

How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: “**Capstone_Stage1**”
3. Replace the text in green

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: marcin-sielski

hazyair

Description

hazyair is a handy application that enables to monitor air quality standards e.g. PM2.5, PM10 and alerts the user if these standards are exceeded. The data used in the applications will come from 3rd party services that owns their air quality monitoring stations such as:

1. <http://powietrze.gios.gov.pl/pjp/content/api> (Polish Government Air Quality Monitoring Service) - at minimum - no access restrictions,
2. <https://airly.eu/pl/api/> (Polish Public Air Quality Monitoring Service) - assuming access token will be granted,
3. ...

Intended User

The application is suitable for everyone who takes care about their health and would like to know when to plan their outdoor activities.

Features

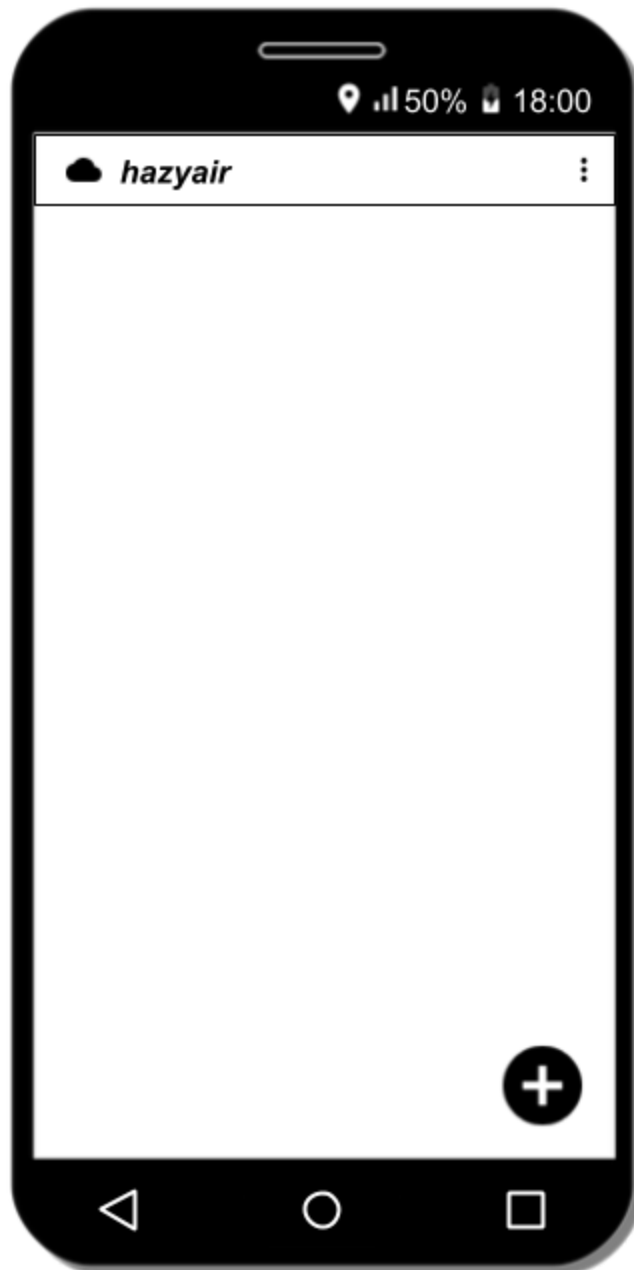
List on the main features of *hazyair* application:

- Presents air quality parameters in the main application (including charts with historical data) and in the widget,
- Integrates data from various sources,
- Incorporates forms to submit information to the authorities about illegal in-house fireplaces,
- Notifies the user if air quality standards are exceeded.

User Interface Mocks

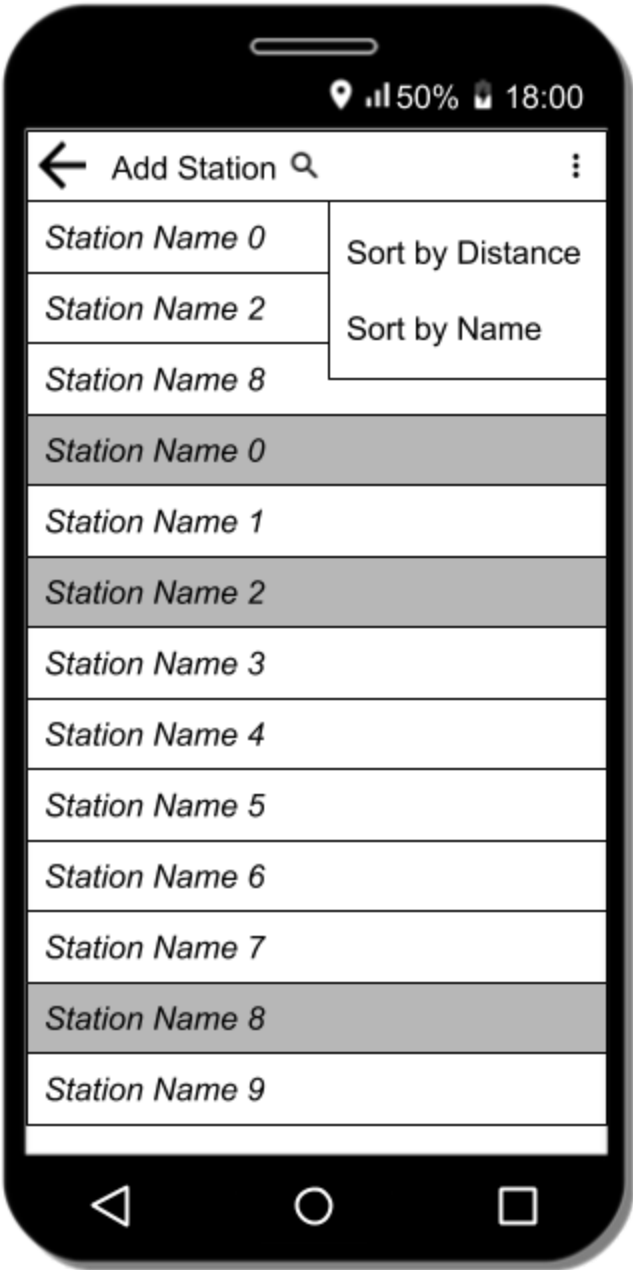
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Open Application



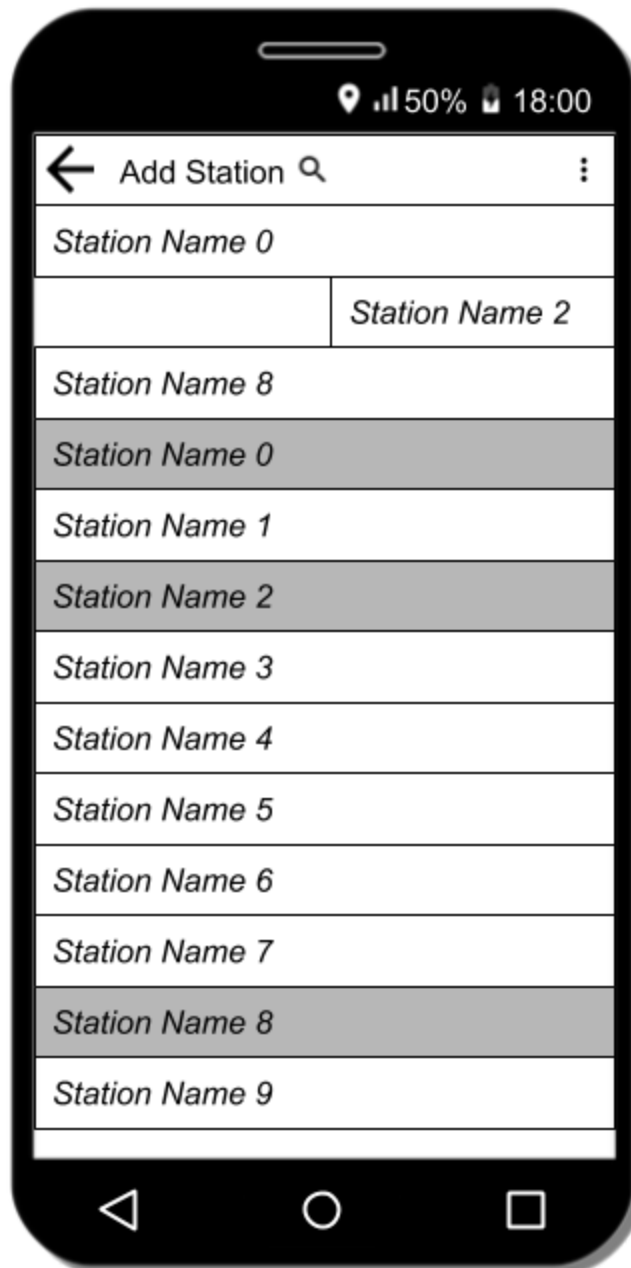
Picture above shows application that is opened for the first time and no air quality monitoring stations are yet configured. Activity enables to add air quality monitoring station with Add Floating Action Button or configure the application.

Add Station



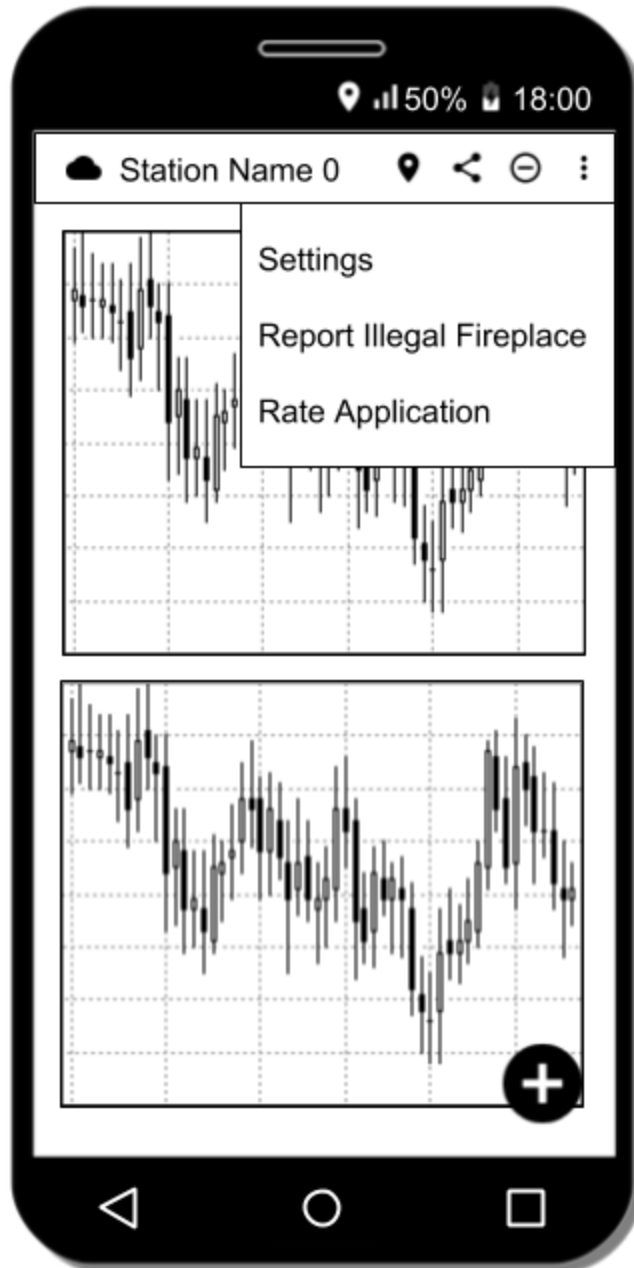
Selecting Add Floating Action Bar opens the list of available air quality monitoring stations. The stations are sorted by distance from current location or by name if current location is not available. This activity enables select the stations that are supposed to be added to the main activity. Selected stations are added to the most top list and are highlighted in different color on the bottom list.

Remove Station



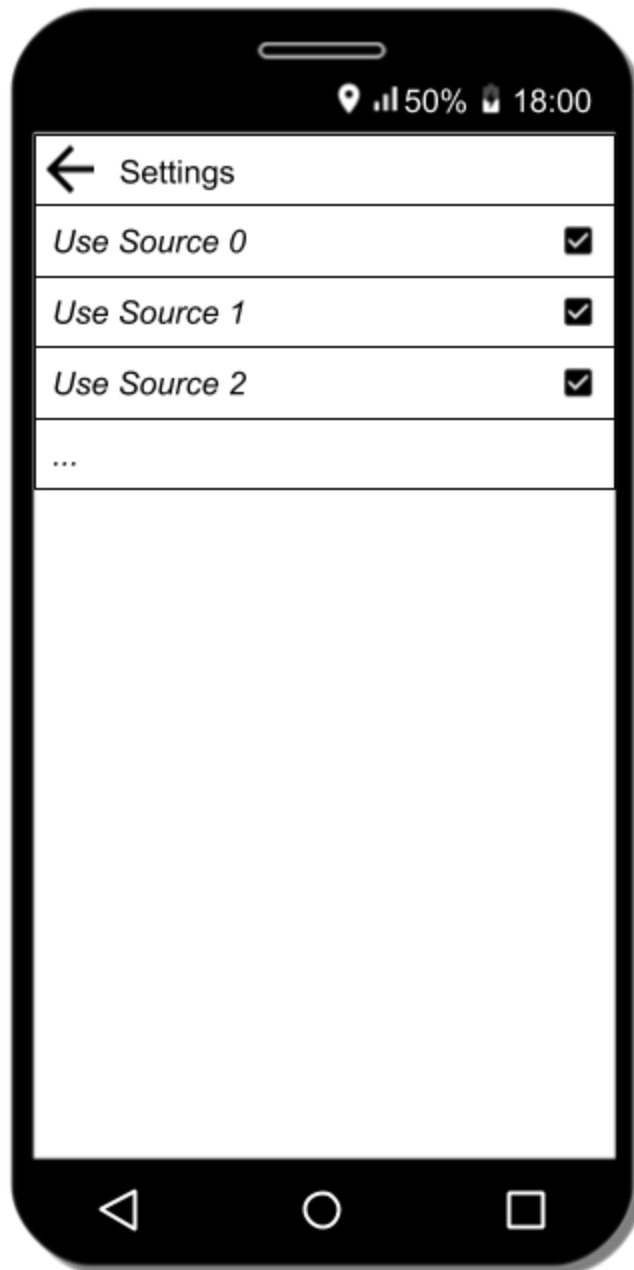
The air quality monitoring stations can be removed from the main activity by swipe as shown in picture above or by deselecting it in the bottom list.

Show Default Or Closest Station



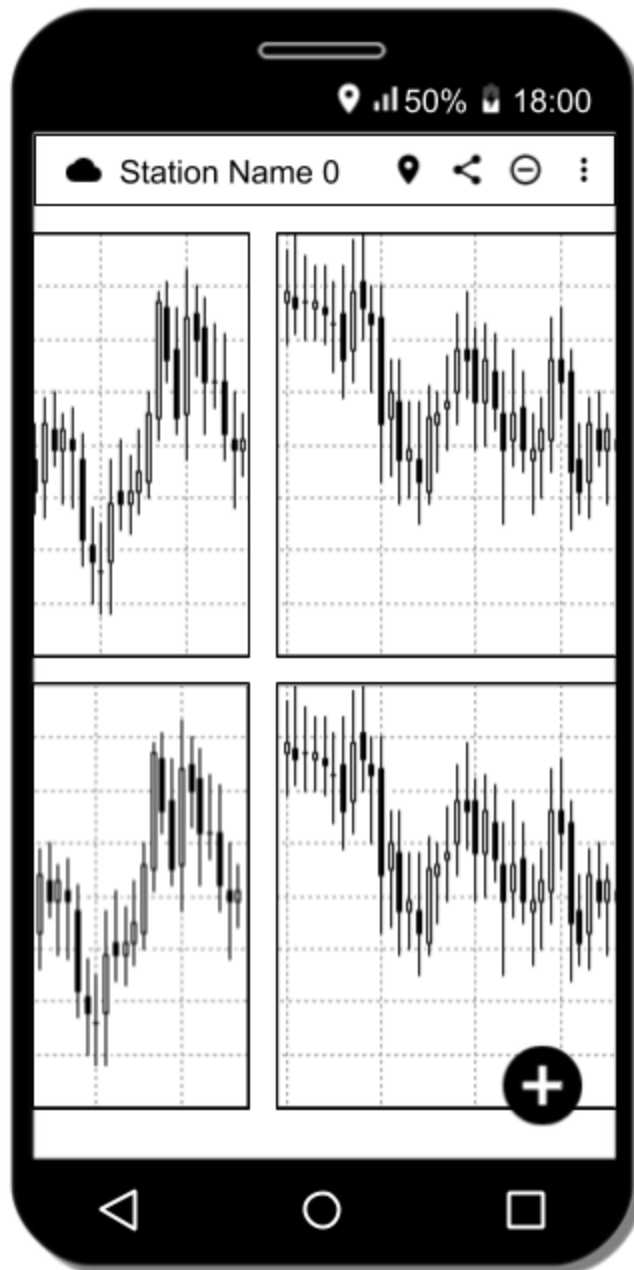
When the stations are configured its details are presented on the main activity in the form of charts and/or textual information depending on station capabilities. The stations are sorted by distance from the current location or by name if location is not available.

Open Settings



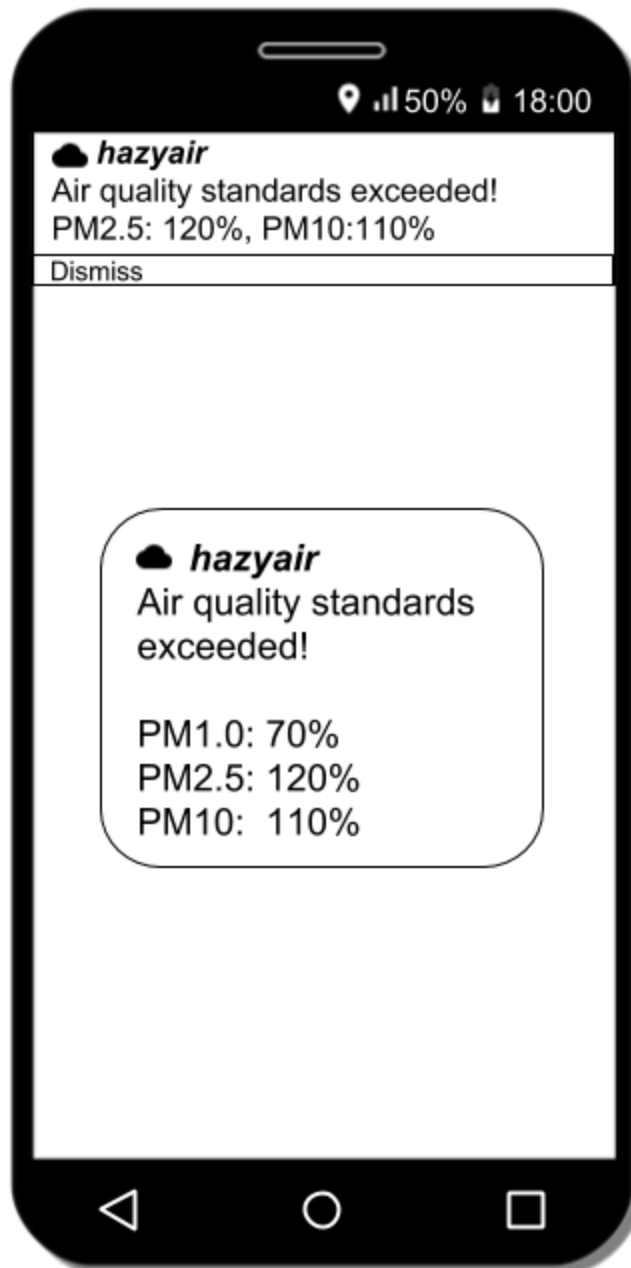
Options may include application settings as well as if various sources shall or shall not be used in the application.

Swipe Between Stations



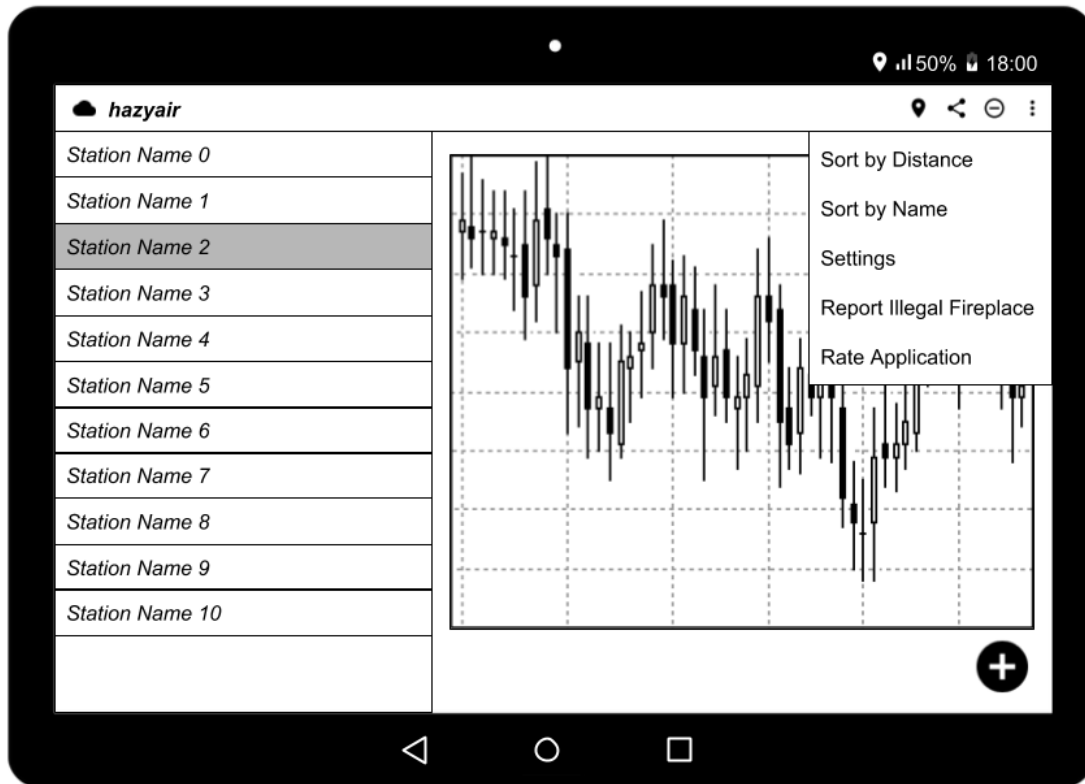
If multiple stations are configured to be displayed on the main activity swipe can be used browse between them. Location icon is used to show the location of the air quality monitoring station on the map. Share icon is used to share the results. Remove icon is used to remove the station from the main activity.

Show Widget And Notification



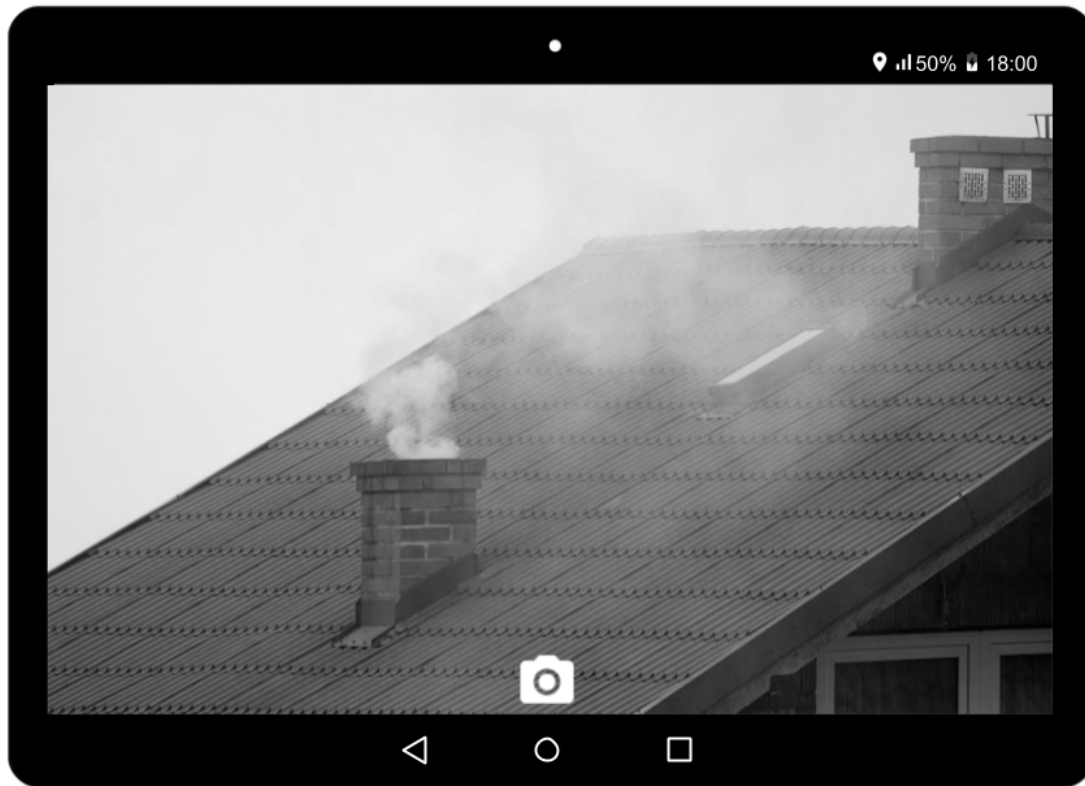
Picture above shows the notification and the companion widget.

Show Master/Detail View on Tablet



Picture above shows the master/detail view on tablet. Report Illegal Fireplace option enables to report illegal fireplace to the authorities.

Report Illegal Fireplace - Take Picture



Reporting Illegal Fireplace process starts from optionally taking picture of the suspicious haze (e.g. smells badly, has blue or dark grey tint).

Report Illegal Fireplace

Report Illegal Fireplace

Form 0

Form 1

Form 2

Form 3

Form 4

Form 5

Form 6

Form 7

Date:

Fireplace Type:

Picture:

Address:

Location:

Sort by Distance

Sort by Name

Settings

Rate Application

>

Number of different forms are available depending on the region. In most cases there are web forms (webview) but sometimes only email messages can be used.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

The measurement data will be stored in the device sql database and will be synchronized periodically with the configured services. Application will implement the content provider. Configuration parameters will be stored in preference persistence.

Describe any edge or corner cases in the UX.

Forms used in to report illegal fireplaces will utilize webviews to leverage forms available in the Internet.

Describe any libraries you'll be using and share your reasoning for including them.

Glide will be used to handle the loading and caching of images.

ButterKnife will be used to annotate views.

Gson will be used to handle json based communication with services.

ShapeOfView will be used to create user friendly interface.

Firebase JobDispatcher will be used to sync periodically the data with services.

Timber will be used for logging service.

MPAndroidChart will be used for charts.

Espresso and Junit will be used for unit tests.

...

Describe how you will implement Google Play Services or other external services.

Polish Government Air Quality Monitoring Service (<http://powietrze.gios.gov.pl/pjp/content/api>) will be used at minimum to collect the measurement data.

Firebase remote config will be used to selectively enable new features of the application.

Firebase crash will be used to report crashes of the application.

Firebase Analytics will be used to collect analytics data.

Places/Maps services will be used to display air quality monitoring station on the map.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

1. Install Android Studio with latest SDK and build tools,
2. Create empty project,
3. Configure libraries in the gradle files,
4. Setup main and test module.
5. Setup appropriate permissions in the Manifest file.
6. Create key store and generate key to be able to Generate Signed APK.

Task 2: Implement UI for Each Activity and Fragment

1. Build UI for MainActivity with use of ViewPager and Fragments for Phone and Tablet,

2. Build UI for AddStationsActivity,
3. Build UI for SettingsActivity,
4. Build UI for ReportIllegalFireplaceActivity,
5. Build UI for ShowStationOnMapActivity
6. Build UI for companion Widget.

Task 3: Implement Persistent Storage

1. Create content provider,
2. Create data contract,
3. Create database helper.

Task 4: Implement Adapters to be able populate the data to the UI

1. Create adapter for the MainActivity,
2. Create adapter for the AddStationsActivity.

Task 5: Setup project in Google Play Service

1. Go to the Google API Console,
2. Create a project,
3. Enable the Google Places API for Android and get an API key,
4. Setup API key in the Manifest.

Task 6: Setup project in Firebase

1. Create Firebase project for Android
2. Setup google-services.json in Android project

Task 7: Implement Services

1. Create Companion widget service,
2. Create data synchronization service that will query online services to obtain measurement data,
3. Integrate application with Crash, Remote Config, Analytics Firebase services,
4. Integrate application with Places/Maps Google Play Services.

Task 7: Implement Unit tests

Add as many tasks as you need to complete your app.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
 - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"