

# Air Pollution Awareness via Photo Sharing (AP)

aka. AirAwareness

System Document by AreYouWorking Team

Computer Engineering, Chiang Mai University

## สมาชิกทีม

1. ธนาดล เดชประภากร 630610734 Developer
2. ธนาวัฒน์ บำเพ็งพันธุ์ 630610736 Tester
3. ธนาตถ์ ตั้งอัน 630610737 Developer
4. ภูริช สินวลแล 630610752 Develop, UX/UI Designer
5. พงศกร รัตนพันธ์ 630610749 SA, Project Manager

Team's website: <https://areyouworking.github.io/>

## Stakeholders

1. รศ.ดร.สรรพวรรณ กันตะบุตร
2. ผศ.ดร.ยุทธพงษ์ สมจิต
3. ผู้ที่ประสบปัญหามลพิษทางอากาศ

## นำเสนอด้วย

ผศ.ดร.ลักษนา ระมิงค์วงศ์  
วิชา 261361 Software Engineering  
ภาคเรียนที่ 2 ปีการศึกษา 2565

# สารบัญ

<b>A1 -Project Proposal</b>	<b>1</b>
<b>A2- Software Requirements Specification</b>	<b>11</b>
1. Introduction	13
2. Overall Description	15
3. External Interface Requirements	20
4. System Features	21
5. Other Nonfunctional Requirements	23
Appendix A: Work Plan	25
<b>A3 - Design Specification (A-DAPT)</b>	<b>28</b>
<b>A4 - Sprint #1 (Presentation slides and Video)</b>	<b>29</b>
<b>A5 - Test Report</b>	<b>52</b>
<b>A6 - Sprint #2 (Presentation slides and Video)</b>	<b>62</b>

# Air Pollution Awareness via Photo Sharing (AP)

## A1-Project Proposal

by AreYouWorking Team

Computer Engineering, Chiang Mai University

### สมาชิกทีม

6. ธนาดา เดชประภากร 630610734
7. ธนาวัตถ์ บำเพ็งพันธุ์ 630610736
8. ธนาตถ์ ตั้งอั้น 630610737
9. ภูริช สีนวลแล 630610752
10. พงศกร รัตนพันธุ์ 630610749

Team's website: <https://areyouworking.github.io/>

### Stakeholders

4. รศ.ดร.สรรวรรณ กันทะบุตร
5. ผศ.ดร.ยุทธพงษ์ สมจิต
6. ผู้ที่ประสบปัญหามลพิษทางอากาศ

### นำเสนอ

ผศ.ดร.ลัชนา ระมิงค์วงศ์

วิชา 261361 Software Engineering

ภาคเรียนที่ 2 ปีการศึกษา 2565

## สมาชิกทีม AreYouWorking

### 1. ธนาดา เดชประภากร 630610734 | Developer

- มีประสบการณ์พัฒนาเกมออนไลน์โดยใช้ Java Spring WebSocket, React
- มีประสบการณ์พัฒนา Android application

### 2. ธนาวัตน์ บำเพ็งพันธุ์ 630610736 | Tester

- มีประสบการณ์พัฒนาแอปพลิเคชันโดยใช้ภาษา Java, JavaScript, Python และ Framework React, Laravel, Svelte
- ออกแบบ UX/UI โดยใช้ Figma, Photoshop
- พัฒนา Front-end เว็บไซต์ด้วย HTML, CSS, TailwindCSS, Bootstrap

### 3. ธนาตถ์ ตั้งอ่อน 630610737 | Developer

- มีประสบการณ์เขียน web-app
- เขียน Rust, TypeScript, JavaScript, Python ได้

### 4. ภูริช สินวัลแล 630610752 | Developer, UX/UI Designer

- มีประสบการณ์เขียน Web-app โดยใช้ React
- ออกแบบ game UI โดยใช้ figma

### 5. พงศกร รัตนพันธุ์ 630610749 | SA , Project manager

- มีประสบการณ์เขียน web-app
- มีประสบการณ์เขียน Java, React, Laravel,
- LeanCanvas

## Project Description

### Air Pollution Awareness via Photo Sharing (AP)

Users can take or upload photos and customize whether to add PM2.5, AQI, heat map or other air pollution data to the photos before saving or sharing.

## Problem Statement

คนเชียงใหม่และคนไทยอยู่ใกล้กับฝุ่นควันมากกว่าที่คิด ฝุ่นควันและคุณภาพอากาศที่ไม่ดี เป็นอันตรายอย่างมากต่อสุขภาพ จึงต้องมีเทคโนโลยีที่ช่วยเพิ่มความตระหนักรู้ เทคโนโลยีที่มี ประสิทธิภาพด้านนี้มากที่สุดคือ Social Media ซึ่งสามารถแจ้งเตือนผู้คนในวงกว้าง ถึงคุณภาพอากาศในบริเวณต่าง ๆ ของเมืองได้ สร้างความรับรู้ในสังคม ตัวอย่าง เช่น Twitter ที่มี hashtag viral ได้ง่าย หรือ การแชร์ข้อมูลสภาพอากาศ ณ ปัจจุบัน ทาง IG story ที่มีความ personal

**Specific Issues** คนเชียงใหม่และคนไทยส่วนใหญ่ไม่ทราบว่าฝุ่นควันและอากาศที่ไม่ดี อยู่ใกล้ ตัวกว่าที่คิด

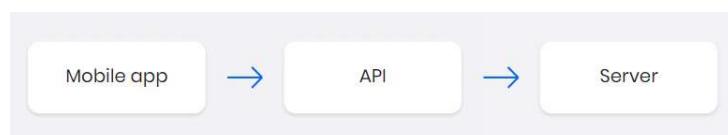
<https://www.bbc.com/thai/thailand-47157580>

<https://www.samitivejchinatown.com/th/health-article/PM-25-Effects-Your-Health>

## Solution

ในตลาดมีแอปพลิเคชันสำหรับการแบ่งข้อมูลสภาพอากาศ ณ ปัจจุบัน ลงบนรูปภาพ การแบ่งข้อมูลการวิ่ง แผนที่การวิ่ง ผลการวิ่ง บนรูปภาพแล้ว เช่น [InstaWeather](#), [runPhoto](#) แต่ยังไม่มีเรื่องของคุณภาพอากาศที่เป็นสิ่งสำคัญเลย การสร้างรูปภาพลักษณะนี้เป็นที่นิยมอย่างมากในสังคมนักวิ่ง มักเห็นได้ทั่วไปบน Facebook, IG

แอปพลิเคชันของเราสามารถแบ่งข้อมูล AQI, PM2.5, hotspot map ลงบนรูปภาพ เพื่อนำไปแชร์ต่อใน Social Media ต่าง ๆ โดยดึงข้อมูล AQI, PM2.5 จาก API (เช่น <https://aqicn.org/api/>) จากจุดวัดคุณภาพอากาศที่อยู่ใกล้มือถือของผู้ใช้งานมากที่สุดผ่านข้อมูล GPS ของมือถือ หรือจากข้อมูล Metadata (GPS) ในรูปภาพ



Air Pollution Awareness via Photo Sharing (AP)  
A1 - Project Proposal

## Contribution

ตาราง 1 : ตารางเปรียบเทียบฟีเจอร์ของ Weathershot, Run Photo, และแอพของเรา.

Features	Application		
	InstaWeather	Run Photo	Our application
ที่มาของข้อมูลสถานที่	Real-time GPS และตัวเลือกของสถานที่ที่แอปให้มา	จากข้อมูลที่ Strava หรือ Garmin ให้มา (น่าจะเป็น real-time GPS)	Real-time GPS และ meta-data ของรูปภาพ (Exif)
ข้อมูลที่แทรก	ภูมิอากาศของสถานที่ที่รู้ปุกถ่าย (ไม่มีมลพิษทางอากาศ)	ข้อมูลการวิ่งจาก Strava หรือ Garmin	ข้อมูล AQI, PM2.5, Hotspot map

ref: [InstaWeather](#) , [Run Photo](#)

เปรียบเทียบ UI ของ 3 แอปพลิเคชัน ด้านบน



## Stakeholders and User Group

Stakeholders ได้แก่

### 1. ผู้ใช้แอปพลิเคชัน

- ใช้งานแอปพลิเคชันเพื่อแชร์หรือบันทึกกรุ๊ปภาพ ใน Social Media บอกให้เพื่อน ๆ รู้ว่าวันนี้อากาศแย่ ฝุ่นเยอะ
- เพื่อเพิ่มความสัมภានด้วยให้แก่ผู้ใช้ในการแต่งรูป ใส่ข้อมูลเกี่ยวกับคุณภาพอากาศ
- เพื่อให้ตัวผู้ใช้ตระหนักรถึงคุณภาพอากาศ

### 2. ลูกค้าที่เป็นเจ้าของแอปพลิเคชัน (รศ.ดร.สรรวรรณ กันทะบุตร, ผศ.ดร.ยุทธพงษ์ สมจิต)

- ได้แอปพลิเคชันที่ช่วยเพิ่มตระหนักรถึงคุณภาพอากาศ นำไปแจกลายให้คนอื่นได้
- เป็นคนกำหนด requirements สิ่งที่ต้องการสำหรับ project นี้
  - แอปพลิเคชันที่มีฟังชันก์ แสดงจุด hotspot ในแผนที่
  - บอกข้อมูลคุณภาพอากาศ และสถานที่
  - ให้ปรับแต่งรูปได้ จัดวางเลเยอร์เอาที่ได้ใส่ข้อมูลคุณภาพอากาศ

### 3. ผู้ที่เห็นรูปจากแอปพลิเคชันของเราใน Social Media

- ได้เห็นข้อมูลคุณภาพอากาศจากการแชร์ของผู้ใช้แอป
- ผู้รับได้รู้ข้อมูลคุณภาพอากาศ
- ตระหนักรถึงคุณภาพอากาศ

### 4. นักพัฒนาแอปพลิเคชัน

- พัฒนาแอปพลิเคชันให้เจ้าของแอปพลิเคชัน(อาจารย์)
- ทำความต้องการของลูกค้า
- ได้เรียนรู้การพัฒนาซอฟต์แวร์

### 5. ผู้เก็บข้อมูลคุณภาพอากาศ

- นำเอาข้อมูลมาให้นักพัฒนาแอปพลิเคชัน

## Technology Feasibility Study

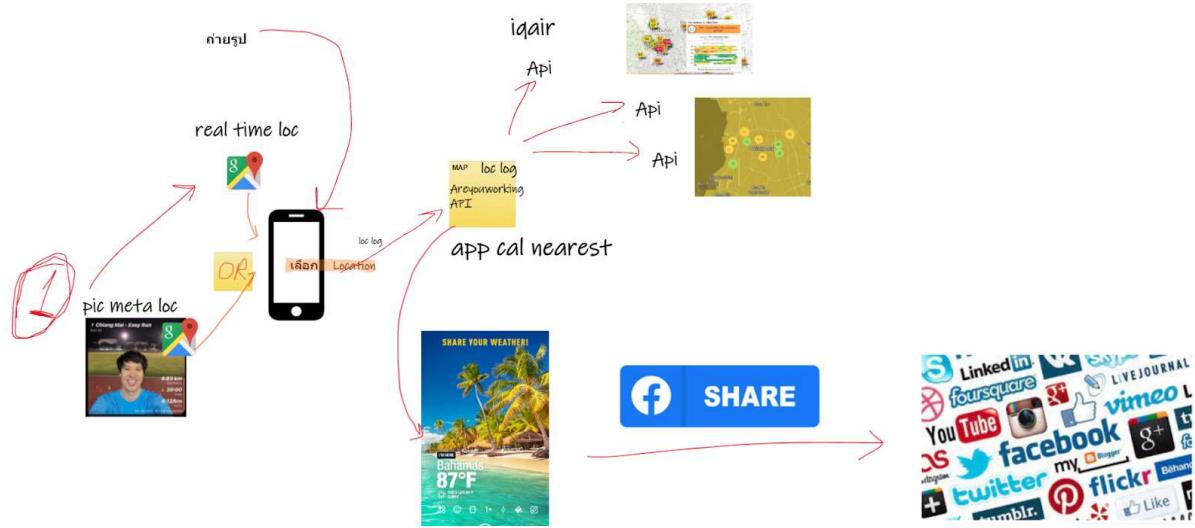
เราจะพัฒนา Mobile Application ด้วยที่ทีมมีขนาดเล็ก เราจึงเลือก Cross-Platform Mobile Application Framework เราได้เลือกใช้ Flutter เพราะ

- Hot reload
- Good documentation
- Touch-responsive animation (60 fps)
- Rendering performance ดีกว่า React Native เพราะไม่มี interpreter เหมือน React Native ที่ค่อย interpret JavaScript ตอน run time Flutter compile มาเสร็จแล้ว ใช้ C++ rendering engine

ระบบของเราประกอบด้วยส่วนที่เกี่ยวข้อง ดังนี้

- Air Quality API: <https://aqicn.org/api/> สามารถดึงข้อมูลคุณภาพอากาศ ว่างอยู่จาก Latitude, Longtitude ได้ ทั้งยังมีข้อมูลสภาพอากาศ พยากรณ์คุณภาพอากาศ 7 วันข้างหน้า
- Latitude, Longtitude สามารถได้มาจากการ GPS ของโทรศัพท์มือถือของผู้ใช้ หรือ Meta Data ของรูปที่ถ่าย
- Fire detection hotspots API: <https://firms.modaps.eosdis.nasa.gov/api/> ดึงแผนที่ ตำแหน่ง hotspots จุดไฟใหม่ จากดาวเทียม VIIRS, MODIS
- Front-end ใช้ widget ของ Flutter เขียนโดยภาษา Dart

## Conclusion



บางครั้งเราแทบไม่รู้ตัวหรือใส่ใจกับคุณภาพอากาศในบริเวณที่เรารอยู่เลย ซึ่งเราอาจอยู่ในพื้นที่ที่มีมลพิษทางอากาศสูงและเป็นอันตรายต่อสุขภาพ เราจึงสร้าง Mobile Application ที่ช่วยเพิ่มความตระหนักรู้ โดยใช้พื้นที่ Social Media และรูปภาพในการแชร์ข้อมูลสภาพอากาศในพื้นที่บริเวณนั้น โดยที่ผู้ใช้งานเปิด Application ของเรา และเลือกรูปภาพพร้อมแทรกข้อมูลคุณภาพอากาศลงใน Social Media ต่าง ๆ เพื่อให้คนที่ผ่านมาเห็นได้รับรู้ถึงคุณภาพอากาศในพื้นที่นั้น ๆ หรือเกิดความตระหนักรถึงคุณภาพอากาศในพื้นที่บริเวณที่ตนเองอาศัยอยู่ ก็จะช่วยให้ผู้ที่ใช้งาน Application หรือผู้ที่ได้รับสาร มีความใส่ใจกับเรื่องของมลพิษทางอากาศ ทั้งการป้องกันตัวเอง หรือการพาทางแก้ไข

Features ของ Application: Air Pollution Awareness via Photo Sharing (AP) คือ

- ผู้ใช้งานสามารถถ่ายรูปหรือเพิ่มรูปภาพจาก Gallery
- จากพิกัดที่อยู่ของผู้ใช้งานโดยใช้ GPS หรือ พิกัดที่ถูกเก็บใน Metadata ของไฟล์รูปภาพในการเลือกดึงจากจุดวัดคุณภาพอากาศที่ใกล้ที่สุด เพื่อดึงข้อมูลทางสภาพอากาศเช่น AQI, PM2.5, Hotspot map
- ผู้ใช้งานสามารถแทรกข้อมูลทางสภาพอากาศลงในรูปภาพ
- แชร์รูปภาพที่มีข้อมูลทางสภาพอากาศ บน Social Media platform

## ภาคผนวก

ชนิดล เดชประภากร( 25% จากจำนวนหน้าที่เขียนและจำนวนบรรทัดที่แก้ไข)

- ศึกษาดู Third-party API ที่จำเป็นต้องใช้ในแอป
- ดูทางเลือกการพัฒนาแอป โดยใช้ Cross-platform framework
- Technical Feasibility
- Cover page, Document formatting, proof-reading

ธนวัตน์ บำเพ็งพันธุ( 17.5% จากการศึกษาและลองใช้เทคโนโลยีที่จะต้องใช้ในการพัฒนาแอป)

- ทำหน้าเว็บของทีมจาก Design ของภริช
- ศึกษาการใช้งาน Android Studio, Flutter, Dart ในการพัฒนา Mobile Application และการทำ Testing
- ลองใช้ Flutter ในการพัฒนา Basic feature อย่าง การดึงค่า Location latitude, longitude ของ ผู้ใช้งานจาก GPS

ธนัตถ์ ตั้งอัน ( 17.5% จากที่กล่าวด้านล่าง)

- ลองใช้ Flutter ในการทำฟีเจอร์ถ่ายรูปและเลือกรูปจากคลังรูปภาพมา
- ศึกษาแอป Run Photo และ WeatherShot ว่าเป็นยังไง มีอะไรที่เหมือนเราบ้าง
- ช่วยเขียนตาราง 1 และตรง Techonoly Feasibility Study nidหน่อย

ภริช สินวัลแอล( 20% เนื่องจากช่วยงานด้านออกแบบ และงานอื่นๆ ดังต่อไปนี้)

- ทำหน้าเว็บ และ ออกแบบ UI ให้เว็บใน [figma](#)
- ออกแบบ UI/UX ให้แอปแบบคร่าว ๆ ใน [figma](#)
- ติดต่ออาจารย์เจ้าของโครงการเพื่อขอ requirement เพิ่มเติม
- ศึกษาการใช้งาน Flutter, ในการพัฒนา Mobile Application
- คอยติดตามผลงานและพูดคุยกับกลุ่ม ถึงปัญหาต่างๆ

พงศกร รัตนพันธ์ (20% จากการใส่เนื้อหาเค้าโครงในทุกๆหน้า และทำงานดังต่อไปนี้)

- ทำหน้าเว็บ <https://areyouworking.github.io/> และ ระบบ fetch Worklog ในหน้าเว็บ
- วิเคราะห์ Requirement เพื่อออกแบบ Technical Requirement
- ทำ Technical Requirement ให้กับคนในทีม
- คอยติดตามผลงานและพูดคุยกับกลุ่ม และช่วยแก้ปัญหาต่างๆ
- กำหนดตาราง โปรเจค
- ศึกษาการใช้งาน Flutter, ในการพัฒนา Mobile Application

## A2 - Software Requirements Specification

---

# Software Requirements Specification

for

## Air Pollution Awareness via Photo Sharing (AP)

Version 0.1

Prepared by  
Tanadol Deachprapakorn  
Purich Seenaullae  
Pongsakorn Ratanapan  
Thanawat Bumpengpun  
Tanat Tangun

Team AreYouWorking

January 2023

# Table of Contents

<b>1. Introduction</b>	<b>12</b>
1.1 Purpose	12
1.2 Scope	12
1.3 Definitions, Acronyms, and Abbreviations	13
1.4 References	13
1.5 Overview	13
<b>2. Overall Description</b>	<b>15</b>
2.1 Product Perspective	15
2.2 Product Functions	15
2.3 User Characteristics	17
2.4 Operating Environment	17
2.5 Design and Implementation Constraints	17
2.6 Assumptions and Dependencies	19
<b>3. External Interface Requirements</b>	<b>19</b>
3.1 User Interfaces	19
3.2 Hardware Interfaces	20
3.3 Software Interfaces	21
3.4 Communications Interfaces	21
<b>4. System Features</b>	<b>21</b>
4.1 Take photo feature	21
<b>5. Other Nonfunctional Requirements</b>	<b>23</b>
5.1 Performance Requirements	23
5.2 Safety Requirements	23
5.3 Security Requirements	23
5.4 Software Quality Attributes	23
<b>Appendix A: Work Plan</b>	<b>23</b>

## Revision History

Name	Date	Reason For Changes	Version
AP	15/01/23	First Version	0.1

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to describe the requirements for the Air Pollution Awareness via Photo Sharing (AP) project. The intended audience includes all stakeholders in the potential system. These include, but are not necessarily limited to, the following: application users, customers, social media photo audience, air quality data providers, and developers.

Developers should consult this document and its revisions as the only source of requirements for the project. They should not consider any requirements statements, written or verbal as valid until they appear in this document or its revision.

## 1.2 Scope

The proposed software product is the Air Pollution Awareness via Photo Sharing (AP) application. It reports the actual and forecasted air quality information in the user's location. Additionally, it allows users to customize their photo to include air quality information and to upload to social media.

The goal for Air Pollution Awareness via Photo Sharing (AP) application is to raise awareness about air pollution, specifically in Chiang Mai and Thailand via photo sharing. Photo sharing has been popular in social media. By enabling users to customize their photo with air quality information easily could help them communicate with others about the air pollution in their communities, potentially raising awareness of air pollution in the communities. People could know how to protect and prepare themselves against air pollution.

## 1.3 Definitions, Acronyms, and Abbreviations

**Table 1.** Definitions

Term/Acronym/Abbreviation	Expansion/Description
Metadata	Information embedded in the image file
Fetch	Requests and retrieves resources from a server
Overlay	A separate layer on top of an existing image that contains air quality information
Widget	Building block of the user interface

**Table 2.** Acronyms/Abbreviations

GPS	Global Positioning System
API	Application Programming Interface
AQI	Air quality index
UI	User interface
AP	Air Pollution Awareness via Photo Sharing
SRS	Software Requirements Specification
CPU	Central Processing Unit
RAM	Random Access Memory
GB	Gigabyte
GHz	Gigahertz
REQ	Requirement
HTTPS	Hypertext transfer protocol secure
JSON	JavaScript Object Notation

## 1.4 References

Template-SRS-iso-iec-ieee-29148-2011.pdf

<https://mango-cmu.instructure.com/courses/1577/files/206713?wrap=1>

A2 - Software Requirements Specification (SRS) Assignment Page

[https://mango-cmu.instructure.com/courses/1577/assignments/8698?module\\_item\\_id=38180](https://mango-cmu.instructure.com/courses/1577/assignments/8698?module_item_id=38180)

AreYouWorking Team's website

<https://areyouworking.github.io/>

## 1.5 Overview

This Software Requirements Specification (SRS) is the requirements work product that formally specifies Air Pollution Awareness via Photo Sharing (AP) project. It includes the results of both business analysis and systems analysis efforts. Various techniques were used to elicit the requirements and we have identified your needs, analyzed and refined them. The objective of this document therefore is to formally describe the system's high level requirements including functional requirements, non-functional requirements and users' needs and constraints. The detail structure of this document is organized as follows:

Air Pollution Awareness via Photo Sharing (AP)

A2 - Software Requirements Specification

Section 2 of this document provides an overview of the functionalities that the proposed Air Pollution Awareness via Photo Sharing (AP) project will support. These include a general description of the product, user characteristics, general constraints, and any assumptions for this system.

Section 3 presents the detailed requirements, including User Interfaces, Hardware Interfaces, and Communication Interfaces. It describes about

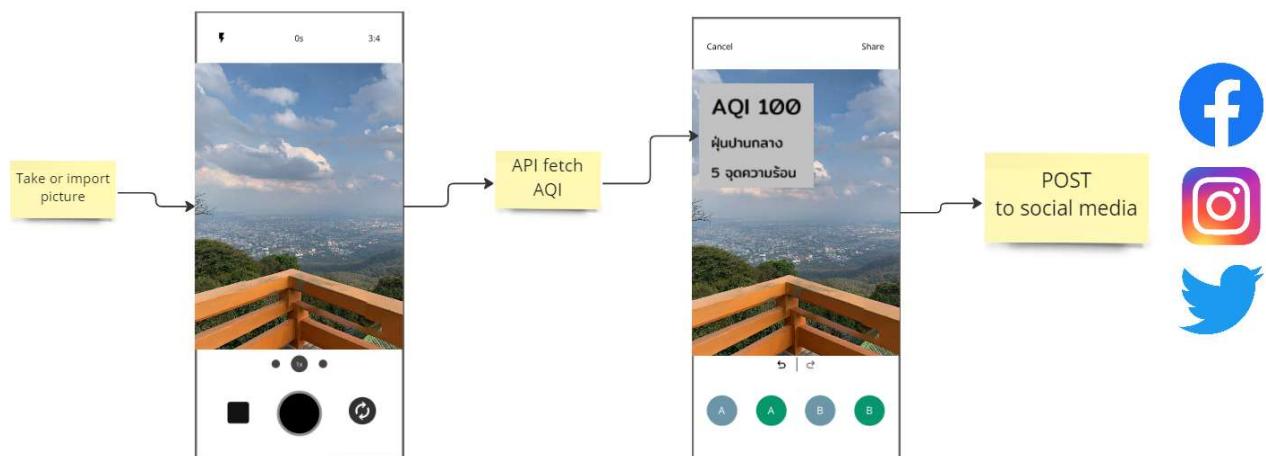
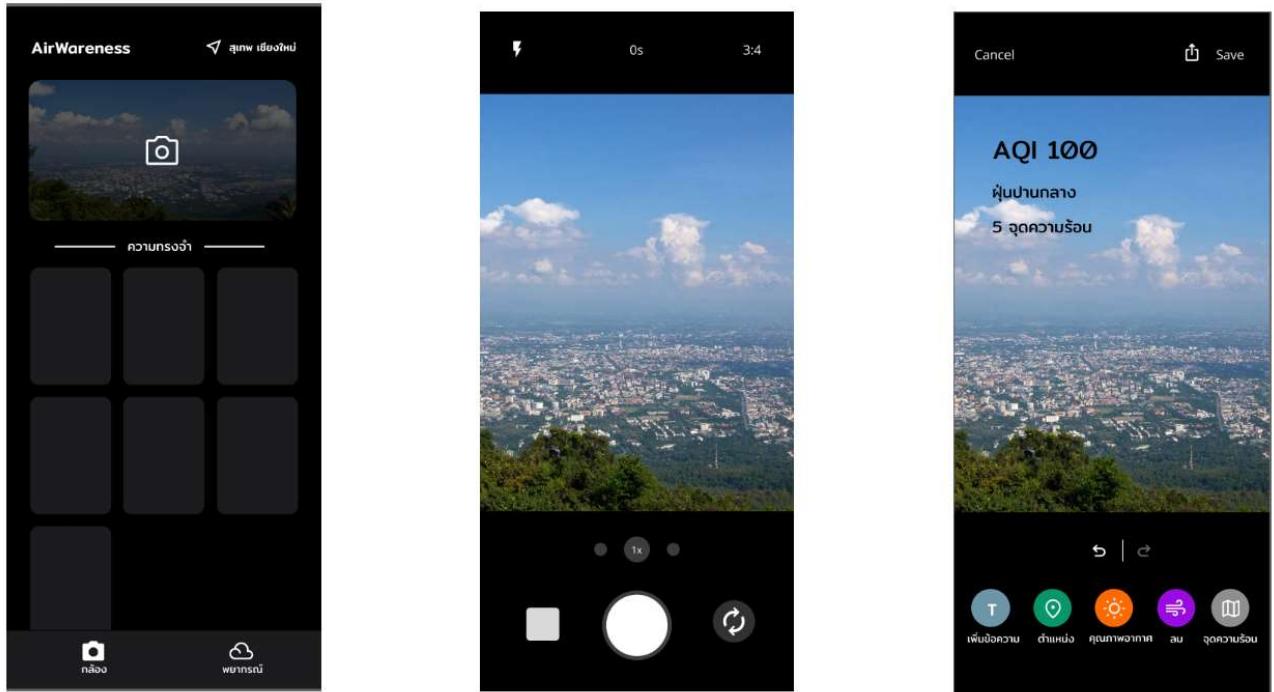
Section 4 describes four features of an app, including a "Take Photo" feature that allows users to take a photo within the app and save it to a designated album or folder, a "Edit Photo with Air Quality Info" feature that allows users to edit photos by overlaying air quality data and customizing the appearance of the overlay, a "Share a Photo to Social Media" feature that allows users to share photos to social media platforms with captions, hashtags, and privacy settings, and a "View Current/Predicted Air Quality" feature that allows users to view the current AQI, predicted AQI, and historical data for a specific location. All of these features are high or medium priority.

Section 5 describes other non-functional requirements for an app, including performance requirements for small size, quick opening and use, and ability to support multiple requests at the same time. Safety requirements include preventing data loss by preventing crashes while editing photos. Security requirements include compliance with terms and conditions of 3rd party APIs, secure and private retrieval of air quality data and allowing users to choose to allow access to device's location, storage, and camera. Additionally, the software quality attributes include easy to use for everyone and fast UI response and loading.

## 2. Overall Description

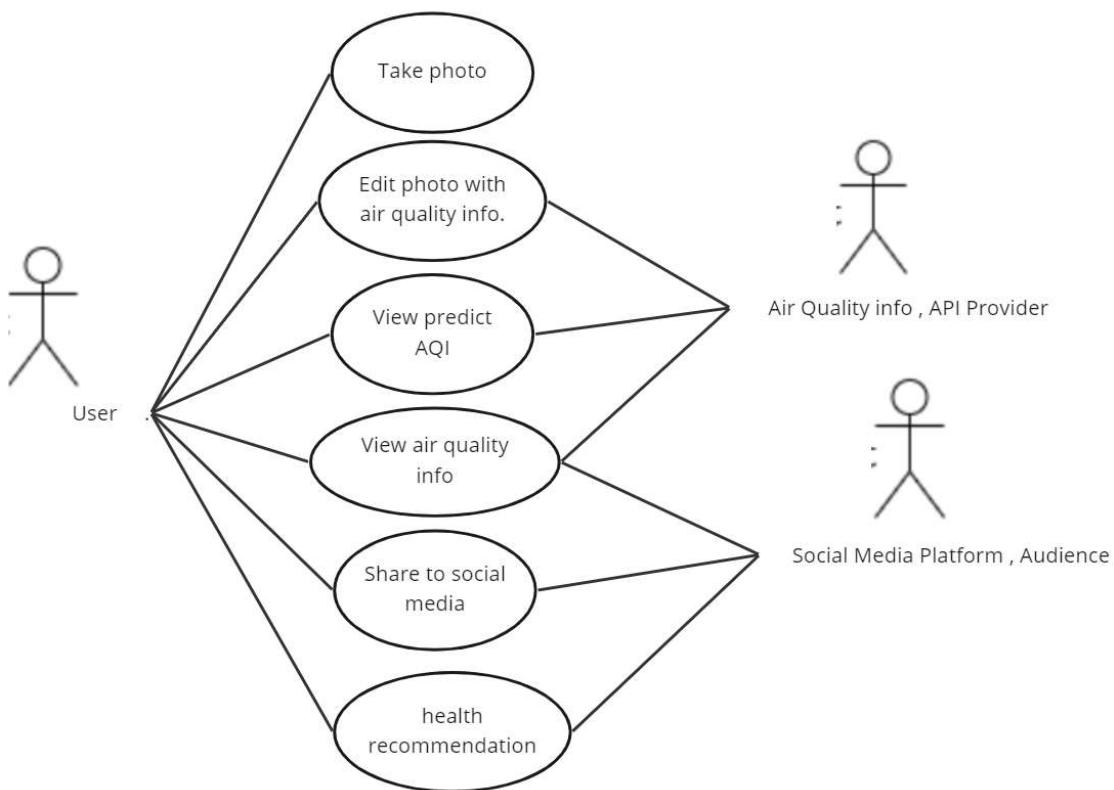
### 2.1 Product Perspective

The Product Perspective of Air Pollution Awareness via Photo Sharing (AP) is an application that aims to raise awareness about air pollution by enabling users to share photos of polluted areas in their communities. The AP platform is designed to be user-friendly and accessible to people of all ages, allowing anyone with a smartphone or other device with a camera to participate.



The AP application includes the ability to take and edit photos with air quality information, viewing air quality forecasting, and history of shared photos.

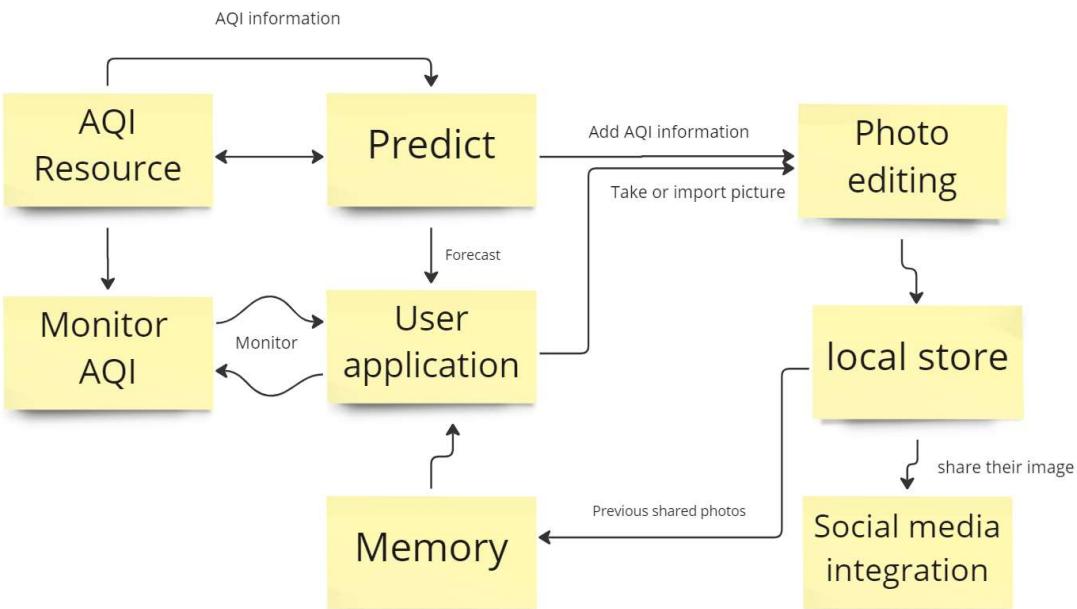
The AP application also includes resources and information about air pollution and its impacts on human health and the environment. This includes information about the causes of air pollution, such as industrial and transportation emissions, as well as strategies for reducing pollution and improving air quality. The following is the use case diagram of the AP application.



Overall, the goal of the AP application is to address and raise awareness about air pollution in their communities. By sharing photos and information about polluted areas, users can help raise awareness about the issue and encourage others to take action to reduce pollution and improve air quality.

## 2.2 Product Functions

- Allow users to view real-time and hourly and daily forecasted air quality data for their current location or any location they choose.
- The users can choose their location using GPS, or searching for their location.
- Air-quality data can be overlaid onto the images taken with the AP app, providing users with a visual representation of the air quality in their area.
- Hot spot information: a feature that allows users to view hot spots, or areas of high heat that may be contributing to air pollution.
- Hot spot maps can also be overlaid onto the images taken with the app.
- Allows users to easily share their images and data to social networks such as Facebook, Twitter, and Instagram. This helps to spread awareness about air pollution and encourages others to join the conversation.
- Allows users to save their edited photos onto their devices.
- Allows users to browse their previous shared photos.



- Easy to use: AP is designed with a user-friendly interface, making it easy for anyone to use. The app includes clear instructions and a simple navigation system, making it accessible to users of all levels of technical expertise.
- Health recommendation for users. This includes information about the causes of air pollution, AQI guide, health impact, how to protect and prevent.

## 2.3 User Characteristics

- Typical Users, such as those who often spend time outside, Instagrammer, social media users.
- Users who like to know more about air pollution.

## 2.4 Operating Environment

Designed to work on Android devices, including smartphones and tablets . The app requires Android version 5.0 or higher to run properly and optimized to work on all screen sizes, including small smartphone screens and larger tablet screens. The app is designed to automatically adjust to the size of the device it is being used on, ensuring that it is easy to use and read on any device.

## 2.5 Design and Implementation Constraints

The AP is developed using Flutter for Android devices. Flutter is an open-source mobile application development framework created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows and the web. Flutter uses the Dart programming language, which was also created by Google. One of the key features of Flutter is its use of a reactive programming model, which allows developers to build high-performance, responsive user

interfaces. Additionally, Flutter includes a rich set of customizable widgets and tools for building beautiful, natively compiled applications for mobile, web, and desktop.

The AP uses 3-rd party Air-quality APIs which provide real-time air-quality data as well as hourly, and daily forecast.

The AP uses 3-rd party location service APIs to provide recommended locations for users to choose from. These APIs provide access to device's location service and allows users to search for a custom location from a 3-rd party database.

The AP must follow the guidelines and best practices for data privacy and security, regarding user's location, photos, storage and API usage.

The user can choose to allow access to their device's location service, camera, and storage. The AP application must continue to work, but limiting functionalities that use the unpermitted service.

The application will have minimum hardware requirements such as a device with at least 1 GB of RAM, and a CPU with a minimum clock speed of 1 GHz.

## 2.6 Assumptions and Dependencies

Assumptions:

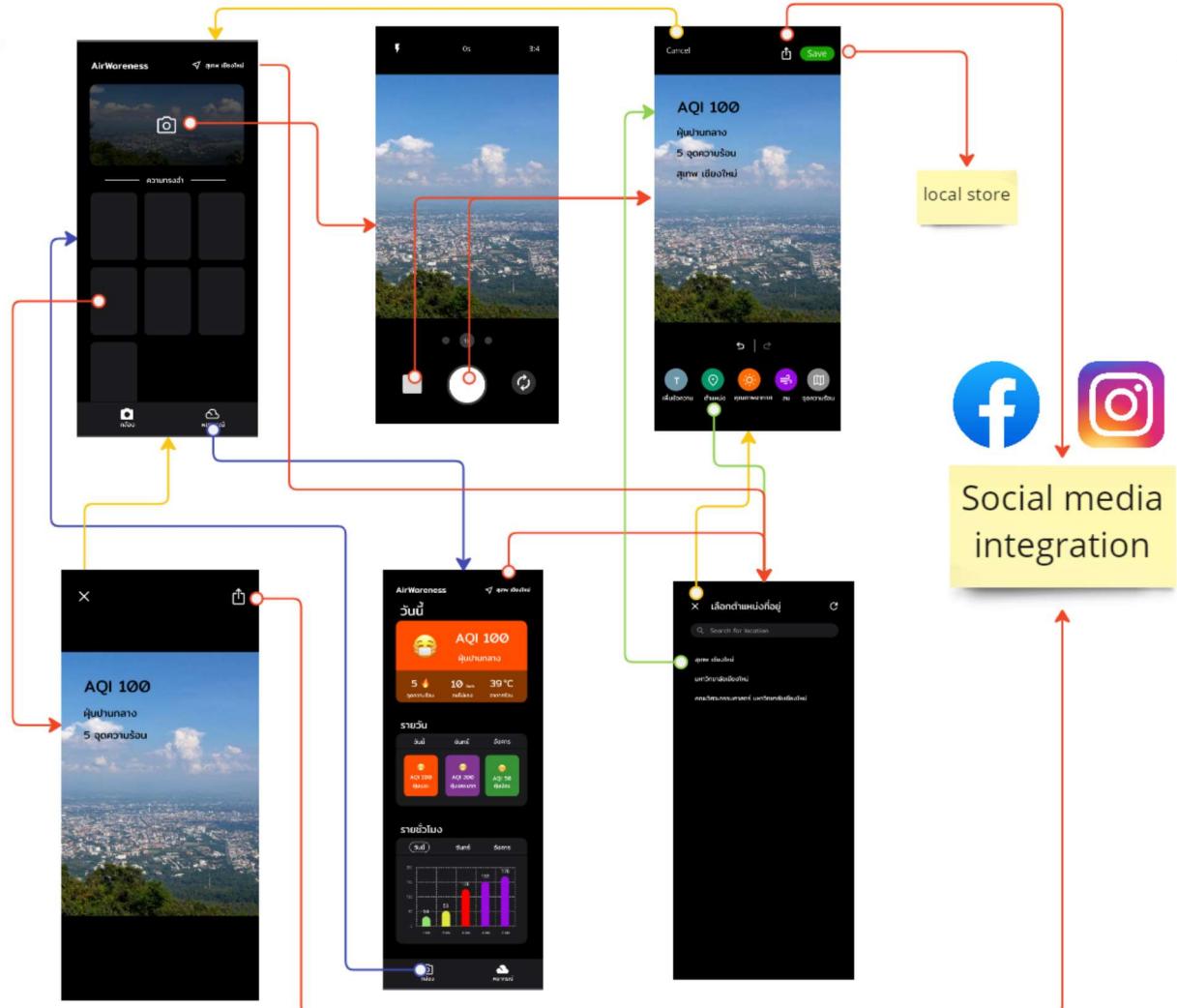
- Users have a device with internet connection to use the application.
- Users have a social media account to share the customized photos.
- The third-party API used for retrieving air quality information is reliable and accessible.
- The location service API is reliable and accessible.

Dependencies:

- Flutter framework for development and building the application.
- A third-party API for retrieving air quality information.
- Location services to provide accurate air quality information.
- Social media integration for photo sharing functionality.
- A device with internet connection to use the application.
- Users must have an account in social media to share the photos from the app.

### 3. External Interface Requirements

#### 3.1 User Interfaces



The first page the user sees when they open the app is the camera screen. This screen lets the user take a photo and browse their previous shared photos. The user can see their photo in full screen when they press on it. The user can switch to the forecast screen using the navigation buttons below. On the top right of both screens shows the user's location. The user can press on that button to search for a location, or select using the device's location service. The forecast screen includes air quality data as shown. It features hourly and daily forecasts. After the user selects a photo from their gallery, or takes a photo, the app will take them to the photo editor screen. This screen lets the user customize their photo with air quality data. The user can save the edited photo onto their device, and share it to social media. The buttons below in the photo editor screen lets the user add more widgets onto the photo. The user can double press on a widget to change appearance. They can select a preset layout to be overlaid on their photo by swiping the screen left and right.

### 3.2 Hardware Interfaces

1. The AP application will require internet connectivity to access the third-party API and retrieve air quality information.
2. The application will also require access to the device's location services in order to provide accurate air quality information.
3. The application will need access to the device's camera and photo library to allow users to customize photos with air quality information.
4. The application will also require access to the device's social media accounts to enable photo sharing functionality.
5. The application will require access to the device's storage to save photos.
6. The AP application is designed for mobile devices, it will be compatible with Android.
7. The application will work on both smartphones and tablets.
8. The application will be tested on various devices to ensure compatibility and optimal performance.
9. The application will be optimized for portrait orientations.
10. The application will be able to run on devices with different screen resolutions and sizes.

### 3.3 Software Interfaces

AP requires Android version 5.0 and above. More information can be on section 2.4 of this document.

### 3.4 Communications Interfaces

- The AP application uses internet connection to retrieve air-quality data, location data, and to share photos in social media.
- The application uses HTTPS for communicating with air-quality API and social media API.
- The application uses JSON format for data exchange with the API.
- The application must be able to handle error and provide appropriate feedback in case of communication error.
- The application must comply with guidelines and best practice for data privacy and security.

## 4. System Features

### 4.1 Take photo feature

Users should be able to take a photo in our app with no need to swap to the camera app. This is HIGH priority.

- REQ-1: The camera should be activated upon pressing the "Take Photo" button or through a designated gesture such as volume button.
- REQ-2: The user should be able to preview the photo on the screen before saving or discarding it.

- REQ-3: The photo should be saved in a designated photo album or folder on the phone's internal storage or external storage.
- REQ-4: The app should support zoom in/out while taking photos.
- REQ-5 :The app can switch to use front or rear cameras.
- REQ-6: The app must support taking photos in both landscape and portrait orientation.

#### 4.2 Edit photo with air quality info feature

Users should be able to edit the imported or taken photo inside the app. This is HIGH priority.

- REQ-1: The user should be able to select a photo from their camera roll or take a new photo to edit.
- REQ-2: The app should be able to access the device's GPS or location services to determine the user's current location or user selected location.
- REQ-3: The app should be able to retrieve real-time air quality data for the user's location from a reputable source, such as a government agency or research organization.
- REQ-4: The app is able to overlay the air quality data, such as the current AQI (Air Quality Index) and the level of pollutants, onto the photo.
- REQ-5: The user should be able to customize the appearance of the air quality data overlay, such as the font, color, and size.
- REQ-6: The app should be able to save the edited photo in a designated folder or album on the phone.
- REQ-7: The app should be able to undo , redo the edit photo state.
- REQ-8: The user can choose their own location from GPS, recommended locations based on GPS, photo's metadata, or their own searched location.

#### 4.3 Share a photo to social media feature

Users should be able to share the edited or taken photo from inside the app to social media like Instagram, Facebook, Twitter etc. This is MEDIUM priority.

- REQ-1: The user should be able to select from a list of social media platforms to share the photo, such as Facebook, Instagram, Twitter, and others.
- REQ-2: The user should be able to add a caption and hashtags on the photo before sharing.
- REQ-3: The app should provide the user with a notification or confirmation when the photo has been successfully shared.

#### 4.4 AirQuality Forecast System

Users should be able to access and view information about the air quality in the specific location.

- REQ-1: The app should be able to access the device's GPS or location services to determine the user's location.

- REQ-2: The app should be able to retrieve real-time air quality data for the user's current location from a reputable source, such as a government agency or research organization.
- REQ-3: The app should be able to display the current AQI (Air Quality Index) and the level of pollutants for the user's location.
- REQ-4: The app should be able to predict the AQI for the next few days, providing the user with a forecast of the air quality.
- REQ-5: The user should be able to select a different location to view the AQI and forecast.
- REQ-6: The app should be able to provide the user with historical air quality data for the location.
- REQ-7: The user can choose their own location from GPS, or their own searched location.

#### 4.5 Previous Edited/Shared Photo (Photo Memory)

Users should be able to view previous edited or shared photos that will look like Instagram story archives (which we'll call memory).

- REQ-1: Users should be able to view previous edited/shared photos on the memory.
- REQ-2: Users should be able to delete previous edited/shared photos from the memory.

### 5. Other Nonfunctional Requirements

#### 5.1 Performance Requirements

The AP application must be small in size and can be opened and used quickly. The 3-rd party APIs must be able to support multiple requests at the same time.

#### 5.2 Safety Requirements

The AP application must not crash while the user is editing a photo to prevent data loss.

#### 5.3 Security Requirements

- User's location must not be kept or publicized.
- The AP application must comply with terms and conditions of 3-rd party APIs.
- The user can choose to allow access to their device's location service, storage, and camera.
- The user's location is used to retrieve air quality data. The retrieval must be done securely and privately.
- The AP application saves the shared photo onto the user's device. The user can choose to delete it anytime.

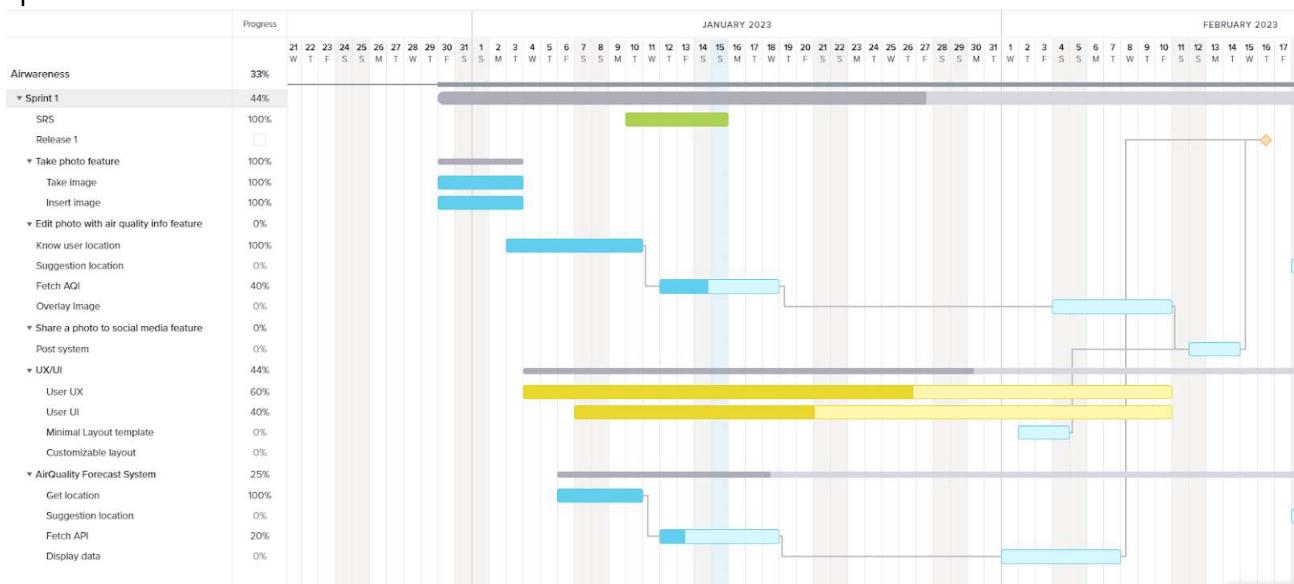
## 5.4 Software Quality Attributes

- The AP application is easy to use for everyone.
- The UI responds fast and loads quickly.

# Appendix A: Work Plan

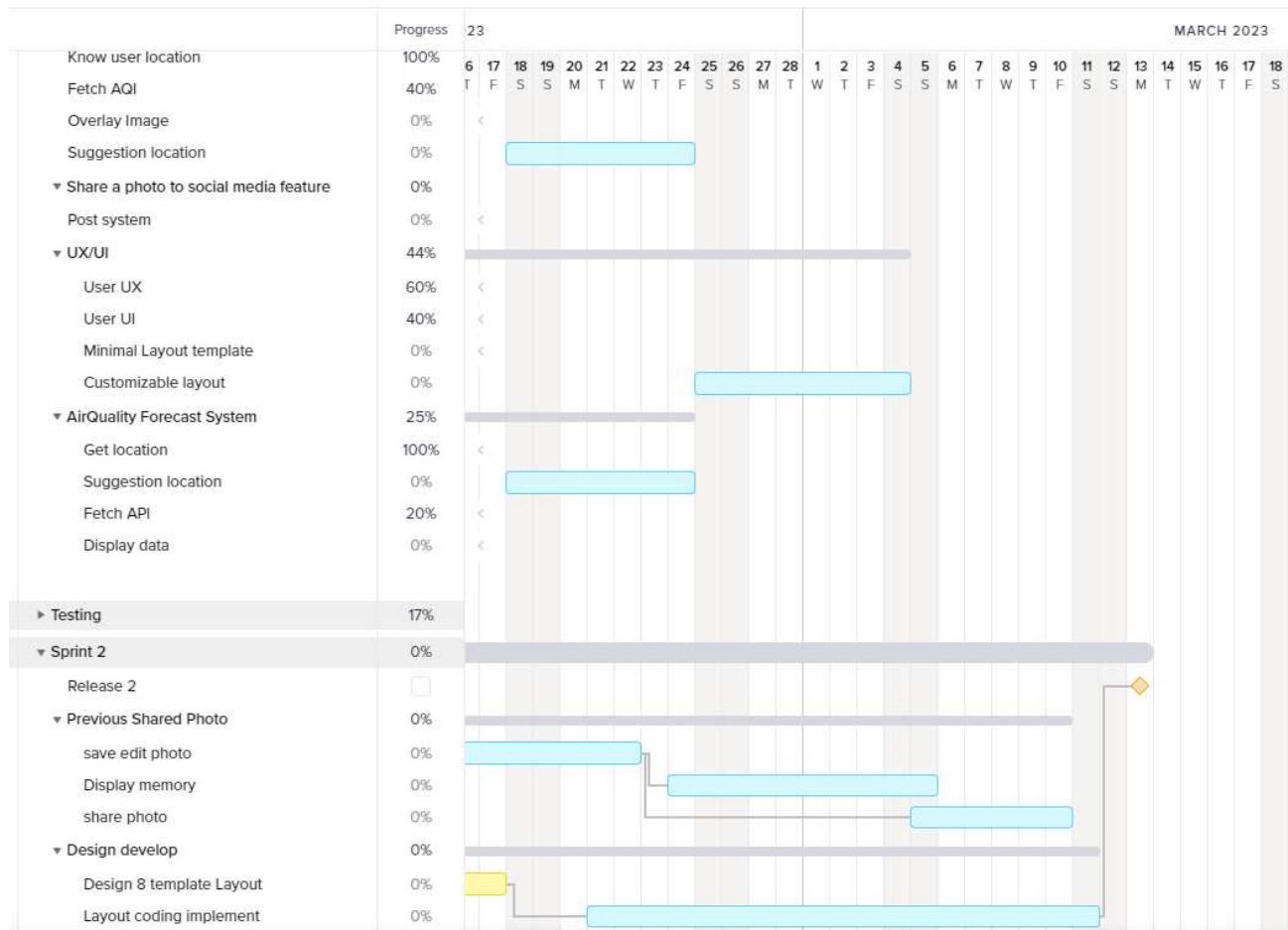


## Sprint 1 :



- Implement all of the requirements of the AirQuality Forecast System in the “System Features” section.
  - Should implement some of “Take photo feature” and “Edit photo feature” requirements.

## Sprint 2:



## Sprint 2

- Implement the remaining requirements of “Take photo feature” and “Edit photo feature”.
- Implement all of the requirements on “Previous Edited/Shared Photo (Photo Memory)” and “Share a photo to social media feature”
- Improve UI/UX on all of the features implemented.

## Appendix B: Contribution

From the number of lines written and the number of manhours.

### **Tanat Tangun 630610737 (20.00%)**

- Write most of the “System Features” section and some of section 1.5.
- Help plan and write on “Appendix A: Work Plan”

### **Tanadol Deachprapakorn 630610734 (22.50%)**

- Write most of section 1, section 2.3, 2.5, 2.6, most of section 3.

### **Thanawat Bumpengpun 630610736 (15.00%)**

- Write section 1.3 Definition/Acronym tables and review section 1,2,3,4.

### **Pongsakorn Rattanapan 630610749 (22.50%) : write most of the**

- Other Nonfunctional Requirements 5.1 5.2 5.3 5.4
- External Interface Requirements and
- Some part in Overall Description”
- review section 2 and 4.
- Meeting and deciding with friends which function should have and which should not have.
- Decide together with friends which functions will not be available in version 1.
- Make a project plan and brainstorm to make it.
- Make all images in this report.

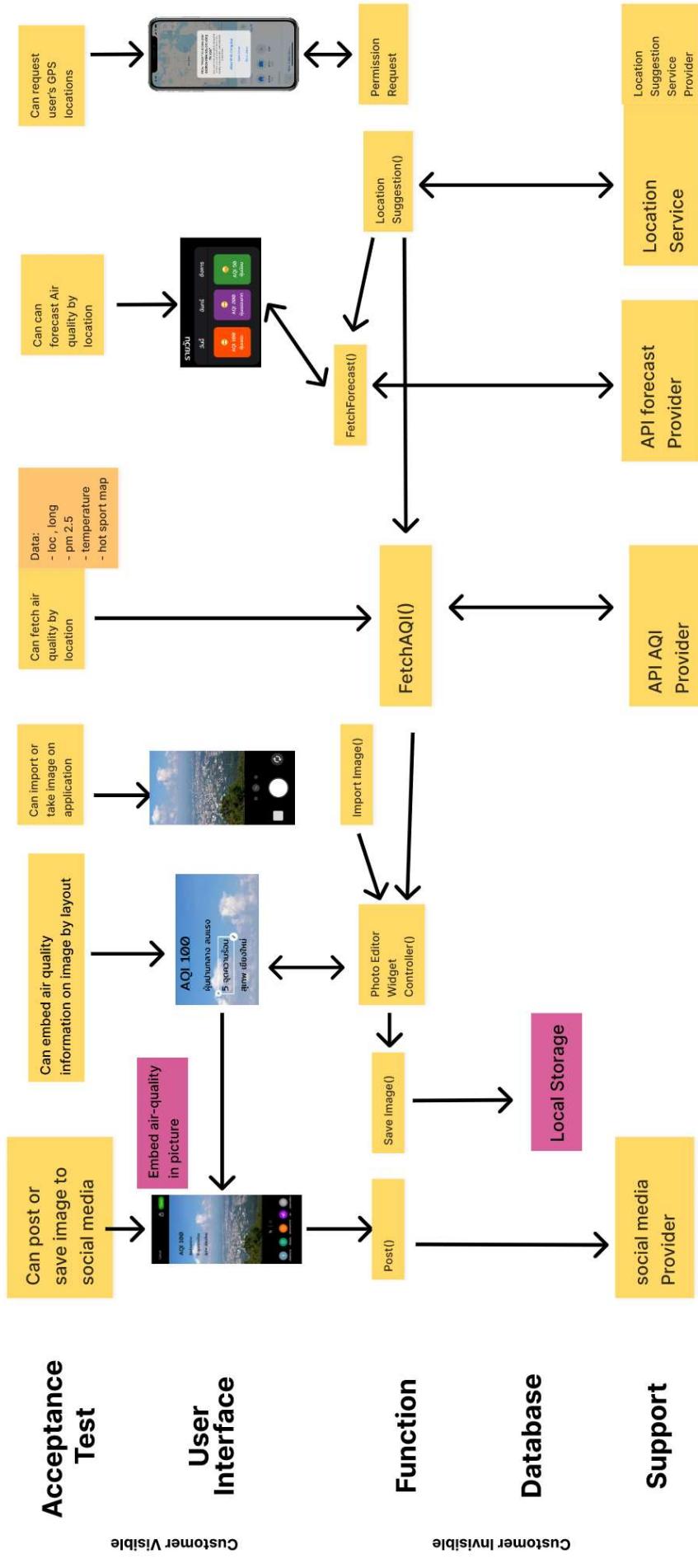
### **Purich Seenuallae 630610752 (20.00 % from the contents I write and wireframe)**

Design a user-friendly and efficient interface for Wideframe, a platform that allows users to quickly create and customize interactive frames for digital content.

I create wireframes and mockups of the interface by Figma. I focus on creating a streamlined flow that allows users to easily create and customize interactive frames within the platform, while also providing advanced options for more experienced users.

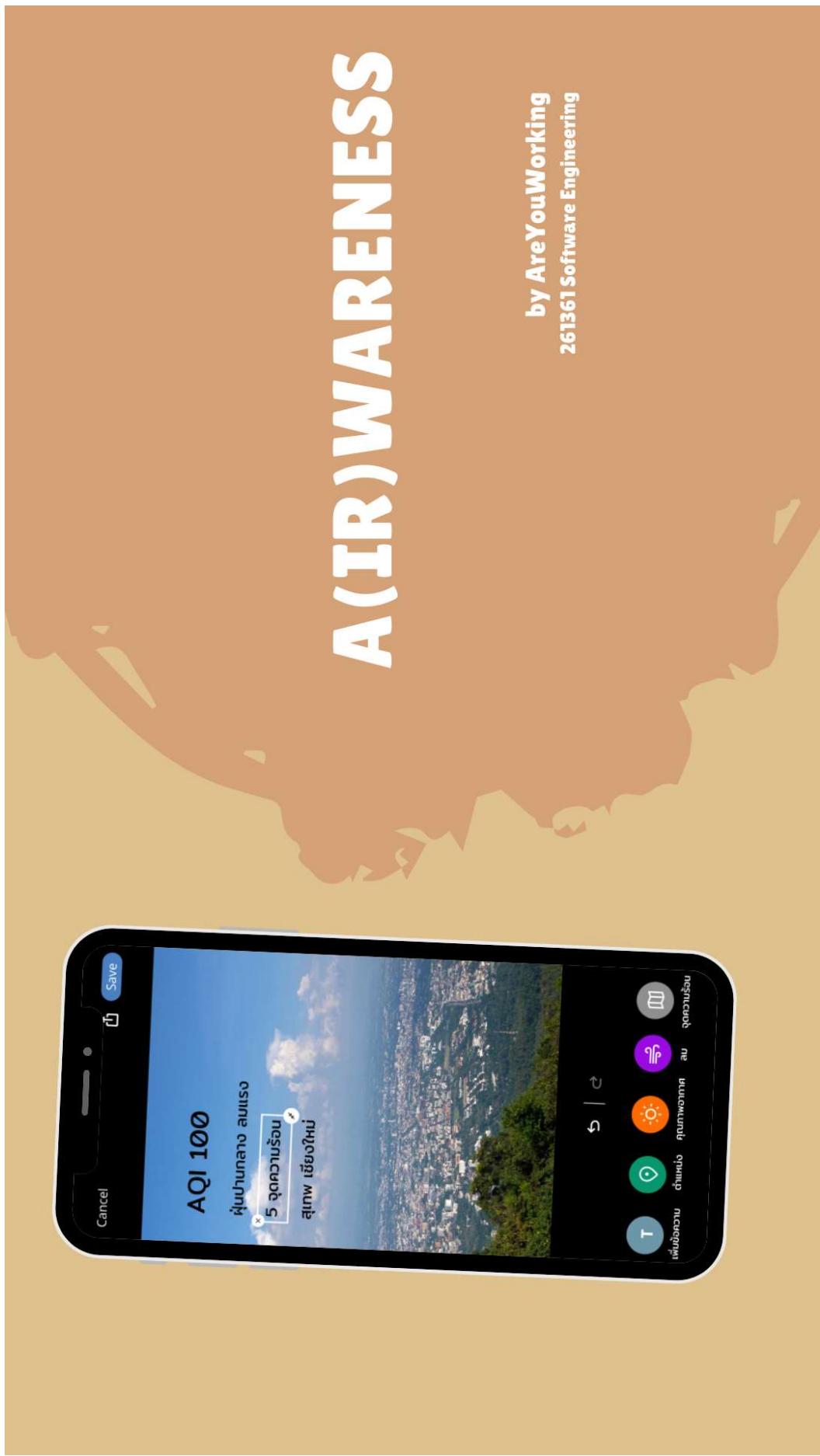
I have written most of the SRS document in Section 2 and have also reviewed the entire document, including sections 1, 2, 3, and 4. I have also attended and presented the wireframes during an online meeting with the team.

## A3 - Design Specification (A-DAPT)



Air Pollution Awareness via Photo Sharing (AP)  
A3 - Design Specification (A-DAPT)

## A4 - Sprint #1 (Presentation slides and Video)



Air Pollution Awareness via Photo Sharing (AP)  
A4 - Sprint #1 (Presentation slides and Video)

## OUR TEAM MEMBER

**PONGSAKORN RATTANAPAN**  
PROJECT MANAGER & SA



**TANADOL DEACHPRAPAKORN**  
DEVELOPER



**TANAT TANGUN**  
DEVELOPER



**PURICH SEENUALLAE**  
DEVELOPER, UX/UI  
DESIGNER



**THANAWAT BUMPENGPN**  
TESTER



# PROBLEM

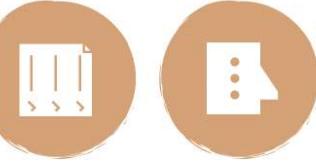
## SOCIAL MEDIA

สร้างการรับรู้ เผยแพร่ข้อมูล



## AIR QUALITY FORECAST

คนเชิงໃຫຍ່ແລະຄາໄຫຍ່ສ່ວນໃຫຍ່ໄປກ່າວນວ່າ  
ຝຶນຄວນແລະອາກາຄາຕີ່ໃນດີ  
ອູ່ໃກ້ຄໍ້ຕັວກວ່າກ່ົດ

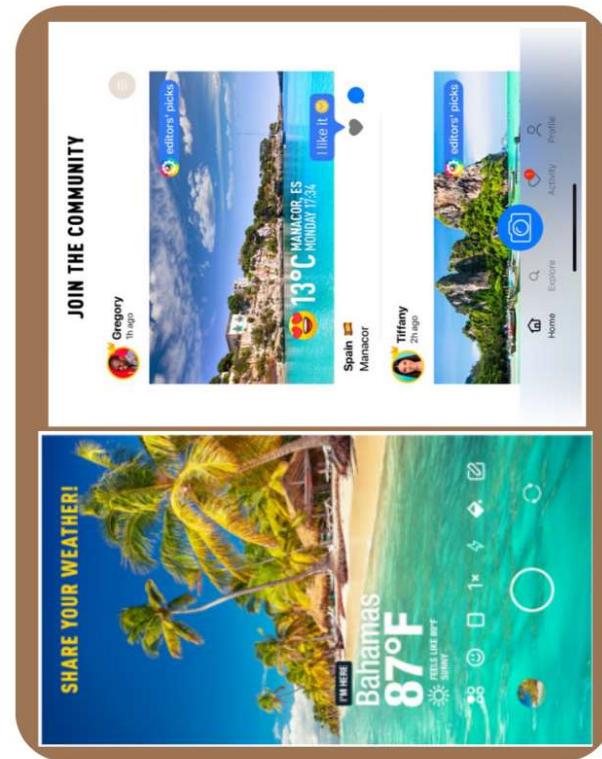




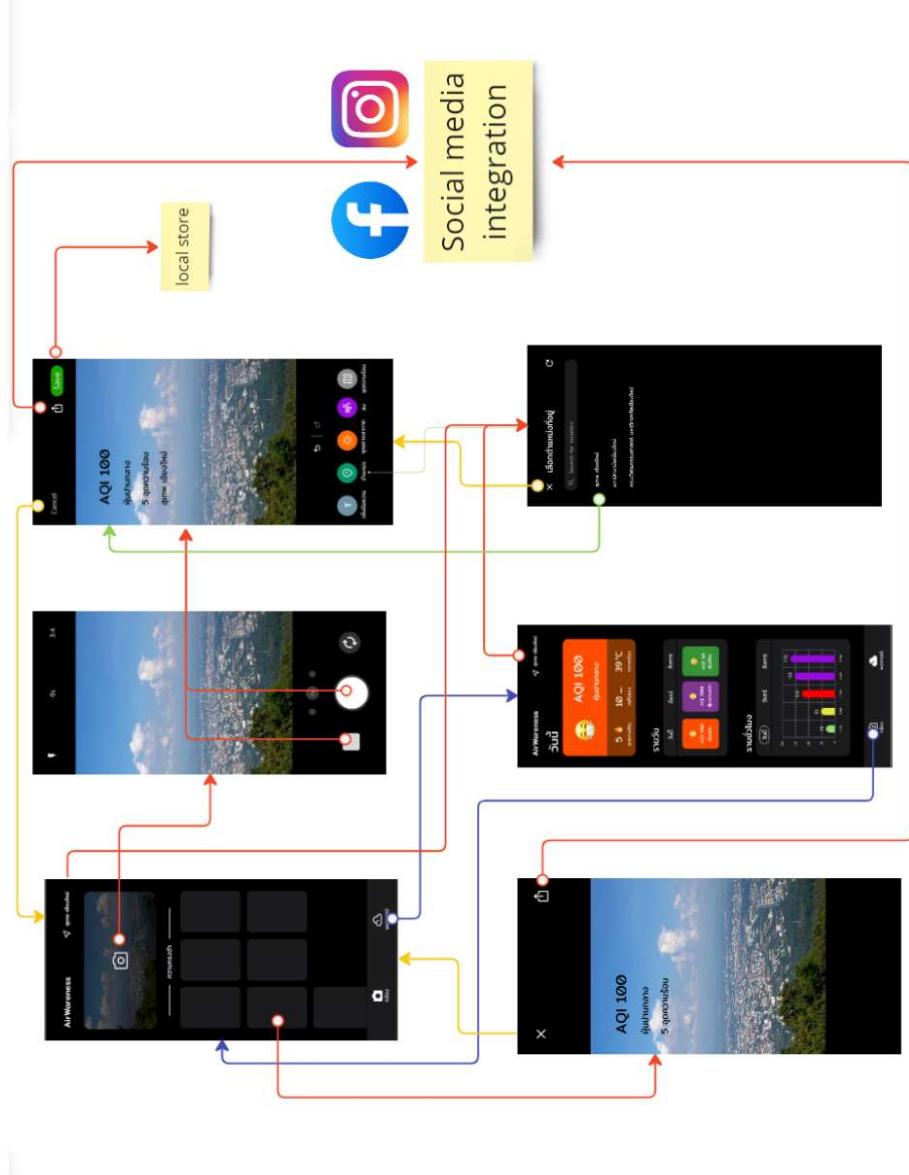
Air Pollution Awareness via Photo Sharing (AP)  
A4 - Sprint #1 (Presentation slides and Video)

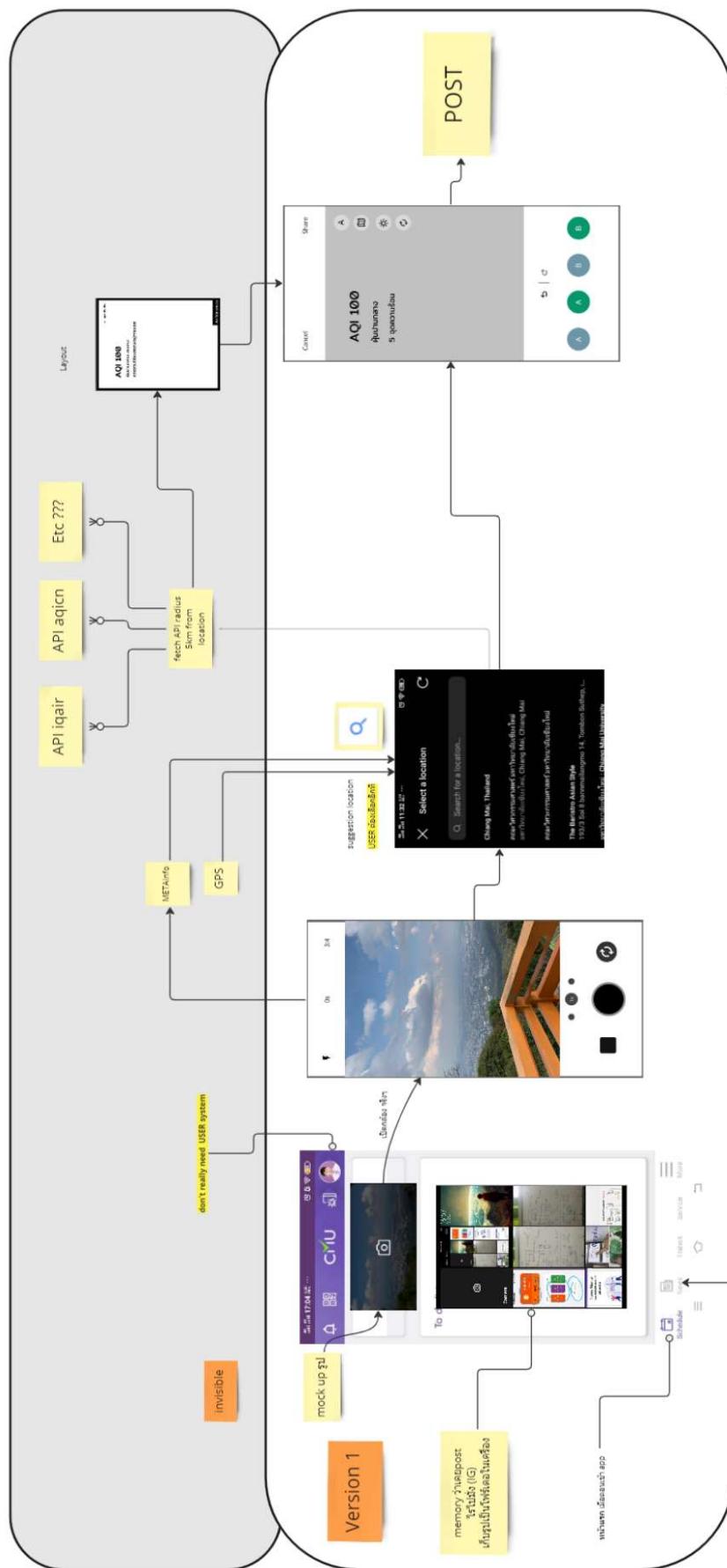
# ABOUT OUR PRODUCT

## INSPIRATION



Air Pollution Awareness via Photo Sharing (AP)  
A4 - Sprint #1 (Presentation slides and Video)





Air Pollution Awareness via Photo Sharing (AP)  
A4 - Sprint #1 (Presentation slides and Video)

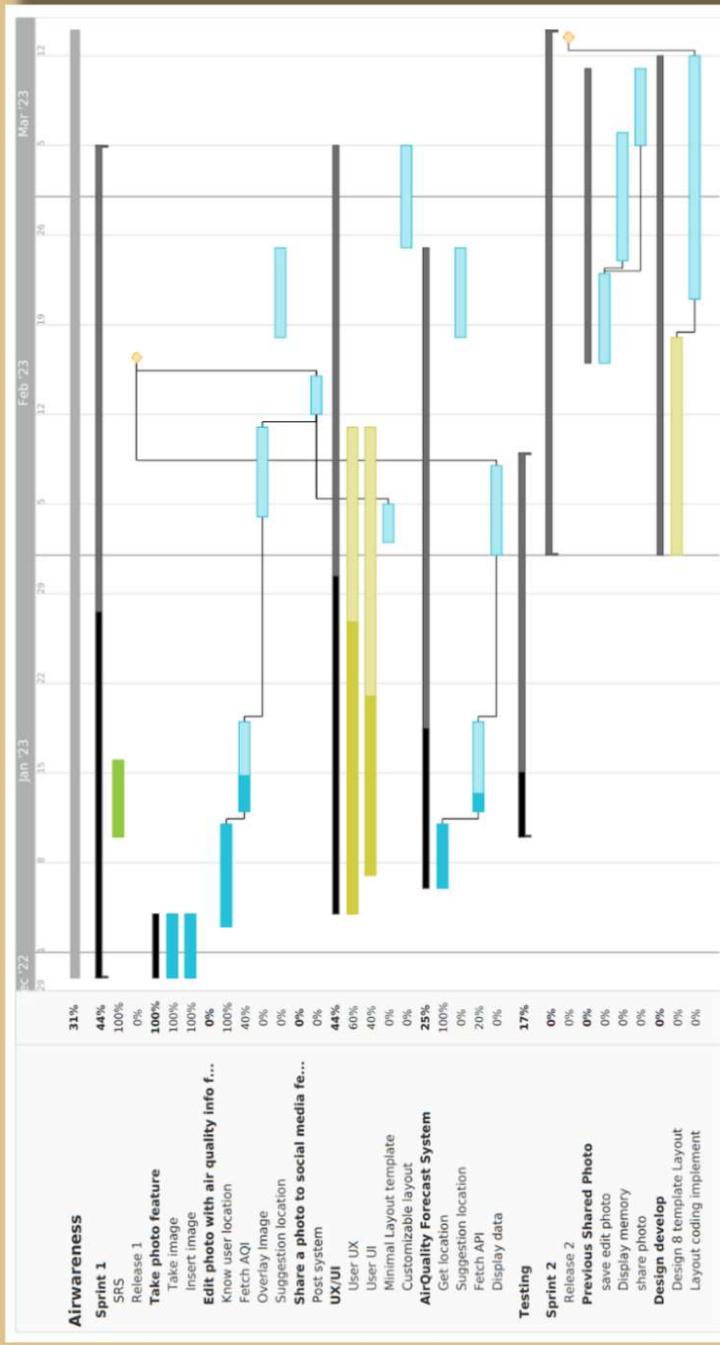
## CODING



## PROJECT MANAGEMENT



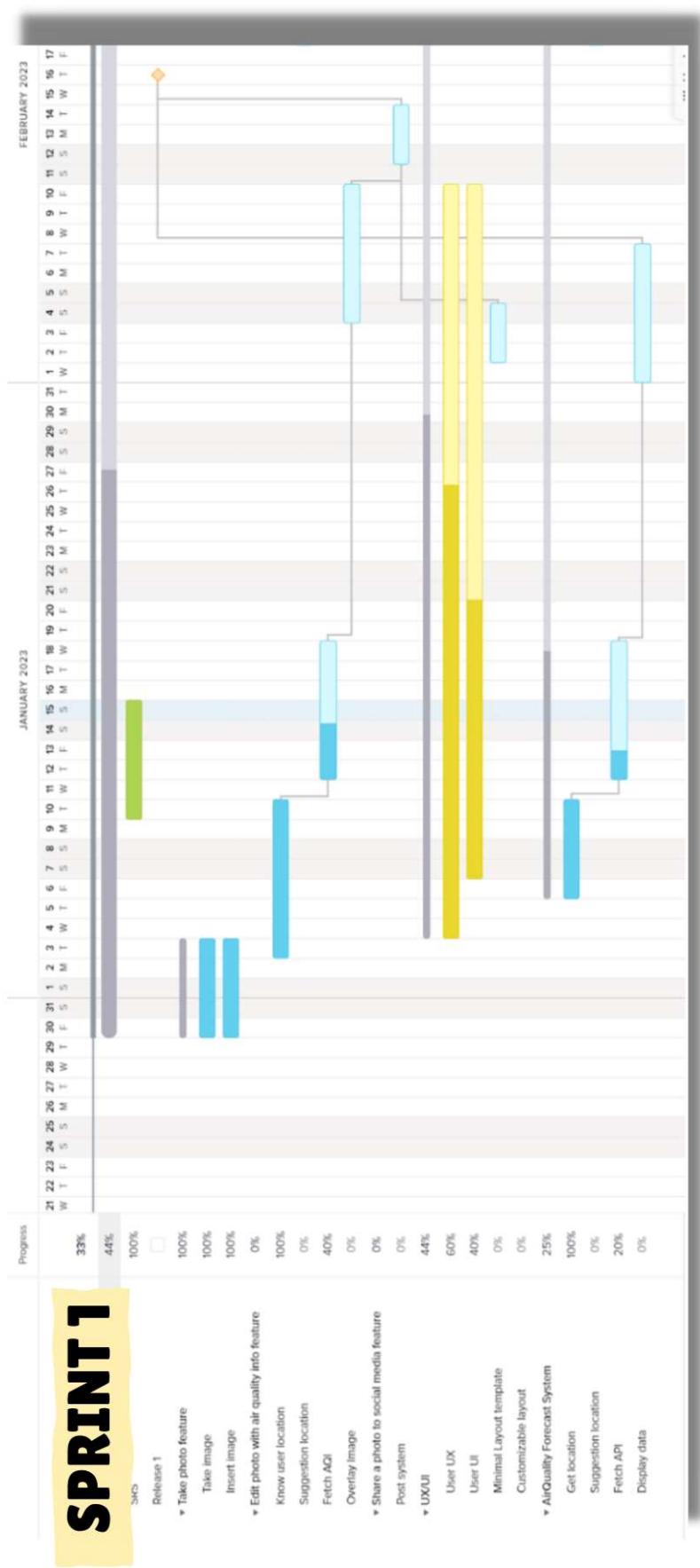
## SCOPE, AND FEATURE LISTS IN SPRINT #1 AND SPRINT #2.



Air Pollution Awareness via Photo Sharing (AP)  
A4 - Sprint #1 (Presentation slides and Video)



## Air Pollution Awareness via Photo Sharing (AP) A4 - Sprint #1 (Presentation slides and Video)



> Implement all of the requirements of the AirQuality Forecast System in the "System Features" section.

> Should implement some of "Take photo feature" and "Edit photo feature" requirements.

## SPRINT 1 - FEATURE LIST

### Forecast

### Choose Location

### Take & import photo

### AQI Overlay Data

### Save Photo



Air Pollution Awareness via Photo Sharing (AP)  
A4 - Sprint #1 (Presentation slides and Video)

## SPRINT 2 – FEATURE LIST

### Photo Memory



### Photo Editor

- Overlay Widget

### Post Photo on Social Media



Air Pollution Awareness via Photo Sharing (AP)  
A4 - Sprint #1 (Presentation slides and Video)

# SPRINT 1 - TESTING

## AUTOMATED TESTING & MANUAL TESTING

### WHITE-BOX TESTING (AUTOMATED UNIT & WIDGET TEST)

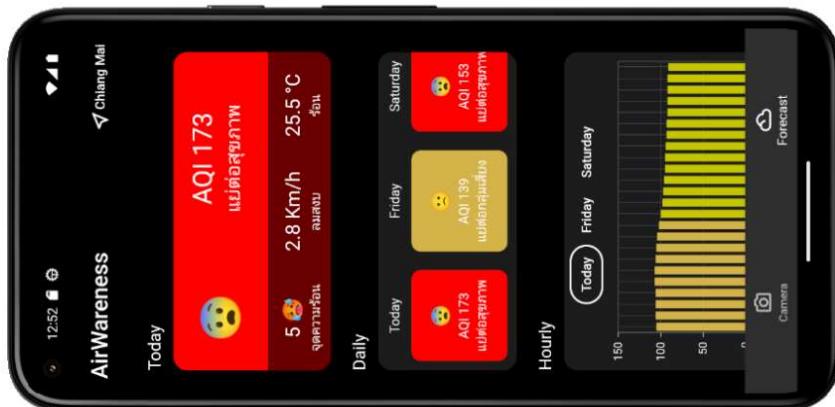
- SUGGEST LOCATION

- AIR QUALITY

- FORECAST PAGE WIDGETS, CAMERA PAGE WIDGETS

### BLACK-BOX TESTING (MANUAL TEST)

- USER'S CURRENT LOCATION
- FORECAST, MEMORY USER INTERACT
- SEARCH LOCATION & SUGGEST LOCATION
- TAKE A PHOTO & IMPORT PHOTO FROM GALLERY
- EDITING PHOTO WITH OVERLAY
- SAVING & VIEWING EDITED PHOTO



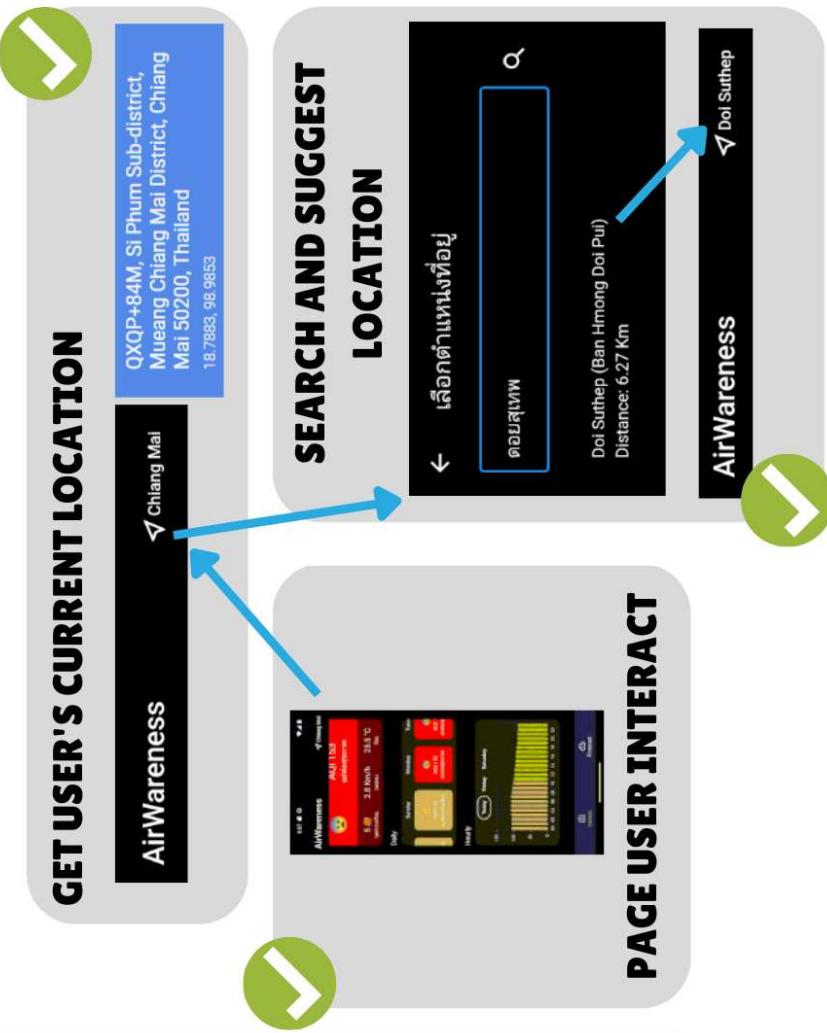
# SPRINT 1 – TESTING

## AUTOMATED UNIT & WIDGET TEST

### RESULT

```
Running tests, 10/10 passed (100%)
  ✓ integration_test/app_test.dart 1/1 passed: 21ms
    ○ [Integration_Test]
      ✓ (tearDownAll) 21ms
  ✓ test/unit_test.dart 5/5 passed: 1.6s
    > ✓ [Unit_Test] [AirQualityApi] 3/3 passed: 1.6s
      ○ fetchAirQuality returns success response 966ms
      ○ getAirQuality5day returns success response 653ms
      ○ fromJson creates valid AirQuality object 11ms
    > ✓ [Unit_Test] [Location] 2/2 passed: 18ms
      ○ fromJson0 should return a Suggestlocation object with correct values 7.0ms
      ○ should return a list of Suggestlocation objects 11ms
    > ✓ test/widget_test.dart 5/5 passed: 2.2s
      ○ [Widget_Test] [Page] 4/4 passed: 2.0s
        ○ should have application name, location 1.4s
        ○ should have Camera, Forecast button 122ms
        ○ forecast page must contains Today, Daily, Hourly 284ms
        ○ camera page must contains memory and camera button 183ms
      > ✓ [Widget_Test] [MainScreen -> Camera] 1/1 passed: 122ms
        ○ can open built in camera and close 122ms
```

## MANUAL TEST



# SPRINT 1 – TESTING

**TAKE A PHOTO & IMPORT  
PHOTO FROM GALLERY**



## EDITING PHOTO WITH OVERLAY



Air Pollution Awareness via Photo Sharing (AP)  
A4 - Sprint #1 (Presentation slides and Video)

# SPRINT 1 - TESTING

SAVING EDITED PHOTO

SAVED TO DCIM/AIRWARENESS

WITH OTHER  
PHOTO SIZE RATIO

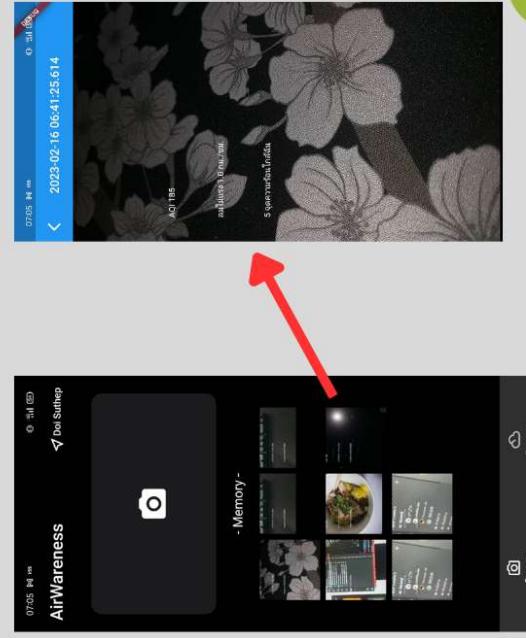


**WITH 9:16  
PHOTO SIZE RATIO**

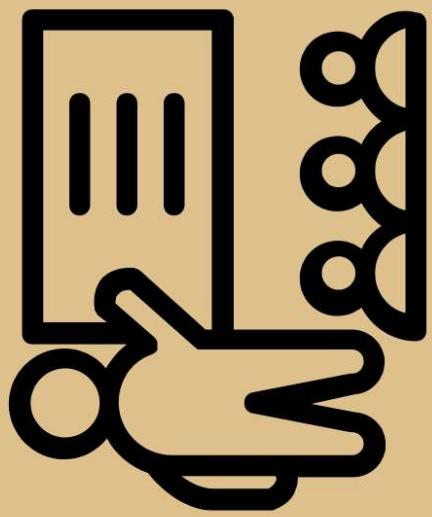


**SWITCHED CAMERA TO OTHER RATIO EXAMPLE 1:1 RATIO  
THE SAVED PHOTO WILL FAIL IF INCLUDED SOME UNWANT BLACK AREA**

## **VIEWING EDITED PHOTO (MEMORY)**



# DEMONSTRATION



Air Pollution Awareness via Photo Sharing (AP)  
A4 - Sprint #1 (Presentation slides and Video)

# RETROSPECTIVE SPRINT 1

## What was good

- Great team spirit
- Good pairing work

## What was bad

- Time management
- Free-tier APIs have limits
- Hourly data (from API) have low accuracy

## IDEAS

- Each backlog needs clear acceptance criteria

## ACTIONS

- More necessary, short, and informative meetings

## Sprint 2 :



> Implement the remaining requirements of “Take photo feature” and “Edit photo feature”.

> Implement all of the requirements on “Previous Edited/Shared Photo (Photo Memory)” and “Share a photo to social media feature”

> Improve UI/UX on all of the features implemented.



**SANPAWAT KANTABUTRA, PH.D.**



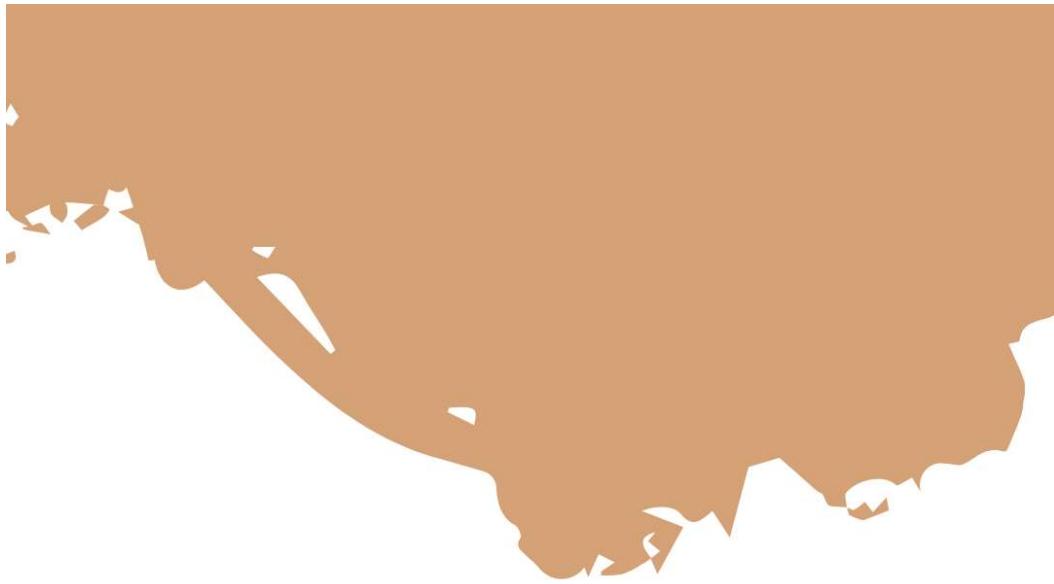
**YUTHAPONG SOMCHIT, PH.D.**

5 Strongly Agree  
1 Strongly Disagree

## SUS SCORE: 70.0

B = Good

I think that I would like to use this application frequently.	5
I found this application unnecessarily complex.	4
I thought this application was easy to use.	5
I think that I would need assistance to be able to use this application.	2
I found the various functions in this application were well integrated.	4
I thought there was too much inconsistency in this application.	2
I would imagine that most people would learn to use this application very quickly.	4
I found this website very cumbersome/awkward to use.	2
I felt very confident using this application.	4
I needed to learn a lot of things before I could get going with this application.	4



#### A(ir)Awareness by AreYouWorking

Air pollution awareness via Photo sharing Users can take or upload photos and customize whether to add PM2.5, AQI, heat map or other air pollution data to the photos before saving or sharing.

"ช้าาาชึ่ง ให้หน่อยไก่กับกุบกุบคุบวนมากกว่าที่เดิม สงอยากจะเพื่อนความดีจะหันเรื่องปุ่นคุณที่เกลี้ด้า โดเบลพาหย์ Social Media"

#### [Areyouworking.github.io](#)



Sprint 1 Video: [https://drive.google.com/drive/folders/18fWZ5Twyboc\\_iXTGC3Q6i5MiT3pxdqn?usp=share\\_link](https://drive.google.com/drive/folders/18fWZ5Twyboc_iXTGC3Q6i5MiT3pxdqn?usp=share_link)

Air Pollution Awareness via Photo Sharing (AP)  
A4 - Sprint #1 (Presentation slides and Video)

## A5 - Test Report

**Software name:** AirAwareness (Air pollution awareness via Photo sharing)

**Team name:** AreYouWorking

**Tester name:** Thanawat Bumpengpun (630610736)

### Software scope

จะต้องรองรับการใช้งานบนระบบ Android และจะต้องมีฟังก์ชันที่ทำงานได้ดังนี้

- สามารถดึงตำแหน่งที่อยู่ ณ ปัจจุบันของผู้ใช้งานจาก GPS ได้, สามารถค้นหาตำแหน่งอื่นๆ และเปลี่ยนไปยังตำแหน่งนั้นได้ โดย Suggest location บริเวณใกล้ๆ GPS หรือคำค้นหา
- สามารถแสดงข้อมูลคุณภาพอากาศโดยอ้างอิงจากตำแหน่งปัจจุบันของผู้ใช้งานหรือตำแหน่งที่ผู้ใช้งานเลือก โดยแสดงข้อมูลในวันปัจจุบัน, คาดการณ์คุณภาพอากาศได้ 7 วันล่วงหน้า, สามารถแสดงข้อมูลคุณภาพอากาศเป็นรายชั่วโมง ในวันปัจจุบัน วันถัดไป และวันมะรืน
- ผู้ใช้งานสามารถเลือกนำเข้ารูปภาพ หรือถ่ายรูปภาพ (กล้องหน้า-หลัง) ด้วยอัตราส่วนรูปภาพขนาด 3:4, 1:1, 9:16, 4:5 และเลือกเปิด-ปิด-ออโต้ แฟลช ได้
- สามารถนำข้อมูลคุณภาพอากาศมาสร้างเป็น Overlay ใส่ลงบนรูปภาพของผู้ใช้งานได้, Overlay สามารถขับ, หมุน, ย่อ-ขยาย, ลบ, เปลี่ยน Style ได้, สามารถเลือก Add overlay ที่มีให้ได้ตามความต้องการ
- สามารถค้นหาและย้ายตำแหน่ง GPS ขณะใส่ Overlay บนรูปภาพได้
- สามารถบันทึกรูปภาพที่มี Overlay ข้อมูลคุณภาพอากาศลงใน Album ของผู้ใช้ได้, สามารถแชร์รูปภาพดังกล่าวไปยัง Social media เช่น Facebook Post, Instagram Story ได้
- สามารถดูรูปภาพที่เคยทำการแชร์ หรือบันทึกลง Album ได้ในหน้า Memory, สามารถเลือกกดเข้าไปดูรูปภาพแบบเต็มจอ และแสดงวันที่และเวลาที่รูปนั้นถูกบันทึก
- User Interface จะต้องรองรับหน้าจอของมือถือ Android ในแนวตั้งได้ทุกขนาด

### Test scope (what is tested and what is not tested)

#### สิ่งที่ทำการทดสอบ (what is tested)

- ฟังก์ชันการดึงข้อมูลคุณภาพอากาศจาก Air Quality API
- ฟังก์ชันการดึงตำแหน่งที่อยู่ของผู้ใช้งานจาก GPS, ระบบ Suggest location จากตำแหน่ง GPS หรือคำค้นหา
- ฟังก์ชันการแปลง Response ที่ได้จาก API ไปเป็น Data object
- การมีอยู่ของ Widgets ต่างๆ เช่น Buttons, Application name, Labels
- ฟังก์ชันการทำงานของแต่ละ Widgets ใน User Interface เช่น กดปุ่มแล้วจะต้องเปลี่ยนหน้า, กดเพื่อเปลี่ยนวัน เช่น ดูข้อมูลคุณภาพอากาศในวันถัดๆ ไป ดูข้อมูลแบบรายชั่วโมง และอื่นๆ
- ฟังก์ชันการสร้างและการทำงานของ Text Overlay สำหรับใส่ลงบนรูปภาพ จากข้อมูลคุณภาพอากาศที่ได้จาก API

- พังก์ชันการนำเข้ารูปภาพ
- พังก์ชันของกล้องถ่ายรูป เช่น การเปลี่ยนอัตราส่วนรูปภาพ, เปิด-ปิด แฟลช, ลับกล้องหน้า-หลัง
- พังก์ชันการเปลี่ยนตำแหน่ง GPS ขณะตกแต่งรูป
- พังก์ชันการบันทึกรูปภาพลงใน Album และการแชร์รูปภาพไปยัง Social Media
- พังก์ชันการตู้รูปภาพที่เคยบันทึกหรือแชร์
- พังก์ชันการร้องขอ Permission เมื่อใช้งานแอปพลิเคชันครั้งแรก เช่น ขอ Permission เข้าถึง GPS และ Storage ของโทรศัพท์
- User Interface มีการรองรับขนาดหน้าจอต่างๆของโทรศัพท์ในแนวตั้ง

#### สิ่งที่ไม่ได้ทำการทดสอบ (what is not tested)

- ความสม่ำเสมอของรูปแบบ Response ที่ได้จาก Air Quality Index API
- ประสิทธิภาพของแอปพลิเคชัน เช่น ความเร็วในการดึงข้อมูลจาก API, ความเร็วในการค้นหาและดึงข้อมูลของระบบ Suggest location

#### Test exit criteria

- ต้องดำเนินการทดสอบทุก Test cases ที่สร้างไว้ และผ่านทั้งหมด
- ต้องไม่พบปัญหาสำคัญใดๆในแอปพลิเคชัน ทั้งบน Android Emulator และบนอุปกรณ์จริง
- User Interface ต้องตอบสนองได้ตามที่ออกแบบไว้

#### Test overview (list of test types performed)

- Unit Testing
- Widget Testing (Automated widget test using Flutter's testing package)
- Manual Testing

#### Activities performed for each test type

##### Unit Testing

1. ดึงข้อมูล Air Quality API
2. ดึงข้อมูล Air Quality API ด้วย Negative Latitude Longitude
3. ดึงข้อมูล Air Quality รายชั่วโมง
4. แปลง JSON response ของ Air Quality ไปเป็น Air Quality Data object
5. แปลง JSON response ของ Suggest location ไปเป็น Location Data object
6. แปลง JSON response หลายตัวของ Suggest location ไปเป็น List ของ Location Data object

##### Widget Testing

1. Application name label ในหน้า Forecast และ Camera
2. City name label ในหน้า Forecast และ Camera

3. ปุ่มกดไปยังหน้า Forecast และ Camera
4. Widget แสดงข้อมูล Air Quality แบบ Today, Daily, Hourly ในหน้า Forecast
5. Section Memory และปุ่มกดถ่ายรูปภาพในหน้า Camera
6. เปลี่ยนหน้าไปยัง Camera กดเปิดกล้องแล้วปิดกล้อง

## Manual Testing

1. แสดง GPS Location Permission Request
2. กด Deny ที่ GPS Location Permission Request
3. กด Allow ที่ GPS Location Permission Request
4. แสดง Storage Permission Request
5. กด Deny ที่ Storage Permission Request
6. กด Allow ที่ Storage Permission Request
7. แสดง Camera Permission Request
8. กด Deny ที่ Camera Permission Request
9. กด Allow ที่ Camera Permission Request
10. Refresh ข้อมูล Air Quality ในหน้า Forecast
11. แสดงชื่อเมืองปัจจุบันในหน้า Forecast
12. แสดงชื่อเมืองปัจจุบันในหน้า Camera
13. แสดงข้อความ Hotspot, Wind, Temperature ใน Section Today
14. แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 44
15. แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 95
16. แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 128
17. แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 168
18. แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 222
19. แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 310
20. แสดงข้อมูล Air Quality Daily เมื่อเลื่อน Section Daily
21. แสดงกราฟและสีแท่งกราฟใน Section Hourly เมื่อเลือก Today
22. แสดงกราฟและสีแท่งกราฟใน Section Hourly เมื่อเลือกวันถัดไป
23. แสดงกราฟและสีแท่งกราฟใน Section Hourly เมื่อเลือกวันมะรืน
24. แสดงหน้าเลือกตำแหน่งที่อยู่จากหน้า Forecast
25. แสดง Suggest location ในหน้าเลือกตำแหน่งที่อยู่
26. แสดง Suggest location เมื่อกดปุ่ม Refresh ในหน้าเลือกตำแหน่งที่อยู่
27. แสดง Suggest locations เมื่อใส่ข้อความ “bang” ในหน้าเลือกตำแหน่งที่อยู่
28. แสดง Suggest locations เมื่อใส่ข้อความ “phit” ในหน้าเลือกตำแหน่งที่อยู่
29. เรียกแสดงหน้า Camera
30. เรียกการใช้งานกล้องถ่ายรูป
31. เรียกการใช้งานการนำเข้ารูปภาพ
32. ปรับอัตราส่วนกล้องถ่ายรูปเป็น 3:4

33. ปรับอัตราส่วนกล้องถ่ายรูปเป็น 4:5
34. ปรับอัตราส่วนกล้องถ่ายรูปเป็น 1:1
35. ปรับอัตราส่วนกล้องถ่ายรูปเป็น 9:16
36. ปรับโหมดแฟลชเป็น ปิด
37. ปรับโหมดแฟลชเป็น เปิด
38. ปรับโหมดแฟลชเป็น ออโต้
39. สลับกล้องหน้า-หลัง
40. เปลี่ยนไปหน้า Photo editor จากการนำเข้ารูปภาพ
41. เปลี่ยนไปหน้า Photo editor จากการถ่ายรูป
42. แสดง Text Overlay ในหน้า Photo editor
43. กดที่ Text Overlay 1 ครั้ง
44. กดที่ Text Overlay 2 ครั้ง
45. กดที่ Text Overlay 3 ครั้ง
46. ลาก Text Overlay ตามนิ่วมือ
47. หมุน Text Overlay ตามนิ่วมือ
48. ย่อ Text Overlay ตามนิ่วมือ
49. ขยาย Text Overlay ตามนิ่วมือ
50. ลบ Text Overlay
51. กดปุ่มเพิ่ม Overlay
52. เลือกเพิ่ม Text Overlay จากการกดปุ่มเพิ่ม Overlay
53. กดปุ่มเปลี่ยน Location ในหน้า Photo editor
54. กดปุ่มบันทึกรูปภาพ
55. กดปุ่มแชร์รูปภาพ
56. กดปุ่มกลับจากหน้า Photo editor
57. กดปุ่มปิดจากหน้ากล้องถ่ายรูป
58. การแสดงผลใน Section Memory ในหน้า Camera
59. การแสดงผลรูปใน Section Memory
60. User Interface ย่อ-ขยายขนาด

## Results

### Unit Testing

NO.	Test Description	Expected Result	Status
1.	ดึงข้อมูล Air Quality API	return status code 200	Pass
2.	ดึงข้อมูล Air Quality API ด้วย Negative Latitude Longitude	return status code 200	Pass
3.	ดึงข้อมูล Air Quality รายชั่วโมง	Air quality object is not null	Pass
4.	แปลง JSON response ของ Air Quality ไปเป็น Air Quality Data object	Air quality object with correct values	Pass
5.	แปลง JSON response ของ Suggest location ไปเป็น Location Data object	Location object with correct values	Pass
6.	แปลง JSON response หลายตัวของ Suggest location ไปเป็น List ของ Location Data object	List of location objects with correct values	Pass

### Widget Testing

NO.	Test Description	Expected Result	Status
1.	Application name label ในหน้า Forecast และ Camera	พบรหัส “AirWareness” ในหน้า Forecast และ Camera	Pass
2.	City name label ในหน้า Forecast และ Camera	พบรหัส “Chiang Mai” ในหน้า Forecast และ Camera	Pass
3.	ปุ่มกดไปยังหน้า Forecast และ Camera	พบรหัส “Camera” และ Icon Camera, พบรหัส “Forecast” และ Icon Forecast	Pass
4.	Widget แสดงข้อมูล Air Quality แบบ Today, Daily, Hourly ในหน้า Forecast	พบรหัส “Today”, “Daily”, “Hourly” และข้อมูล	Pass
5.	Section Memory และปุ่มกดถ่ายรูปภาพในหน้า Camera	พบรหัส “Memory” และ Icon Camera	Pass

6.	เปลี่ยนหน้าไปยัง Camera กดเปิดกล้องแล้วปิดกล้อง	พบรหัส text label “Camera” และ Icon Camera แตะที่ Icon Camera แล้วกล้องเปิด	Pass
----	---	---	------

## Manual Testing

NO.	Test Description	Expected Result	Status
1.	แสดง GPS Location Permission Request	แสดง Location Permission Request เมื่อเข้าแอปพลิเคชันครั้งแรก	Pass
2.	กด Deny ที่ GPS Location Permission Request	แสดงการ Loading ข้อมูลคุณภาพอากาศ	Pass
3.	กด Allow ที่ GPS Location Permission Request	แสดงการ Loading และข้อมูลคุณภาพอากาศที่ได้เมื่อโหลดเสร็จ	Pass
4.	แสดง Storage Permission Request	แสดง Storage Permission Request เมื่อเข้ามาที่หน้า Camera ครั้งแรก	Pass
5.	กด Deny ที่ Storage Permission Request	ไม่แสดงรูปภาพจาก AirAwareness Album ใน Section Memory	Pass
6.	กด Allow ที่ Storage Permission Request	แสดงรูปภาพหากมีรูปภาพจาก AirAwareness Album ใน Section Memory	Pass
7.	แสดง Camera Permission Request	แสดง Camera Permission Request เมื่อเรียกใช้งานกล้องถ่ายรูปครั้งแรก	Pass
8.	กด Deny ที่ Camera Permission Request	ไม่แสดงกล้องถ่ายรูป	Pass
9.	กด Allow ที่ Camera Permission Request	แสดงกล้องถ่ายรูปและสามารถใช้กล้องถ่ายรูปได้	Pass
10.	Refresh ข้อมูล Air Quality ในหน้า Forecast	เมื่อแตะ และลากจ�ล  จะ Refresh ข้อมูล Air Quality ใหม่	Pass
11.	แสดงชื่อเมืองปัจจุบันในหน้า Forecast	แสดงชื่อเมืองถูกต้องตามตำแหน่งปัจจุบัน หรือตำแหน่งที่เปลี่ยนตามที่ค้นหา	Pass

12.	แสดงชื่อเมืองปัจจุบันในหน้า Camera	แสดงชื่อเมืองถูกต้องตามตำแหน่ง ปัจจุบันหรือตำแหน่งที่เปลี่ยนตามที่ค้นหา	Pass
13.	แสดงข้อความ Hotspot, Wind, Temperature ใน Section Today	แสดงจำนวนจุด Hotspot, ความเร็วลม, และ อุณหภูมิที่ถูกต้อง	Pass
14.	แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 44	แสดงพื้นหลังสีเขียว, ข้อความ "Good" และอีโมจิ "😊"	Pass
15.	แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 95	แสดงพื้นหลังสีเหลือง, ข้อความ "Moderate" และอีโมจิ "😐"	Pass
16.	แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 128	แสดงพื้นหลังสีส้ม, ข้อความ "Unhealthy for some" และอีโมจิ "🙁"	Pass
17.	แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 168	แสดงพื้นหลังสีส้ม-แดง, ข้อความ "Unhealthy" และอีโมจิ "阽"	Pass
18.	แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 222	แสดงพื้นหลังสีม่วง, ข้อความ "Very Unhealthy" และอีโมจิ "😱"	Pass
19.	แสดงพื้นหลังและข้อความใน Section Today, Daily เมื่อ AQI = 310	แสดงพื้นหลังสีแดง-ชมพู, ข้อความ "Hazardous" และอีโมจิ "ஓ"	Pass
20.	แสดงข้อมูล Air Quality Daily เมื่อเลือก Section Daily	แสดงพื้นหลัง, ข้อความ, อีโมจิ ของ ข้อมูลวันถัดไปได้ถูกต้อง	Pass
21.	แสดงกราฟและสีแท่งกราฟใน Section Hourly เมื่อเลือก Today	มีการเปลี่ยนแปลงข้อมูล และแสดงค่ากราฟและสีแท่งกราฟได้ถูกต้อง	Pass
22.	แสดงกราฟและสีแท่งกราฟใน Section Hourly เมื่อเลือกวันถัดไป	มีการเปลี่ยนแปลงข้อมูล และแสดงค่ากราฟและสีแท่งกราฟได้ถูกต้อง	Pass
23.	แสดงกราฟและสีแท่งกราฟใน Section Hourly เมื่อเลือกวันมะรืน	มีการเปลี่ยนแปลงข้อมูล และแสดงค่ากราฟและสีแท่งกราฟได้ถูกต้อง	Pass
24.	แสดงหน้าเลือกตำแหน่งที่อยู่จากหน้า Forecast	เมื่อกดที่ชื่อเมืองจากหน้า Forecast แล้วแสดงหน้าเลือกตำแหน่งที่อยู่	Pass
25.	แสดง Suggest location ในหน้าเลือกตำแหน่งที่อยู่	แสดง Suggest location ที่สัมพันธ์ กับตำแหน่งปัจจุบันในหน้าเลือกตำแหน่งที่อยู่	Pass
26.	แสดง Suggest location เมื่อกดปุ่ม Refresh ในหน้าเลือกตำแหน่งที่อยู่	Refresh Suggest location ที่ สัมพันธ์กับตำแหน่งใหม่ในหน้าเลือก	Pass

		ตำแหน่งที่อยู่	
27.	แสดง Suggest locations เมื่อใส่ ข้อความ “bang” ในหน้าเลือก ตำแหน่งที่อยู่	แสดง Suggest location “Bangkok” และระยะห่างเป็น ผลลัพธ์ในหน้าเลือกตำแหน่งที่อยู่	Pass
28.	แสดง Suggest locations เมื่อใส่ ข้อความ “phit” ในหน้าเลือกตำแหน่ง ที่อยู่	แสดง Suggest location “Phitsanulok” และระยะห่างเป็น ผลลัพธ์ในหน้าเลือกตำแหน่งที่อยู่	Pass
29.	เรียกแสดงหน้า Camera	กดปุ่ม Camera ที่แถบ Navigate แล้วเปลี่ยนหน้าไปหน้า Camera	Pass
30.	เรียกการใช้งานกล้องถ่ายรูป	กด Icon Camera แล้วเรียกใช้งาน กล้องถ่ายรูปภายในแอปพลิเคชัน	Pass
31.	เรียกการใช้งานการนำเข้ารูปภาพ	กด Icon Image แล้วเรียกการนำเข้า รูปภาพ	Pass
32.	ปรับอัตราส่วนกล้องถ่ายรูปเป็น 3:4	แสดง Camera Preview ด้วย อัตราส่วนรูป 3:4, ถ่าย และ Edit รูป ด้วยอัตราส่วน 3:4, บันทึกและแชร์ ด้วยรูป 3:4	Pass
33.	ปรับอัตราส่วนกล้องถ่ายรูปเป็น 4:5	แสดง Camera Preview ด้วย อัตราส่วนรูป 4:5, ถ่าย และ Edit รูป ด้วยอัตราส่วน 4:5, บันทึกและแชร์ ด้วยรูป 4:5	Pass
34.	ปรับอัตราส่วนกล้องถ่ายรูปเป็น 1:1	แสดง Camera Preview ด้วย อัตราส่วนรูป 1:1, ถ่าย และ Edit รูป ด้วยอัตราส่วน 1:1, บันทึกและแชร์ ด้วยรูป 1:1	Pass
35.	ปรับอัตราส่วนกล้องถ่ายรูปเป็น 9:16	แสดง Camera Preview ด้วย อัตราส่วนรูป 9:16, ถ่าย และ Edit รูปด้วยอัตราส่วน 9:16, บันทึกและ แชร์ด้วยรูป 9:16	Pass
36.	ปรับโหมดแฟลชเป็น ปิด	ไม่มีการใช้งานแฟลชเมื่อถ่าย	Pass
37.	ปรับโหมดแฟลชเป็น เปิด	มีการใช้งานแฟลชเมื่อถ่าย	Pass
38.	ปรับโหมดแฟลชเป็น ออโต้	มีการใช้งานแฟลชในที่มีด แสง และไม่ใช้งานเมื่อสว่าง	Pass
39.	สลับกล้องหน้า-หลัง	เมื่อกด Icon Swap แล้วจะสลับ	Pass

		กล้อง	
40.	เปลี่ยนไปหน้า Photo editor จากการนำเข้ารูปภาพ	เมื่อนำเข้ารูปภาพ จะเปลี่ยนหน้าไปยัง Photo editor พร้อมกับการ edit รูปที่นำเข้านั้น	Pass
41.	เปลี่ยนไปหน้า Photo editor จากการถ่ายรูป	เมื่อถ่ายรูป จะเปลี่ยนหน้าไปยัง Photo editor พร้อมกับการ edit รูปที่ถ่ายมาตามอัตราส่วนนั้น	Pass
42.	แสดง Text Overlay ในหน้า Photo editor	Text Overlay แสดงข้อความค่า AQI และ Recommend text ถูกต้อง	Pass
43.	กดที่ Text Overlay 1 ครั้ง	จาก Style 1 เปลี่ยน Style ของ Text Overlay เป็น Style 2	Pass
44.	กดที่ Text Overlay 2 ครั้ง	จาก Style 1 เปลี่ยน Style ของ Text Overlay เป็น Style 3	Pass
45.	กดที่ Text Overlay 3 ครั้ง	จาก Style 1 เปลี่ยน Style ของ Text Overlay เป็น Style 4	Pass
46.	ลาก Text Overlay ตามนิ้วมือ	แตะที่ Text Overlay แล้วขยับตามตำแหน่งนิ้ว	Pass
47.	หมุน Text Overlay ตามนิ้วมือ	แตะที่ Text Overlay แล้วหมุนตามตำแหน่งนิ้ว	Pass
48.	ย่อ Text Overlay ตามนิ้วมือ	แตะที่ Text Overlay แล้วย่อขนาดเมื่อหุบนิ้ว	Pass
49.	ขยาย Text Overlay ตามนิ้วมือ	แตะที่ Text Overlay แล้วขยายขนาด เมื่อกางนิ้ว	Pass
50.	ลบ Text Overlay	แตะที่ Text Overlay แล้วลากมาที่ Icon ถังขยะ Text Overlay นั้นจะถูกลบ	Pass
51.	กดปุ่มเพิ่ม Overlay	แสดงแถบเพิ่ม Text Overlay ที่มีให้	Pass
52.	เลือกเพิ่ม Text Overlay จากการกดปุ่มเพิ่ม Overlay	สร้าง Text Overlay ลงบนรูป	Pass
53.	กดปุ่มเปลี่ยน Location ในหน้า Photo editor	สร้าง Text Overlay ลงบนรูปใหม่ โดยอ้างอิงจากชื่อมูลคุณภาพอากาศใน Location นั้นได้ถูกต้อง	Pass
54.	กดปุ่มบันทึกรูปภาพ	บันทึกรูปภาพที่มี Text Overlay	Pass

		ตามอัตราส่วนได้อย่างถูกต้องใน AirAwareness Album	
55.	กดปุ่มแชร์รูปภาพ	แชร์รูปภาพที่มี Text Overlay ตาม อัตราส่วนได้อย่างถูกต้องใน Social Media	Pass
56.	กดปุ่มกลับจากหน้า Photo editor	กลับมายังหน้ากล้องถ่ายรูป แล้ว กล้องถ่ายรูปทำงานได้	Pass
57.	กดปุ่มปิดจากหน้ากล้องถ่ายรูป	กลับมายังหน้า Camera และแสดง Section Memory ได้ถูกต้อง	Pass
58.	การแสดงผลใน Section Memory ใน หน้า Camera	แสดงรูปภาพทั้งหมดที่อยู่ใน AirAwareness Album ได้อย่างถูก ต้อง	Pass
59.	การแสดงผลรูปใน Section Memory	เมื่อกดที่รูป จะแสดงรูปเต็มจอ และแสดงข้อมูลวันและเวลาที่บันทึกของ รูปภาพ	Pass
60.	User Interface ย่อ-ขยายขนาด	User Interface ขยายตามขนาด หน้าจอ Resizable Android Emulator	Pass

#### Feedback and Suggestion

- UI/UX Feedback: Icon สำหรับปุ่มการเข้าไปยังหน้าเลือกตำแหน่งที่อยู่ ยังสื่อสารได้ไม่ชัดเจนพอ, สีพื้นหลังของ Section Today ระหว่างแบบ AQI และข้อมูล Hotspot, Wind, Temperature อาจปรับเปลี่ยนให้มีความ Contrast เพื่อการมองเห็นที่ดีขึ้นได้

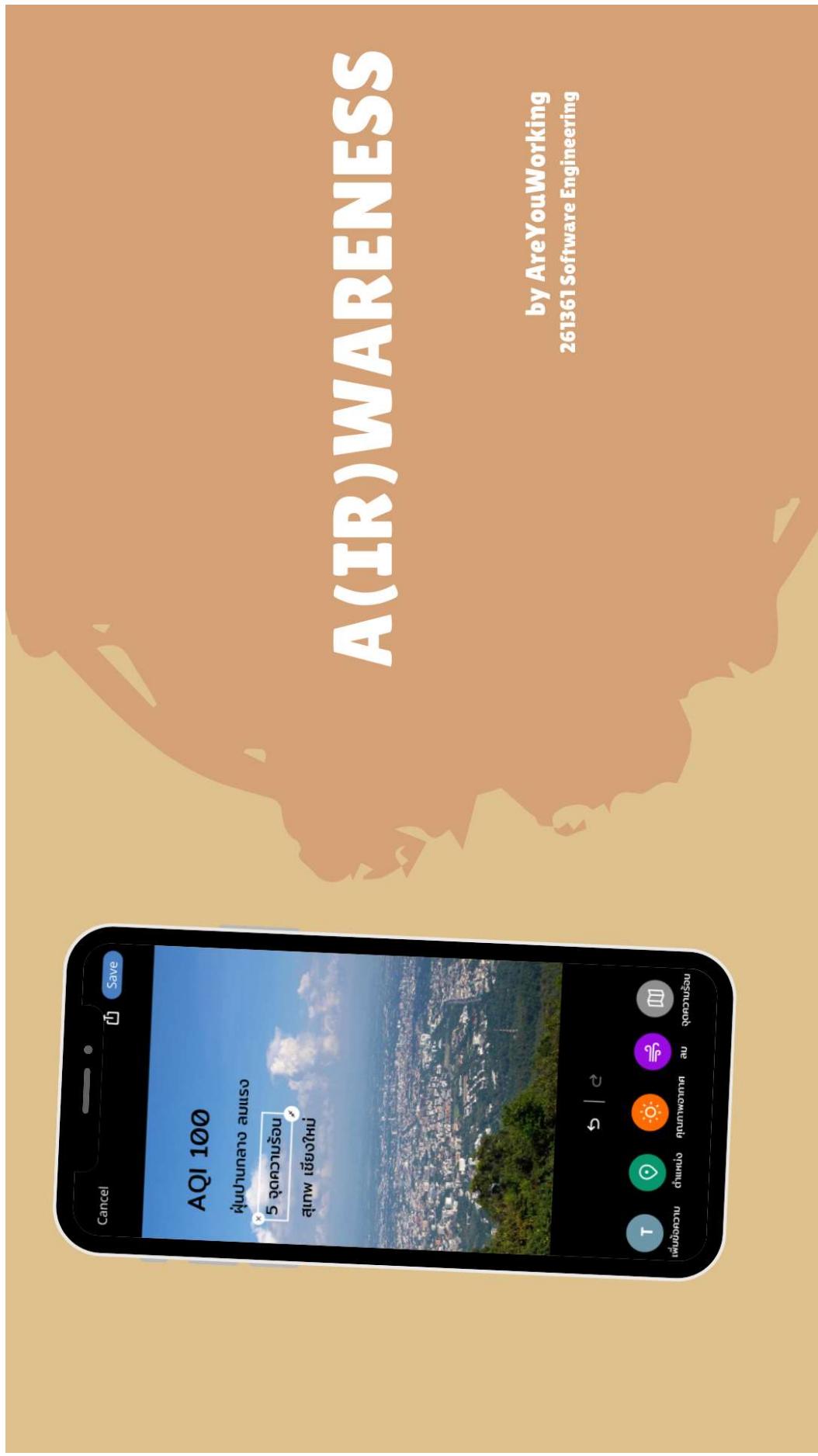
#### Recommendation

ควรทำการทดสอบ ความสม่ำเสมอของ Response ที่ได้จาก API ที่เราใช้งานด้วย แม้ว่า จะไม่ใช่หน้าที่การรับผิดชอบของเรา และการทำงานของฟังก์ชันเราก็ไม่ได้ผิด แต่ Response ที่ ไม่สม่ำเสมอของ API ที่เราใช้งานบางครั้งอาจทำให้ฟังก์ชันไม่ทำงานไม่ตามที่ควรจะเป็น

#### Summary

จากการทดสอบทั้ง Unit Testing, Widget Testing และ Manual Testing ทั้งในระบบ Android Emulator และบนอุปกรณ์จริง เมื่อพับเฉพาะปัญหาและทำการแก้ไขปัญหาให้ผ่านทั้งหมด ตาม Test exit criteria แล้ว ทำให้ตอนนี้ทุกฟังก์ชันที่ออกแบบไว้ตามใน Software scope ทำงานได้ครบถ้วนและเรียบร้อยดี

## A6 - Sprint #2 (Presentation slides and Video)



Air Pollution Awareness via Photo Sharing (AP)  
A6 - Sprint #2 (Presentation slides and Video)

## OUR TEAM MEMBER

**PONGSAKORN RATTANAPAN**  
PROJECT MANAGER & SA



**TANAT TANGUN**  
DEVELOPER



**PURICH SEENUALLAE**  
DEVELOPER, UX/UI  
DESIGNER

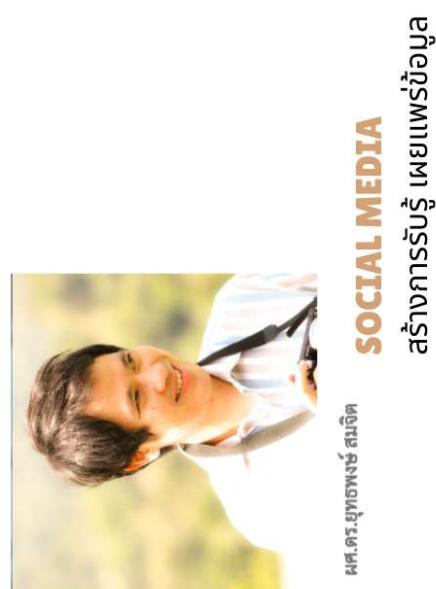
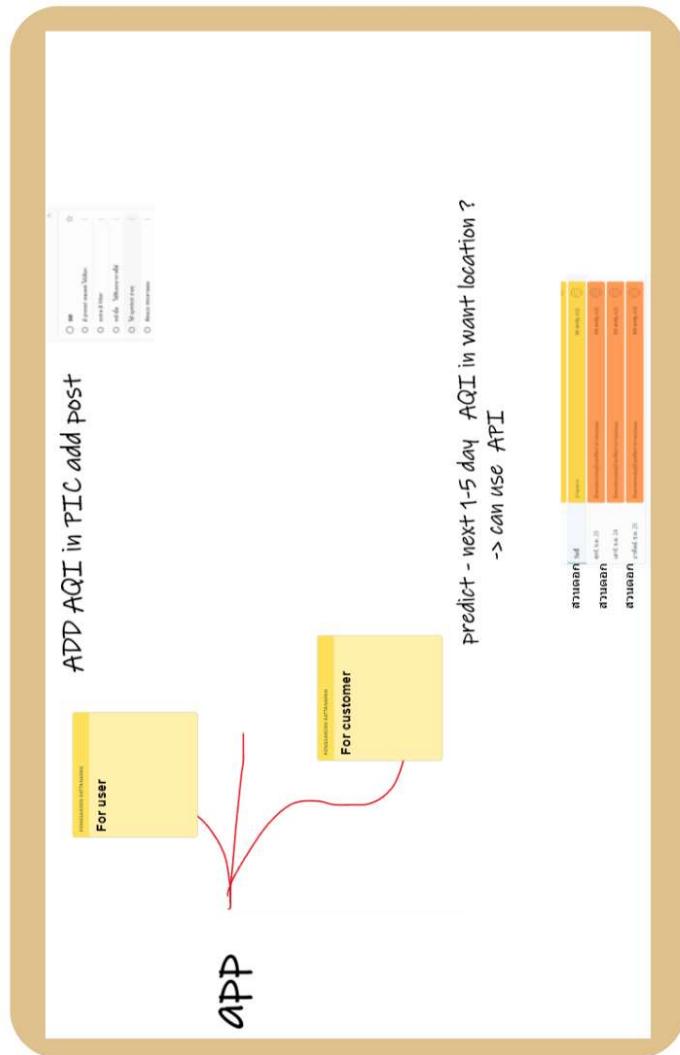


**TANADOL DEACHPRAPAKORN**  
DEVELOPER



**THANAWAT BUMPENGUN**  
TESTER

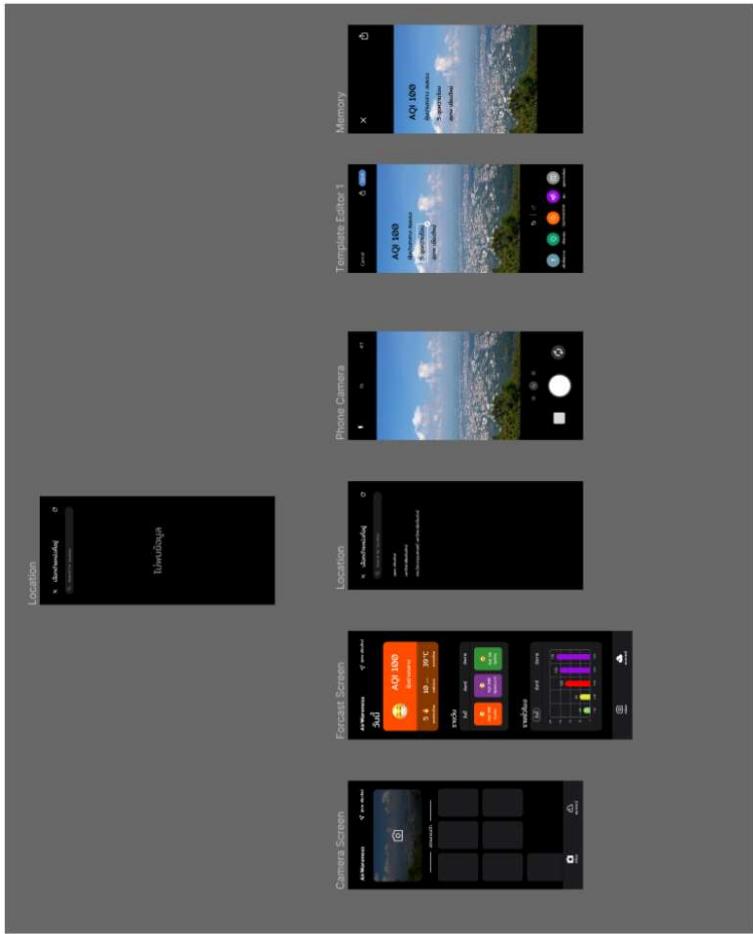
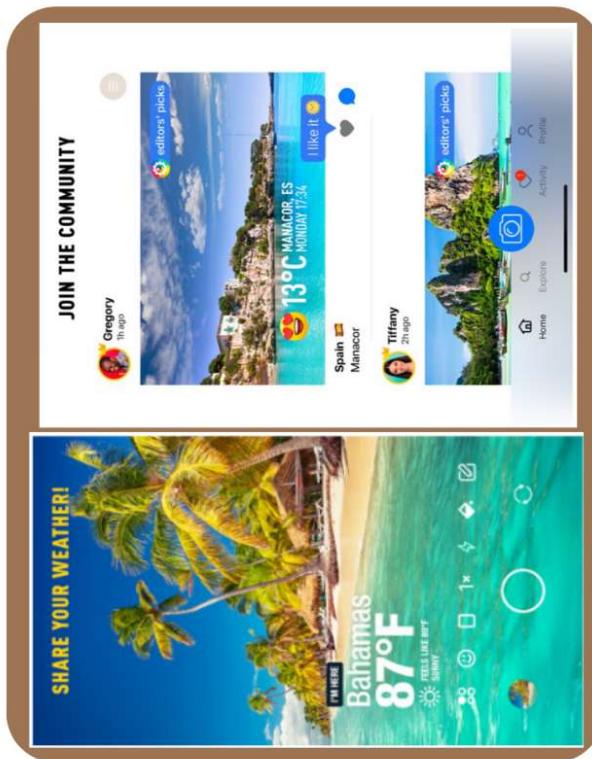




Air Pollution Awareness via Photo Sharing (AP)  
A6 - Sprint #2 (Presentation slides and Video)

# ABOUT OUR PRODUCT

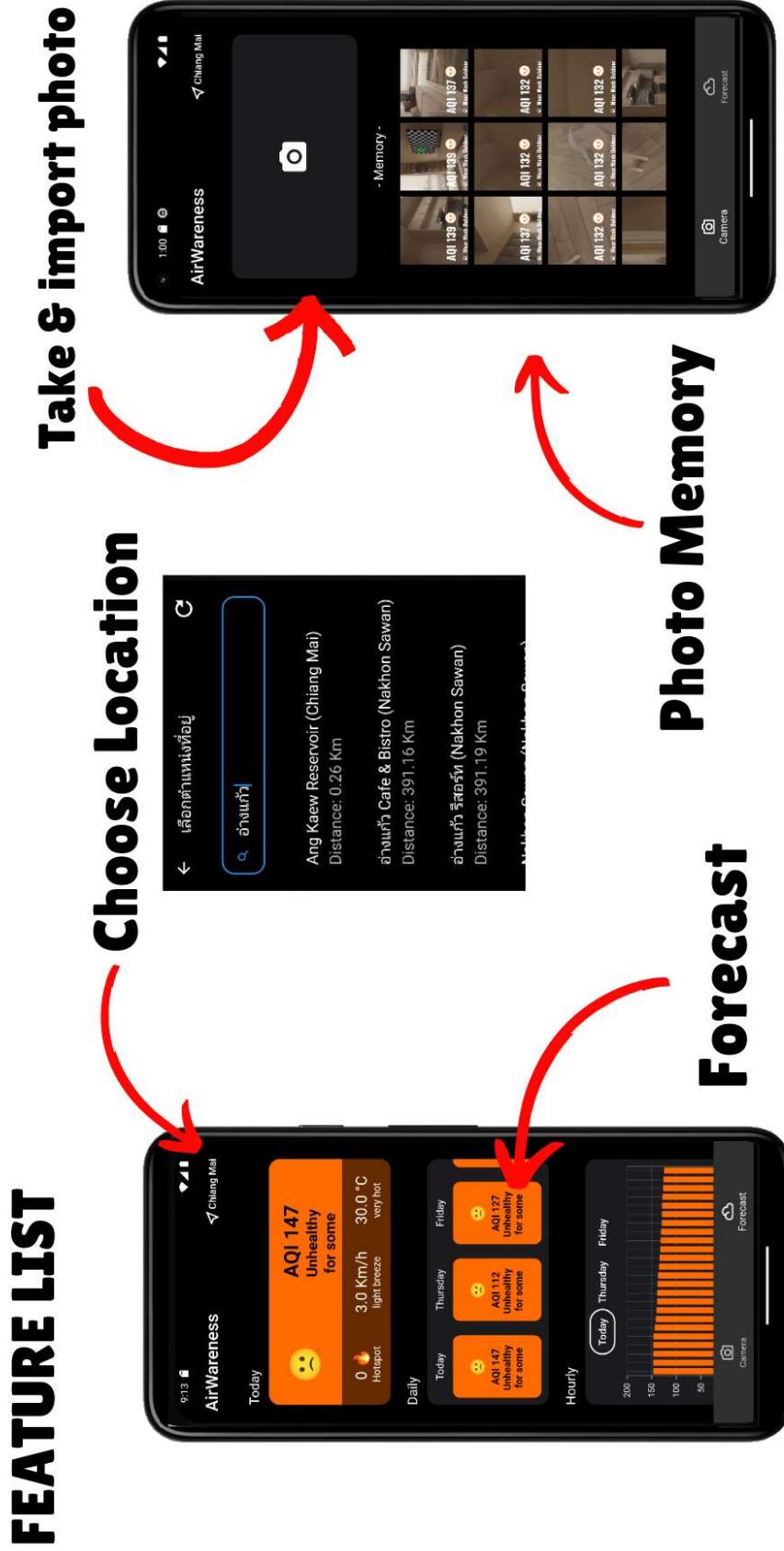
## INSPIRATION



Air Pollution Awareness via Photo Sharing (AP)  
A6 - Sprint #2 (Presentation slides and Video)

# Feature list

**UI****Forecast****Camera**



# Today AQI

# Forecast

-Daily  
-Hourly



# Phuket



Air Pollution Awareness via Photo Sharing (AP)  
A6 - Sprint #2 (Presentation slides and Video)

# Lampang



# chiang mai



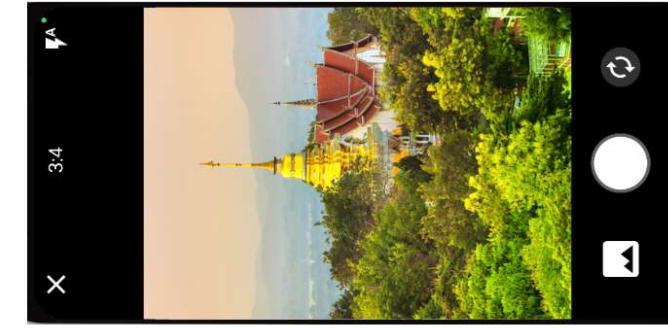
## Take & import photo



## Photo Memory

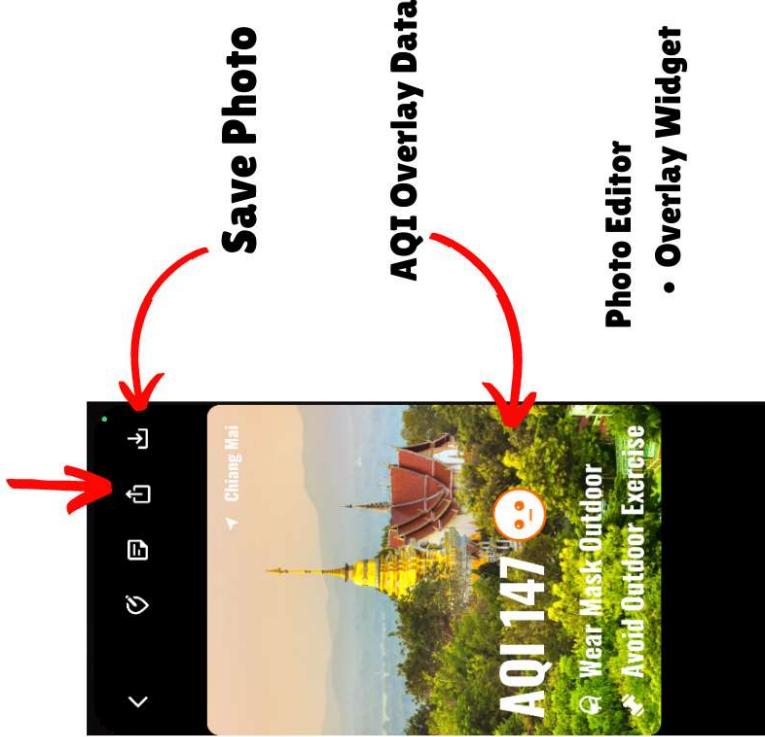
Air Pollution Awareness via Photo Sharing (AP)  
A6 - Sprint #2 (Presentation slides and Video)

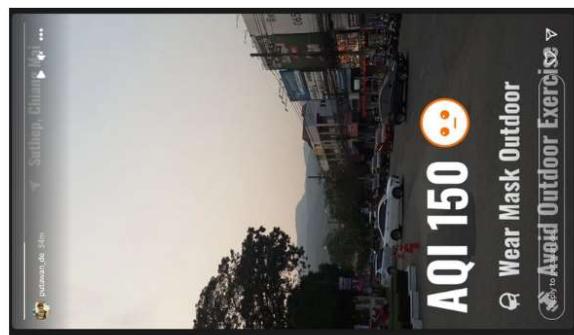
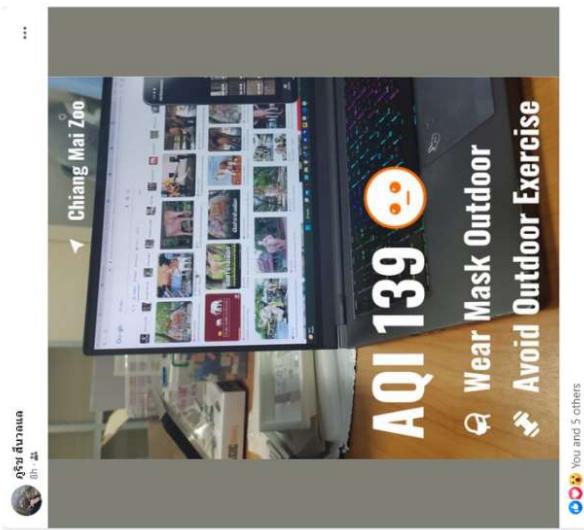
## FEATURE LIST



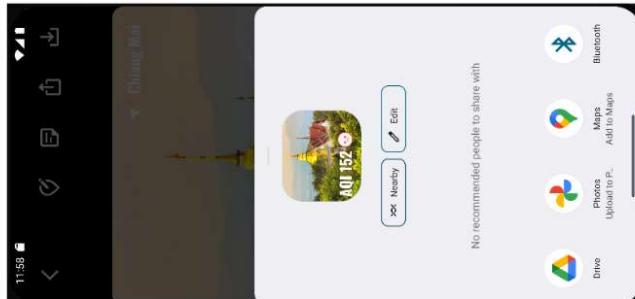
**Take & import photo**

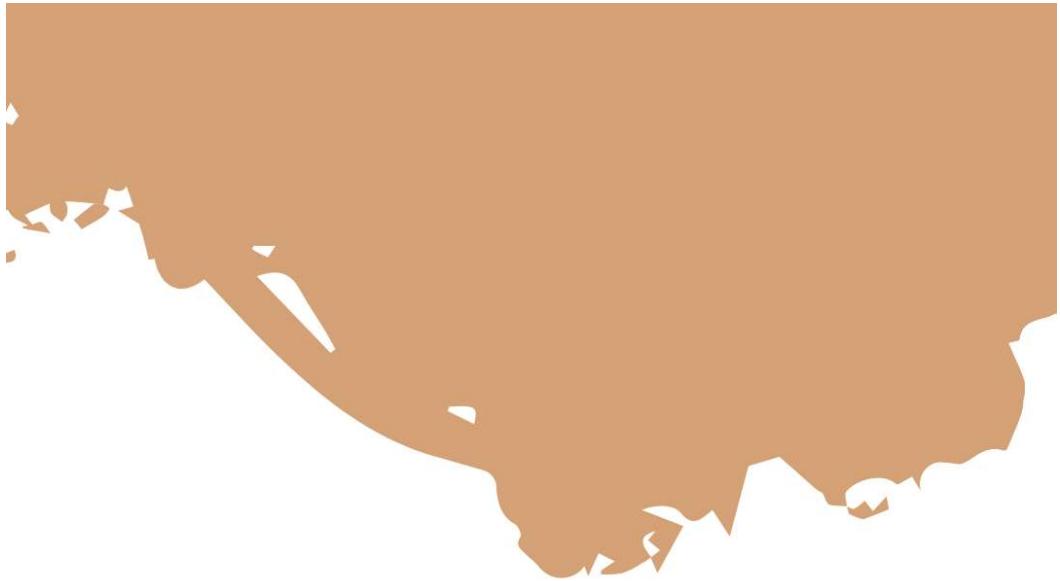
## Post Photo on Social Media





share image





#### A(ir)Awareness by AreYouWorking

Air pollution awareness via Photo sharing Users can take or upload photos and customize whether to add PM2.5, AQI, heat map or other air pollution data to the photos before saving or sharing.

"งานซึ่งในปัจจุบันมีมากกว่าที่คิด จึงขอ焉จะเพิ่มความตระหนักรู้ของผู้คนวันที่ใกล้ตัว โดยเฉพาะบน Social Media"

#### [AreYouWorking.github.io](https://AreYouWorking.github.io)



Sprint 2 Video: [https://drive.google.com/drive/folders/1D-Tz6q8hcspRzcFxp0Xp5qKmBhdvzxNS?usp=share\\_link](https://drive.google.com/drive/folders/1D-Tz6q8hcspRzcFxp0Xp5qKmBhdvzxNS?usp=share_link)

Air Pollution Awareness via Photo Sharing (AP)  
A6 - Sprint #2 (Presentation slides and Video)