap
		G-01	Bike Rentals	Level 1: Information via GIS data	OpenStreetMap
GC-04	Transportation	G-01	Air Travel	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		G-02, PUC1-17	Sights	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
GC-05	Cultural	G-02, PUC1-17	Museums	Level 1: Information via GIS data	OpenStreetMap
		G-02, PUC1-17	Venues	Level 1: Information via GIS data	OpenStreetMap

Table 2: Information-related requirements (General information) addressed by the GIS Services

ID	Category	Info ID	Subcategory	Level Infromation	Source of Information
		PUC1-16, PUC1-17	Warnings	Level 2: Research by location scout	-
		PUC1-16, PUC1-17	Flooded Reports	Level 2: Research by location scout	-
		PUC1	Risk Areas	Level 2: Research by location scout	-
		-	Civil Protection	Level 2: Research by location scout	-
		PUC1-03	Civil Protection Distribution Places	Level 2: Research by location scout	-
AAC-01	Disaster Management	PUC1-03	Safety Areas	Level 1: Information via GIS data Level 2: Research by location scout	OpenStreetMap
		PUC1-02	Manholes	Level 1: Information via GIS data Level 2: Research by location scout	OpenStreetMap
		PUC1-02	Pipelines	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		PUC1-02	Embankments	Level 1: Information via GIS data	Vicenza.gov.it
		PUC1-02	Power Grids	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		PUC1-02	Factories	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		PUC1-02	Antennas	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
AAC-02	Infrastructure	PUC1-02	Aqueducts	Level 1: Information via GIS data	Vicenza.gov.it
AAC-03	Flood Forecasts	-	Flood Results From	Level 2: Research by location scout	-

			Model		
		PUC1-07	Elements at Risk	Level 2: Research by location scout	-
		PUC1-01	Water Bodies	Level 1: Information via GIS data	Vicenza.gov.it
		PUC1-03	Parks	Level 1: Information via GIS data	OpenStreetMap
AAC-04	Natural Elements	PUC1-03, PUC2-04, PUC2-05	Natura 2000 Sites	Level 2: Research by location scout	-
		-	Nurseries	Level 1: Information via GIS data Level 2: Research by location scout	Vicenza.gov.it
		-	Kindergartens	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		-	Primary Schools	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
AAC-05	Education	-	Secondary Schools	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it
		-	Universities	Level 1: Information via GIS data	OpenStreetMap, Vicenza.gov.it

Table 3: Information-related requirements PUC1 (Disaster management)

ID	Category	Info ID	Subcategory	Level Infromation	Source of Information
		PUC2-14	Source of Pollution	Level 2: Research by location scout	-
DWC-01	Production Planning	PUC2-04, PUC2-05	Restricted Areas	Level 1: Information via GIS data	OpenStreetMap

		PUC2-11	Props and Gear	Level 2: Research by location scout	-
		New	Snapshots	Level 2: Research by location scout	-
		PUC2-10	Cafés	Level 1: Information via GIS data	OpenStreetMap
		PUC2-10	Restaurants	Level 1: Information via GIS data	OpenStreetMap
DWC-02	Food and Dring	PUC2-10	Bars and Pubs	Level 1: Information via GIS data	OpenStreetMap
		-	Hotels	Level 1: Information via GIS data	OpenStreetMap
		-	Apartments	Level 1: Information via GIS data	OpenStreetMap
DWC-03	Accommodation	-	Campsites	Level 1: Information via GIS data	OpenStreetMap
		PUC2-08	Groceries	Level 1: Information via GIS data	OpenStreetMap
		PUC2-08	Clothing	Level 1: Information via GIS data	OpenStreetMap
		PUC2-08	Electronics	Level 1: Information via GIS data	OpenStreetMap
DWC-04	Commerce	PUC2-08	Other Shops	Level 1: Information via GIS data	OpenStreetMap
		PUC2-08	ATMs	Level 1: Information via GIS data	OpenStreetMap
DWC-05	Finance	PUC2-08	Exchange Offices	Level 1: Information via GIS data	OpenStreetMap
		PUC2-08	Banks	Level 1: Information via GIS data	OpenStreetMap

Table 4: Information-related requirements for PUC2 (Media production planning)

2 REVIEW

A review of existing works in outdoor AR applications and the devices that explicitly support AR and MR experiences was presented in section 2 of deliverable D4.1. In the following, we expand our initial review on the frameworks for developing XR applications.

2.1 AR / MR Frameworks

There are different frameworks for developing XR applications. Most of them have a strong correlation with the selected device for deploying an application, some of them are closed-sourced. VR frameworks are out of scope here; the research is targeted to AR / MR frameworks.

ARKit[1] is Apple's own AR development platform and exclusive to iOS devices. It offers advanced features such as face tracking, realistic rendering, and environment understanding using its ARKit World Tracking. Developers can also use ARKit's physics engine to simulate realistic interactions between virtual objects and the real world. Additionally, ARKit supports image detection and tracking, making it possible for developers to create AR experiences with real-world images.

Google has developed **ARCore**[2] for Android devices. It offers similar features as ARKit, such as motion tracking, environmental understanding, and light estimation using its ARCore Motion Tracking and ARCore Light Estimation. ARCore also supports image detection and tracking, allowing developers to create AR experiences with real-world images.

Vuforia[3] is a popular AR SDK that supports both iOS and Android platforms. It offers advanced image and object recognition capabilities using its VuMark technology, as well as support for multiple targets and simultaneous tracking via its Smart Terrain technology. The SDK also provides support for 3D model recognition and tracking, allowing developers to create interactive experiences with virtual objects. Vuforia also offers a cloud-based recognition system, allowing developers to scale their AR applications to a larger user base.

Lightship ARDK[4] is another framework and was developed by Niantic, the company behind ”Pokémon Go”, the first widely available AR game. ARDK supports spatial mapping for mobile, world segmentation, and integrates its own VPS to specific areas of the world, for world scale AR experiences.

EasyAR[5] is a cross-platform AR SDK that supports both iOS and Android. It offers features such as image and object recognition, tracking, and 3D rendering. EasyAR supports multiple target

[1] https://developer.apple.com/augmented-reality/arkit/

[2] https://developers.google.com/ar

[3] https://www.ptc.com/en/products/vuforia

[4] https://lightship.dev/products/ardk?hl=en

[5] https://www.easyar.com/

tracking and can recognize different types of targets such as images, QR/Barcodes, and 3D objects.

MRTK is a (Microsoft-driven) open source project for the Unity game engine and provides tools and features for rapid prototyping development of MR applications on various devices, HoloLens among others.

The differences between the AR / MR devices and the supporting frameworks create difficulties when it comes to cross-platform application development. Lately, Unity has started to provide **AR Foundation**[6] , a set of tools and API's built on top of the Unity Engine that allows developers to create AR experiences that work across multiple platforms and devices. It uses the underlying technology of ARKit and ARCore and provides a common API for developers to access the features of those platforms. This helps developers create AR experiences that will run on both iOS and Android devices with minimal modifications. The AR Foundation also includes features such as plane detection, image tracking, and point cloud visualization, facilitating developers to create interactive AR experiences.

Other remarkable AR frameworks are: **WebXR**[7] an open-source JavaScript API that allows developers to create and run XR experiences on the web and **SolAR**[8]**,** an open-source framework to solve computer vision related problems for AR.

2.2 AR Framework of AR app

Unity's AR Foundation was selected due to its versatile nature. AR Foundation offers a unified framework for AR development across both iOS and Android devices, reducing the need for platform-specific development. This makes it easier and more efficient for developers to focus on the development of features within the app rather than solving platform specific issues.

Based on the feedback received from the project partners, a mobile device is selected as the device of augmentation as it seems to be more comfortable for first responders in the field to use phones instead of HMDs. Furthermore, phones allow for a more natural and intuitive UX, given their built-in sensors and input methods. This makes it easier for first responders to interact with the app – and it eliminates the need for additional hardware or accessories.

[6] https://unity.com/unity/features/arfoundation

[7] https://immersiveweb.dev/

[8] https://solarframework.github.io/

3 AR APPLICATION USE CASES

This section provides a comprehensive overview of the use case scenarios supported by the AR app. The purpose of this presentation is to explain the intended usage of the AR application and resolve any misunderstandings that may arise during the evaluation process. A diagram representing all use cases is displayed in Figure 1, and each use case is analyzed in greater detail below.

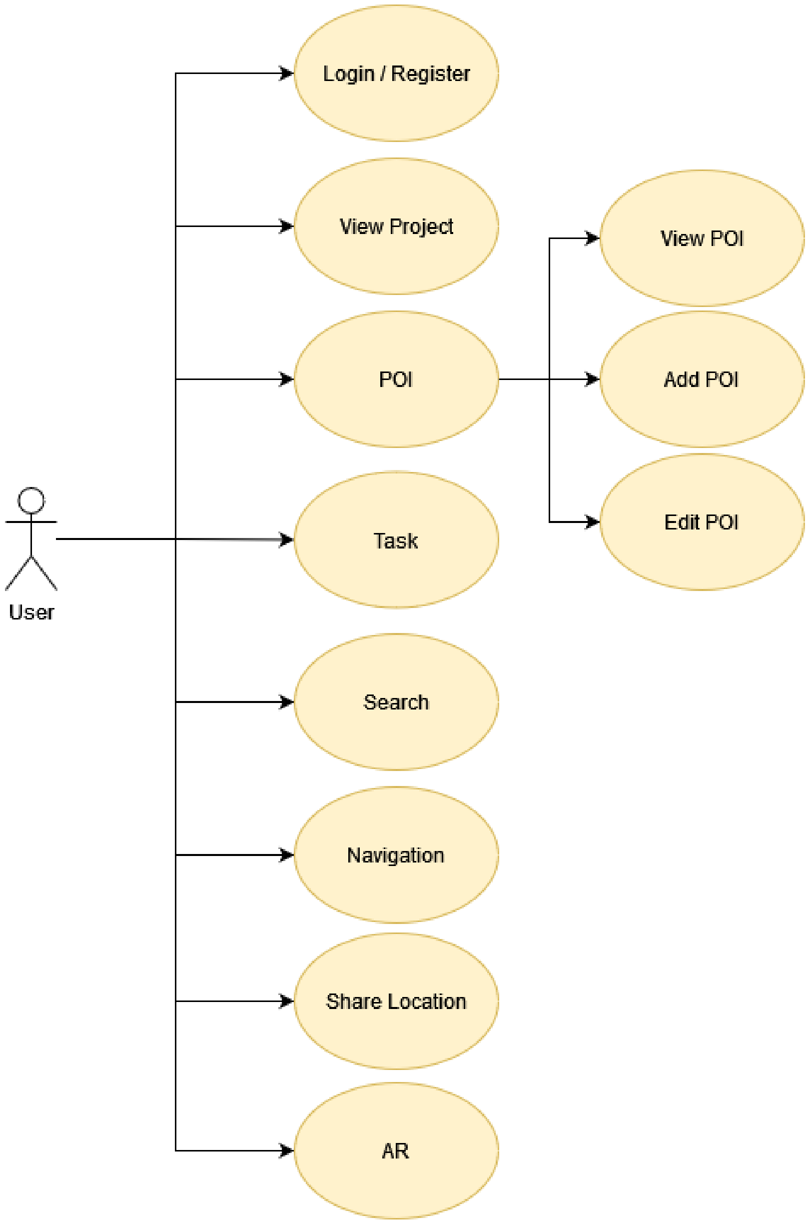

Figure 1: ARapp Use Cases

3.1 Register / Login

Upon launching the application, the user is required to enter their credentials in order to access the features of the application. The user can enter their registered credentials to log in or create a new account by pressing "Sign Up" and filling out the required form. The application will redirect the user back to the login view after a new account is registered. Figure 2 displays the respective login screens.

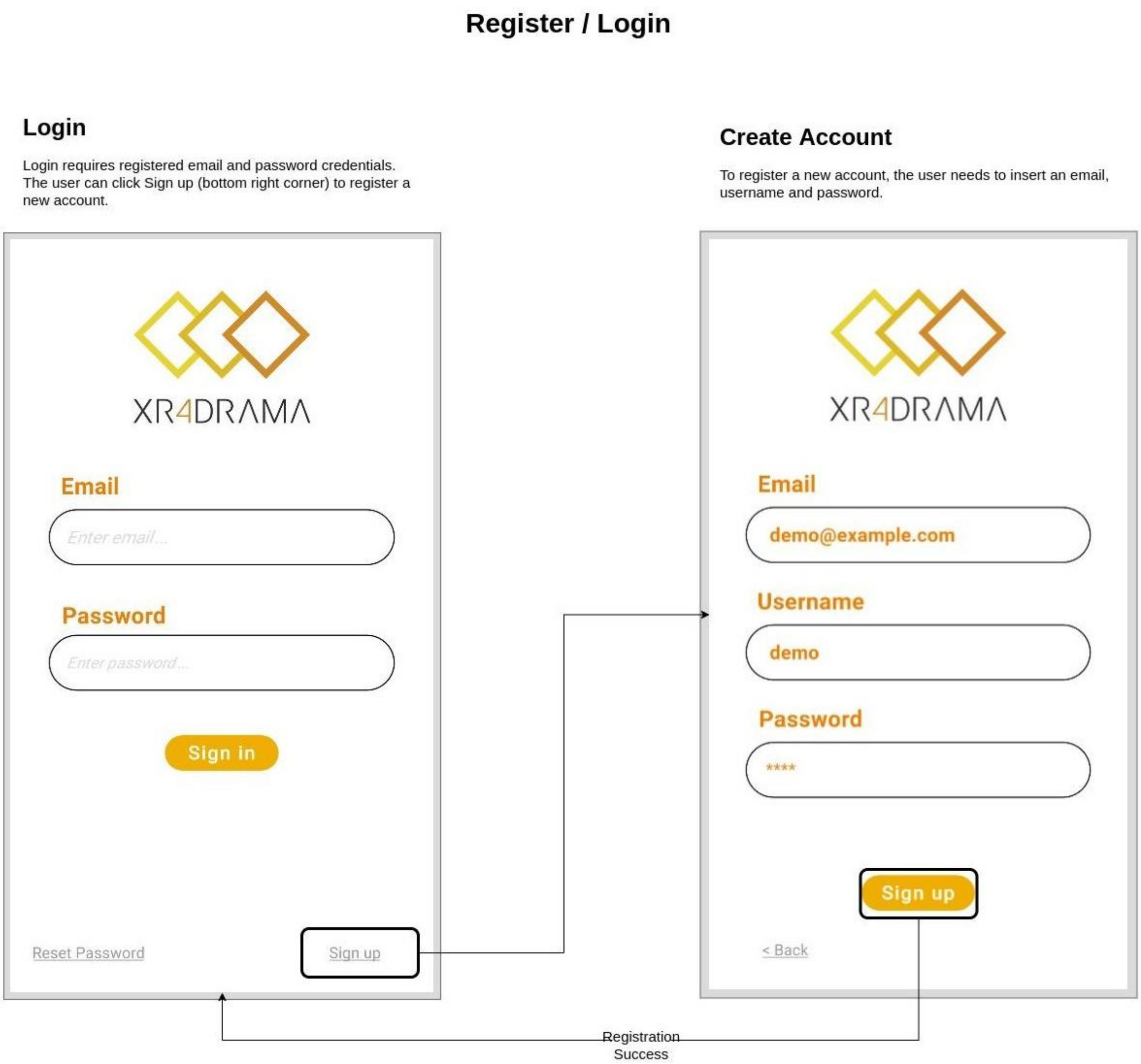

Figure 2: ARapp: Register / Login

3.2 View Project

Once logged in, the user can view all projects they have registered for. The projects are classified as either within or outside the user's GPS location area, and the user has full control and unrestricted access to all projects within their location area, while access to projects outside of the location area is limited.

By clicking on a specific project, the user is able to view detailed information regarding a selected project. This information includes an overview, files and contacts. More specifically:

- **Overview:** Comprises general information related to the project.
- **Files:** Encompasses media files specific to the project.
- **Contacts:** Contains the relevant contacts associated with the project.

The user can also enter the map view of the selected project by clicking on the map icon on the bottom menu.

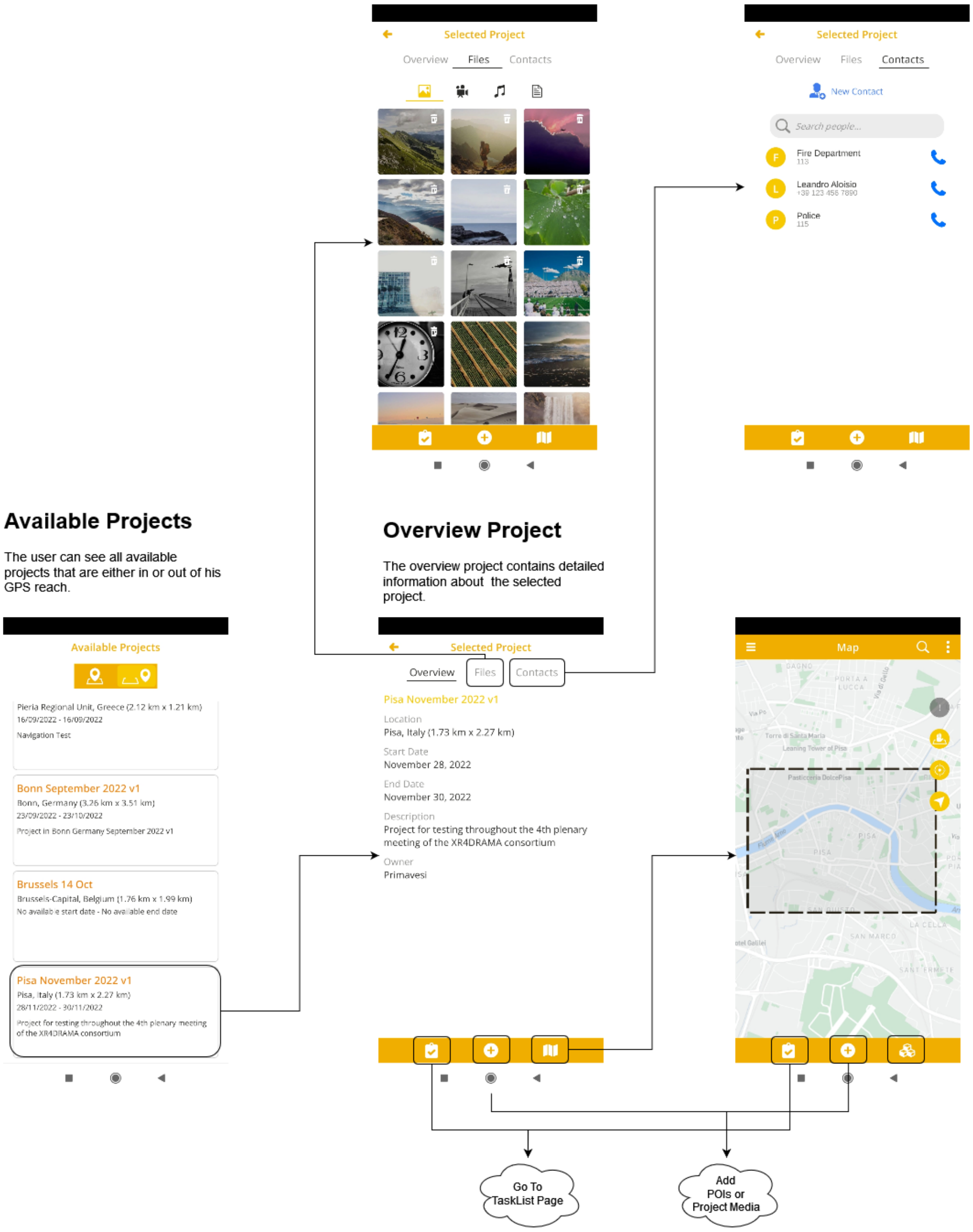

Figure 3: ARapp: View Project

3.3 POIs

The POI Retrieval System is designed to gather data from GIS sources, such as OSM and AWAA's data pool. The system presents the retrieved POIs with both a short and a general overview.

The short view includes the name, subcategory, thumbnail, and quick access buttons for navigation, task-list, call, email, and original geometry. The display also shows the date of creation or last update, and the individual responsible for the change.

The general overview provides a comprehensive view of each POI, including its properties, media files, comments, and attributes. Users are able to edit existing POIs, add new ones and modify their properties, media, comments and attributes as needed.

3.3.1 View POIs

In the **Map view,** the user has the ability to display a selection of POIs on the map by utilizing the *Categories* panel on the top-right. The panel offers a variety of categories and subcategories that can be chosen by the user. It is important to note that the capability of simultaneously selecting multiple categories is accessible at all times. This allows for a flexible and efficient way to display relevant POIs on the map.

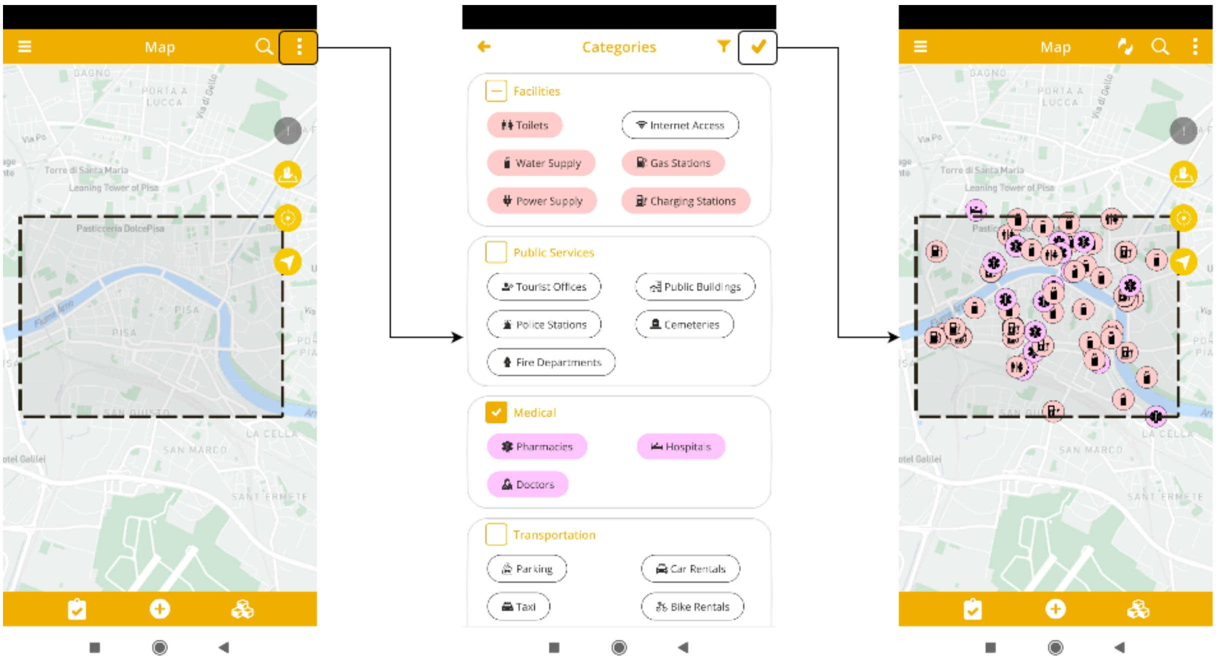

Figure 4: ARapp: View POIs

3.3.2 Edit POI

The POI Retrieval System provides the user with the ability to modify an existing POI. To do so, the user must first select the POI in question. After selection, the POI is displayed in the center of the screen, along with a short view. This view provides key information about the

POI, including its name, subcategory, thumbnail, and quick access buttons for navigation, task-list, call, email, and original geometry. For a more comprehensive view of the POI, the user must perform a swipe up gesture. The general overview, which is accessed in this manner, provides a detailed account of the POI's properties, media files, and attributes. The attributes provide specific information about the POI and can be modified by the user as necessary.

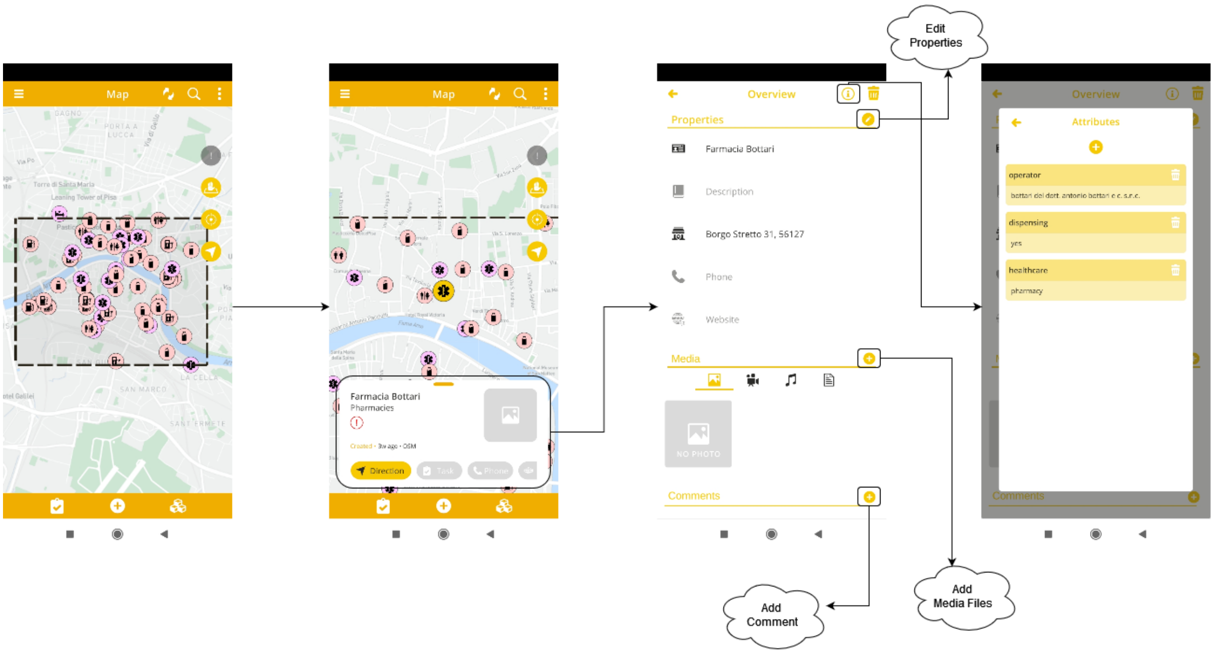

Figure 5: ARapp: Edit POI

3.3.3 Add POI

The Map view in the POI Retrieval System allows the user to create a new POI by pressing the *Add* button located at the bottom-center of the screen. The user is prompted to select a category and subcategory for the POI, which will then be displayed on the map. After choosing the appropriate category, the user must select the geometry for the POI, which may be in the form of a point, line, or polygon.

Once the geometry has been defined, the user can finalize the creation of the POI by pressing the *Check* button located at the top right of the screen. The overview of the newly created POI will then be displayed, allowing the user to make any desired modifications.

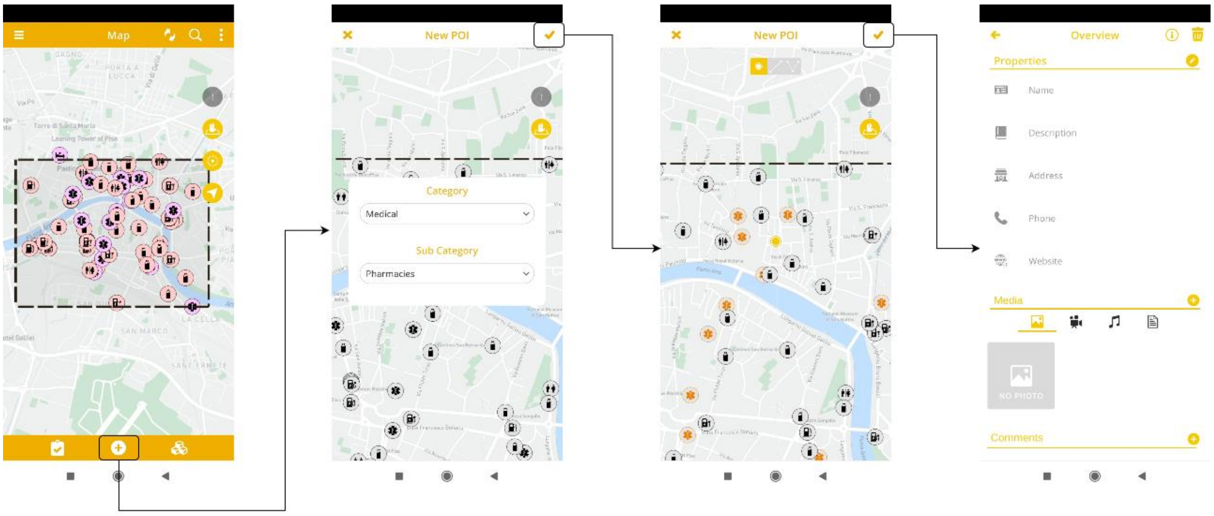

Figure 6: AR app: Add POI

3.4 Task List

The Task List is an integral component of the app and can be accessed by pressing the Task List button located at the bottom-left of the screen. This feature displays tasks that are related to a specific project including details such as the task creator and assigned worker, headline, description, priority, comments, status, and type.

Completed tasks include a report that comprises a message and an audio file. Tasks are designed with four statuses, with varying levels of access:

- **Requested**: The task has not yet been assigned to a worker and is available to all users in the project.
- **Appointed to:** The task has been assigned to a worker, but has not been initiated (i.e. started).
- **Initiated:** The worker assigned to this task is currently working on it.
- **Completed**: The worker has finished the task and submitted the relevant report.

Tasks with the *"requested"* status are accessible to all users that have access to the project, while the other statuses are only accessible to the creator and the designated workers. The status of a task can be altered by executing a swipe gesture to the left or right. A swipe to the left changes the task from its current status to the next status in the sequence (e.g. Assigned → Initiated), while a swipe to the right changes the task status back to the previous status.

The Task List feature supports three different types, each with distinct functionalities within the application:

- **Edit POI:** A task related to a POI, providing the user with quick access to the POI.
- **Add Project File:** A task that allows the user to submit media files in a related project.

- **Add POI**: This task does not have any specific capabilities within the application, but it provides extensive information for creating general tasks.

The creator can create a task in the application, related to a POI or submitting media files in a project.

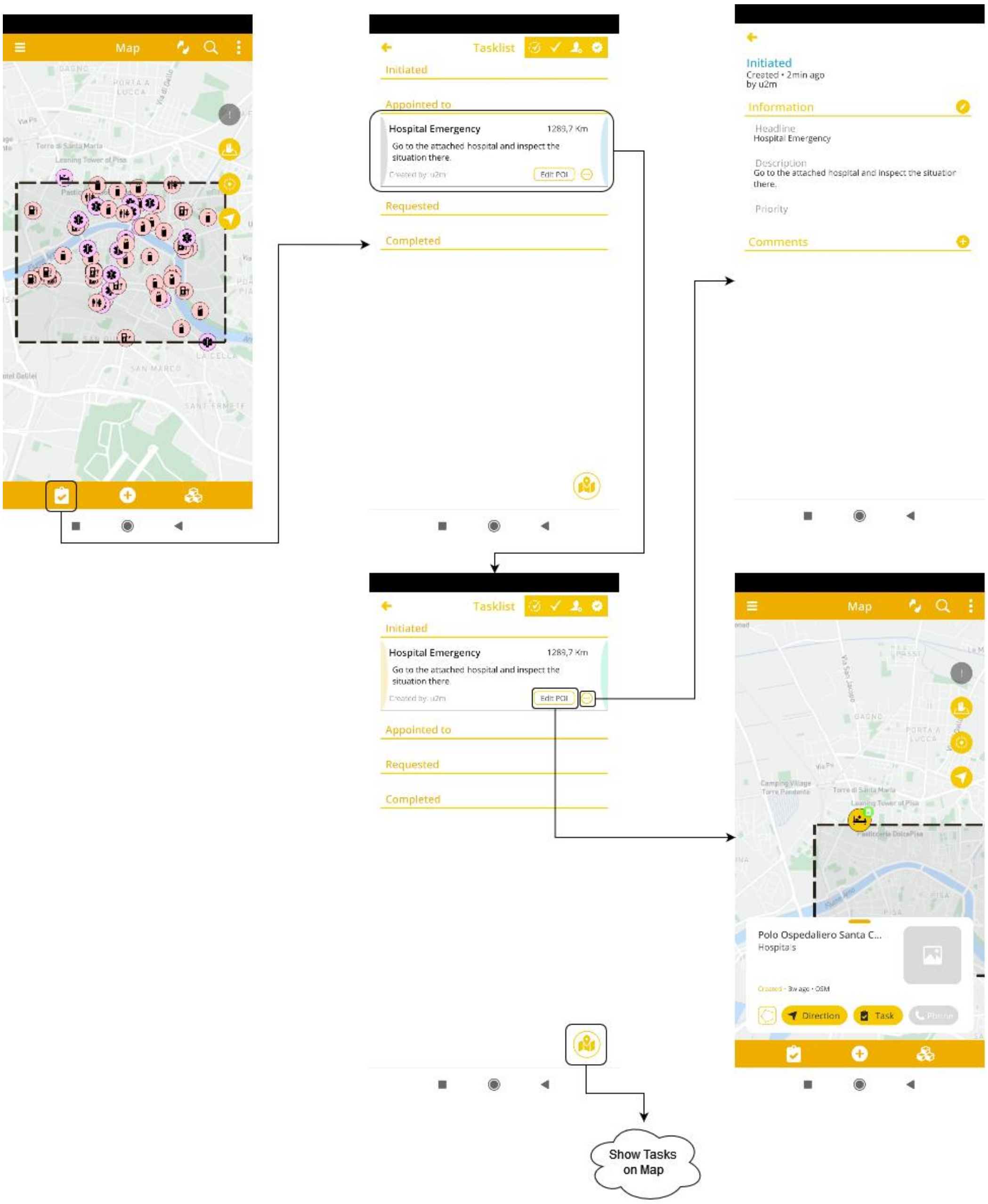

Figure 7: AR app: Task List

3.5 Search

In the Map view, users can perform a search by pressing the *"Search"* button located at the top-right of the screen. This search function enables users to find POIs, places and roads based on their name. In particular:

- **POIs:** The user can search for POIs within the current project.
- **Places:** The user can search for places that are included in the project.
- **Roads**: The user can search for streets within the project.

The search results include the full name match, the type of match (POI, place, or street), the distance from the user, and additional relevant information, as shown in Figure 8. To create a POI related to a particular search result, the user can select that result from the search results list.

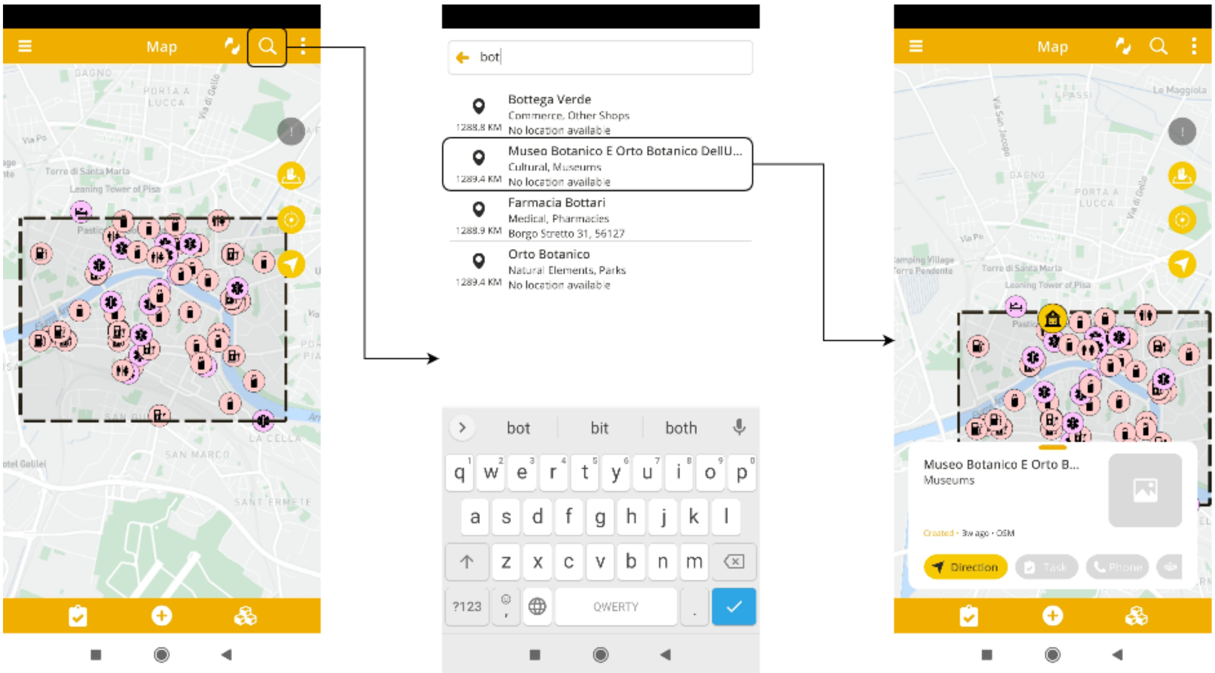

Figure 8: AR app; Search

3.6 Navigation

The navigation service in the application provides a convenient and effective way for users to find their way from one location to another within the project. There are two methods of navigation available in the application, allowing users to either choose a non-predefined location on the map or navigate directly towards a selected POI.

The first way to initiate navigation is by clicking the Navigation button located in the middle-right of the screen. The user will then place the yellow marker on the desired location for navigation. It is also possible to navigate directly to a selected POI by using the Navigation button in the short view of the POI.

When navigation is initiated, the application sends the user's current location and the desired destination to the GIS service and receives the most efficient, quickest, and safest route to the destination. The calculation takes into account any hazardous areas within the project and provides an alternative safe route if necessary. Once the route has been calculated, the application visualizes the path for the user, providing a clear and easy to follow guide to their destination. The *"Start"* button begins on the target location and the estimated time, residual distance and arrival time are displayed at the bottom of the screen.

Finally, once the user reaches their desired destination, the application will send out a notification message to inform them the journey has been successfully completed.

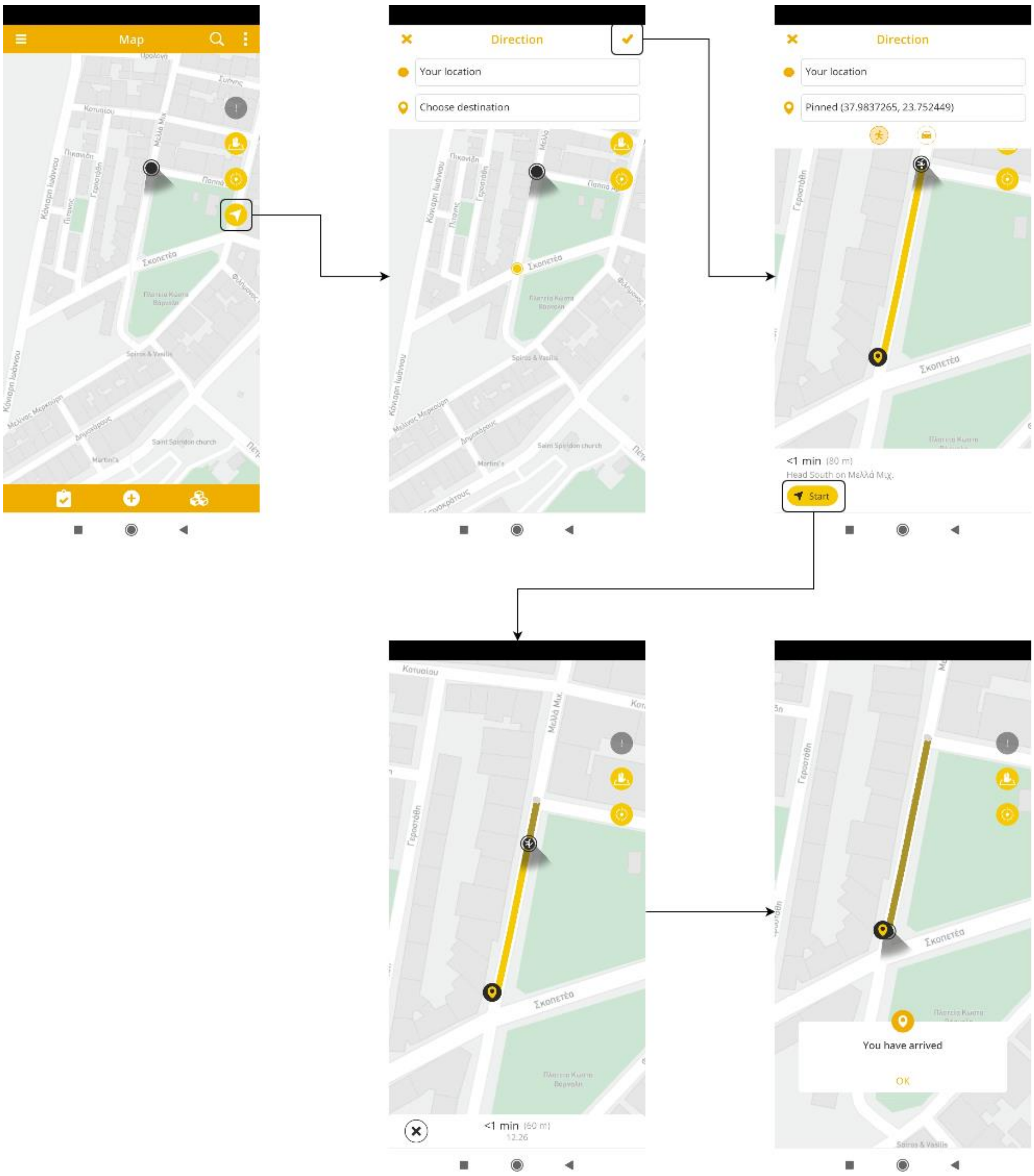

Figure 9: ARapp: Navigation

3.7 Danger Zone

The app displays the danger zones defined for each project. The danger zones are first established by the AT and then represented on the map view for the user's reference. Whenever a POI is located within one of the established danger zones, it is identified with a danger symbol in the short view. The navigation service of the app is designed to guarantee the safety of the user by avoiding routing through any danger zone. As a result, the user will be directed along the quickest, most efficient, and safest route, considering any potential dangers within the project, and offering an alternative safe route if required. The app continually updates the estimated arrival time, residual distance, and time to the destination.

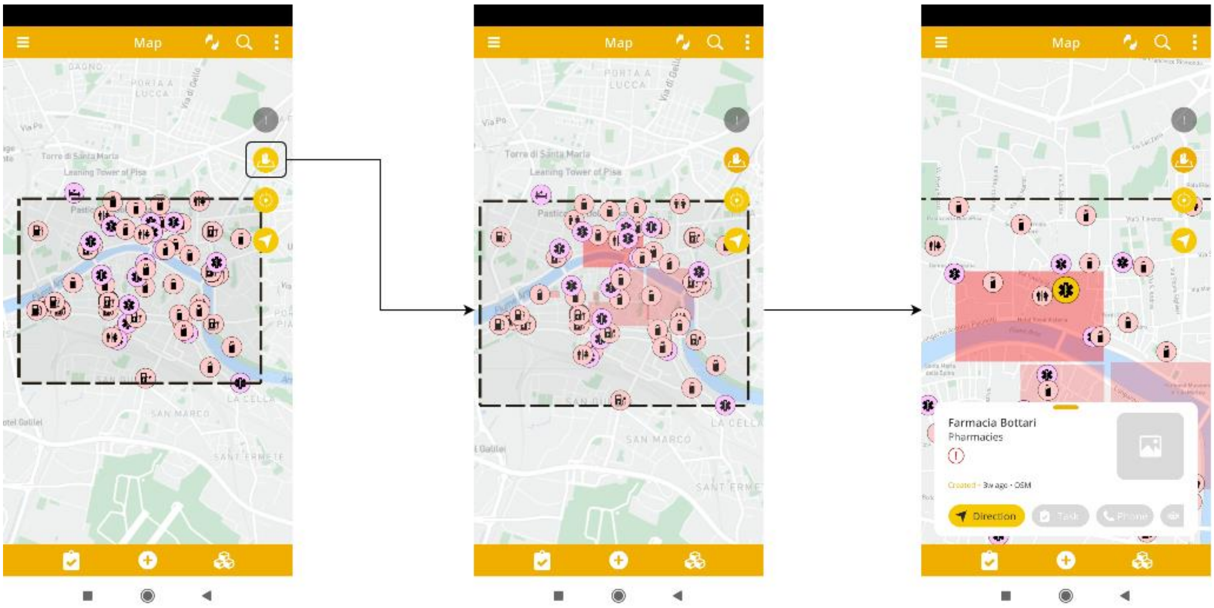

Figure 10: ARapp: Danger Zone

3.8 Share Location

The app includes a location sharing feature, which can be accessed through the side menu where the user can active or deactivate it. On the share location panel, the user is able to observe a list of registered users for the current project and can adjust the settings of the location sharing service. When the service is activated, the user's location will be visible to others on the map, indicated by a green icon. This information is only available to users who have also enabled the location sharing service. The map displays a marker that features the first letter of the user's name for each user sharing their location, which updates every 5 seconds and is immediately transmitted to all other active users.

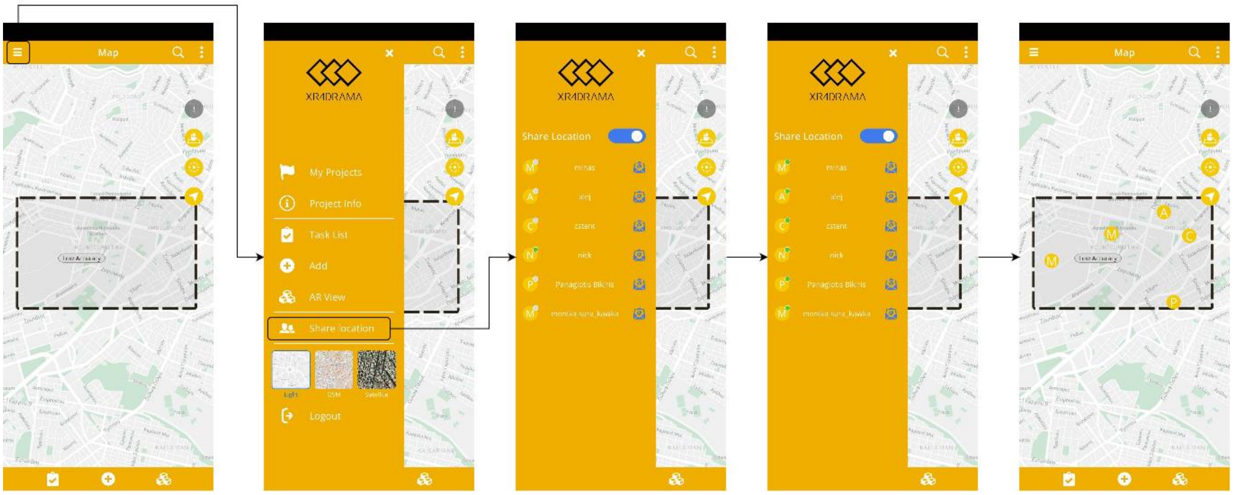

Figure 11: ARapp: Share Location

3.9 Augmented Reality (AR)

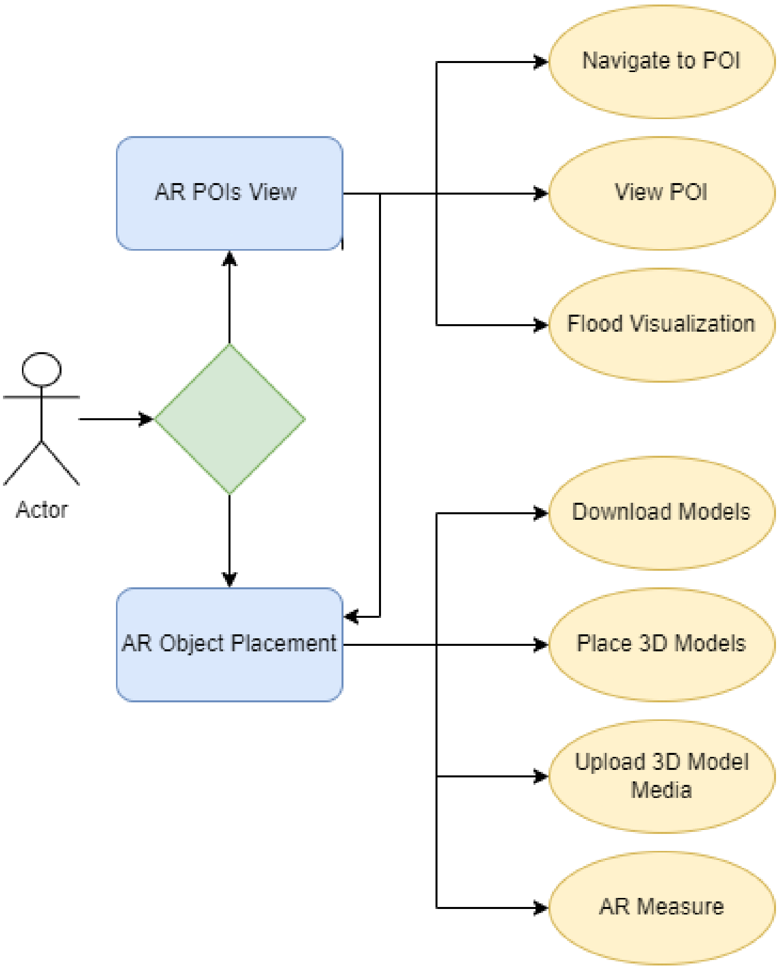

Figure 12: AR Mode: Use Cases

A major feature of the AR app is the support of AR functionalities served by the "AR POI View" and "AR Object Placement" features.

3.9.1 Navigation to POI

In AR mode of the app, navigation is supported by displaying a virtual line on the screen that points to the next POI. Actually, multiple POIs are supported. Upon entering the "AR POI View", the pre-selected POIs can be viewed on top of the real-world environment, positioned based on their geographical coordinates. The POIs are updated constantly to align with the user's orientation view. The mini map at the bottom of the screen also rotates in accordance with the user's orientation.

The user has the capability to adjust the range in which POIs will be displayed on the map by manipulating the arrows on the compass dial in either left or right direction. A direct line connects the user with any displayed POI. The user can also modify the level of transparency for the connected lines at any time by pressing the upwards or downwards arrows.

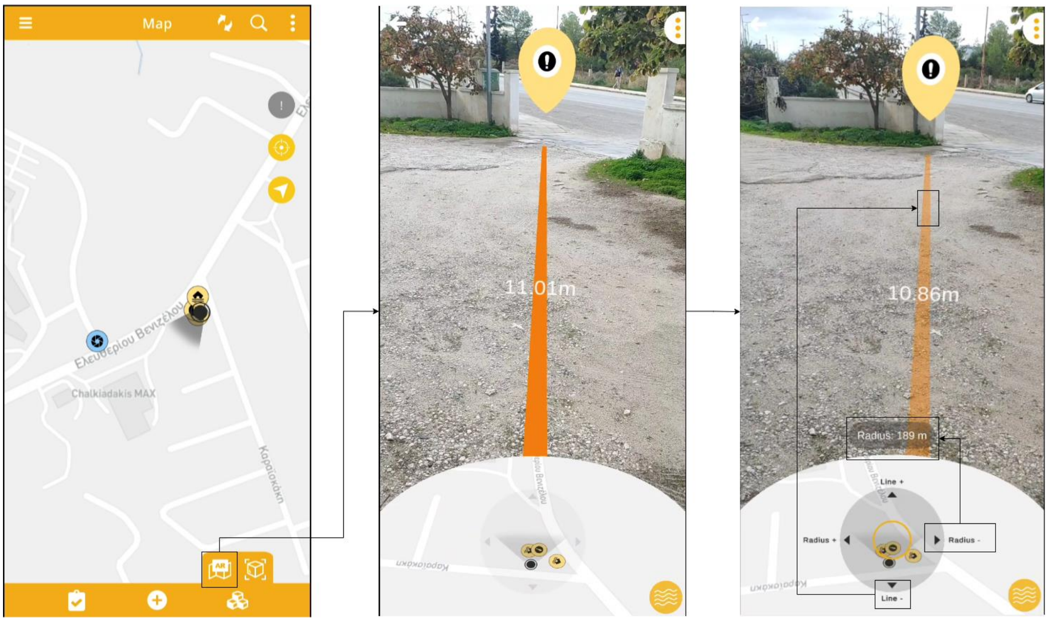

Figure 13: AR Mode: User enters "AR POI View"

3.9.2 View POI

When the user arrives at a POI (range: < 20 m), they can interact with it by clicking on its mark. After clicking, three buttons become available:

- **Info button**: displays a description and relevant information about the POI.
- **Comments button**: displays comments made by other users regarding the POI.
- **Media button**: displays media files (photos, videos, audio) related to the POI.

If the POI includes geometry data, a fourth button is enabled, which (when pressed) displays the geometry data on the mini-map.

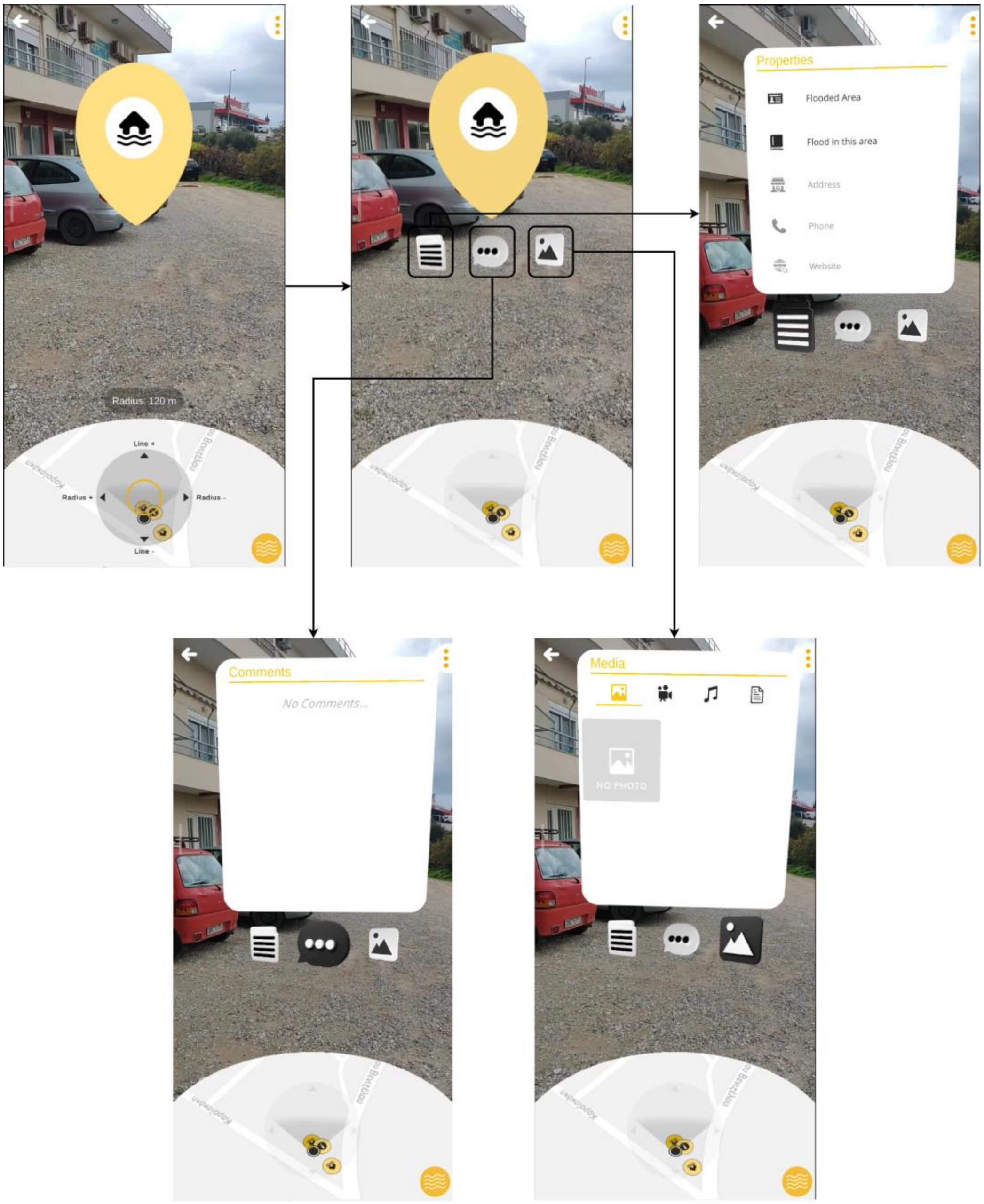

Figure 14: AR Mode: User views POI's information

3.9.3 **Flood Visualization**

If flood layer data is available, a corresponding button is enabled in the bottom right corner of the screen. By pressing it, the application prompts the user to scan the ground. After a successful scan, a message is displayed indicating the expected height of a flood in this area.

Subsequently, the user views the real-world environment augmented by virtual water, which rises to the level of the expected flood, based on the available data for the area.

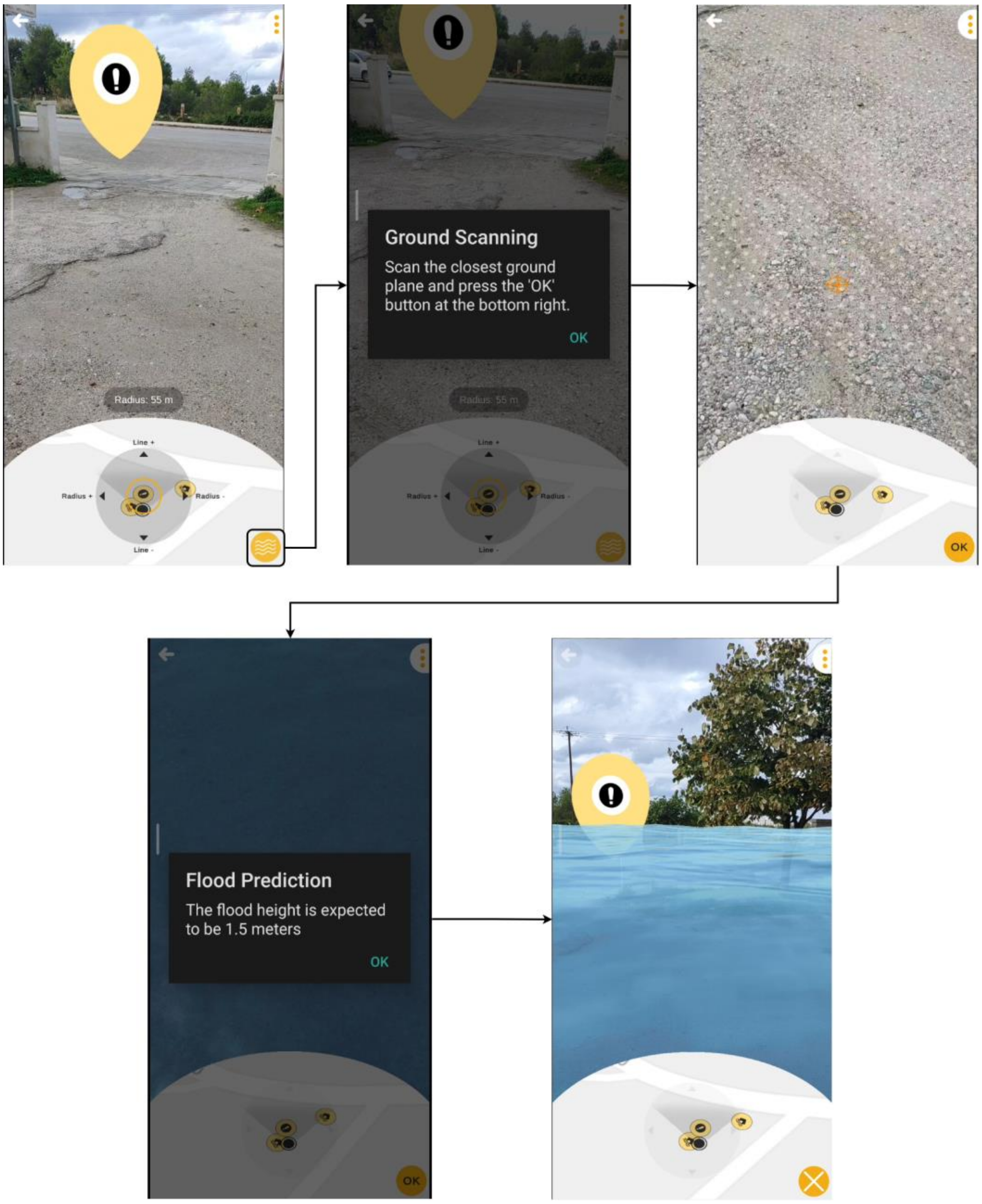

Figure 15: AR Mode: Flood visualization

3.9.4 **3D Model Download**

Upon entering the "AR Object Placement" feature, a side panel displaying available 3D models becomes visible, as shown in Figure 16. The type, quality, and number of objects depend on the individual project. Soon after the initial activation of the placement feature, the 3D models chosen by the user need to be downloaded. After a successful download, the object is saved on the device and can be used at any time. User can access “AR Object Placement” either through the map or the “AR POI View”.

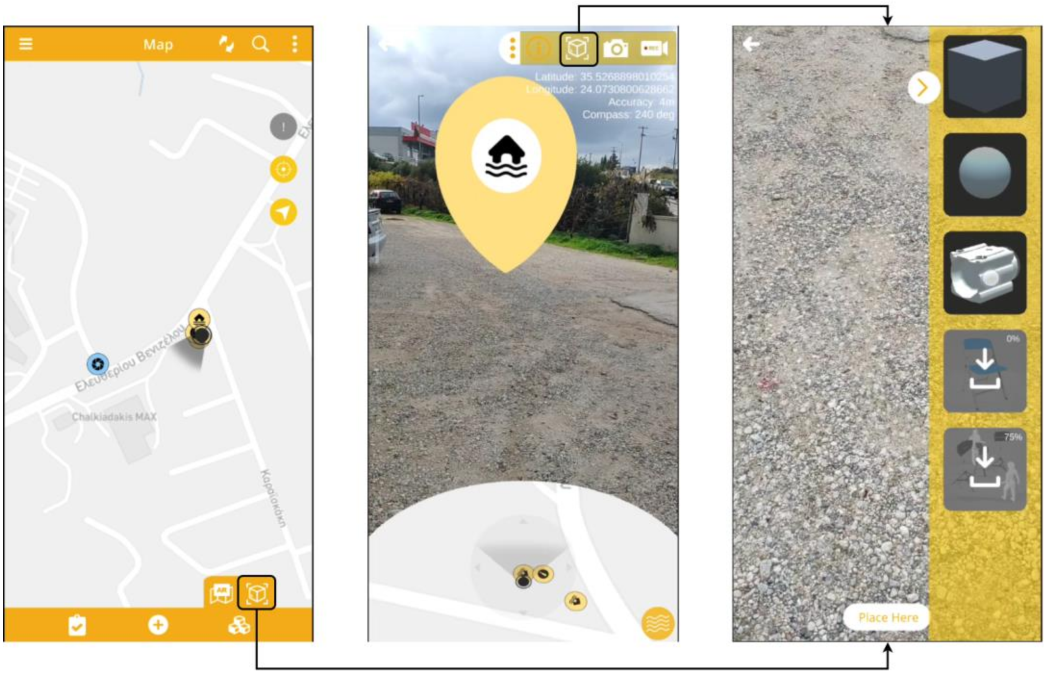

Figure 16: AR Mode: User enters “AR Object Placement”

3.9.5 Placement of 3D Models

The user can place downloaded models in the real-world environment. Upon selecting a model, the application will prompt the user to scan a floor surface for proper placement. The user can adjust the position, orientation, and size of objects as needed. Additionally, any previously placed object can be removed if desired.

There is a unique type of object, namely the aquadike, which can be lined up to create a virtual barrier for planning purposes.

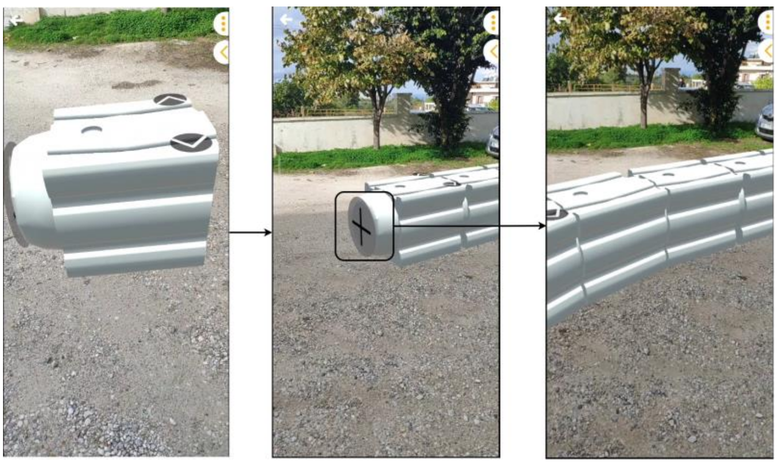

Figure 17: AR Mode: User places aquadikes

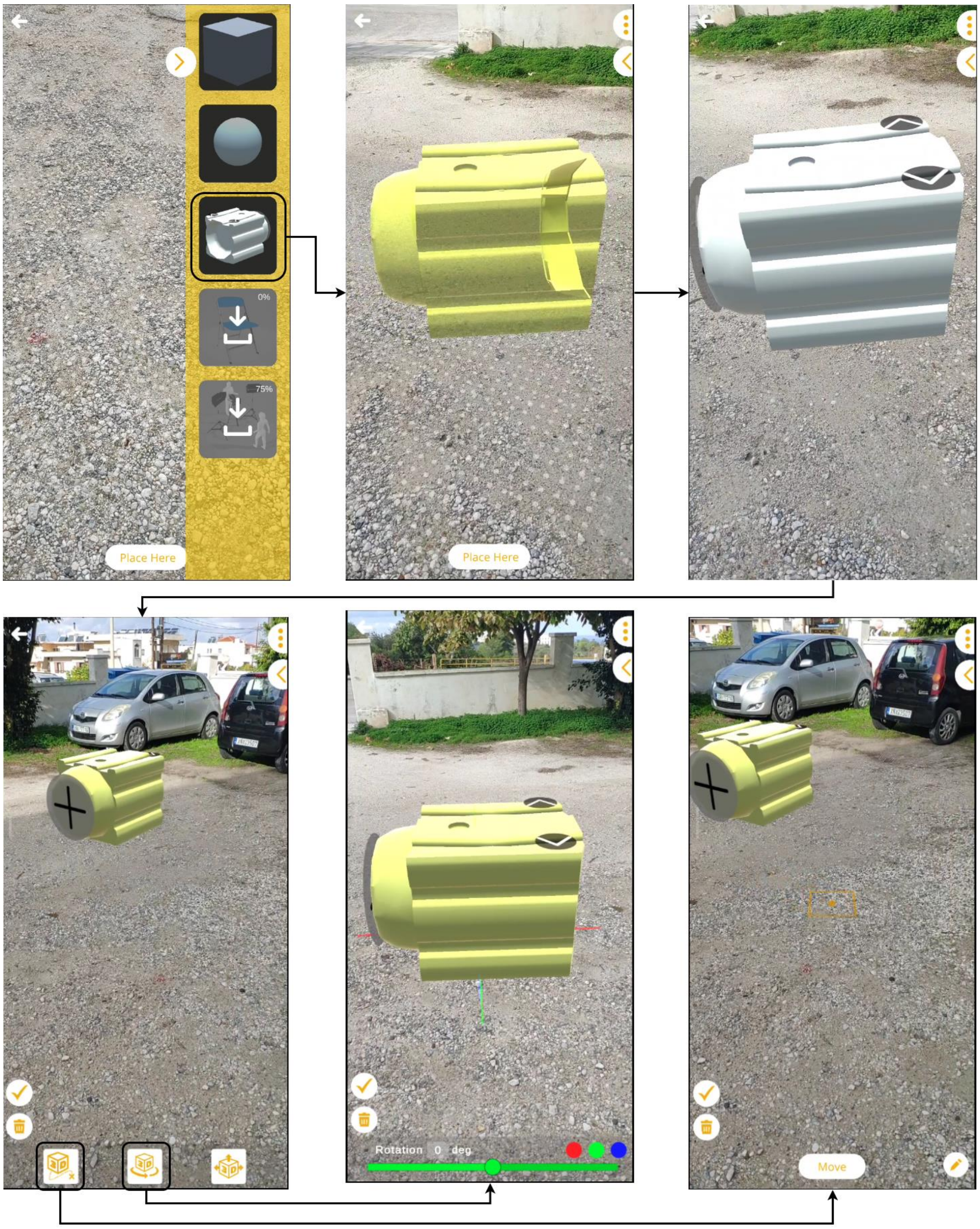

Figure 18: AR Mode: User places 3D model and moves/rotates it

3.9.6 **3D Model Media Upload**

This feature allows users to upload media files (e.g. screenshots or videos) that document the placement of 3D models in the real-world environment. This upload functionality can be accessed from the tools panel in the upper-right corner, which opens a menu of available media options.

Once the collection of media is complete, a new POI is created with the exact geo-location of the user at the time of the upload, under the category *"Production Planning"* and subcategory *"Snapshots".* The media can be viewed both on the map and in the "AR POI View".

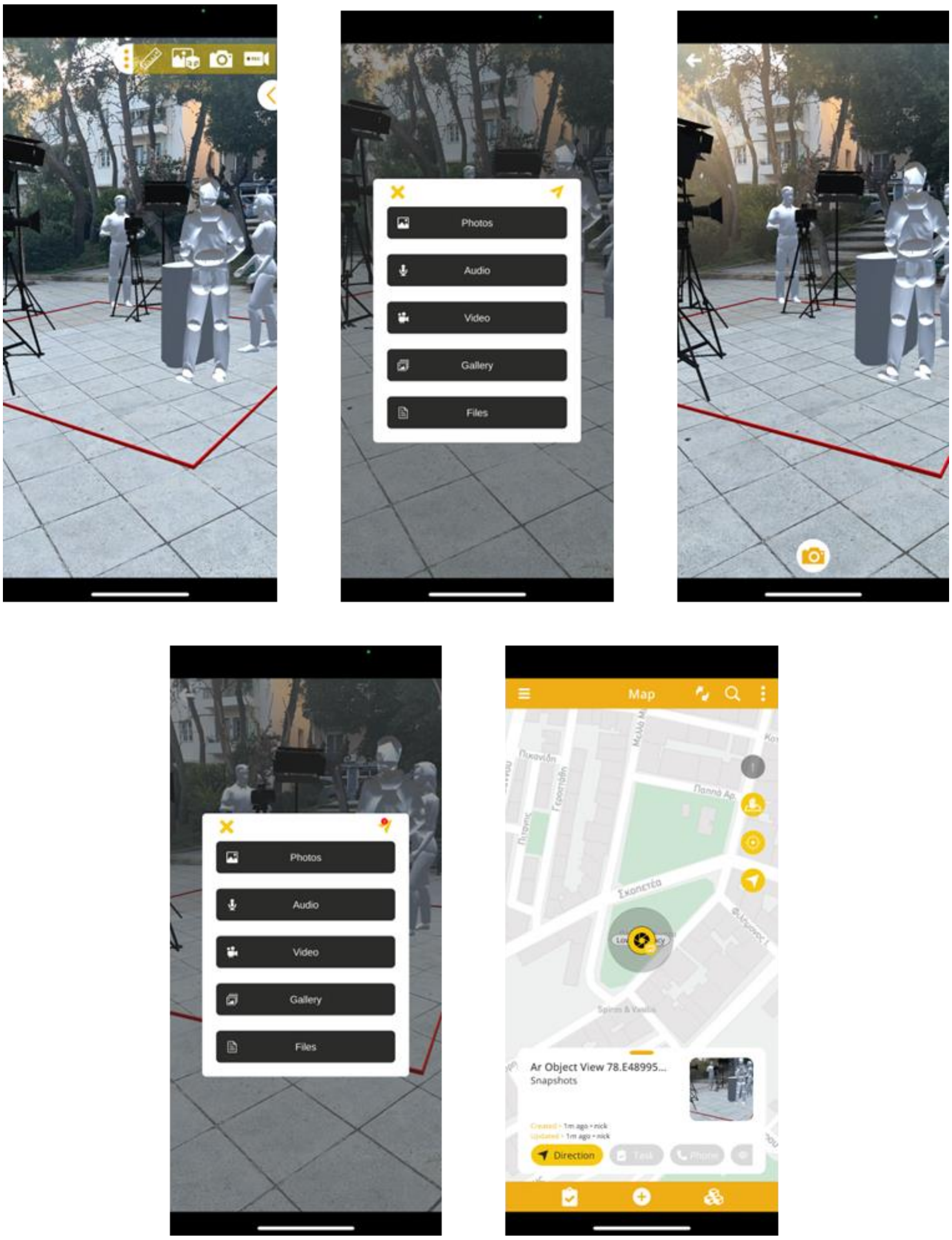

Figure 19: AR Mode: User uploads media showcasing the position of virtual models into the real world

3.9.7 **AR Measuring**

The measuring tool in the application allows the user to measure distances in the real world by creating measuring lines with multiple points. The user can access this tool from the toolbar on the top right. The summary of the line is displayed on the bottom right corner; multiple lines can be used.

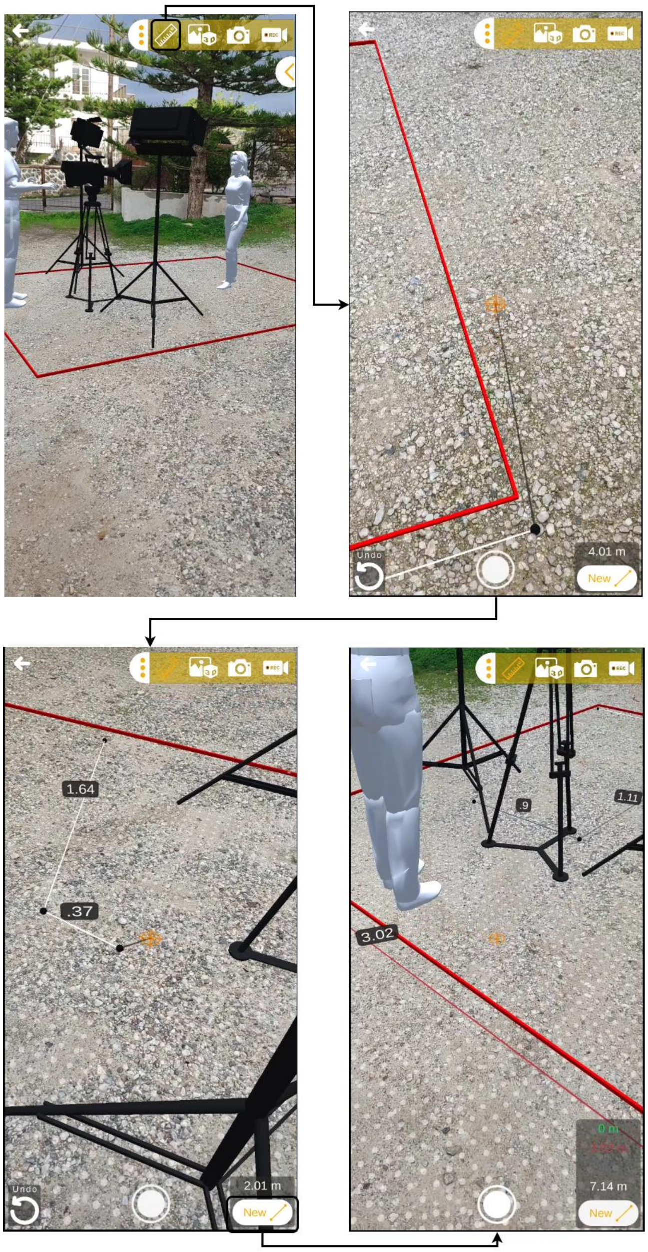

Figure 20: AR Mode: User takes measurements

3.10 Detect Outdoor Scenery

The XR4DRAMA AR app supports real-time outdoor scene object detection. The following steps are required to access this feature:

- Navigate to the map view within the application
- Press the AR button located at the bottom of the screen
- Press the object detection button to enter the object detection mode

Once inside the object detection mode, the user has the option to utilize the utility toolbar to capture a screenshot or record the screen view.

Figure 21: Outdoor scenery object detection mode

4 IMPLEMENTATION

4.1 Hardware

The XR4DRAMA AR app was deployed on a mobile phone after considering users preferences and testing capabilities. Mobile phones seamlessly offer GNSS and other sensors, as well as integrated connectivity to the internet. They are also easy to provide and cost effective, at least compared to a dedicated hardware solution like the HoloLens headset.

Furthermore, it was proven after relevant testing and prototype development that the outdoor usage of holographic devices is uncomfortable in bright sunlight. On the other hand, headsets provide a more immersive experience and give the user free hands (because they do not have to hold a device). However, the use of a headset during an outdoor procedure that requires extensive movement carries many risks and safety issues.

The minimum requirements for a mobile phone device to run the XR4DRAMA app are listed in Table 5.

	Requirements (Android)	Requirements (iOS)
Operating System	Android 8.0 (Ver. 26) +	IOS 14+
CPU power	1,8 GHz or higher	1.8 GHz or higher
CPU model	Snapdragon 636 or higher	A9 or higher
RAM	3 GB or higher	2 GB or higher
Disk storage	16 GB or more	16GB or more
Battery storage	4000 mAh	3000 mAh
Internet connection with or without WIFI	YES, 4G connection at least	YES, 4G connection at least
AR Support	ARCore (Required)	ARKit (Required)
Depth Support	Depth API (Optional)	Lidar devices (Optional)

Table 5: Minimum mobile phone system requirements for the ARapp

4.2 Software

4.2.1 AR Technology Stack

Figure 22 illustrates the technology stack for the AR app. The Unity XR plugin offers a comprehensive set of Unity packages that enable native integration with VR and AR platforms and devices. It provides a common API for accessing VR and AR devices and allows developers to create cross-platform experiences with minimal modifications. It also supplies a set of XR-specific features and tools for creating immersive experiences, such as support for tracking, input, and rendering.

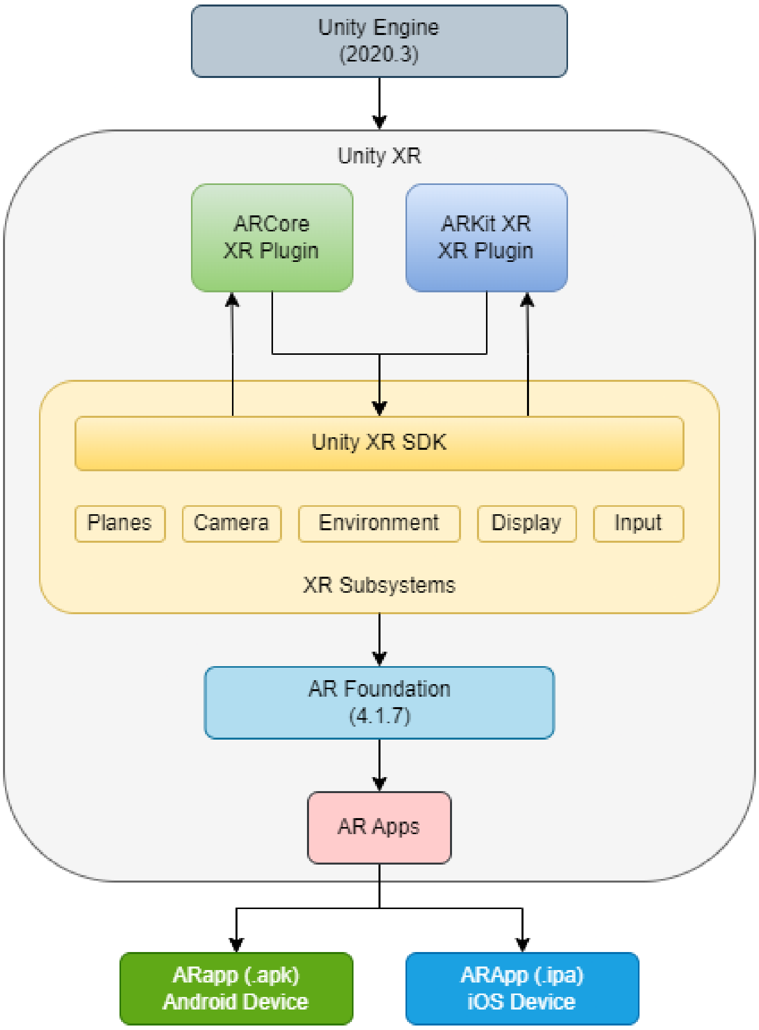

Figure 22: AR Technology Stack Diagram

The Unity AR Foundation allows for the development of AR experiences on both iOS and Android platforms using the Unity Engine. It provides a common API for accessing ARKit and ARCore and can be utilized to create cross-platform AR experiences with minimal modifications. It provides a wide range of features and tools for creating immersive AR experiences such as:

- Support for image tracking, plane detection, and point cloud visualization.
- Support for lighting estimation, which helps to match the lighting of virtual objects to real-world lighting conditions.
- Support for environmental probing, which allows the creation of realistic shadows and reflections on virtual objects.
- Support for face tracking.

The AR app is an AR application based on both ARCore and ARKit plugins and is compatible with both Android and iOS devices through the Unity XR plugin.

4.2.2 AR POIs System

The AR Foundation requires several key components for tracking the environment, with the most vital component being the **AR Session Origin**, which serves as the foundation of the AR tracking system. This component manages the **AR Session**, which is responsible for tracking the device's movements and anchoring virtual objects in the real world.

In addition to the AR Session Origin, plane detection also requires an **AR Plane Manager** component. This component is responsible for detecting and tracking flat surfaces in the real-world and is an essential component for placing virtual objects on a surface. The AR Plane Manager leverages data from the AR Session to track surfaces and can be configured to detect specific types of surfaces, such as horizontal or vertical planes. To ensure accurate plane detection, it is crucial to have a clear view of the environment and sufficient lighting.

The system combines GPS data from the device with tracking performed by AR Foundation. The AR tracking adjusts the camera's position to align with the real-world movements of the device. The system refines the position and orientation estimations from the GPS sensor in relation to the user’s camera. To accomplish this, it requires an object known as the **AR Location Origin** in the scene. This object must be part of the **AR Camera** used by the AR tracker and should contain both an **AR Location Manager** and an **AR Location Orientation** component.

These components ensure that the AR Location Origin is aligned with geographical directions, which allows for accurate positioning of objects. A component referred to as the **Location Provider** handles the incoming **GPS and Compass** data from the device. All components responsible for positioning objects in the scene listen to events emitted by these foundational components to make updates. Figure 23 depicts the system’s architecture.

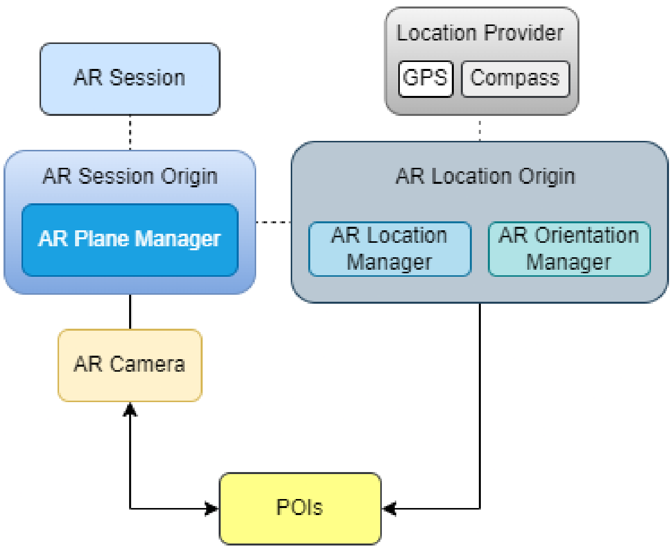

Figure 23: AR POIs Architecture

4.2.3 AR Objects System

The **Occlusion Manager** is a component in the AR Foundation framework for building AR experiences and is responsible for managing occlusion, which refers to the process of hiding virtual objects that are behind real-world objects. In the AR experience, the Occlusion Manager leverages depth information from the device's sensors and cameras to determine the relative positions of virtual and real-world objects. If a virtual object is behind a real-world object, the Occlusion Manager will hide or partially hide the virtual object, creating a more realistic and immersive experience for the user.

Download Manager is a system responsible for requesting from the server the available models in the current project. The server responds with the location of the models in the database. Using the link, the system downloads and imports on runtime the models from the server and saves them on the local storage for later use. The manager adapts the downloaded models to ensure the best and most accurate usage.

AR Objects Manager is responsible for object placement and manipulation in the scene. It gathers the data from AR Plane Manager and using the AR Raycast Manager determines the position of the model. **AR Raycast Manager** leverages data from the device's sensors and cameras to track the user's gaze, and casts a ray from the device's position into the real world. This ray is used to detect real-world objects and surfaces, in order to place virtual objects in the scene. The system is presented in Figure 24.

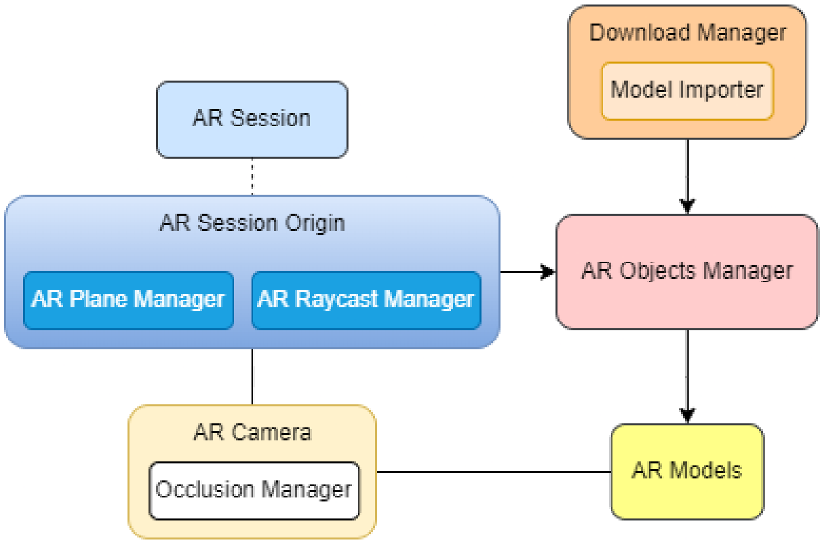

Figure 24: AR Objects Architecture

4.2.4 Search

The search engine in the application allows users to search for points within the designated project area. As the user types in their search query, the application transmits the characters to the server and receives a list of POIs that match the user's input. This list may also include names of roads, villages, and municipalities. The application presents the list to the user, allowing them to choose from the available results. Upon selection, the related point on the map is visualized.

4.2.5 Share Location

The Share Location service communicates with the server through web sockets. The first user creates a unique channel for the project, and all subsequent users subscribe to this channel. The application sends updates of users' locations to the channel every 5 seconds and broadcasts this information to all other connected users. The application also manages events related to adding or updating user locations and disconnecting users who leave the channel.

In particular, the following events occur:

Open Connection: The user is successfully added to the channel and gets updates on the existence of other connected users.

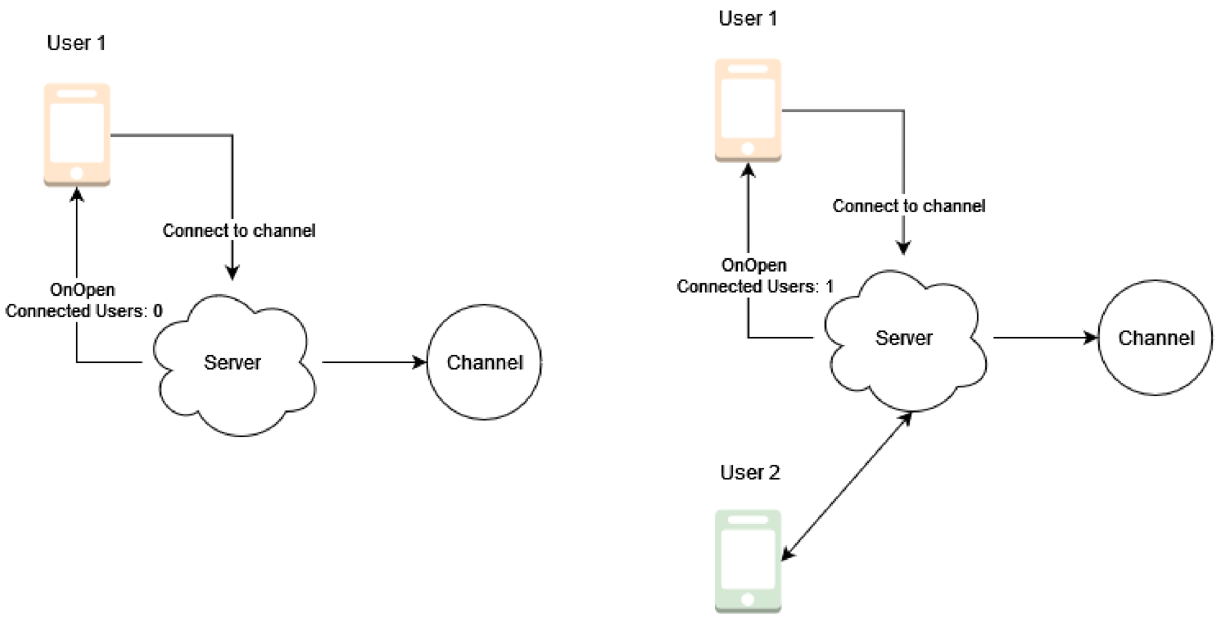

Figure 25: ARapp open connection

Add User: The user who has already been added can view the location of a new user when he subscribes to the same channel.

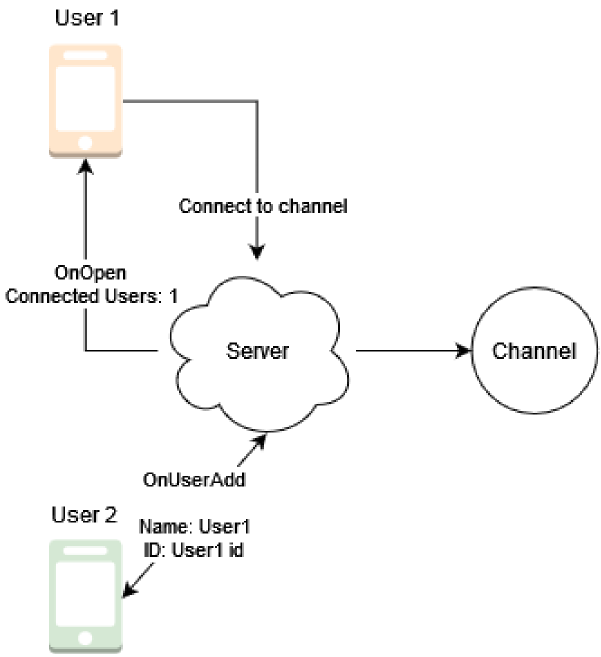

Figure 26: AR app is updated for the existence of new connected user

Update Location: The user’s location is updated, and the application updates the user's icon on the map of each connected user. Update occurs every 5 seconds.

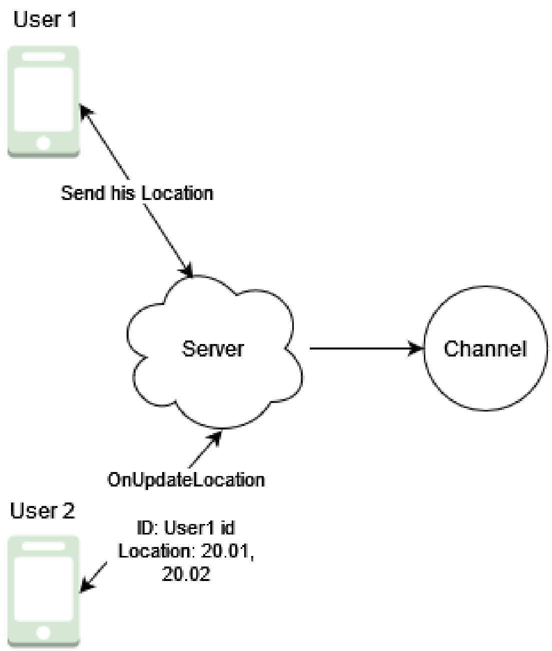

Figure 27: AR app send updates for the location of other users in the project

Leave: The user leaves this channel, and the application removes the user's icon from the map of active users. This user no longer receives updates from other connected users.

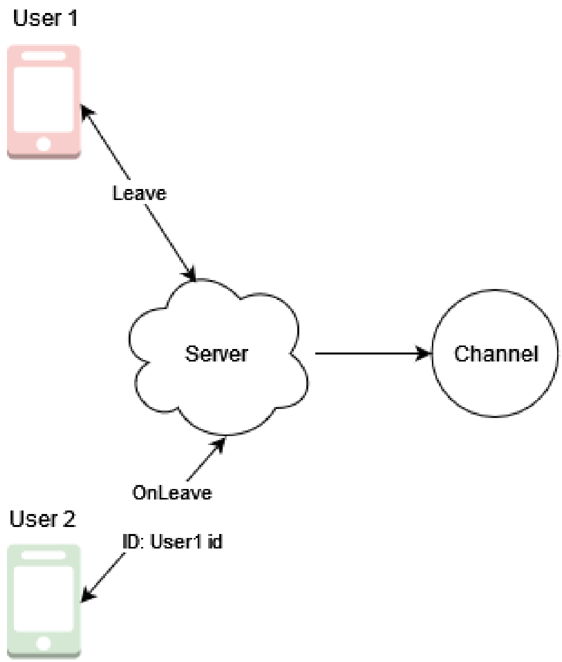

Figure 28: AR app stops streaming the location of a user that left the channel

Note that if the application goes into the background, it will automatically disconnect from the channel and reconnect once the application comes back into focus again. While the app is not active other users won't be able to view the location of this specific user.

4.2.6 Navigation

The Navigation feature of the application allows users to find the most efficient route to their destination. The application sends the current location of the user and their desired destination, along with their preferred route type, to the server. The server then returns a list of nodes forming the shortest path, the total distance, and estimated travel time based on the chosen route type. The application uses these nodes to display the path on the map, and a panel shows the updated distance, travel time, and estimated arrival time as the user moves.

However, it should be noted that the accuracy of the GPS on mobile devices is limited, with an average accuracy 4-5 meters, and can be influenced by factors such as the device's hardware, the surrounding environment, and weather conditions.

The initial implementation of the navigation feature relied solely on raw GPS location data, which was found to be unreliable and resulted in inaccurate calculations of distance and travel time, making it difficult to determine the user's exact location along the route.

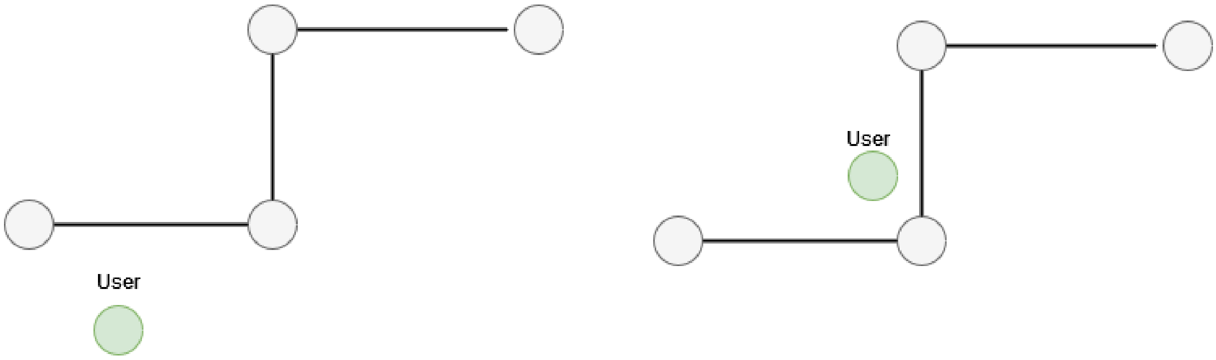

Figure 29: AR app Navigation without projection

To improve the accuracy of the navigation, an algorithm that projects the user's location onto the path has been implemented. This algorithm calculates the percentage of distance travelled by the user and updates the display of the distance and travel time accordingly. The short path is a line of nodes. The nearest node of the route is determined, and if the user is near the first or last node, they are considered to be on the start or ending line. Finally, the projection is calculated using the user 's location and the line formed by the nearest node and the next of the nearest node.

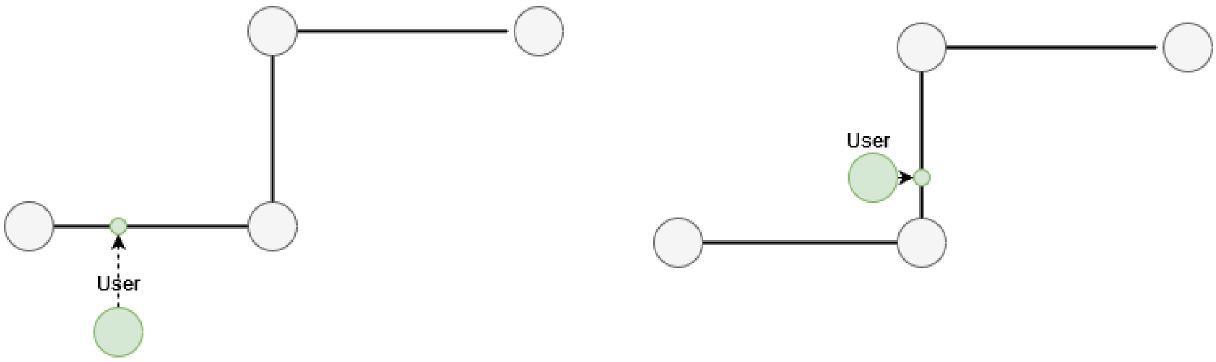

Figure 30: AR app Navigation with projection

4.2.7 **GPS**

GPS on mobile devices is subject to limitations in terms of accuracy (typically limited to over 4 meters), which can be affected by various factors such as the generation of the device, outdoor conditions, and weather. However, it has been observed that coordinates with an accuracy of less than 20 meters provide a sufficient approximation of the actual location. To address this issue, the AR application implements a restriction on the use of GPS coordinates, utilizing only those with an accuracy of less than 20 meters.

Due to the frequent update of metrics on mobile devices every 1-5 seconds, the use of median and average algorithms is not deemed suitable for this application. This results in a slow processing time to obtain an accurate representation of reality. The frequency of new coordinates depends on the specific mobile device.

In AR mode, GPS continuously updates the position of the POIs. Even if GPS measurements are good, the orientation might seem a little off. The application calculates the orientation of the device based on compass metrics which seem to have large inaccuracies. In some devices the compass only works properly under specific conditions e.g., the phone being in vertical orientation. To counter this, the update of the POIs only takes place after sufficient data have been received in a small frame of time, leading to a smoother experience. The compass is updated continuously with a range of metrics. The application applies the orientation based on the median of 100 discrete measurements.

4.2.8 **File Manager**

To ensure compatibility with both Android and iOS, a mechanism was implemented for the hybrid implementation of the AR app. The photos, videos, and audio recordings created through the AR app are saved in two directories in the device file manager: the standard gallery, where the user keeps a media backup on his device, and the ‘persistent data’ directory, an individual folder for each application. The files in the ‘persistent data’ directory are used for uploading to the server and are deleted once the upload is successful.

The media files retrieved from the gallery for upload purposes have different implementations for its device platform. For instance, Android provides the original paths of selected media for uploading, whereas iOS creates a copy of the selected media in the “persistent data” directory and provides the paths of copies of files.

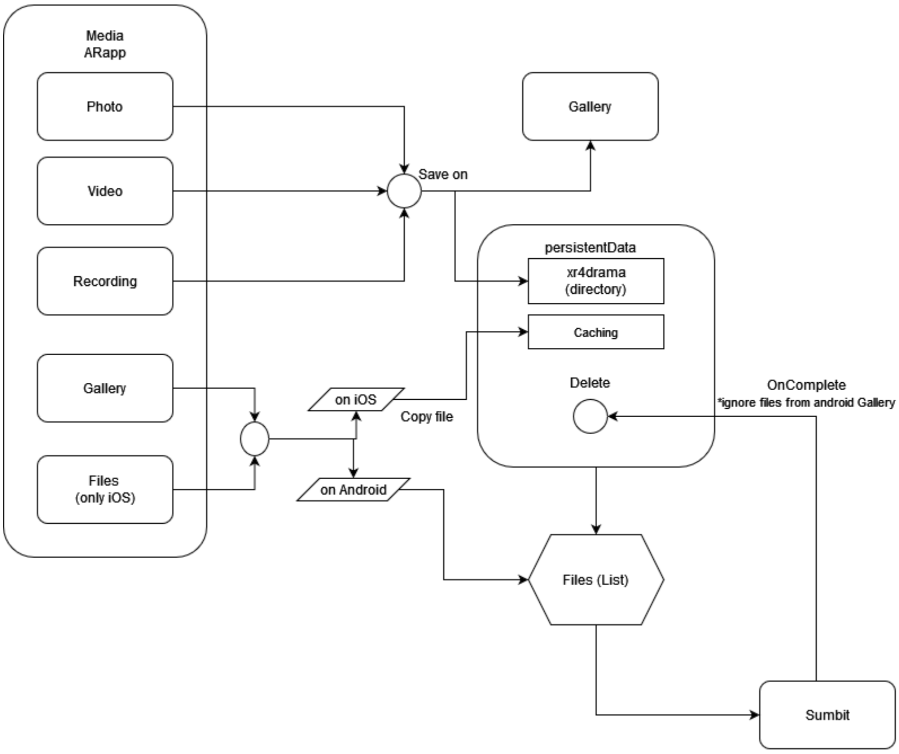

Figure 31: File Manager Architecture

4.3 Communication with the GIS Service

The communication process between the AR app and the GIS Service is illustrated in the following figures. Figure 32 shows the communication process between the AR app and the GIS Service when a user creates a new POI or edits/adds information to existing POIs. The GIS Service has POIs that have been created through geoportals and the user has the option to modify the existing information and add additional information.

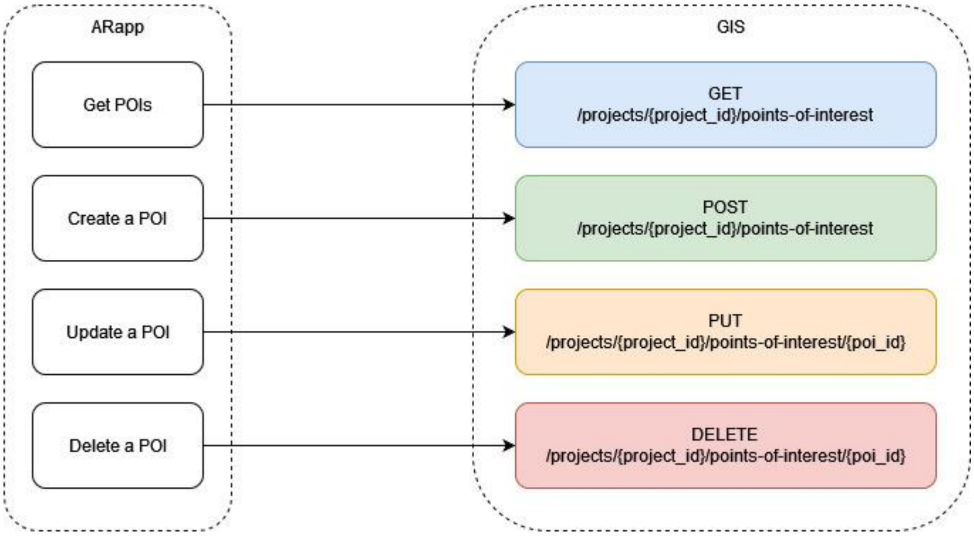

Figure 32: Get POIs and Create, Update Delete a POI

Figure 33 despicts the communication process between the AR app and GIS Service when uploading a media file or adding a comment to existing POIs. Users can gather media and feedback related to the POIs and share them with the application, which supports various types of media files such as photos, videos, audio recordings, and text files. These files, along with user comments, will be made available to all users once successfully uploaded.

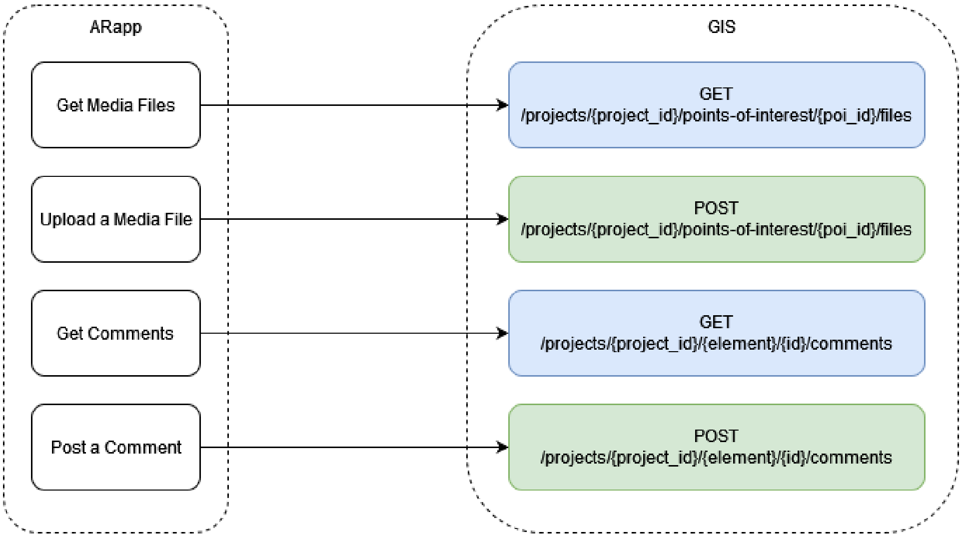

Figure 33: Get, Create Media and Comments

Figure 34 illustrates the interaction between the AR app and GIS Service when a user manages a task. The mobile application allows registered users to view tasks assigned to them and tasks without an assigned worker, which can be claimed by any user. The user can view the details of the tasks and the task supervisor can monitor its progress, while the user can update the task status. Upon completion of the task and its status update, the assigned worker must submit a report with a description and an audio report.

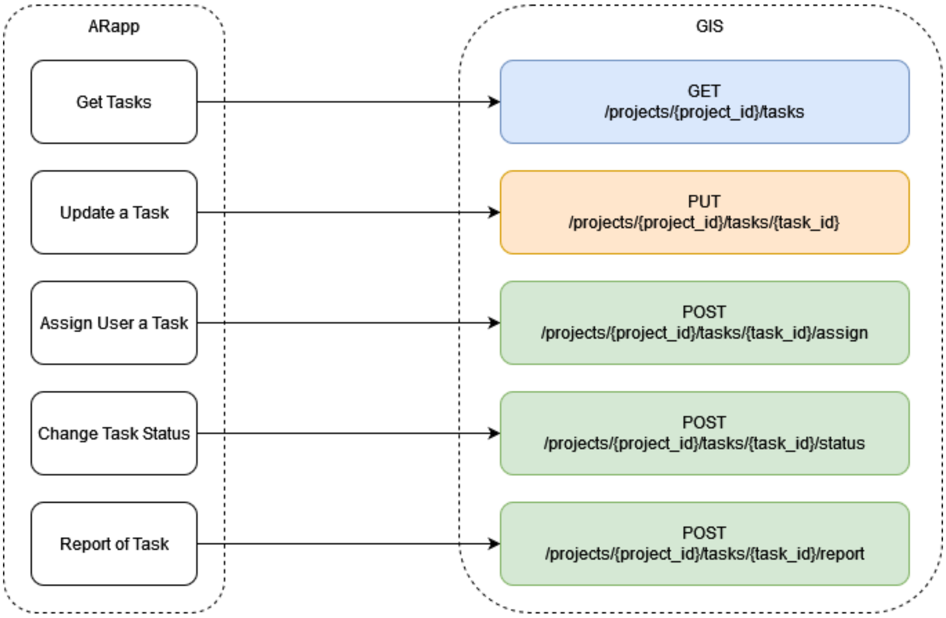

Figure 34: Tasks Management

4.4 Outdoor Scene Detection

The Outdoor Scene Detection feature of the XR4RAMA AR app is designed to provide real-time detection of objects in outdoor environments. The detection is achieved through the use of the phone camera, which feeds images into the detection module that provides real-time feedback on the scene decomposition to assist the user in their current surroundings.

The implemented object-detection system consists of two discrete software modules:

- The python-based object detection research module, which aims to develop and finetune the scene detection algorithm.
- The Unity interface module, which is responsible for seamlessly integrating the trained object detection model in the XR4DRAMA AR app.

4.4.1 **Object detection review**

Object detection is a computer vision and image processing technique that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Methods for object detection generally fall into either neural network-based or non-neural approaches.

Architecture

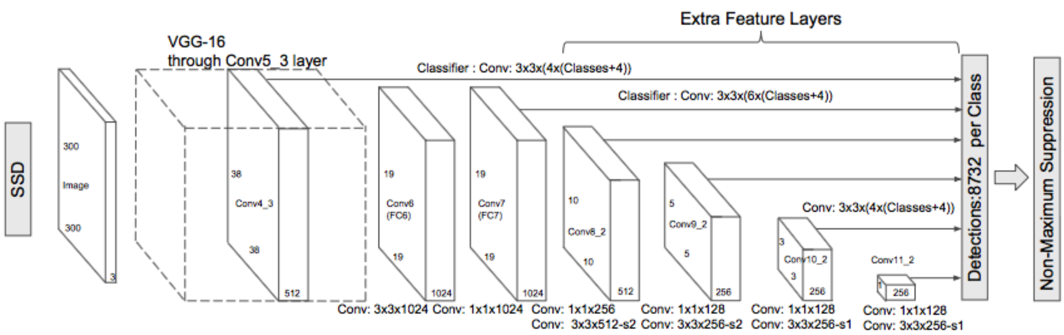

Figure 35: Model Architecture

The SSDLite neural network architecture was selected for the implementation of the object detection functionality. This neural network architecture is specifically tailored for mobile and resource constrained environments. It takes as an input a low-dimensional compressed representation which is first expanded to high dimension and filtered with a lightweight depthwise convolution. Features are subsequently projected back to a low-dimensional representation with a linear convolution.

In general, the lite version is a replacement of all the regular convolutions with separable convolutions (depthwise followed by 1×1 projection) in SSD prediction layers. This design is in line with the overall design of MobileNets and is seen to be much more computationally efficient. Compared to regular SSD, SSDLite dramatically reduces both parameter count and computational cost.

This convolutional module was selected since it is particularly suitable for mobile designs. It allows to significantly reduce the memory footprint needed during inference by never fully materializing large intermediate tensors. This reduces the need for main memory access in many embedded hardware designs that provide small amounts of very fast software-controlled cache memory.

4.4.2 Datasets

Object detection requires well-curated datasets to train and evaluate models. Two commonly used datasets in the field are the PASCAL VOC 2012 dataset and the BDD100K: A Large-scale Diverse Driving Video Database. Another widely used dataset is Cityscapes.

4.4.2.1 VOC: Visual Object Classes

The PASCAL VOC 2012 dataset contains 20 object categories including vehicles, household items, animals, and others. Each image in this dataset includes pixel-level segmentation annotations, bounding box annotations, and object class annotations. The dataset has been widely used as a benchmark for object detection, semantic segmentation, and classification tasks and is split into three subsets: 1.464 images for training, 1.449 images for validation and a private testing set. The classes included in the dataset are: airplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, TV/monitor, bird, cat, cow, dog, horse, sheep, and person.

Figure 36: VOC dataset sample

4.4.2.2 BDD100K: A Large-scale Diverse Driving Video Database

BDD100K is the largest driving video dataset with 100K videos and 10 tasks for evaluating the progress of image recognition algorithms in autonomous driving. The dataset possesses geographic, environmental, and weather diversity, making it useful for training models that are less likely to be surprised by new conditions. BDD100K covers more realistic driving scenarios and captures more of the “long-tail” of appearance variation and pose configuration of categories of interest in diverse environmental domains. The categories included in the dataset are pedestrian, rider, car, truck, bus, train, motorcycle, bicycle, traffic light, and traffic sign.

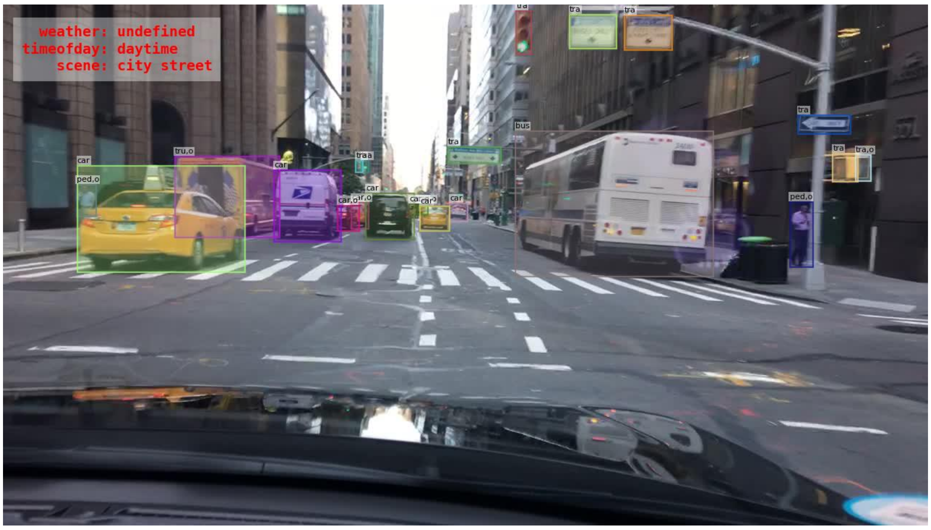

Figure 37: BDD100k dataset sample

4.4.2.3 Cityscapes

Cityscapes is a dataset comprised of a large, diverse set of stereo video sequences recorded in the streets of 50 different cities. 5000 of these images have high quality pixel-level annotations; 20000 additional images have coarse annotations to enable methods that leverage large volumes of weakly-labeled data. The categories included in the dataset are person, rider, car, truck, bus, train, motorcycle, and bicycle.

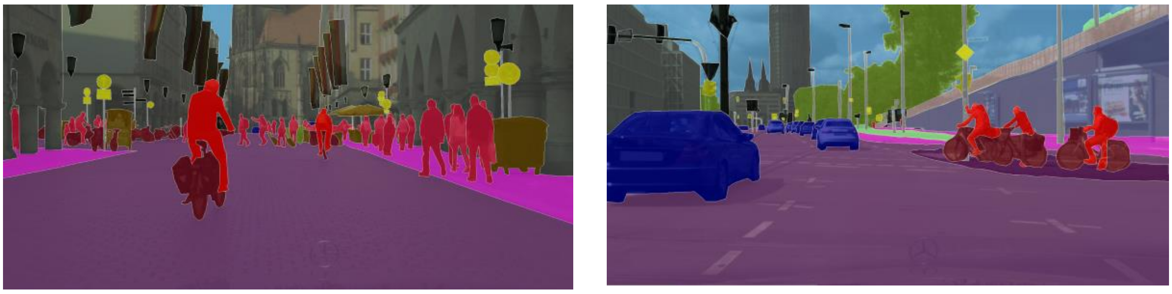

Figure 38: Cityscapes dataset sample

4.4.3 Training Dataset

The training set used for the object detection module of the AR app consists of VOC and BDD100K transformed into VOC format. After merging the two datasets, the instances of each class in the training set is described in the following table.

Class	Merged training set
train	792

truck	29971
traffic light	186117
traffic sign	239686
rider	4517
person	101478
bus	12309
bike	8000
car	713211
motor	3753

Table 6: Instances of each class in the training set

4.4.4 **Experiments**

Experiment with learning rates

Experiments with learning rates were conducted in order to monitor and evaluate the training process of a neural network. The Tensorboard tool was used to track and visualize various metrics, such as loss, during the training process. Initial results indicated that the suggested learning rate (0.01) resulted in an abnormal trajectory of validation loss, as shown in Figure 39. However, upon dividing the former learning rate by 10, a monotonic validation loss was observed, resulting in a smoother training curve. The class-wise results of the experiment can be viewed in the following chart.

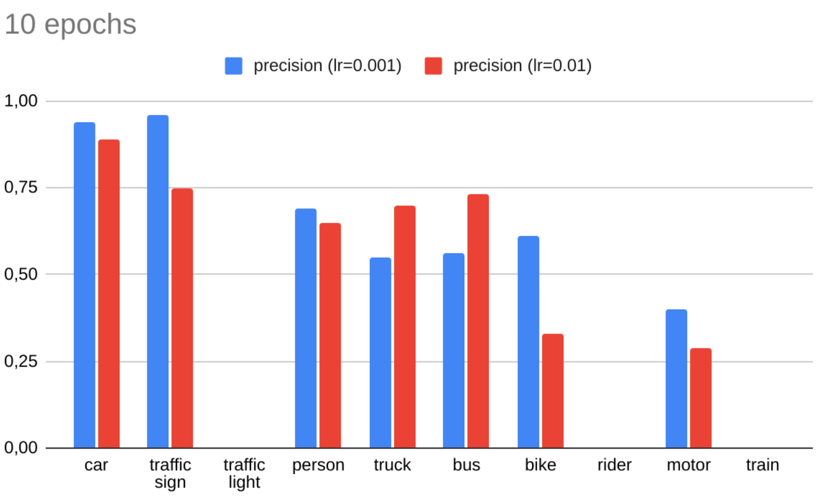

Figure 39: Learning rate experiments

Experiment with complex model

Mobilenet SSD

In order to identify whether the selected neural network model (SSD Lite) has reached a bottleneck during the training process, it was compared and evaluated against a larger and more complex model (Mobilenet SSD). This comparison showcased that the selected model needed further training.

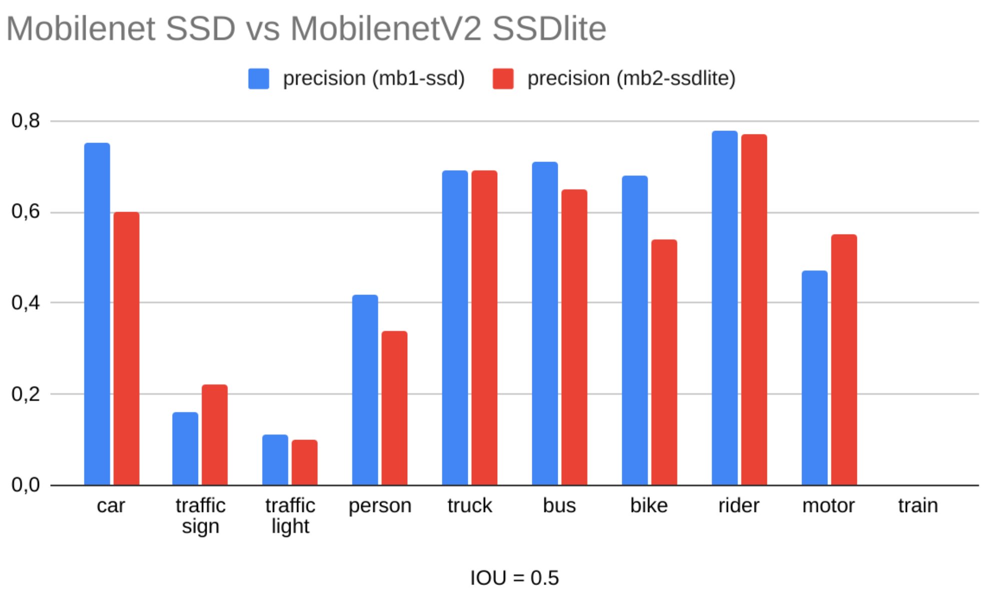

Figure 40: Backbone architecture experiments

Experiment with optimizers

Additional experiments were conducted using different optimizers.

SGD optimizer

SDG is an iterative method for optimizing an objective function with suitable smoothness properties. It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data).

Adam optimizer

Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iteratively based on training data. The optimizer is called Adam because it uses estimations of the first and second moments of the gradient to adapt the learning rate for each weight of the neural network. Adam is proposed as the most efficient stochastic optimization which only requires first-order gradients where memory requirement is too little.

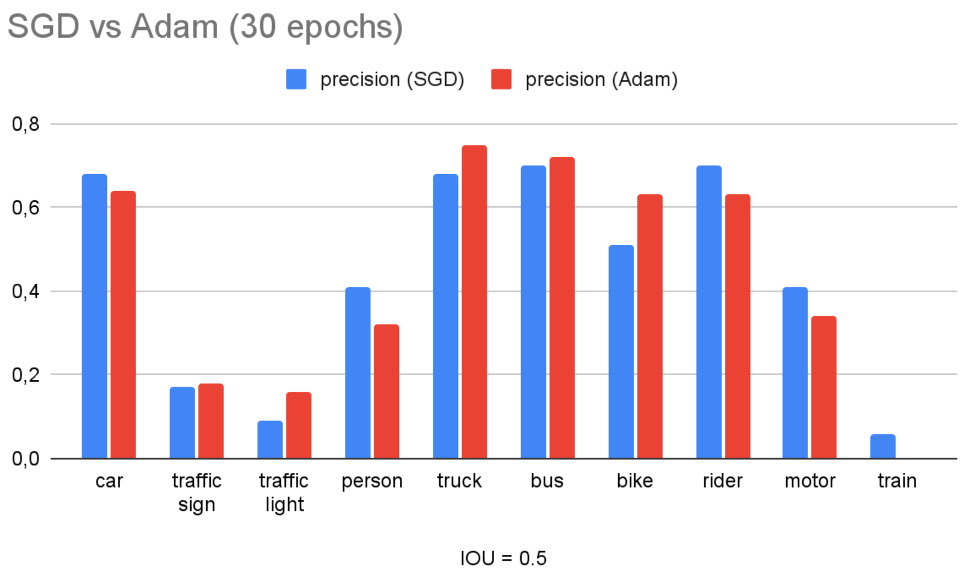

Figure 41: Learning optimizers experiments

In our case, the Adam optimizer failed to learn the train class, which has fewer data than every other class in our dataset. Therefore, the Stochastic Gradient Descent (SGD) optimizer was selected.

Experiment with dataset samplers

The custom dataset that we exploit contains classes with a few instances and therefore some classes cannot be detected. To overcome this deficiency, a sampler that is more likely to sample from low frequency classes was used. The weights of the sampler were calculated with the following methods:

Summing all weights

The weight of each image is the sum of all weights of the image.

Minimum of all weights

The weight of each image is the weight of the most infrequent object in the image.

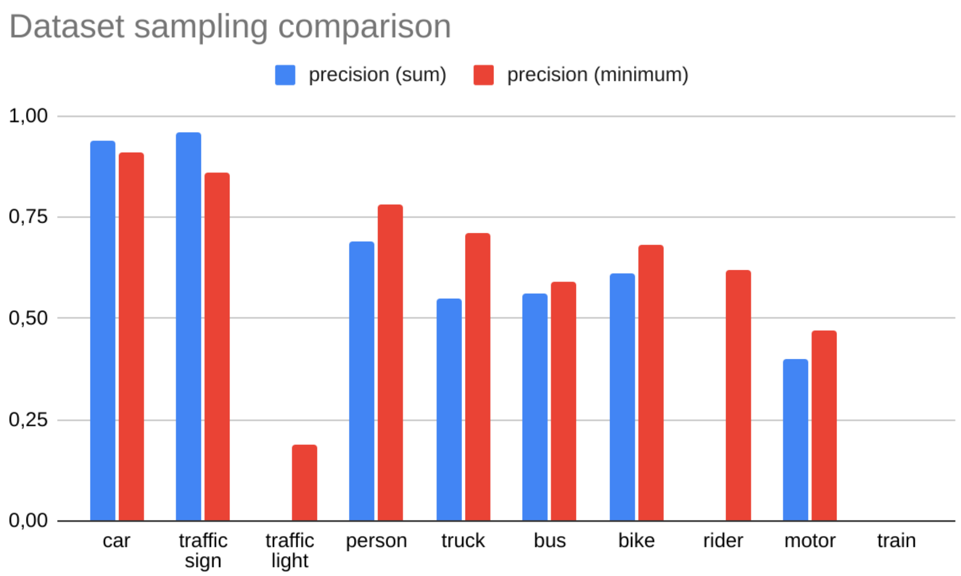

Figure 42: Dataset sampling experiments

The disadvantage of summing the weights is that if an image contains the most infrequent object and the most frequent, the weight that is calculated is neutral. This is the reason for choosing the minimum weight for each image and proof of this claim is that using the minimum weight, the traffic light class can be predicted, whereas if we prefer to sum the weights, traffic lights are left undetected.

Experiment with losses

Another experiment was to adjust the weights used in the Loss estimation function to favor either the classification loss or the regression loss.

Loss = w1 Lclass +w2 Lreg

Classification loss

Training loss is calculated in favor of classification loss.

The weights used were w1 = 0.8 and w2 = 0.2.

Regression loss

Training loss is calculated in favor of regression loss.

The weights used were w1 = 0.2 and w2 = 0.8.

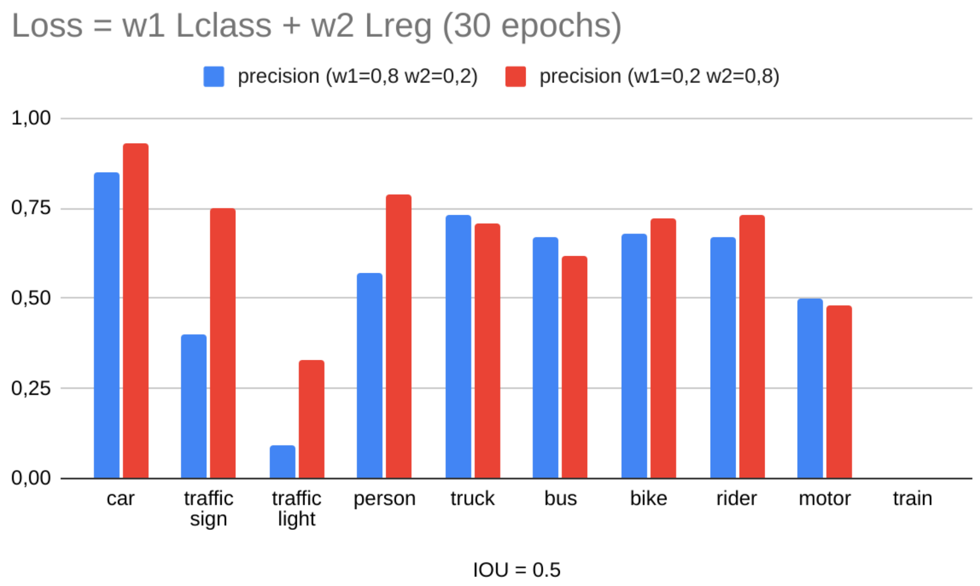

Figure 43: Loss functions weights experiments

Judging by the results, the detection of classes that contain a small amount of images are enhanced when the total loss is more focused on the regression loss. This may occur because the model is more prone to error when calculating the bounding boxes.

4.4.5 **Unity Interface**

In order to interconnect the trained object-detection model with the XR4DRAMA AR app, NatML deep learning Unity API was used. NatML is a highly efficient deep learning API for c#, that allows for seamless integration of deep learning models on mobile and desktop Unity applications.

The implemented Unity architecture that provides a real-time webcam feed to the detection module, processes it and visualizes the results on-screen, is presented below:

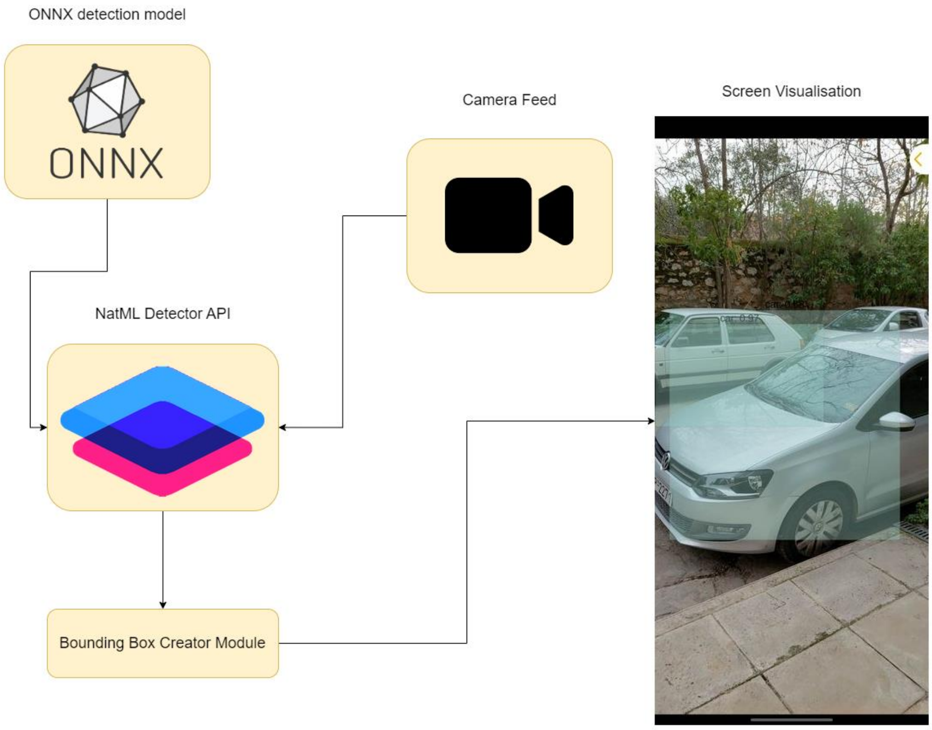

Figure 44: AR app architecture for outdoor scene detection

- The web camera module is streaming constantly and feeds image data to the NatML detector
- The detector pre-processes the provided input images and predicts normalized bounding box vectors corresponding to the trained categories
- The bounding-box-creator module accepts the normalized prediction vectors, constructs the visualized bounding boxes and scales them to screen size
- The ending result is displayed on-screen in realtime and shows the predicted objects of the outdoor scene

5 DEMO

Name/Version	Link
ARApp_final_version_demo.mp4	https://1drv.ms/u/s!Ao1ZozKtfn2JooQWDRAe9hWy2bL8kQ?e=rYi4sN

Table 7: Demonstration Videos

6 CONCLUSION

In conclusion, this technical deliverable presents the development and implementation of a mobile AR application (for outdoor use) with object detection functionalities to support both disaster management and media planning. The objectives and user requirements were identified and successfully fulfilled through the application's various use cases.

The report provides an overview of the AR frameworks used in the development of the application. The application provides various functionalities to aid in the planning and management of different scenarios of disaster and media planning. The hardware and software used in the implementation of the application as well as the communication with the GIS Service were also discussed in detail.

In addition, the demonstration of the application showcases its effectiveness in addressing the challenges faced in the XR4DRAMA use cases and providing a comprehensive solution for improving actions and tasks in the field. The integration of AR technology proves to be a valuable tool for first responders in disaster management and location scouts in media planning, providing real-time information and improving the accuracy of decision-making processes.

All in all, the development of this AR application represents a significant advancement in the fields of disaster management and media planning. The app provides an innovative solution to the challenges faced in these fields, has the potential to revolutionize the way they are approached, and may serve as a model for future developments.

A Appendix

XR4DRAMA AR app flowchart