**ASInt – MEEC**
**2017/2016**
**Project**

In this project students should develop a web application used to manage dissemination of messages to users working in particular physical locations.
In an organization such as IST some rooms are available for public use by members of the community. In case of emergency some notifications should be forwarded through digital means to users in specific rooms.
The system will be operated by two different class of users: admins and regular users. Admins can observe current/historic room occupancy and generate messages to be sent to users occupying specific rooms. Regulars users can check-in/check-out in rooms and receive messages.

# 1 Web application

The system will be implemented as a web application, accessible using a browser. A suitable REST API should be implemented to guarantee that the system could be accessed from remote clients (web, mobile or desktop applications).

## 1.1 Administrator

The administrator can see who is checked-in at that instance in each room and can also see a history of all the check-ins/check-outs since the installation of the system.

The administrator can also send a text message to the regular users that are at a specific room.

## 1.2 Regular users

Regular users should be able to browse rooms available at IST and check-in to one of those rooms when entering it. Regular users can also check-out when leaving the room.

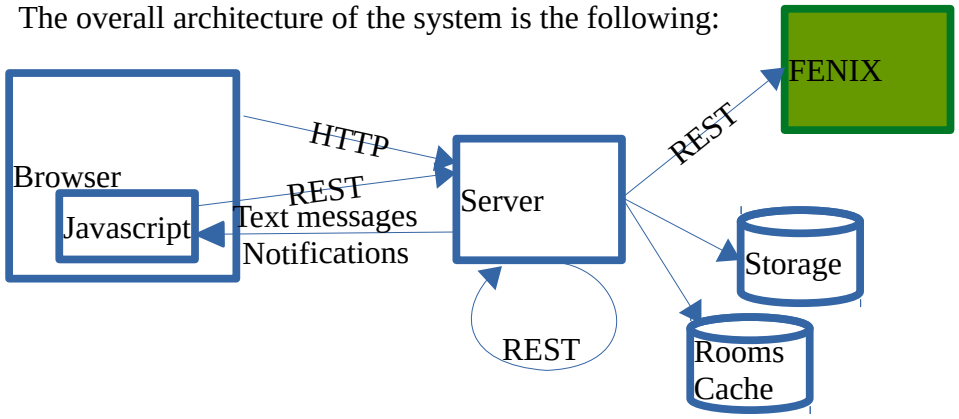
A regular user can only check-out from the last room he has checked-in. The student can only be in one room at a time: checking-in without first checking-out will implicitly perform a check-out.

A regular user can see who is at that moment in the same room.

Students should login in the system using the FENIX Authentication (istID and password).

# 2 Architecture

The overall architecture of the system is the following:



# 3 Login

The login of the regular users should be done using the FENIX API:

- http://fenixedu.org/dev/tutorials/use-fenixedu-api-in-your-application

The administrator should do a local login to the system (username: **admin** / password: **123**).

# 4 Room search

When checking-in, regular users should fill a form that will trigger a search for a room with name similar to the text written.

FENIX provides a REST API to search and browse all the rooms in the various *campi* (Campus do Tagus, Campus da Alameda, e Campus Tecnológico e Nuclear). The FENIX API should be used to provide the system with a list of rooms.

The administrator should only see the rooms with regular users inside.

To speed responses, the system should cache all room searches previously done.

# 5 Implementation

Students can use any technology and programming language (python, php, java, node.js) to develop the system. The selection the programming language will not affect the final grade.

The server should be deployed in the Cloud (for instance in the Google App Engine infrastructure)

The storage should be in the Cloud.

# 6 Data persistence

In order to simplify implementation and future deployment in the cloud, students should define one class for each of the data type necessary (Users, rooms, check-ins/check-outs) and store them in different dictionaries.

In the final version data should be stored in a Cloud database.

# 7 Evaluation

Student should implement the maximum described functionality and architecture requirements. Alternative implementations will influence the final grade:

- Use of JavaScript on the browser.
- Implementation of a suitable REST API on the server.
- Interaction with the FENIX system

- Deployment on the cloud
- Optimizations

## 7.1 Report

The students should write a report describing:

- The system architecture
- Implemented REST API
- Used technologies/libraries
- User interface of the system
- Cloud deployment
- FENIX integration
- Data manipulation

Students should describe in detail one of the external systems/libraries used. In this description students should highlight the benefits from using it, how it is internally works, and how the systems could have been implemented without it.

The report should explicitly described the points for evaluation (Section 8).

## 7.2 Submission date

Although students should submit the projects before Christmas, this date can be scheduled to after the Christmas holidays. The final date for the submission of the work should be defined and agreed with the teacher.

# 8 Further information

**Jquery**

- https://jquery.com/

**Google app engine**

- https://cloud.google.com/appengine/ (browse the page for free limits)
  - https://cloud.google.com/appengine/quotas
- https://cloud.google.com/pubsub/
- Python
  - https://cloud.google.com/appengine/docs/python/
  - https://cloud.google.com/appengine/docs/python/download
- Cloud datastore

    - https://cloud.google.com/appengine/docs/python/datastore/
- Node.js
    - https://cloud.google.com/nodejs/
- php
    - https://cloud.google.com/appengine/docs/php/

**FENIX API**

- http://fenixedu.org/dev/tutorials/use-fenixedu-api-in-your-application/
- http://fenixedu.org/dev/api/
-