

Location-based services with MongoDB

Initial Plan

Daniel Huntley: c1544004

Supervisor: Alia I Abdelmoty

Moderator: Chris B Jones

Module: CM3203

CM3203 One Semester Individual Project 40 Credits

Project Description

With the increasing need for availability, flexibility and performance in modern applications NoSQL databases have seen a massive uptake in recent years. One of the most popular NoSQL database programs is MongoDB which is a document-orientated database program providing users with a flexible and highly available storage solution.

In addition to this MongoDB allows the storage and retrieval of geospatial data which is a core component of many of the services businesses offer which includes maps, geo-tagging and the storing of locations, making it an advantageous tool for developers to adopt and use. However, being able to create and understand these queries requires an in-depth knowledge of the database program which can be overwhelming and confusing especially to new developers. Interacting with a MongoDB instance through the command line or using Compass¹ are both limited in their support for geospatial functions, making the generation of geospatial queries challenging especially for new developers.

In this project, I will be producing a solution which is aimed at solving this, by allowing new developers to run geospatial queries against their chosen datasets uploaded to MongoDB. A map interface will be used to generate the queries and display the results, making it easier for new developers to gain an understanding of the different geospatial query operators.

Additionally, the program will illustrate a breakdown of the query produced explaining each component and its purpose, further supporting the development of new developers in understanding and querying of geospatial data in MongoDB.

¹Compass is a visualisation tool used to visualise the structure of data in your MongoDB database

Project Aims and objectives

The following aims and objectives outline the deliverables to be achieved by the end of the project.

Aims

Design and develop a web-based application to allow users to run geospatial queries against their chosen datasets, uploaded to MongoDB. A map interface will be used to generate the queries and display the results in conjunction with outlining and explaining each component of the query.

Objectives

- **Study and research different tools and techniques** used to visualise, interact and understand geospatial data. Including research into,
 - Different languages and libraries best suited to develop the application
 - * Languages: NodeJs and Python.
 - * Libraries: D3 and Mapbox.
 - Existing papers and work relating to this topic. Including,
 - * geojson.io.
 - Datasets which can be used for testing, including the,
 - * Yelp open Dataset.
 - * MongoDB sample Geojson datasets.
 - * Ordnance Survey datasets inc. Points of interest and Open Roads
 - * GeoJson available through Data.gov
 - User-interface standards and best practice.
 - Techniques which can be used to visualise large amounts of Geospatial data.
- **Design and develop a solution for new developers to create and understand geospatial queries.**
 - Provide a web-based interface allowing users to connect and visualise there MongoDB geospatial data.
 - Allow users to interact with a map interface to generate the queries and display the results.
 - Explain each component of the query generated using the map, supporting the users understanding of geospatial queries.
- **Test and evaluate the usability of the system**
 - Test the application using different datasets sourced during research, e.g.the Yelp Open Dataset.
 - Evaluate the usability of the application, testing with a sample of real novice developers.

Preliminary set of requirements

These requirements are subject to change and will be refined through research and investigation into the project.

- The user must be able to connect there MongoDB instance using the applications user-interface.
- The user must be able to visualise there imported MongoDB Geospatial Data on the applications web-based interface.
- The user must be able to generate MongoDB Geospatial through interaction with the visualised Geospatial data. Including query selectors,
 - \$geoIntersects (Returns geometries that intersect other GeoJson objects)
 - \$geoWithin (Returns geometries entirely within a specified shape)
 - \$near (Return Geospatial Geometrics objects in proximity to a point)
 - \$nearSphere (Return Geospatial Geometrics objects from nearest to farthest using spherical geometry)
- And Geometry Specifiers,
 - \$maxDistance Specifies a maximum distance to limit the results.
 - \$minDistance Specifies a minimum distance to limit the results.(Docs.mongodb.com, 2019)[1]
- The user must be able to test the MongoDB query generated against the connected dataset, which includes returning and displaying a subset of 10 results.
- The user must be able to see a breakdown of the query generated, including outlining and explaining each component of the query.
- The application must be tested with real novice developers to assess the effectiveness of the project in supporting users generate geospatial queries and understand them.
- The application must be tested and evaluated with multiple MongoDB datasets of varying size.

Desirable Deliverables

- The user should be able to visualise large datasets (collections) of at least 50000 documents.
- The application should use cookies to store the database connection details.
- The application should be compatible with different browsers and versions.
- The application should adhere to good coding standards.
- The applications user-interface should adhere to usability standards including Shneidermans principles.

Time permitting Deliverables (Could)

- The system could provide a sample repository to be used to test the application.
- The user could instead upload a JSON file containing MongoDB Geospatial data in the applications web-based interface (over the connecting of a MongoDB instance).
- The application could include full end-to-end automated tests testing all paths of the application's functionality.

Work Plan

Below is have outlined a week by week plan, demonstrating my aims, objectives and deliverables over the next term. Also, a Gantt chart is included illustrating the project's schedule and high-level overview of the project's timeline.

Throughout the project, I intend on using the agile methodology where I will work iteratively on the design, coding and testing of individual tasks. Using this methodology allows me to prioritise valuable features first and supports change should an approach or idea not work in practice. Furthermore, it supports the ability to refine and test the implementation continually.

[2](Murugaiyan, 2012)

Week 1

Tasks:

- Research and investigate languages and libraries best suited for developing this application (Languages include: NodeJS and Python - Libraries include: D3 and Mapbox).
- Plan and create Initial Plan.
- Learn about the platform chosen.

Milestones:

- Final Version of initial Plan. (1)

Week 2

Tasks:

- Continued. Research and investigate languages and libraries best suited for developing this application.
- Construct an architecture design for the application and provide brief reasoning into the decisions made, including architectural structure, languages and libraries.
- Produce write up of report introduction.

Deliverables:

- Produce write up of report introduction.
- Produce architecture plan of the application inc. structure, languages and libraries.

Week 3

Tasks:

- Source suitable data to be used for testing the application, e.g. MongoDB geospatial data.

- Research best practices for user interface design including, Shneidermans principles.
- Construct basic UI workflow and design.
- Implement the backend architecture based on the architecture design.

Deliverables:

- Produce initial implementation of the backend of the application.
- Produce Initial design of the User Interface based on research and UI standards.

Week 4

Tasks:

- Import sourced test data into MongoDB test instance.
- Produce a brief summary of findings on UI standards.
- Produce a write up of report Background and approach.
- Begin implementation of UI design.

Deliverables:

- Connection of MongoDB test instance to the backend of the application.
- Write up of report background and approach.

Week 5

Tasks:

- Work on implementation of visualising geospatial data from the MongoDB instance.
- Write up of notes on report implementation.
- Further research into visualising geospatial data.

Deliverables:

- Visualisation of geospatial data from MongoDB test instance.

Weeks 6-7

Tasks:

- Add the ability for end-users to be able to generate MongoDB geospatial queries through interaction with the interface.
- Implement integration tests.

Deliverables:

- Be able to generate MongoDB geospatial queries in the user-interface.
- Write up of report implementation section.
- Generation of MongoDB geospatial queries in the user-interface.

Milestones:

- Test system different datasets. (2)

Week 8

Tasks:

- Add the ability for users to test the geospatial queries against the connected repository and return a subset of the results.
- Add the ability for users to see a breakdown of the generated query explaining individual components of the query produced.

Deliverables:

- Running of queries generated in the user interface.
- Visualise a breakdown explaining components of the query generated.

Week 9

Tasks:

- Add the ability for users to be able to filter visualised data.
- Complete extensive usability, functionality and consistency tests across the different browsers and version, including Chrome, Firefox and Edge.
- Add support of caching of connection details using cookies (time permitting).
- Look at adding the ability for users to be able to upload a JSON file containing MongoDB GeoSpatial data (time permitting).
- Add further tests on newly added components.

Deliverables:

- Finalise frontend development.

Milestones:

- Finalise and complete implementation of the project. (3)

Week 10

Tasks:

- Make any necessary minor tweaks or bug fixes in the application.
- Test the application with a sample of real novice developers and assess using their feedback whether the project has achieved its aim of allowing users to generate MongoDB geospatial queries and helped there understanding of geospatial queries.

Deliverables:

- Conduct and write up of real user tests assessing feedback and fulfilment of aims.

Milestones:

- Complete real user testing (4)
- Finish the initial draft of final report. (5)

Week 11 (inc. Easter Break)**Tasks:**

- Make any necessary minor tweaks or bug fixes in the application.
- Write up of remaining sections in the report including, Results and evaluation, future work, conclusion and reflection on learning.
- Meet with supervisor and work to correct outstanding areas brought up regarding the report.

Week 12**Tasks:**

- Make any final amendments or corrections to the report.
- Validate and check references.

Milestones:

- Completion of the final report. (6)

	28/01/18	04/02/18	11/02/18	18/02/18	25/02/18	04/02/18	11/03/18	18/03/18	25/03/18	01/04/18	08/04/18 - 05/05/18	06/05/18
Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Initial Plan	[Green bar]											
Architecture Plan	[Green bar]											
Introduction and background (Report)	[Green bar]											
UI Design	[Green bar]											
Backend of application	[Green bar]											
Background and approach (Report)	[Green bar]											
Data Visualisation (Frontend)	[Green bar]											
Implementation (Report)	[Green bar]											
Generation of Geospatial Queries	[Green bar]											
Integration Tests	[Green bar]											
Running (Testing) query against connected DB	[Green bar]											
Breakdown explaining generated query	[Green bar]											
Addition of frontend features	[Green bar]											
Conduct and write up of real users tests assessing feedback and fulfilment of aims	[Green bar]											
Remaining sections in final Report	[Green bar]											
Final Report Submissions	[Green bar]											

Key:
Milestone Completion = x

Bibliography

- [1] Docs.mongodb.com. (2019). Geospatial Query Operators MongoDB Manual. [online] Available At: <https://docs.mongodb.com/manual/reference/operator/query-geospatial/> [Accessed 29 Jan. 2019].
- [2] Murugaiyan, M. (2012). WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC. [online] Jitbm.com. Available at: <http://jitbm.com/Volume2No1/waterfall.pdf> [Accessed 30 Jan. 2019].