

# Data Management System for A Series of Kinematic Movement in Football Analytics

Nur Athirah Umairah Binti Mohd Hafizul, Muhd Azimin Ab. Ghani, Khairul Imran Sainan\*

School of Mechanical Engineering, College of Engineering,  
Universiti Teknologi MARA, 40450 Shah Alam, Selangor  
\*imransainan@uitm.edu.my

## ABSTRACT

*A web application was developed to visualize player performance. The web app requires a comprehensive database to store the performance data. The primary aims of this study were to develop a platform for the user to download and upload the data, and a database. Meanwhile, the second objective is to manage the stored data according to the front-end requirement. Data from an actual match was used as the sample of the study. A platform for the user to download and uploading files was created. The downloader was designed for the pre-formatted raw data file. The file consisted of 12 columns of x-y coordinate data. The number of rows formed based on the total number of players. With this, the user re-uploads the file that has been filled through the uploader in the web app to be stored in the database. Next, a folder is created in the cloud database. The plugin in the hosting platform is used to connect the form to the database to store the uploaded file. Another server calculated the uploaded file uploaded by the user. A new folder was created to store the calculated data. There are two front-end requirements. For “real-time” analysis, the stored data were made available online using cloud computing database. Meanwhile, for periodical updates, the data were managed accordingly based on the NoSQL Document Oriented Database Management System (DODBMSes) approach. As a result, the online data stored in the database was sent to another server for processing and displaying the results, such as graphs and heatmaps. In conclusion, this study has successfully proven that uploaded files are stored in the database and managing existing data in the database to be sent to other servers.*

**Keywords:** *Cloud Database; JSON Approach; Kinematic Movement; Football Analytics*

## **Introduction**

Modern professional sports such as soccer, football, or basketball are increasingly supported by analysing quantities of data about players, performances, history matches, and various additional info about the sports team [1-3]. Sport analytic through current and past data is a growing field with various implementation methods from various situations [4]. This study uses a Web Application platform to show the results of analytic data. The database is required to store and manage data. Hence, the requirement for the front end is achieved.

Third parties are involved in the database management task to achieve the first front-end requirement, periodical analysis. Traditional PC-based framework securing cards cannot handle high image transformation and high data transmission [5]. The processor chip is based on von Neumann engineering calculations, with the restriction that there is not a lot of workloads in both hardware and programming. [6,7]. A Cloud database is a suitable platform for third parties to access the database since it is an online platform. Through the network, multiple low-cost computers are integrated into a powerful computing capability system. A cloud database is a cloud computing data storage and retrieval resource. A cloud database is a large data system in which some data is stored in the cloud and others are stored on a medium-sized service provider's server. [8]. The distributivity and dynamics of cloud databases make cloud databases extensive storage resources.

Next, real-time data analysis is the next front end. In the context of extensive data analysis, a multitude of NoSQL solutions to store, manage and extract information have been implemented. NoSQL Document Oriented Database Management Systems (DODBMSes) with data integrity, better data analysis, and high efficiency are becoming more popular [14, 23]. These characteristics are essential for real-time web apps and Big Processing of data [9]. With the introduction of Big Data platforms, where data can be transferred over networked computers into JSON format, NoSQL DODBMSes have emerged as viable solutions for storing and retrieving device semi-data structures. The use of DODBMSes has the advantage of fast query performance [15]. It includes query engines for 'CREATE,' 'READ,' 'UPDATE,' and 'DELETE' operations, as well as economic indicators for better data retrieval. DODBMSes store XML and JSON data using documents. The JSON structure is used by JSON Database Management System (JDBMSes) to store data in the collection and to model the documents. Initially, the data is obtained in the form of delimited text files. Then, a multi-paradigm programming language is used to convert delimited text files to JSON documents. The previous study states that it will be easier to create analysis reports and create model testing [20]. JSON system uses JSON navigation instruction to query JSON documents using programming language and

JSONPath query language [22]. Furthermore, JSON libraries in scripting languages provide an iterator for issuing instructions to perform a task. [18,19].

To sum up, this study aims to meet the front-end requirement to be displayed in web applications. Therefore, a cloud computing database is used for the first front-end requirement. For the following front-end requirement, a NoSQL solution is used.

## **Methodology**

The project methodology presented focuses on the technique used to achieve the objective and the method used to fulfil the front-end requirement.

### **Platform design and implementation**

A platform was developed based on the requirement to get data from the client. A software component called a plugin is used to add features to existing web pages in web development, thus enabling customization [21]. The platform in the web application was created using a plugin. The platform used is to download and upload files. A preformatted data file contains a header of 12 columns was created. The preformatted data file was subjected to the client to download and fill in the player's data. Files containing player data are uploaded to the platform in delimited text file format. The plugin used to create the file upload platform has a function to store data in a database as shown in Figure 1. A database was used to store the uploaded file.

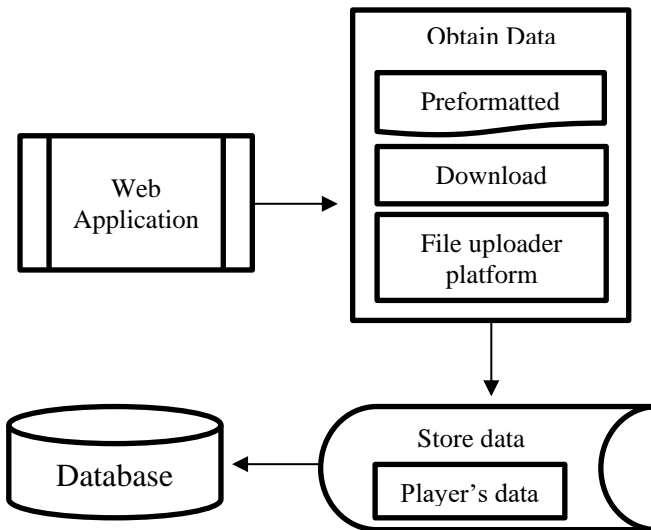


Figure 1: Data store in database flowchart.

### Store and manage data

The uploaded file needs to be saved in data storage. Various data storage centers are created for significant data challenges. Cloud database systems are used to meet user requests for ample data storage [10, 11, 12, 13]. A Cloud database is a built-in service for databases accessed through cloud platforms. Cloud information provides many frameworks with various advantages to users [24]. Cloud Database Management System is an online data storage concept. The purposes of this method are to allow third parties to extract data to meet front-end requirements. Based on Figure 2, two folders were created based on two requirements to be met. The first file stores the data that the client uploads for periodical data use. While the second file stores the data that has been calculated for the use of real-time data analysis. Through this, the data stored in the database can be accessed by third parties to meet the requirements.

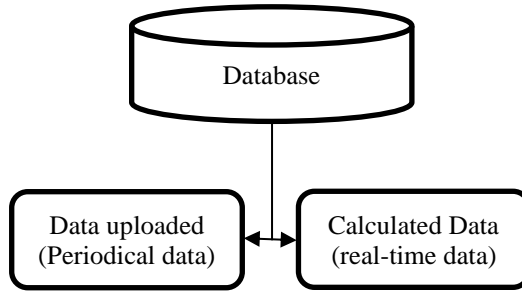


Figure 2 : Database setup for front-end requirement.

### Dataset extraction and deployment

A multi-paradigm programming language is used for this method. A multi-paradigm program language supports extensive data and functions in one structure [17]. It is used to convert data files in delimited text file format to JSON format as shown in Figure 3. The JSON format is used because it is a compact data format based on the JavaScript programming language. [22]. Data in JSON format is used in real-time data analysis requirements.

```
import pandas as pd

df = pd.read_csv (r'Path where the CSV file is saved\File Name
.csv')

df.to_json (r'Path where the new JSON file will be stored\New
File Name.json')
```

Figure 3: Programming language used to convert delimited text file format to JSON format.

Document-Oriented Database Management Systems (DODBMSes) store JSON data using documents. DODBMSes are used to store, retrieve, and manage semi-structured data thus, to provide flexibility for data modeling [14]. JSON Database Management System (JDBMSes) is a subcategory of DODBMSes. It has a system that uses JSON format for document encoding. The transformed JSON data is a well-attribute that helps preserve the machining-interpretable purpose's original attributes [16]. Figure 4 represent the JSON document is deployed as an API URL to display real-time data analysis results.

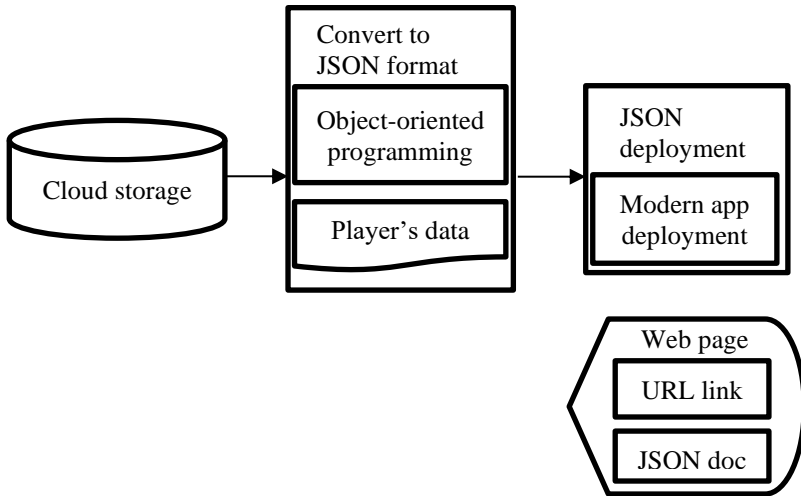


Figure 4: Data retrieval flowchart

## Results and Discussion

The results presented focus on the data storage and data retrieval for front-end requirements.

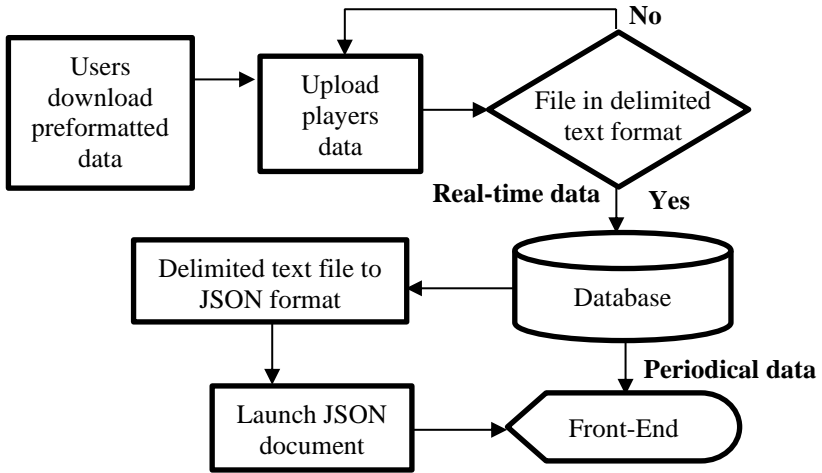


Figure 5: Flowchart of overall process.

The platform for download and upload was created using a plugin; thus, it links to a database for data storage as represent in Figure 5. Cloud databases were used for data storage. Cloud database provides ease for third parties to access data than the MySQL database that acts as localhost. MySQL also has a limitation where data in the form of files is difficult to display in the database table. The data storage needs to be stored online and provide extensive storage to meet the front-end requirement. Therefore, a Cloud database was used since it can store extensive data and the database works online. Data in the cloud database was retrieved using the query process. The data was used for periodic updates using an online dashboard. Since the dashboard works online, it is much easier to access the cloud database.

The deployment of a JSON document into a webpage was a requirement for displaying real-time analysis results. The real-time analysis method uses a system that uses URL links to pull data in the webpage. First, the document in JSON format was launched into a website. Then, the URL from the launch webpage was used to retrieve JSON data. Figure 7 shows the sample of players data that have been converted to JSON format.

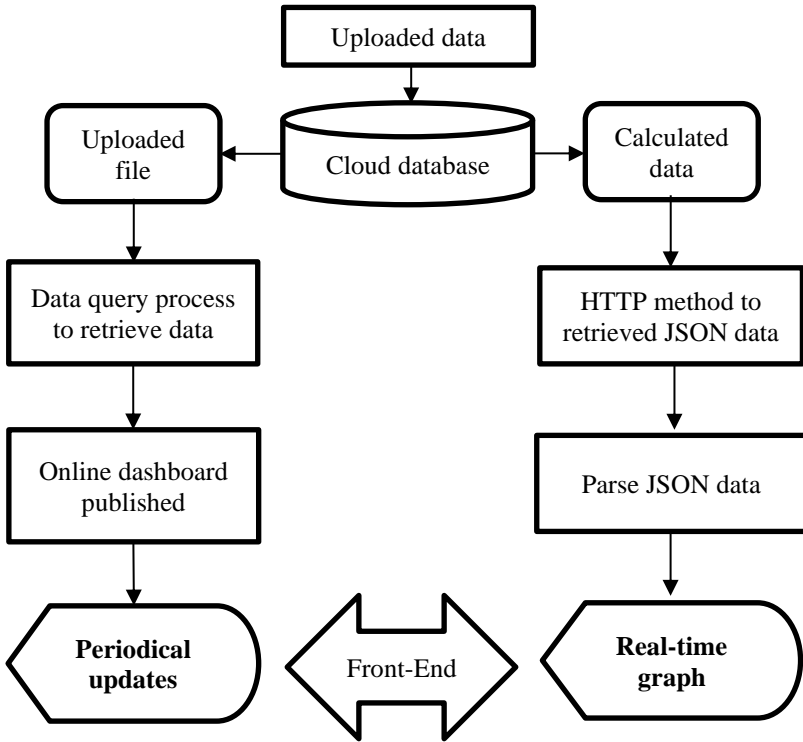


Figure 6: Pipeline to meet the front-end result.

```

{"Match":{"0":"SSSA vs. MONSTA FC","1":" SSSA vs. MONSTA FC","2":"
SSSA vs. MONSTA FC"},
  "Home":{"0":"SSSA","1":" SSSA ","2":" SSSA "},
  "Away":{"0":" MONSTA FC","1":" MONSTA FC","2":" MONSTA FC"},
  "Player ":{0:"Mukhlis (1)","1":" Mukhlis (1)","2":" Mukhlis
(1)"},
  "Club":{"0":" SSSA ","1":" SSSA ","2":" SSSA "},
  "Time":{"0":0.0,"1":0.5,"2":1.0},
  "x(1) ":{0":4.14,"1":5.4,"2":5.71},
  "y(1) ":{0":35.3,"1":34.85,"2":34.61},
  "x(2) ":{0":4.06,"1":4.75,"2":4.66},
  "y(2) ":{0":35.64,"1":34.99,"2":34.89},
  "x(3) ":{0":4.47,"1":4.25,"2":4.85},
  "y(3) ":{0":35.12,"1":34.85,"2":35.03}}
    
```

Figure 7: Sample of players data in JSON format.

## Conclusion

The study has satisfied the objective of data storage and retrieval to fulfil the front-end requirement. The results show that data storage and data retrieval techniques can display the front-end requirement for football analytics. However, the method needs to be further investigated to ensure that the data management system in football analytic is beneficial.

## Acknowledgement

The authors would like to thank the School of Mechanical Engineering College of Engineering, University Teknologi MARA, for supporting this manuscript.

## References

- [1] T. Hayduk, *The future of sport data analytics*, 1st ed. Routledge, 2020. doi: 10.4324/9780367854454-14.
- [2] H. C. Shih, "A Survey of Content-Aware Video Analysis for Sports," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 5, pp. 1212–1231, May 2018, doi: 10.1109/TCSVT.2017.2655624.
- [3] L. Morra, F. Manigrasso, G. Canto, C. Gianfrate, E. Guarino, and F. Lamberti, "Slicing and dicing soccer: Automatic detection of complex events from spatio-temporal data," *International Conference on Image Analysis and Recognition*, vol. 12131, pp. 107–121, 2020, doi: 10.1007/978-3-030-50347-5\_11/COVER.
- [4] B. Gerrard, "Moneyball and the role of sports analytics: A decision-theoretic perspective," *North American Society for Sport Management Conference*, pp. 2010–2012, 2016.
- [5] C. Yang and M. Chen, "Design of High Performance Image Acquisition and Processing Platform Based on DSP and FPGA," *Proceedings of 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications, AEECA 2020*, pp. 684–688, Aug. 2020, doi: 10.1109/AEECA49918.2020.9213543.
- [6] P. K. Han et al., "Free-Breathing Three-Dimensional T1 Mapping of the Heart Using Subspace-Based Data Acquisition and Image Reconstruction," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4008–4011, Jul. 2019, doi: 10.1109/EMBC.2019.8856511.
- [7] W. P. Segars, B. M. W. Tsui, J. Cai, F. F. Yin, G. S. K. Fung, and E. Samei, "Application of the 4-D XCAT Phantoms in Biomedical



- Imaging and Beyond,” *IEEE transactions on medical imaging*, vol. 37, no. 3, pp. 680–692, 2018, doi: 10.1109/TMI.2017.2738448.
- [8] Y. H. Zhang, L. Feng, and Z. Yang, “Optimization of Cloud Database Route Scheduling Based on Combination of Genetic Algorithm and Ant Colony Algorithm,” *Procedia Engineering*, vol. 15, pp. 3341–3345, Jan. 2011, doi: 10.1016/J.PROENG.2011.08.626.
- [9] C. O. Truică, E. S. Apostol, J. Darmont, and T. B. Pedersen, “The Forgotten Document-Oriented Database Management Systems: An Overview and Benchmark of Native XML DODBMSes in Comparison with JSON DODBMSes,” *Big Data Research*, vol. 25, Jul. 2021, doi: 10.1016/J.BDR.2021.100205.
- [10] V. Anand and C. M. Rao, “MongoDB and Oracle NoSQL: A technical critique for design decisions,” *International Conference on Emerging Trends in Engineering, Technology and Science*, Oct. 2016, doi: 10.1109/ICETETS.2016.7602984.
- [11] B. Subramaniam and W. C. Feng, “On the energy proportionality of distributed NoSQL data stores,” *International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems*, vol. 8966, pp. 264–274, 2015, doi: 10.1007/978-3-319-17248-4\_14/COVER.
- [12] A. Thusoo *et al.*, “Hive - A petabyte scale data warehouse using hadoop,” in *International Conference on Data Engineering*, 2010, pp. 996–1005. doi: 10.1109/ICDE.2010.5447738.
- [13] M. N. Vora, “Hadoop-HBase for large-scale data,” *Proceedings of 2011 International Conference on Computer Science and Network Technology*, vol. 1, pp. 601–605, 2011, doi: 10.1109/ICCSNT.2011.6182030.
- [14] P. Atzeni, F. Bugiotti, L. Cabibbo, and R. Torlone, “Data modeling in the NoSQL world,” *Computer Standards & Interfaces*, vol. 67, p. 103149, Jan. 2020, doi: 10.1016/J.CSI.2016.10.003.
- [15] E. Gallinucci, M. Golfarelli, and S. Rizzi, “Schema profiling of document-oriented databases,” *Information Systems*, vol. 75, pp. 13–25, Jun. 2018, doi: 10.1016/J.IS.2018.02.007.
- [16] Z. Dong, S. Paul, K. Tassenberg, G. Melton, and H. Dong, “Transformation from human-readable documents and archives in arc welding domain to machine-interpretable data,” *Computers in Industry*, vol. 128, p. 103439, Jun. 2021, doi: 10.1016/J.COMPIND.2021.103439.
- [17] R. Drechsler *et al.*, “Panel: Future SoC verification methodology: UVM evolution or revolution?,” in *Design, Automation & Test in Europe Conference & Exhibition*, Apr. 2014, pp. 1–5. doi: 10.7873/DATE.2014.385.
- [18] E. Botoeva, D. Calvanese, B. Cogrel, and G. Xiao, “Expressivity and Complexity of MongoDB Queries,” in *21st International Conference*

- on *Database Theory*, 2018, no. 9, pp. 1–23. doi: 10.4230/LIPIcs.ICDT.2018.9.
- [19] K. Ong, Y. Papakonstantinou, and R. Vernoux, “The SQL++ Semi-structured Data Model and Query Language: A Capabilities Survey of SQL-on-Hadoop, NoSQL and NewSQL Databases,” *ArXiv*, 2014.
- [20] G. Renuka, “Monitoring the state of materials in verification environment for IP architectures using python based verification mechanism,” 2021.
- [21] J. Sterne, “Plug-in Software,” *Britannica*, 2022. <https://www.britannica.com/technology/plugin>
- [22] P. Bourhis, J. L. Reutter, and D. Vrgoč, “JSON: Data model and query languages,” *Information Systems*, vol. 89, 2020, doi: 10.1016/j.is.2019.101478i.
- [23] M. Stonebraker, “SQL databases v. NoSQL databases,” *Communications of the ACM*, vol. 53, no. 4, pp. 10–11, Apr. 2010, doi: 10.1145/1721654.1721659.
- [24] J. Ding, “Cloud computing database and remote system for real-time image acquisition application in english classroom teaching,” *Microprocessors and Microsystems*, vol. 82, p. 103916, Apr. 2021, doi: 10.1016/J.MICPRO.2021.103916.