



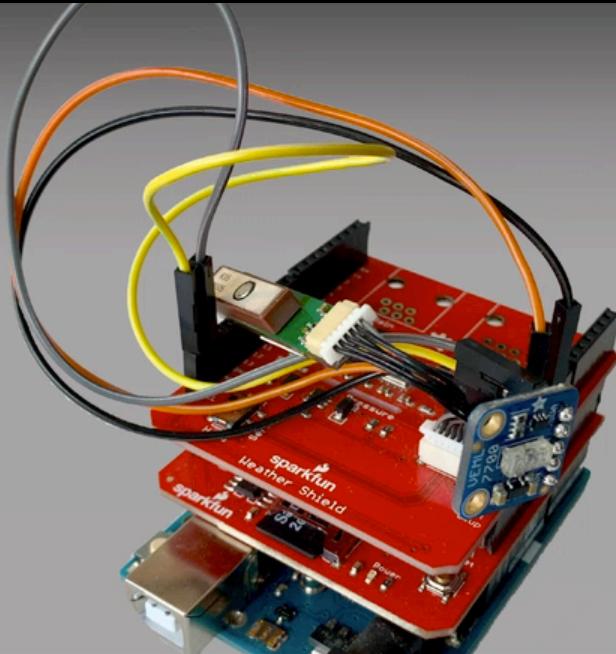
POLITECNICO
MILANO 1863

MOQA Monitoring Outdoor Quality of Air

Giudici Mathyas

Design and Implementation of Mobile Applications

Luciano Baresi



**How to get and store data from
an Arduino board?**

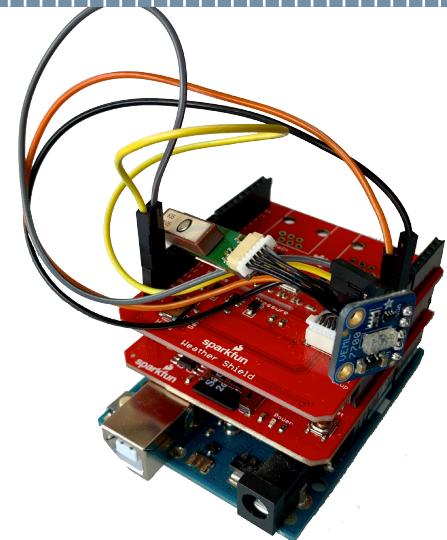
Agenda



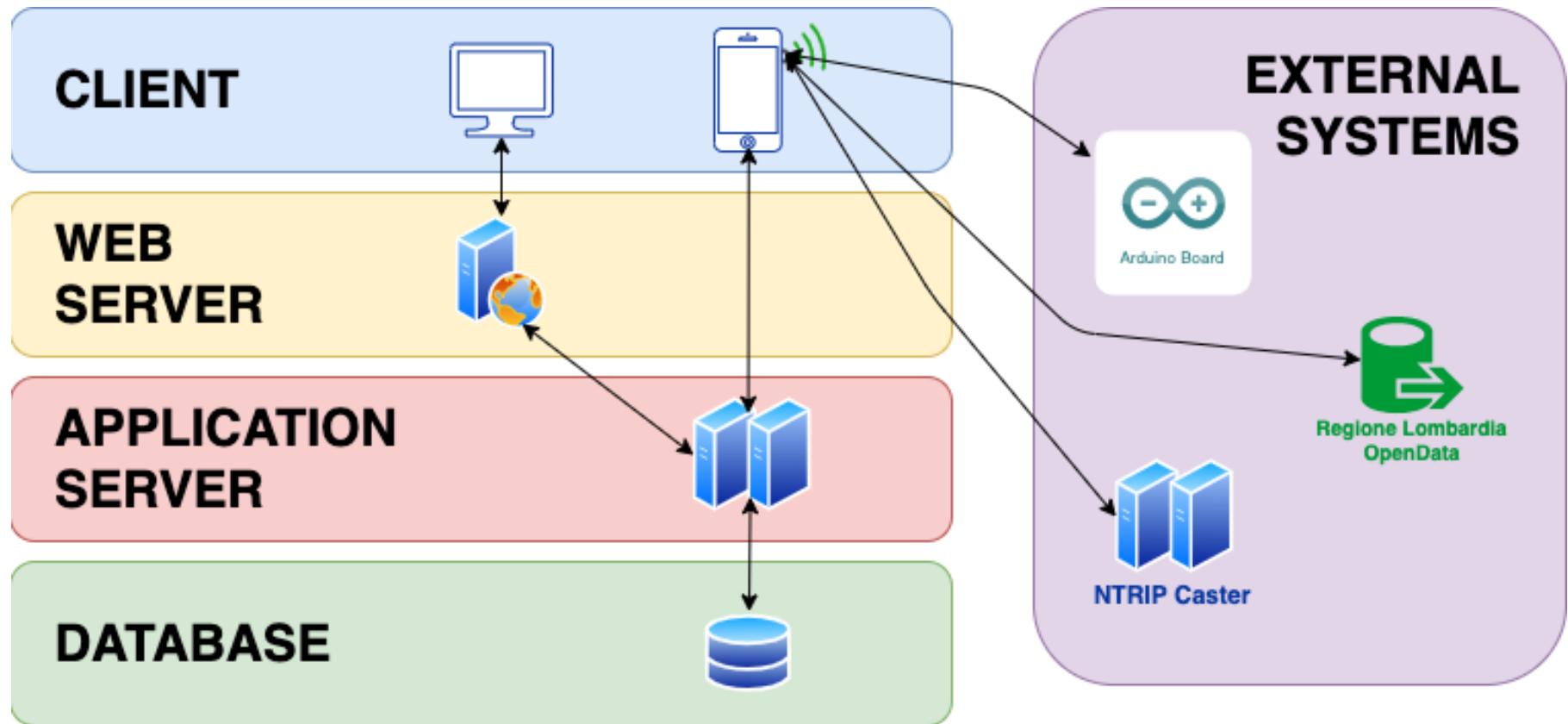
- What is MOQA
- Architecture
- Sequence diagram
- User Interface
- Testing
- Future works
- References

What is MOQA

- Get weather and air quality data from the **Arduino** board
- Push data on a remote server
- Visualize data on a **map**
- Visualize data on a **chart**
- Compare Arduino data with ARPA data



Architecture



Frameworks

Client



Server



HEROKU



KNEX.JS



Frameworks

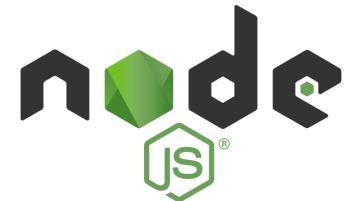
Client



Server



HEROKU

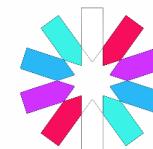


Swagger

Supported by SMARTBEAR

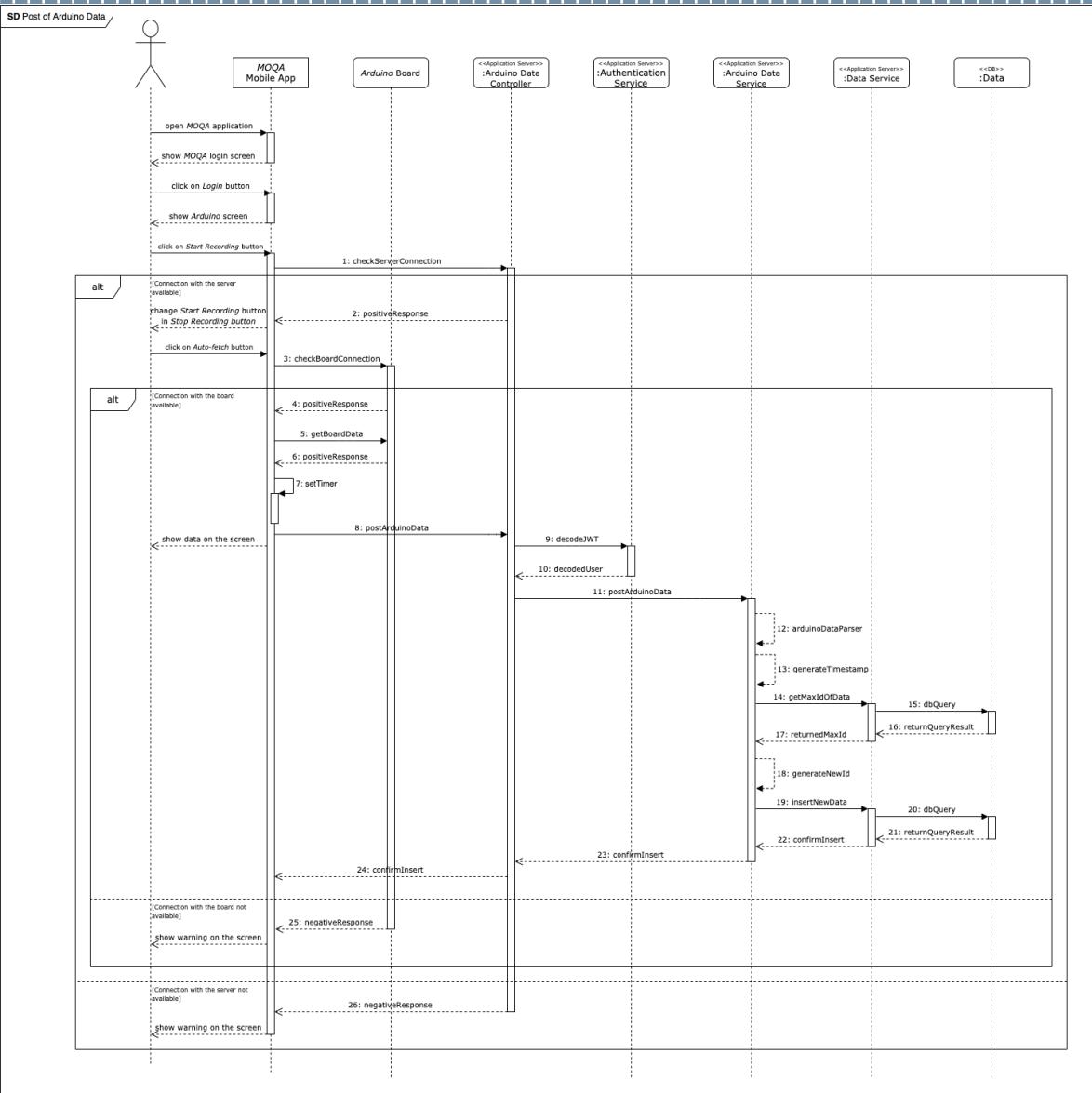


KNEX.JS



J W U T

Sequence diagram



Human Interface Guidelines



User Interface

1:00

Temperature 26.5 °C
Relative Humidity 29 %
Pressure 102700 Pa
Altitude 122 m
TVOCs 600 ppb
eCO2 3999 ppm
PM0.5 0 µm
PM1 1 µm
PM2.5 2 µm
PM4 4 µm
PM10 8 µm

Stop recording

Fetch data

Auto-fetch data

Visualize live data

Arduino Maps Chart Settings

Arduino Data

12:58

Maps

Filter C

Arduino Maps Chart Settings

1:06

Chart

29.90
19.93
9.97
0.00

15/4/2020 00:00 15/4/2020 16:50

Arduino Data ARPA Data

Temperature's deviation

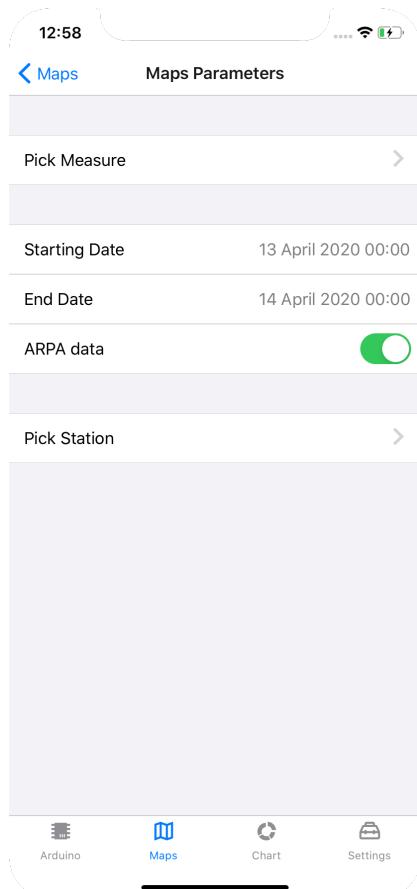
Quartiles	Arduino	ARPA
1st Quartile	0 °C	11.8 °C
2nd Quartile	20.4 °C	11.8 °C
3rd Quartile	24.8 °C	12.3 °C

Filter

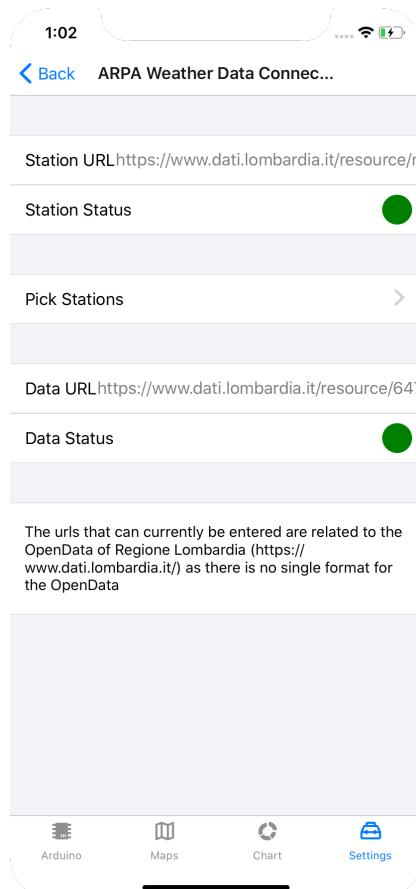
Arduino Maps Chart Settings

Chart

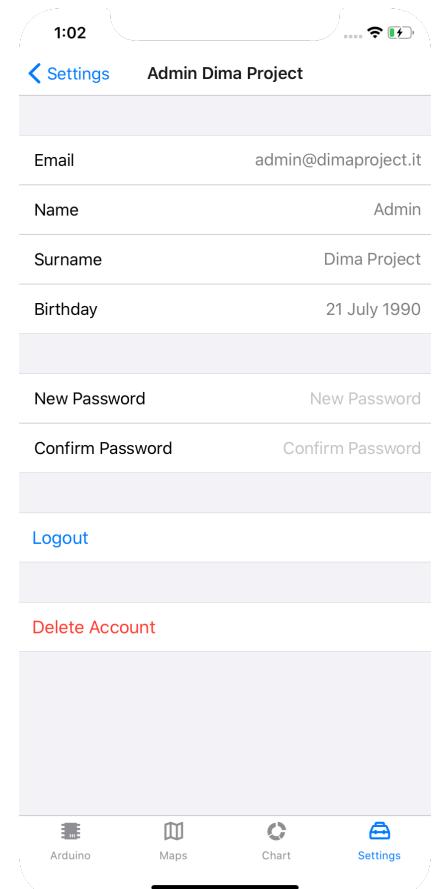
User Interface



Filter



Settings



User

Testing

Test Case	<i>Map Tracking</i>
Goal	Visualize live data on the map.
Input	The user logs-in in the application, enables on Auto-fetch data, enables Visualization of live data, goes to Map screen.
Expected outcome	At every new data from the board the Map screen refreshes.
Actual outcome	CORRECT: After the login in the application, enabling the Auto-fetch data and Visualize live data toggles: the application starts to automatically fetch data; going on the Map screen, every time a new data arrives the map is updated with a new circle.
	Expected outcome
Test Case	<i>Chart Tracking</i>
Goal	Visualize live data on the chart.
Input	The user logs-in in the application, enables on Auto-fetch data, enables Visualization of live data, goes to Chart screen.
Expected outcome	At every new data from the board the Chart screen refreshes.
Actual outcome	CORRECT: After the login in the application, enabling the Auto-fetch data and Visualize live data toggles: the application starts to automatically fetch data; going on the Map screen, every time a new data arrives the chart screen is updated.
Test Case	<i>Data Sampling</i>
Goal	Sample and send to the server Arduino data.
Input	The user logs-in in the application, clicks on Start Recording, then clicks on Fetch data or Auto-fetch data.
Expected outcome	Data are sampled and sent to the server.
Actual outcome	CORRECT: After the login in the application, a click on Start recording and Auto-fetch data: the application starts to automatically fetch data and send the to the server. If there are are problems in the connection with the server or with the board an alert is thrown.

Future works

- Web-interface to provide the features now available only with the mobile application
- In the Chart Screen visualize data coming from different stations (ARPA) or correlate different measures available

References

- Code repository
<https://github.com/MathyasGiudici/polimi-dima-moqa>
- Documentation repository
<https://github.com/MathyasGiudici/polimi-dima-moqa-documentation>
- Utils repository (Server and board code)
<https://github.com/MathyasGiudici/polimi-dima-utils>
in collaboration with @Antonino96 and @michelepilia

Thank you!
Questions?