

# UNIVERSITÀ DEGLI STUDI “ROMA TRE”



**Facoltà di Ingegneria  
Corso di laurea in Ingegneria Informatica  
A.A. 2019-2020**

## **TECH4CH PROJECT PROPOSED BY THE PROFESSOR TSVI KUFLIK FOR THE ADVANCED TOPICS IN COMPUTER SCIENCE COURSE**

**Students:**

**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
- Hipster4J, to create the graph and a search algorithm
- D3.js, to animate all transition between points
- Amazon AWS Educate, to deploy the web service

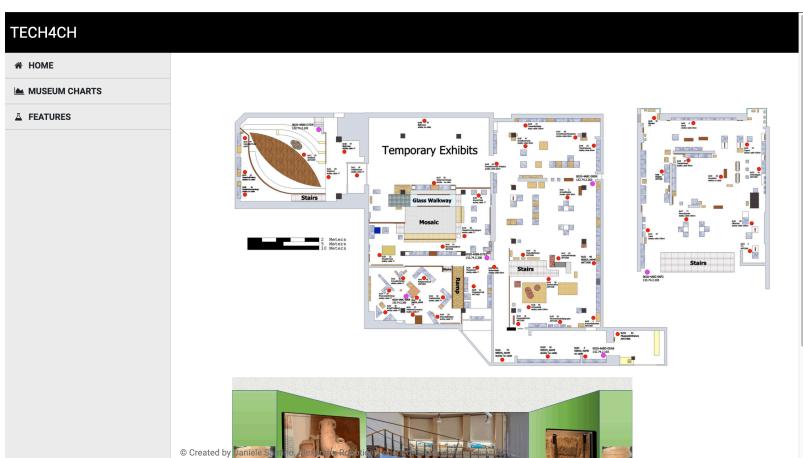
## System structure

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



## 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](#)

[FEATURES](#)

### Best attraction power POIs

A horizontal bar chart titled "Best attraction power POIs". The x-axis is labeled "Visitors" and ranges from 0 to 300 with major ticks every 50 units. The y-axis lists eight POI names: GlassOvenVessels, BronzeTools, JewishCoffins, DuckBoxIvories, Phoenicians, MaterialCultures, JerusalemPhoto, and SymbolsJewishMenorah. Each name is associated with a horizontal bar of a different color. The bars are ordered by visitor count, with SymbolsJewishMenorah having the highest count (around 280) and GlassOvenVessels the lowest (around 130). The chart is created using CanvasJS.

POI	Visitors
GlassOvenVessels	~130
BronzeTools	~140
JewishCoffins	~150
DuckBoxIvories	~180
Phoenicians	~200
MaterialCultures	~220
JerusalemPhoto	~240
SymbolsJewishMenorah	~280

Trial Version      [CanvasJS.com](#)

© Created by Daniele Santino, Alexandru Ropotica, Federico Gaglio using CanvasJS

## 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](#)

[FEATURES](#)

### Best holding power POIs

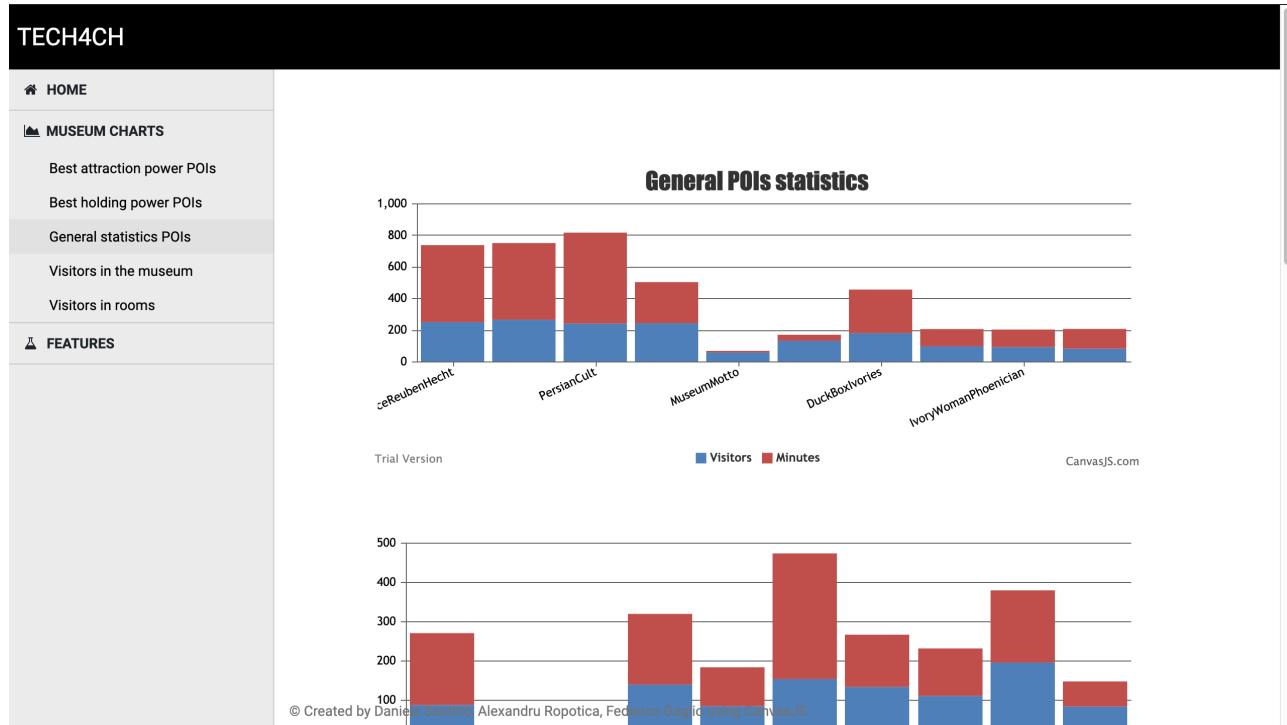
A horizontal bar chart titled "Best holding power POIs". The x-axis is labeled "Minutes" and ranges from 0 to 600 with major ticks every 100 units. The y-axis lists nine POI names: LeadCoffinMosaic, PhoenicianWriting1, CraftsAndArts, DuckBoxIvories, MosaicfromSynagogue, ShipBack, SymbolsJewishMenorah, PersianCult, and another entry for SymbolsJewishMenorah. The bars are ordered by time, with PersianCult having the longest duration (around 550 minutes) and LeadCoffinMosaic the shortest (around 180 minutes). A tooltip for the "MaterialCultures" bar indicates a value of 330. The chart is created using CanvasJS.

POI	Minutes
LeadCoffinMosaic	~180
PhoenicianWriting1	~190
CraftsAndArts	~210
DuckBoxIvories	~240
MosaicfromSynagogue	~270
ShipBack	~320
SymbolsJewishMenorah	~380
PersianCult	~550
SymbolsJewishMenorah	~550

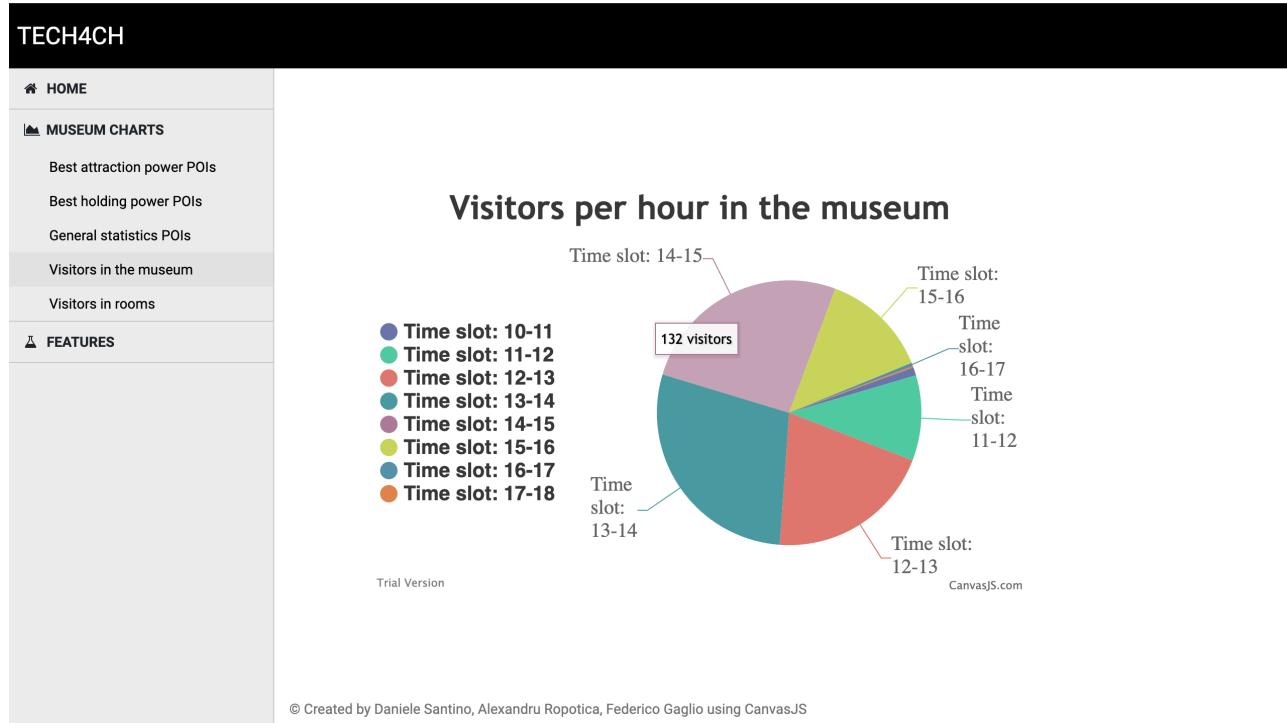
Trial Version      [CanvasJS.com](#)

© Created by Daniele Santino, Alexandru Ropotica, Federico Gaglio using CanvasJS

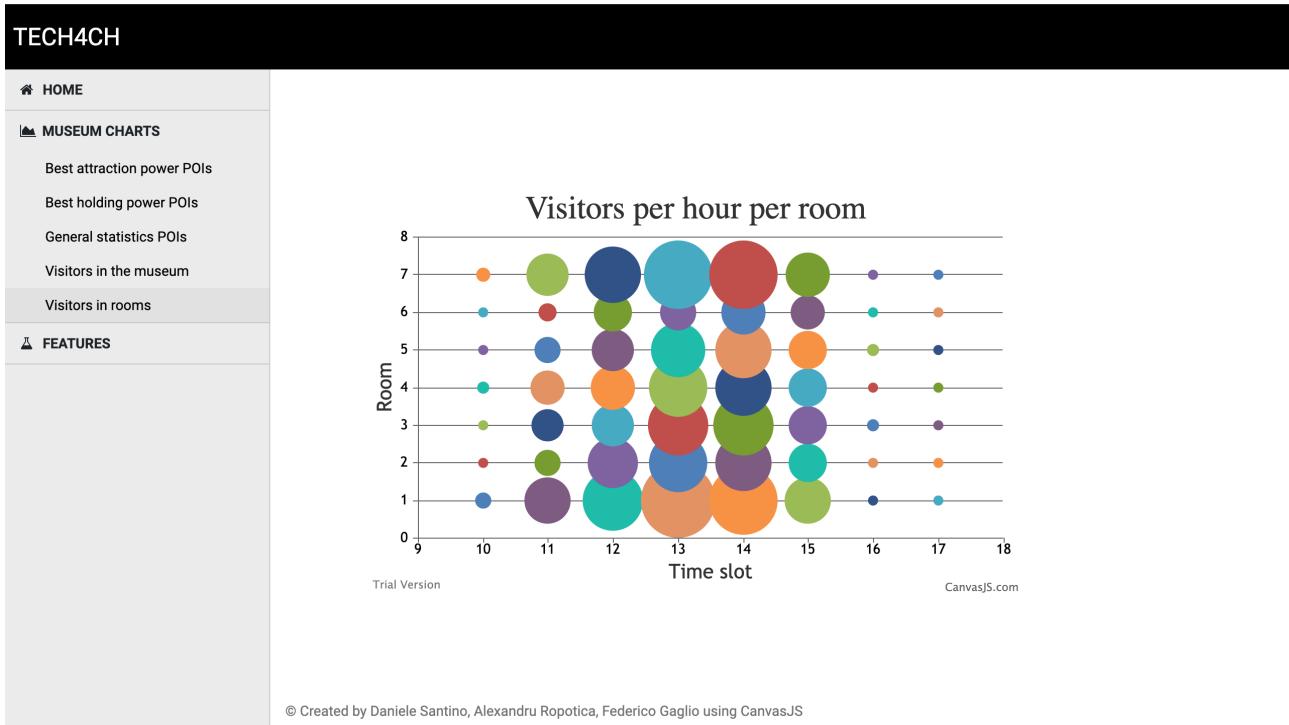
# General Statistics



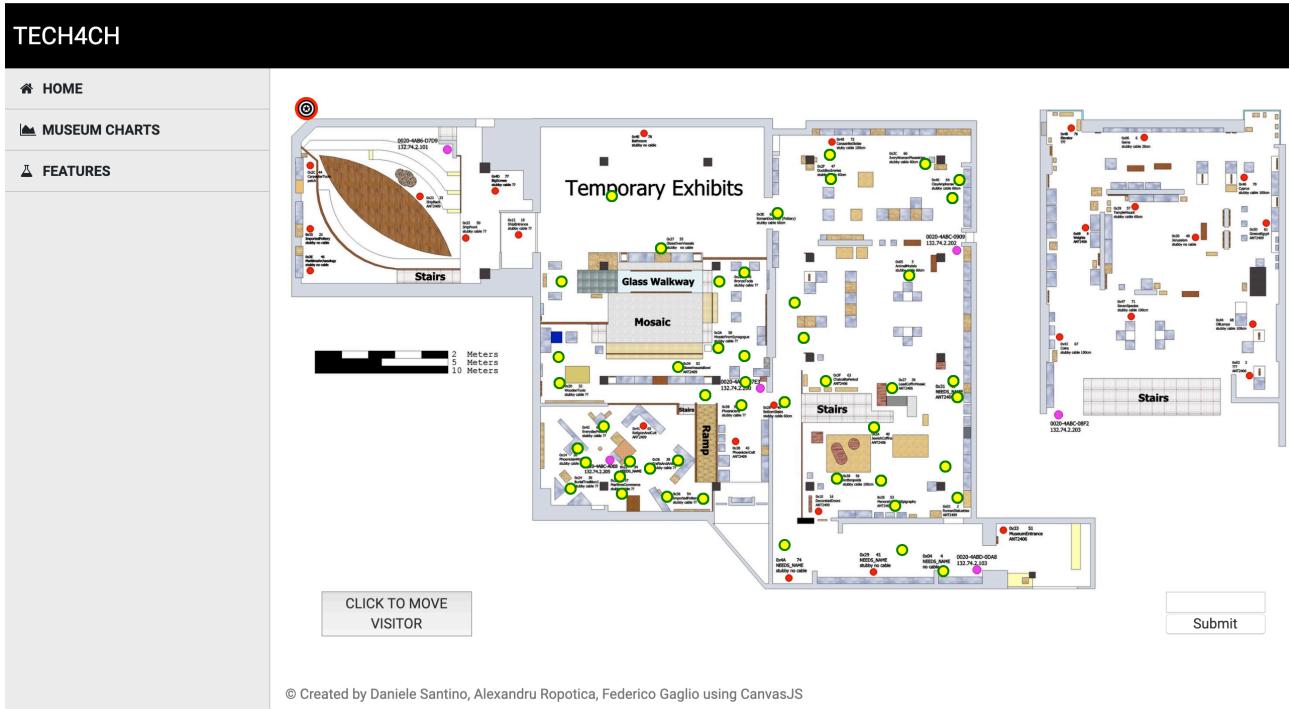
## Visitor per hour in the museum



# Visitor per room per hour in the museum



## Revisit a visit feature



## Visitor summary

TECH4CH

[HOME](#)

[MUSEUM CHARTS](#)

[FEATURES](#)

**GROUP 1**

Visitor 1 watched:

- Presentation about DuckBoxIvories for 0 minutes, ended by System. Average time: 7 minutes.
- Presentation about MaritimeCommerce for 0 minutes, ended by System. Average time: 2 minutes.
- Presentation about CraftsAndArts for 1 minutes, ended by System. Average time: 10 minutes.
- Presentation about BurialTradition2 for 0 minutes, ended by System. Average time: 4 minutes.
- Presentation about PhoenicianWriting1 for 0 minutes, ended by System. Average time: 7 minutes.

Visitor 2 watched:

- Presentation about DuckBoxIvories for 1 minutes, ended by System. Average time: 7 minutes.
- Presentation about Phoenicians for 1 minutes, ended by System. Average time: 4 minutes.
- Presentation about JewishCoffins for 1 minutes, ended by System. Average time: 10 minutes.
- Presentation about ShipFront for 2 minutes, ended by System. Average time: 13 minutes.

Visitor 3 watched:

- Presentation about ShipBack for 6 minutes, ended by System. Average time: 17 minutes.
- Presentation about DuckBoxIvories for 0 minutes, ended by User. Average time: 7 minutes.
- Presentation about CraftsAndArts for 0 minutes, ended by System. Average time: 10 minutes.

This group viewed 4 presentations. Average presentation viewed: 3

© Created by Daniele Santino, Alexandru Ropotica, Federico Gaglio using CanvasJS

## 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 CanvasJS to render the statistics, coding the Controller classes and JSP pages
- Graph, Federico got all the POIs position with a script, Alexandru created the graph and applied Dijkstra algorithm
- Points animation, Federico and Daniele made the animation between points
- Amazon AWS, Daniele deployed the project

## Challenges/problems encountered

The main problem was to implement the Revisit a visit feature.

Given the museum floor map, we got all the POIs position from the image using a JavaScript script, create a graph introducing auxiliary nodes between POI nodes to avoid collisions with the walls and apply a pathfinding algorithm using Hipster4J library, that given a POI list of a visitor it will show on the map its movement.

## **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](#)

[GoogleDrive](#)

[Web Service](#)