# Università degli studi “Roma Tre”


**Facoltà di Ingegneria**

**Corso di laurea in Ingegneria Informatica**

**A.A. 2019-2020**

# TECH4CH

*Daniele Santino 496958*
*Federico Gaglio 499609*
*Mihail Alexandru Ropotica 499275*

## System requirements and goals

The project goal is to replicate a study that was published at PUC, which studies the behavior of visitors in the Reuben Hecht museum, analyzing their movements and their time spent near multiple points of interest. In order to replicate this study, we needed to implement a web service, that given the museum floor plan and a set of visitors logs, shows some statistics about the museum and can *play back* a visit of a selected group or individual.

## Software used

- MYSQL database, to store data about visitors
- Spring Boot, to create a web service following the MVC design path
- CanvasJS, to render statistics
- Amazon AWS Educate, to deploy the web service using Amazon RDS to instantiate our database, Elastic Beanstalk to load the WAR file, Amazon EC2 to run out instance

## System structure

We created a MYSQL database importing the .csv samples file and storing information about visitors. Then, we built our Model:

- Poi, this class stores information about the points of interest in the museum
- Visitor, this class stores information about the visitors information
- Room, this class stores information about the rooms in the museum
- Museum, this class stores all the previous information

We implemented Generator classes which retrieve data from the database and create our model classes. After this, we coded some auxiliary methods to create some statistics about the visitors: best holding power pois, best attraction power pois, general statistics about pois, museum visitors per hour, museum visitors per room per hour. We decided to generate charts for the statistics, so we used CanavasJS library to code the controller and to render the JSP pages.

# Home

TECH4CH

HOME

MUSEUM CHARTS




# Best attraction power POIs

TECH4CH

HOME

MUSEUM CHARTS

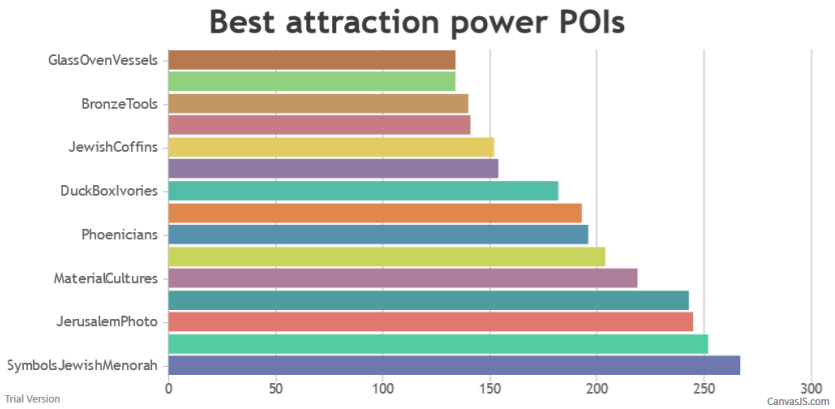
Best attraction power POIs

Best holding power POIs

General statistics POIs

Visitors in the museum

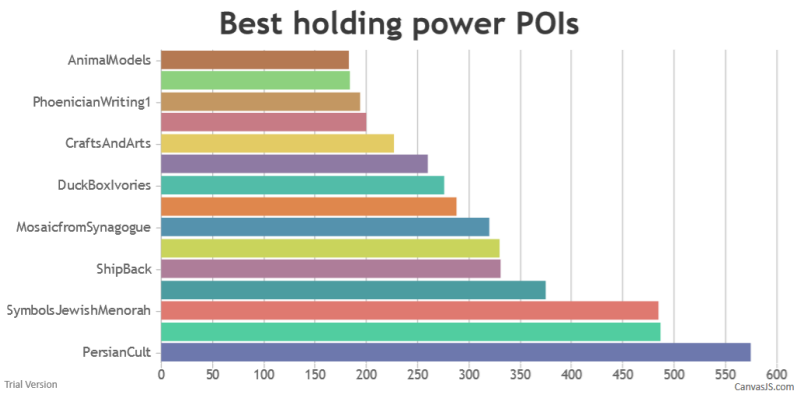
Visitors in rooms



# Best holding power POIs

TECH4CH

- HOME
- MUSEUM CHARTS
  - Best attraction power POIs
  - Best holding power POIs
  - General statistics POIs
  - Visitors in the museum
  - Visitors in rooms

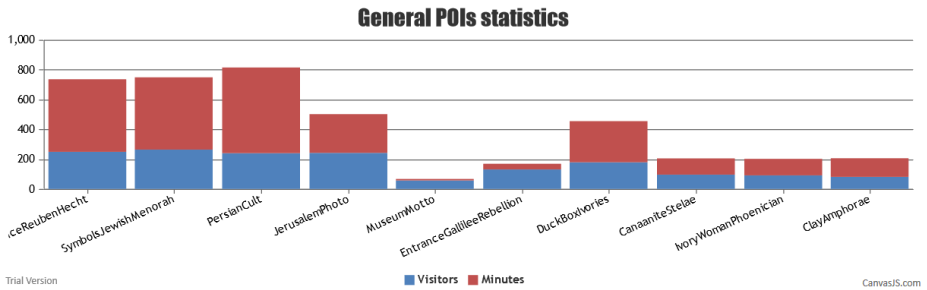


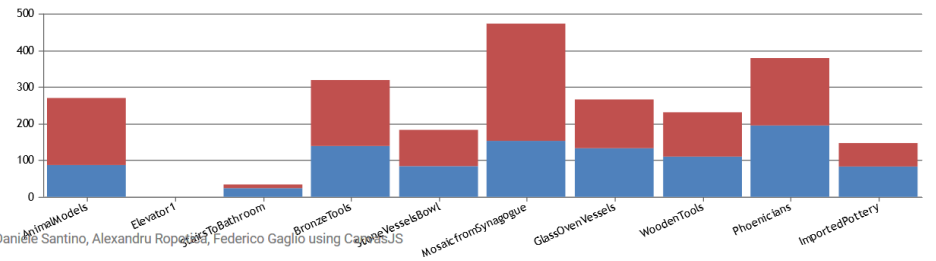



# General Statistics

TECH4CH

- HOME
- MUSEUM CHARTS
  - Best attraction power POIs
  - Best holding power POIs
  - General statistics POIs
  - Visitors in the museum
  - Visitors in rooms



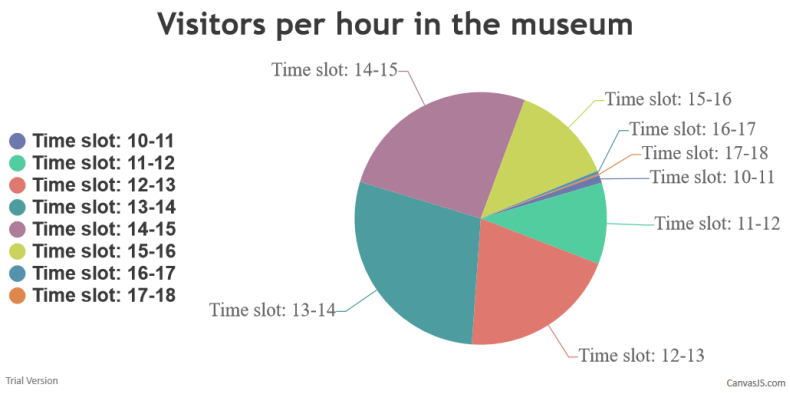
# Visitor per hour in the museum

TECH4CH

HOME

MUSEUM CHARTS

- Best attraction power POIs
- Best holding power POIs
- General statistics POIs
- Visitors in the museum
- Visitors in rooms





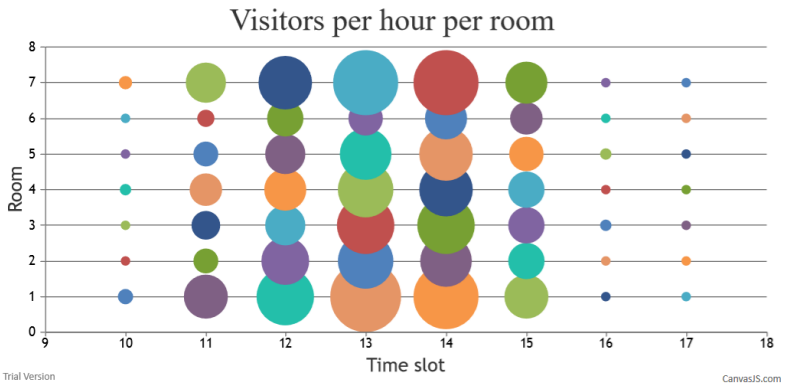

# Visitor per room per hour in the museum

TECH4CH

HOME

MUSEUM CHARTS

- Best attraction power POIs
- Best holding power POIs
- General statistics POIs
- Visitors in the museum
- Visitors in rooms



## Work distribution

- Database, Daniele and Federico designed and implemented the database structure and Alexandru helped formatting the csv files with AWK scripts
- Dynamic Web Project, Alexandru implemented the structure of the project, Daniele and Federico reviewed the code and made some tweaks
- Charts, Federico and Daniele used CanavsJS to render the statistics, coding the Controller classes and JSP pages
- Amazon AWS, Daniele deployed the project using Amazon AWS

## Challenges/problems encountered

The main problem was to implement the Revisit a visit feature and we tried to do it with JavaScript.

### *JavaScript problem*

Given the museum floor map, our idea was to get all the POIs position from the image, create a graph introducing auxiliary nodes between POI nodes to avoid collisions with the walls and apply a pathfinding algorithm, that given a POI list of a visitor it will show on the map its movement. The first problem was getting all the positions, so we wrote a script that prints the position of the cursor in an image on mouse click, but this is highly dependent on the size of the image because the coordinates of every position changes. The second problem was to animate a point on an image, we tried to code it, but even reading some StackOverflow discussions, it didn't work.

## What did we get from this assignment

Most of the patterns we used to implement this service was already studied with our Prof. Paolo Merialdo in his course Sistemi Informativi sul Web. The new things that we learned is coding in JavaScript, making animation and charts.

## Useful links

GitHub Repository
Google Drive
Web Service