# Location-based services with Mongodb
# Initial project plan

Cardiff University School of Computer Science and Informatics
CM3203 One Semester Individual Project: 40 credits
Author: Nojus Lenciauskas
Project supervisor: Dr Alia I Abdelmoty
Moderator: Dr Natasha Edwards

# Table of contents

# Project Description

Nowadays, millions of people are consuming and sharing information on the Internet. Old technologies for data storage and retrieval are no longer a viable option. Relational databases used to be the prefered technology by many, but with increasing data volumes new solution had to be developed. NoSQL databases were introduced to the market in order to tackle this kind of problem. Mongodb database is one of the most popular NoSQL databases. It provides powerful capabilities without sacrificing flexibility, therefore used by many businesses worldwide.

Being familiar with the surroundings in your environment was always a crucial aspect in human existence. However, increasing urbanisation, constant changes and many different locational options had an impact on people's ability to develop such environmental sense. Technologies can aid in preventing such case, in order to be informed about their neighbourhood more in depth, or acknowledge them what is happening in places they are not familiar with.

Perfect example for information and geospatial data sharing, would be internet. Mongodb allows querying on geospatial data. Simple api provides powerful capabilities in storing and retrieving data at high volumes with great performance.

The main goal of the project is to create an interface which allows users to generate Mongodb query for geospatial data, use that query to get data from the database and then visually represent the information. Main challenge is to make the interface feel intuitive to the user. In order to make interface easy to use, familiar industry standards should be used together with minimalistic design and modern web development techniques. Geospatial data can be hard to interpret directly, because coordinate system is just angles between earth points, hence using the map of the world would the best way in converting geospatial data to any human knowledge accessible interface.

The interface will allow users to express all possible queries related to geospatial data in Mongodb database. It is going to demonstrate how Mongodb database could be used to help customers to find and discover locations.

Overall aim of the project is to create an application which makes it easy to demonstrate and visualise Mongodb geospatial querying capabilities for people who are not familiar with the database.

# Project Aims and Objectives

**Aims**

The aim of this project is to create a system that demonstrates the functionality and capabilities of the Mongodb database system using the Yelp open data set as an example. The Yelp data set contains geospatially referenced documents and will allow the demonstration of the storage and querying capabilities of the Mongodb system. The system will let users to generate different types of spatial queries over the data set and display the results on a map as appropriate. The usability and scalability of the system will be considered, to ensure the interface is intuitive to use and the system is able to handle large data sets.

**Objectives**

1. Examine and identify system requirements:
    a. Identify functional, non-functional requirements, system scope and boundaries, legal, social, ethical and professional issues, use cases.
    b. Asses benefits and risks.
    c. Analyse system's users and similar projects.
    d. Produce test cases.
2. Design the system:
    a. Download and analyse Yelp open data set.
    b. Produce UML diagram.
    c. Design interface with flexible querying capabilities and familiar data presentation, using well established industry standards.
3. Research technologies required:
    a. Research how to host Mongodb database on the cloud.
    b. Study Mongodb indexing.
    c. Study modern web developing techniques such as typescript, webpack, vuejs, graphql.
4. Implement the system:
    a. Prepare for project development by installing needed applications and dependencies, version control, etc..
    b. Upload appropriately formatted data from Yelp open data set to the database.
    c. Create user interface prototype.
    d. Evaluate prototype.
    e. Implement method to help user generate geospatial queries using a map.
    f. Implement interface to presented data returned from queries on the map and in the list.

      g. Implement graphql server to query database and validate user requests.

      h. Implement all possible filtering and geospatial querying options on Yelp open data set.

      i. Index the database.

      j. Test user interface usability.

5. Upload the system on university's cloud:
   a. Get cluster on university's cloud.
   b. Implement system on the cluster.
   c. Make system accessible through Internet.

6. Evaluate the system:
   a. Evaluate application.
   b. Do application testing.
   c. Do user testing.

**Ethics**

The project does not require ethical approval. Only publicly available data is going to be used, therefore no ethical issues need to be considered. User-based evaluations are going to be conducted anonymously.

**Constraints**

System is created for demonstration purposes, therefore it would not allow new data to be uploaded, because it creates security and implementation problems that are beyond this project scope.

# Work Plan

**Milestones**
- Submit initial report.
- Develop interface prototype and have meeting with supervisor to evaluate the prototype.
- Develop working system and have meeting with supervisor to evaluate user interface.
- Upload system on the university's cloud and have meeting with supervisor to evaluate the system.
- Do user testing.
- Submit final report.
- Submit application source code.

**Deliverables**
- Initial report.
- Final report.
- Application source code.
- Make the system accessible on the Internet.
- UML diagram.
- Interface prototype drawings.
- Interface evaluation using Nielsen's Heuristic evaluation.
- Test Cases.
- Description of personas.
- Project benefit and risk assessment.
- Usability evaluation.
- Functional and non-functional requirements.
- System scope and boundaries.
- Legal, social, ethical and professional issues.
- Use cases.
- User testing results.

**Important dates**
- 04/02/2019: Submit initial report for "Location-based services with Mongodb".
- 10/05/2019: Submit final report for "Location-based services with Mongodb".

**The plan**

**Week 1**(28/01 - 01/02):
- Write and submit initial plan.
- Draw interface prototype.
- Create system prototype.
- Implement graphql server
- Prepare database.
    - Get familiar with data.
    - Upload data to database.

**Week 2**(04/02 - 08/02):
- Identify and describe personas for application.
- Identify and describe use cases.
- Produce test cases for application.
- Produce UML diagram.
- Produce project benefit and risk assessment.
- Produce functional and non-functional requirements.
- Identify and describe system scope and boundaries.
- Identify and describe Legal, social, ethical and professional issues.

**Week 3**(11/02 - 15/02):
- Implement graphql api for data querying.
- Improve user interface.
- Connect user interface to the server.
- Evaluate user interface with supervisor.

**Week 4**(18/02 - 22/02):
- Fix user interface.
- Adapt server logic to fit user interface needs.
- Do database indexing.
- Develop method to pass query plan to interface.
- Write test for the system.

**Week 5**(25/02 - 01/03):
- Improve system.
- Fix bugs.

**Week 6**(04/03 - 08/03):
- Improve system.
- Fix bugs.
- Interface evaluation using Nielsen's Heuristic evaluation.
- Do user-based testing.
- Evaluate project with supervisor.

**Week 7**(11/03 - 15/03):
- Evaluate interface using test cases.
- Improve system.

- Fix bugs.

**Week 8**(18/03 - 22/03):

- Improve system.
- Fix bugs.
- Usability evaluation.
- Evaluate project with supervisor.

**Week 9**(25/03 - 29/03):

- Upload system on the university's cloud.
- Do testing with users.
- Report writing.

**Week 10**(01/04 - 05/04):

- Report writing.

**Week 11(**08/04 - 12/04):

- Report writing.

**Easter recess**(13/04 - 05/05)

**Week 12**(06/06 - 10/05):

- Report writing.
- Submit final report.
- Submit application's source code.

## Gantt Chart

| TASKS | | WEEKS | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | TASK | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 0.1 | Write and submit initial plan | X | | | | | | | | | | | |
| 1.1 | Create system prototype | X | X | | | | | | | | | | |
| 1.2 | Prepare database | X | X | | | | | | | | | | |
| 2.1 | Identify and describe system requirements | | X | X | | | | | | | | | |
| 3.1 | Implement graphql api for data querying | | | X | | | | | | | | | |
| 3.2 | Connect user interface to the server | | | X | | | | | | | | | |
| 3.3 | Evaluate user interface with supervisor | | | X | | | | | | | | | |
| 3.4 | Adapt server logic to fit user interface needs | | | | X | | | | | | | | |
| 3.5 | Do database indexing | | | | X | | | | | | | | |
| 3.6 | Develop method to pass query plan to interface | | | | X | | | | | | | | |
| 3.7 | Write test for the system | | | | X | | | | | | | | |
| 4.1 | Improve system | | | | X | X | X | X | | | | | |
| 4.2 | Interface evaluation using Nielsen's Heuristic evaluation | | | | | | X | | | | | | |
| 4.3 | Do user-based testing | | | | | | X | | | | | | |
| 4.4 | Evaluate project with supervisor | | | | | | | X | | | | | |
| 4.5 | Evaluate interface using test cases | | | | | | | X | | | | | |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4.6 | Usability evaluation | | | | | | | | ▓ | | | | | |
| 4.7 | Evaluate project with supervisor | | | | | | | | ▓ | | | | | |
| 4.8 | Upload system on the university's cloud | | | | | | | | | ▓ | | | | |
| 4.9 | Do testing with users | | | | | | | | | ▓ | | | | |
| 5.1 | Report writing | | | | | | | | | ▓ | ▓ | ▓ | | ▓ |
| 6.1 | Submit final report | | | | | | | | | | | | | ▓ |
| 6.2 | Submit application's source code | | | | | | | | | | | | | ▓ |