COSMOS

Cultivate Resilient Smart Objects for Sustainable City Applications

The IoT Challenge @NTUA

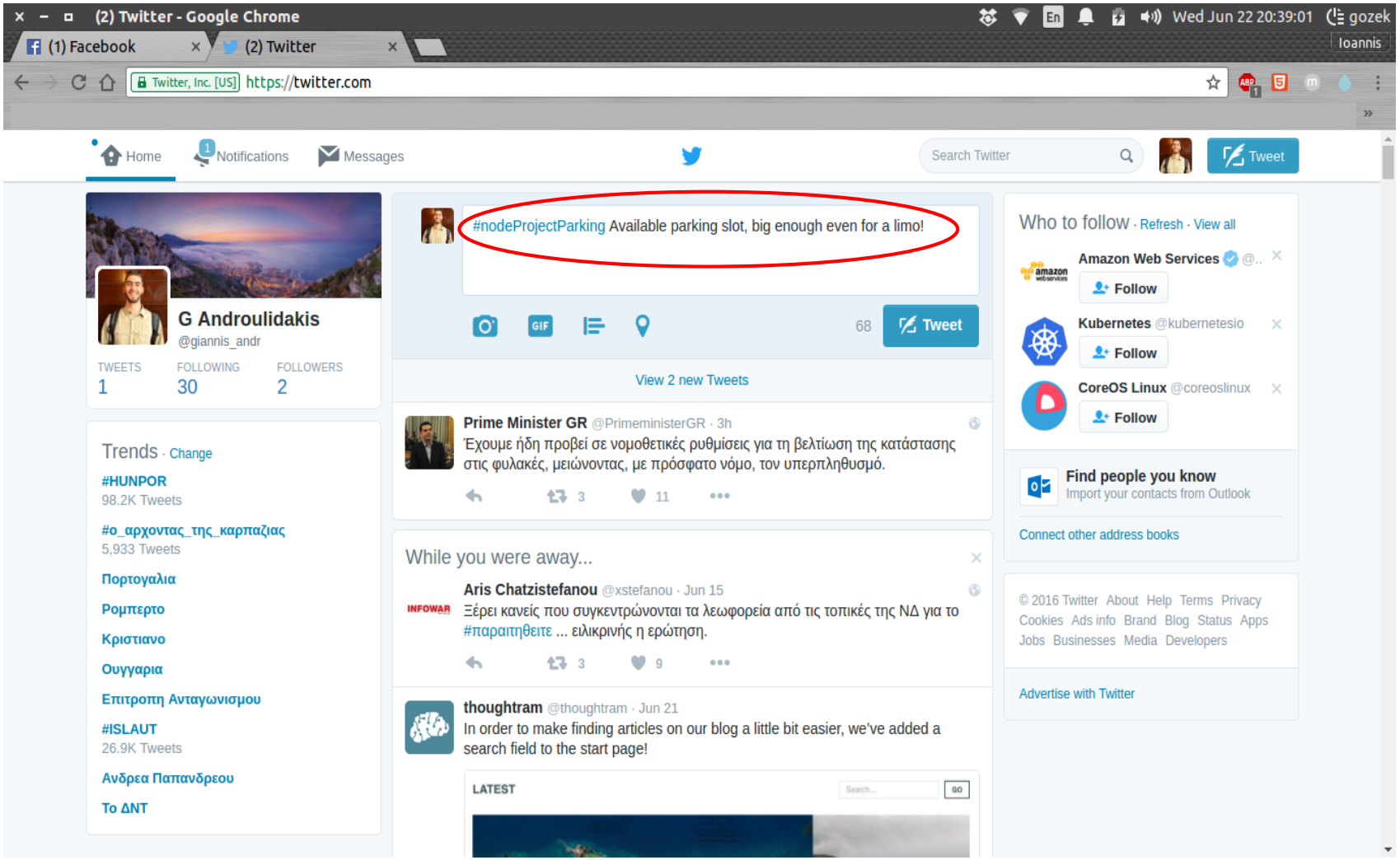
June 24th 2016

John Androulidakis, George Theodorakis

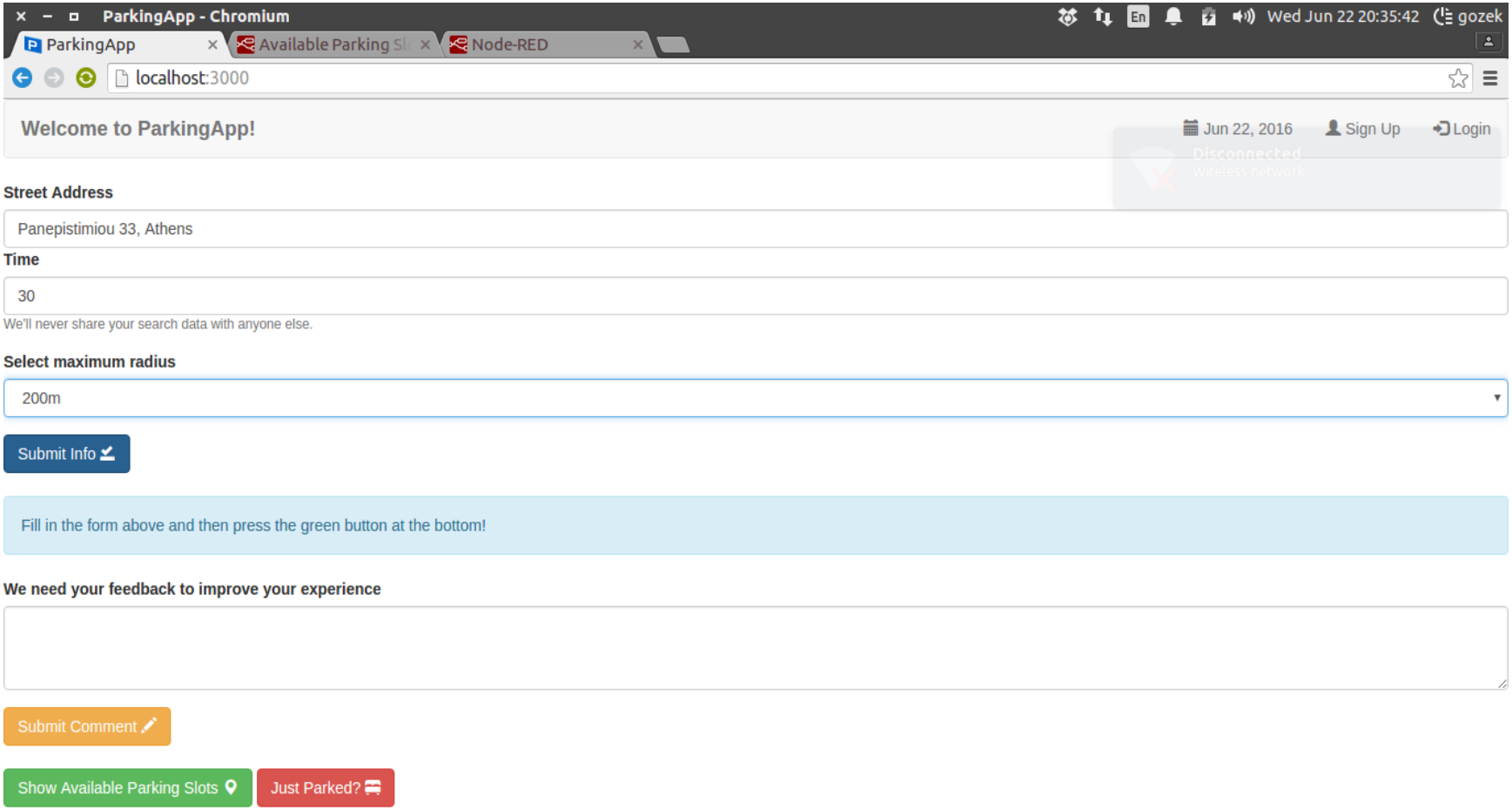
The Scenario (1/4)

- We chose scenario B. We created a parking application. End-users can search for empty parking slots in their current location with a user friendly application. There are two type of users:
  - **End-Users:** They can post empty parking slots from their current position using twitter **(#nodeProjectParking blah blah**) with a comment of the slot (e.g. slot for small car/ many positions for free all day). When they want to search for an empty parking slot, they can use the parking application on their phone(or computer) and get the nearest available parking slots with the comments about them.
  - **Administrator:** They manage manually the application with Node-RED and receive comments from the end-users (in an collection named "comments") about the application. Those comments are processed with the **LeanBigDataSentiment** in order to categorize them and improve the application in the future.

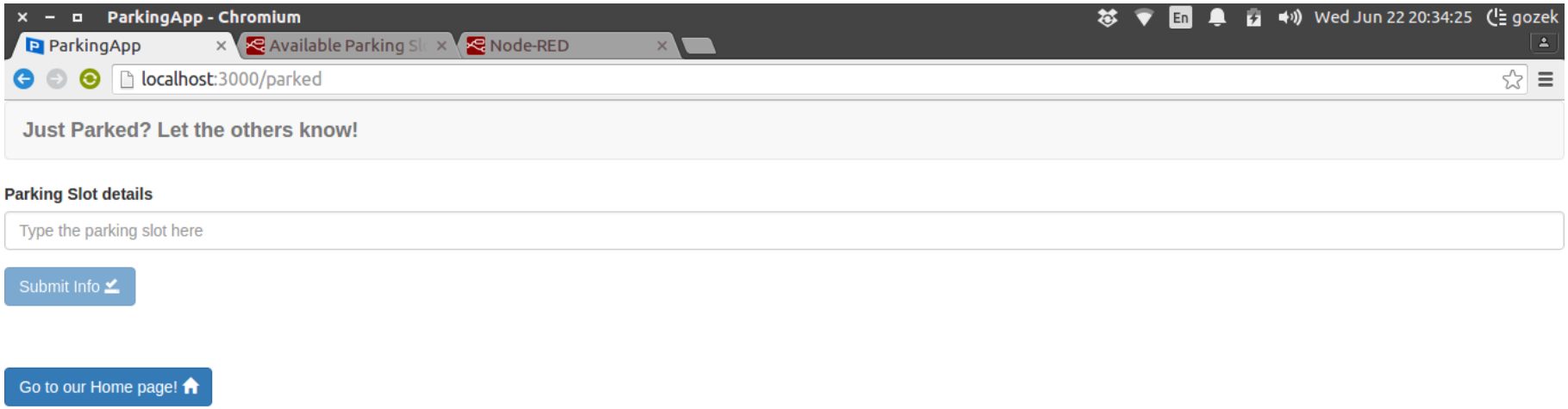
The Scenario (2/4)



The Scenario (3/4)



The Scenario (4/4)



The Procedure followed (1/2)

- We searched over the internet about “hot” issues that smart cities face. Then we thought about using the combination of **social networks** and **mobile applications** in order to create a user friendly and easy to use application. We used :
  - LeanBigData **Sentiment Analysis** for comments classification for further improvements.
  - **Node RED** for creating the server side of our application.
  - **Twitter API** for Node-RED (node-red-node-twitter) for the communication with Twitter.
  - **Google API** for Node-RED (node-red-node-google) for using various Google services(e.g. GeoCode).
  - **MongoDb API** (node-red-node-mongodb). We chose it for the json format it uses and the fast response times in BigData that makes it great for real-time IoT applications.
  - **Angular 2.0 MVC** framework for building a mobile application

The Procedure followed (2/2)

- Simple Use Case Scenario:
  - Some users tweet a comment about an empty parking slot from their current position, which is located automatically by our system.
  - A driver gives the address near which he/she wants to park in the application with these three parameters:
    - address,
    - radius of search and
    - the last slot filtered by time (e.g. “empty slot in the last hour” for more up-to-date results)
  - The user selects the slot he/she parked at, which then gets deleted.
  - The user can comment about the application.

The final Results

- We run many simulations and tests for the application. The main features described above have been implemented and run correctly. Further **improvements** have to be implemented:
  - The users could insert empty parking slots with google plus, facebook and the main mobile applications (not only twitter).
  - GPS instructions to the selected parking slot could be provided.
  - The time option is not yet available for the searches.
  - The visualization of the empty parking slots result is minimal.
  - The user should delete a slot that he/she parked with radio button option and not by writing it's description.
  - We could categorize the comments by their sentiment analysis and extract valuable information about the users of our application.
  - The application is written in Angular 2.0 and is running as a web application. In the future it can be packed as a mobile app.
  - After contacting local municipalities, we could install sensors on the streets in order to detect empty parking slots automatically.

Thank you!

www.iot-cosmos.eu

leanbigdata.eu

coherentpaas.eu

The research leading to these results has received funding from the EC Seventh Framework Programme FP7/2007-2011 under Grant Agreement n° 609043