MigDB
Relational to NoSQL Mapper

Project ID: 16-015
Project SRS Document

B.Sc. Special (Honors) Degree in Information Technology
Submitted on 01/04/2016
IT13001476

Project ID : 16-015

Project Group Members:

Student ID	Name	Signature
IT13001476	Liyanaarachchi L.A.G.C	
IT13088156	Madhusanka K.P.L.K	
IT13093938	Kothalawala K.R.M.N	
IT13075422	Padmasiri H.R.K.L	

Supervisor

……

Dr. Anuradha Karunasena

Co-Supervisor

……

<Name of the Co-supervisor>

DECLARATION

I declare that this is my own work and this project proposal does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Student ID: IT13001476

Student Name: L.A.G.C Liyanaarachchi

Signature: .....

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Purpose

The purpose of this document is deliver comprehensive information about a data migration research project. This document has capability to get complete understanding about requirements of the problem considering functional and nonfunctional categories, designed architecture to resolve problem, features of the system, technologies chosen for this project, expected milestones, expected test execution plan, mockups of user interfaces and projected outcomes of the system.

1.2 Scope

This document define for a group research of migrating data from relational database to NOSQL database especially for [1] MongoDB database. This project was named as MigDB. This project is consisting with relational database analyzing module, from relation tables to collections mapping module, collection management module. There have been number of researches available in data migration and all the awareness collected at the literature review has added to this document.

User of the system can configure connection to the database and establish connection with databases. System has intelligent to analyze relational data dump and write created Mongo database to the disk using connection manager. MigDB collection management module has capability to read Mongo database and load database structure to a user interface and enable user to manipulate existing data or add new data. MigDB collection management module create databases, update databases and delete databases in Mongo database with user's instructions. The main intention of this project is creating standalone application to migrate existing relational database to the MongoDB host without any error along with previous relationships injected to the collections. This document has been using for the development of the system by the developers, testing of the system by test engineers and the evaluation of the system by clients.

1.2.1 Definition of the System

This system is designed to function as a desktop application and system gain more functionalities through cloud application interface. This system is contain four major assignments. These assignments have been completing individually by the research team members.

This document is especially describe about what is semantic network, how to implement semantic network with diagrammatical expressions, how to learn a sematic network using supervised learning methodology, what are the expected inputs to the network and what are the outputs. This network is a cloud service and expose internal functionalities through a secured application interface to be accessed by desktop application to get decisions while database collections are mapping. System is powered by incremental machine knowledge to map collections with the support of experienced users of the system.

Rest of the assignment is on MigDB management module and create interactive user interface to represent collections in Mongo database to the user as tables. These tables are very sensitive with user events on screen and user has capability to customize entire Mongo database structure by simple drag and drop events. This module is highly focused on user experience and user friendliness to make user events much easier. Final fragment of this project is implementing logic to convert existing referencing mapping in collection to embedded mapping.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Acronyms

RAM	Random Access Memory
JDK	Java Development Kit
JRE	Java Run Time Environment
SQL	Sequential Query Language
NoSQL	Not only SQL
DB	Data Base
1:N	One to Many
N:M	Many to Many
GUI	Graphical User Interface
API	Application Programming Interface
CSV	Comma Separated Values

Table 1: Acronyms

1.4 Overview

1.4.1 Overview of Software

MigDB system was sliced into four categories to enhance development velocity with consideration of complexity, dependents, technologies, performance and volume of workload.

- Connection Manager, SQL DB Analyzer, One-to-one Mapping, Graphical Query Generator
- SQL DB Modification Evaluator, One-to-one Mapping, SQL Query Converter
- **Semantic Network for Collection Mapping, Creation of MongoDB Collection Modifier UI, Implementation of Converting of Referencing to Embedding Logic**
- Many to Many Mapper, conversion of embedded mapping to referencing, MongoDB Modification Evaluator, MongoDB Data Manipulation Module

The main goal of MongoDB system is to migrate relational database into a Mongo Database with relationships between collections as relational database table relationships alone with converted existing SQL queries into NoSQL commands.

Sub goals of this system can be listed as sequence of requested processes of the system.

- Create a connection manager to operate connections with databases.

- Create module to analyze table structure of relational database and initialize internal JSON file.
- Create module to customize relational database tables, columns and add or remove from migration list. This module has intelligent to evaluate user changes and restrict harmful modifications.
- **There are two options to use for 1:N and N:M mapping events. Mapping model can be referencing or embedding. This system has semantic network to decide best mapping model considering table information within small time period. This module has capability to trigger scheduled learning tasks using CSV files and return back to the functioning state.**
- Create module to map all 1:1, 1:N and N:M relationships to collections and migrate all data. This module has capability to write newly created database to Mongo database through connection manager.
- **Create interactive user interface to represent Mongo database collections and enable user to customize collection structure.**
- Create module to evaluate user changes on collections and restrict harmful changes and rollback to error free state.
- **Create module to analyze existing referencing mapping between collections and transform it to embedding mapping with user's request.**
- Create module to analyze existing embedding mapping between collections and transform it to referencing mapping with user's request.
- Create module with graphical representation of data in Mongo database and enable user to manipulate data through the graphical interface.
- Create module to generate NoSQL commands to handle queries of a Mongo database with graphical interface.
- Create module to convert existing SQL queries into NoSQL commands without user interaction.

To achieve above goals system need to follow below sequence of tasks.

1. Design architecture, breakdown workload, plan timeline and select technologies.
2. Implement Relational database analyzing module and map entire database to the intermediate JSON file.
3. Arrange all the data in relational database into Mongo database collections with mapped relationships and write them to the disk.
4. Create Mongo database collection management module.
5. Create test cases considering functionalities and test system against test cases and SRS document.

The target audience of this product is software developers, database administrators, individual software product owners and software organizations. MigDB have capability to behave as a support software product in most cases of data migration with relationships.

1.4.2 Organization of SRS Document

This document consisting with major four section.in the initial section this document describe introduction to data migration from relational database to MongoDB. In this

section document describe purpose and scope with system definition. Second Section provide overall description about software requirements, hardware requirements, user interfaces, communication interfaces, system features, development environment, user characteristics and constraints. Third section define in detail information about above topics. Final section provide support referred URLs and documents to clear idea about the content of the document.

2 Overall Descriptions

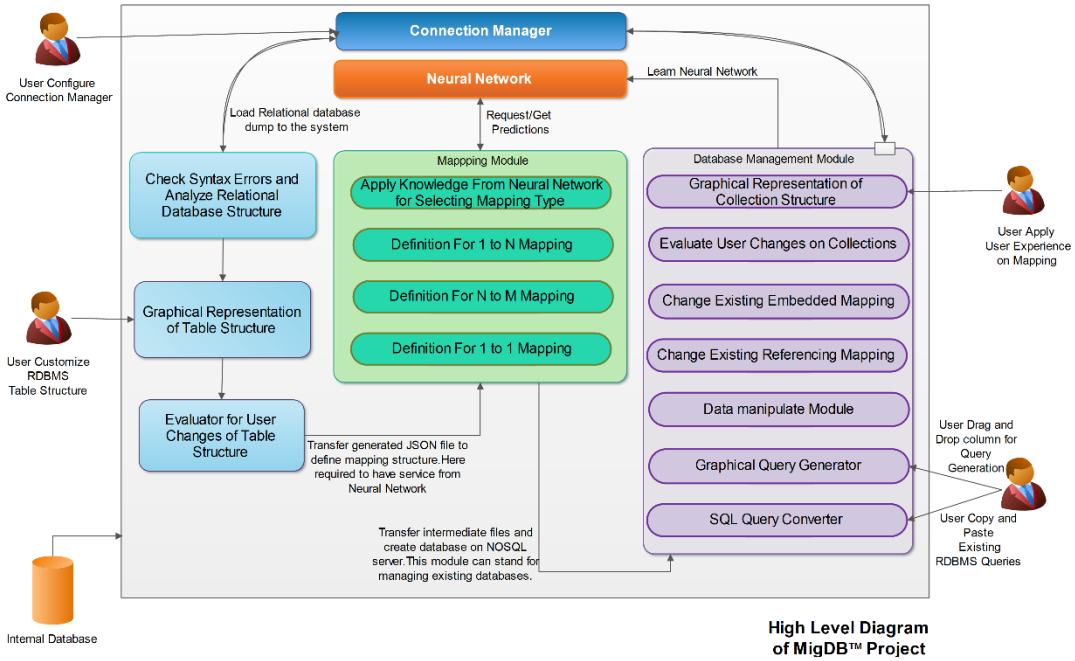

Figure 1: High Level Diagram

MigDB system is a desktop application designed to migrate existing relational database into Mongo database along with data and converted SQL queries into NoSQL commands. MigDB system has capability to map previous relationships between relational tables into relationships between collections. User has functionalities to provide connection information to connect with relational database and Mongo database and establish connection with these databases. Relational database analyzing module has capability to download backup file from selected database by the user. Relational database analyzing module read database dump and initialize intermediate JSON file including all the table information, columns and relationships between tables. Before start migration process system graphically display relational database table structure and enable user to add or remove tables, column from migration list. This module has intelligent to restrict harmful user changes and rollback system to the error free state with providing notifications to user about the error. Mapping module take intermediate JSON file as input and analyze all the table information. Firs begin to map 1:1 relationships and create second intermediate JSON file and write created collections and write data of those tables into collections. When mapping 1:N and N:M mapping system needs to select collection mapping model. N:M mapping is more complex as system need to select which collection need to be embedded or reference.

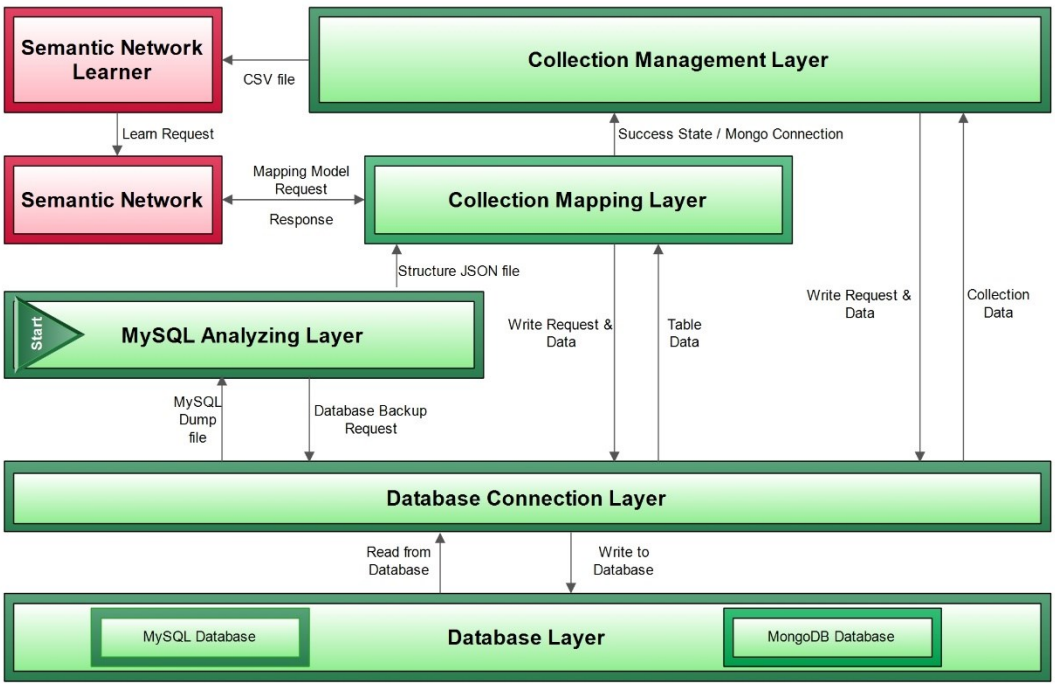

Figure 2: Layered Diagram of System

MigDB system is consisting with [2] Semantic network to take this decision without having procedural classic programming methodology a semantic network powered by incremental machine knowledge. This module require table, column count, data type count as input and it provide mapping model and numeric range to select embedding or referencing table. This module is a cloud service which expose functionalities to client applications through a Restful API via the internet. MigDB cloud API contain security gateway to check authentications of the requests. If the security gateway validate request and pass through then request move to appropriate mapping module. This module validate provided parameters against standards and push validated request to load balancer. Load balancer control parallel requests at same time and manage load on the neural network. Neural network take inputs and process through internal nodes and return output value. Post analyzing modules of semantic network take this output and define decisions and create well-structured and pre-defined JSON file to deliver to client end. This semantic network has self-learning module. This module has major two components. First one is user interaction information evaluator. This component analyze user changes on collection mapping with respect to the column count and data type count and write them to a CSV file. System was designed to use supervised learning methodology and all the input sets and output sets extract from CSV file. Neural networks are incapable to learn incrementally. So system has scheduled learning task trigger module to start learning process and drop current neural network status and learn from beginning including all the newest updates from the CSV file.

MigDB system turn to the collection management module after finishing all mapping and writing those mapping to the Mongo database. System generate graphical representation of newly created Mongo database structure to the user in very user friendly manner and enable user to do changes to the collection structure. This user interface is a highly sensitive interface that react user actions on window. This module has capability to restrict harmful user actions and prevent database from errors. These

evaluated user changes apply to the database via mapping changing modules. First module has capability to convert existing referencing mapping in database to embedding mapping and restore data. Second module has capability to convert existing embedding mapping to referencing mapping and restore data. Collection management module has capability to read data and display in graphical interface. User can update data through this interface. Another feature of management module is converting SQL queries to NoSQL commands without knowledge of user. This module can create new commands using graphical interface and enable user to select collections and provide output as NoSQL commands. This module has intelligent to convert existing SQL queries into NoSQL command and provide these commands to the user.

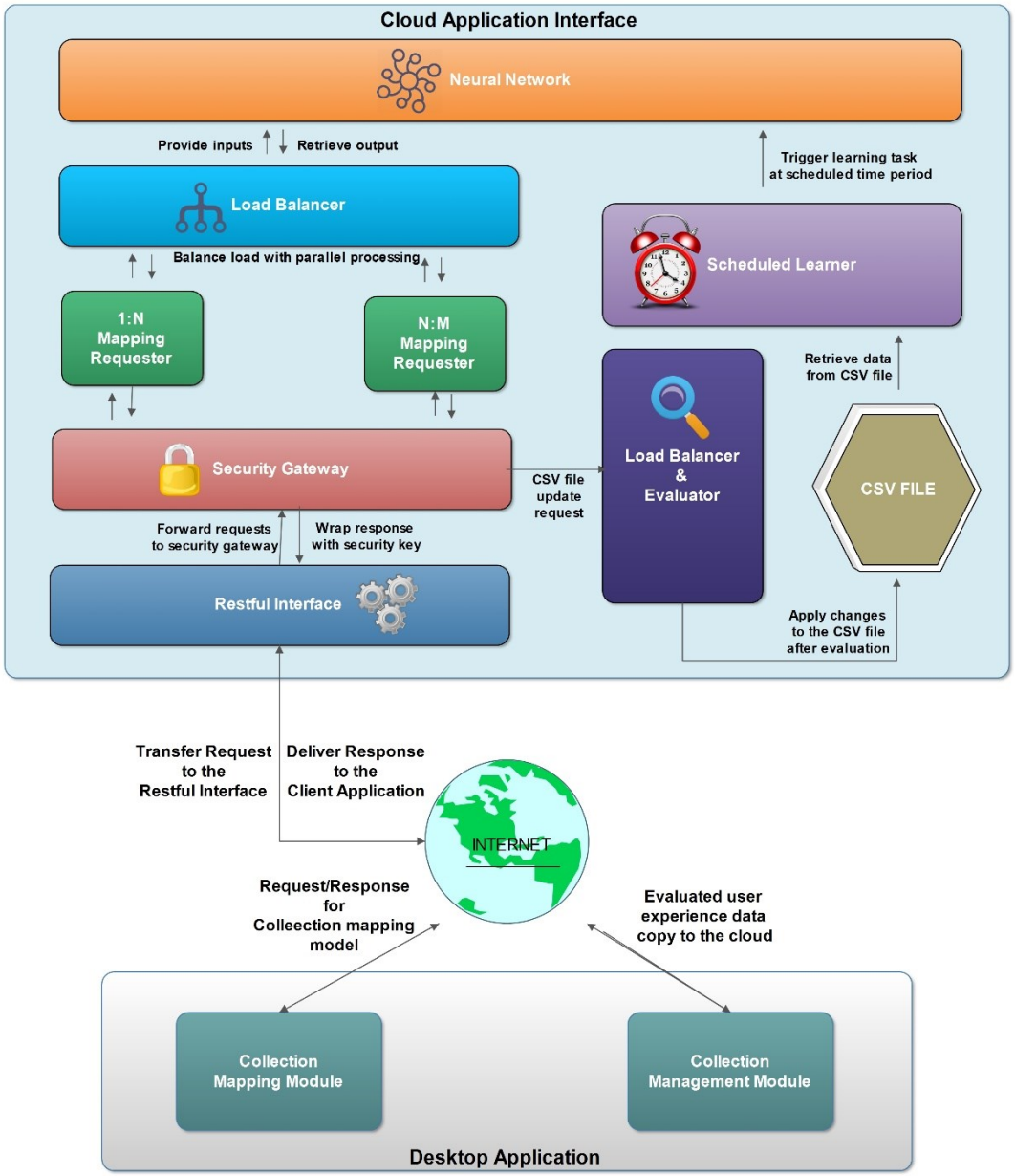

Figure 3: Infrastructure of Cloud Service

2.1 Product perspective

The research [3] by Gansen Zhao, Weichai Huang, Shunlin Liang and Yong Tang describe how to convert relational model to NoSQL model using embedding mechanism to improve query speed of NoSQL databases. This method is not feasible because number of foreign keys increase with respect to the size of the database.

Pelica Migrator [4] is a commercial product widely uses in industry to migrate from relational database to NoSQL databases. Pelica Migrator supports to migrate table structure along with data. The drawback of this software is it not maintain relationships between tables into relationships between collections.

Robo Mongo [j] is an open source product to manipulate mongo database data through a graphical interface. The drawback of the Robo Mongo tool is it require previous MongoDB shell command knowledge to manage data.

2.1.1 System interfaces

2.1.1.1 Development Environment

1. Windows or Linux operating system.
2. Eclips IDE v4.5.2
3. Java 8 JDK
4. JavaFX binaries
5. Oracle SceneBuilder v2.0 IDE
6. Apache Tomacat v8 server
7. Ant builder
8. Sqlite binaries
9. MongoDB binaries
10. MySQL binaries
11. Neuroph library binaries
12. Git tool binaries

2.1.1.2 Production Environment

1. Windows or Linux operating system
2. MongoDB binaries
3. MySQL binaries
4. Java 8 JRE

2.1.2 User interfaces

MigDB system consist with two applications. First application is desktop application and second application is cloud application. Cloud application provide service to client applications via a software interface and it does not have any graphical interface. This document is mainly focus on sematic network and collection management module. Desktop application has multiple interactive user interfaces to get user inputs and provide system state to the user. First interactive interface is relational database structure presentation view. This view graphically represent data and enable user to select which tables and columns need to be migrate. Second interface is Mongo database collection structure presentation interface. User has capability to change

mapping model through this interactive user interface. Third user interface is Mongo database collection data presentation interface. This interface enable user to update data. Final interface is SQL to NoSQL converting interface. This interface enable user to convert existing queries into NoSQL commands.

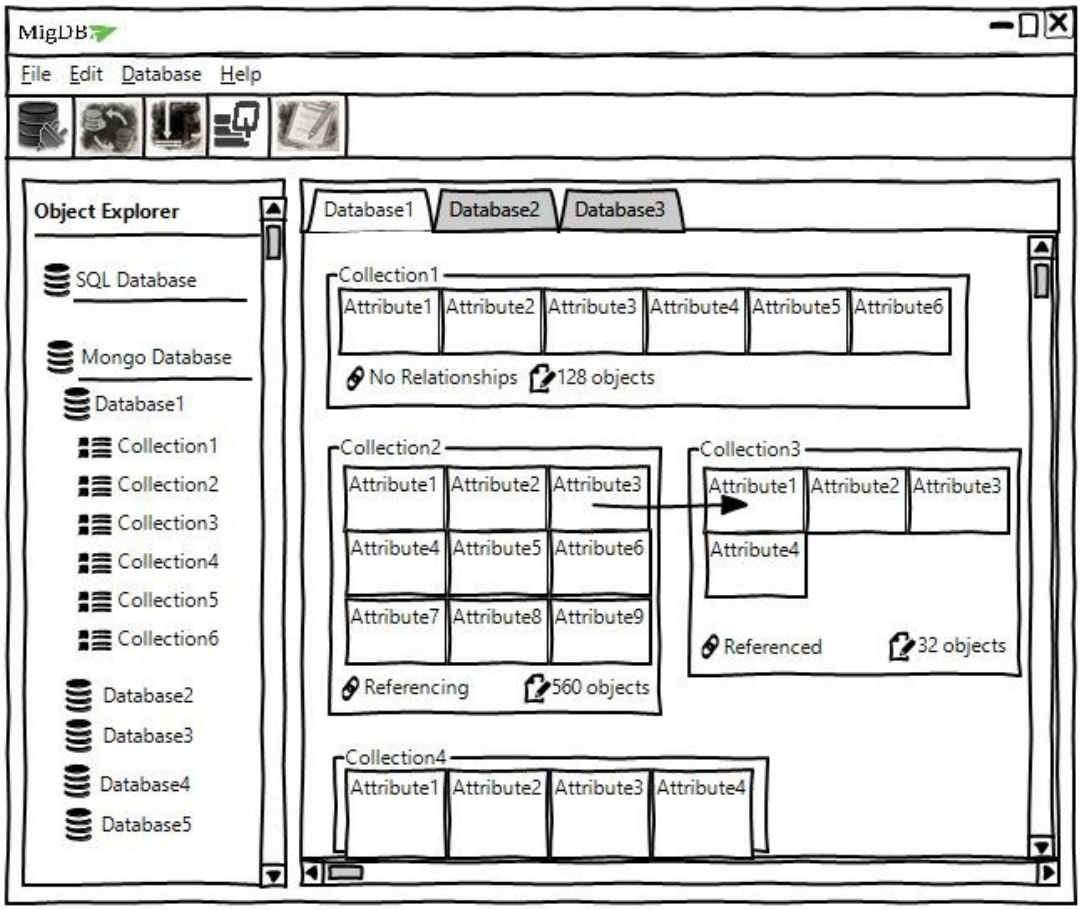

Figure 4: Collection Management Interface

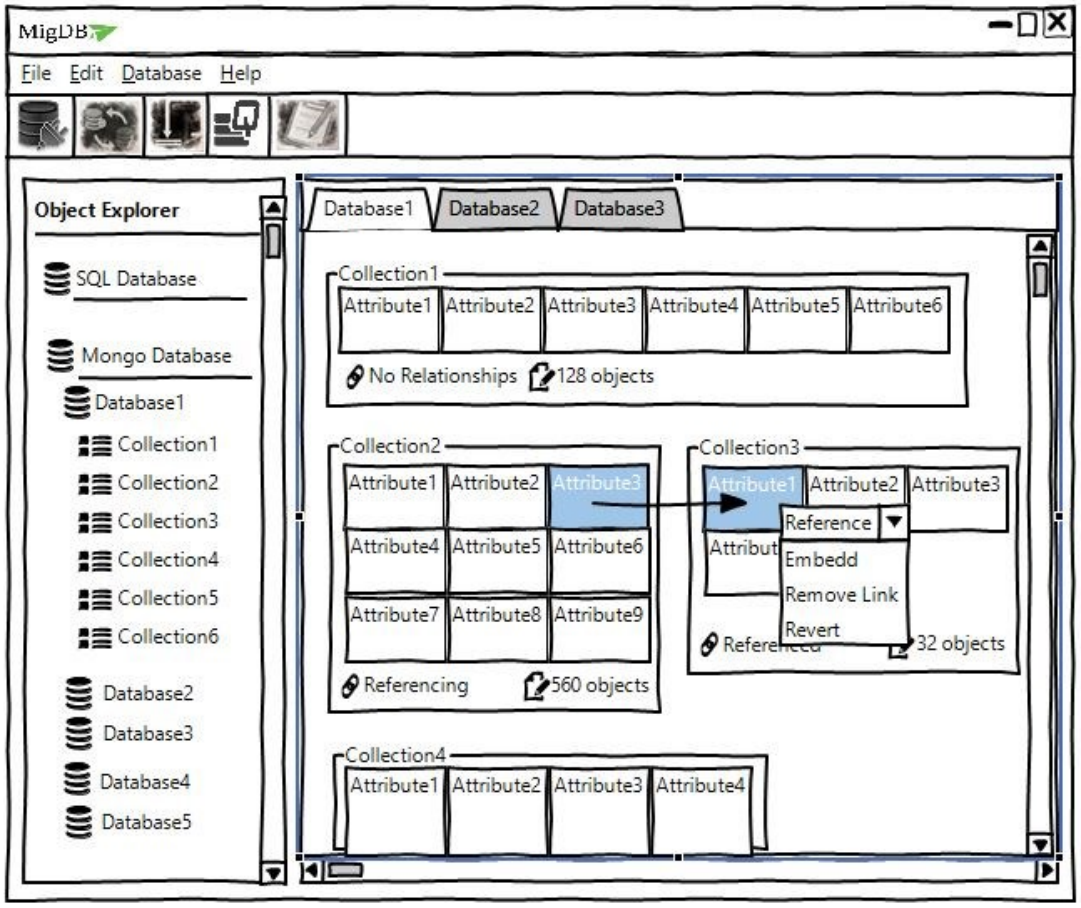

Figure 5: Change of Mapping Model

2.1.3 Hardware interfaces

- Desktop/ Laptop Computer: MigDB desktop application runs on a computer with any operating system.
- Apache Server: MigDB cloud application interface host on remote server and make available for client applications.
- Internet Connection: Dialup, Modem, 3G or 4G internet connection between desktop computer and internet.

2.1.4 Software interfaces

MigDB data migration application was designed to use in all platforms since it use Java programming language for the development. User interfaces create using JavaFX technology. MigDB use Git repository management tool for source code version control. Most of the software tools are freely available on internet as open source products.

2.1.5 Communication interfaces

MigDB desktop application needs continues internet connection via any internet technology .e.g. Modem, 3G, 4G or Wi-Fi (IEEE 802.11b,g,n)

2.1.6 Memory constraints

For the Desktop Application:

MigDB system require to have minimum 2GB (4GB preferred) Ram to install application. All the temporary files are loading to the memory while the migration process is going on. Best case is small relational database dump. But worst case can be very large database dump file which size is about 100GB.

For the Cloud Application:

Sematic network load to the memory of the server and start functioning. It requires minimum 1GB (2GB preferred) free memory on server to operate without any conflicts.

2.1.7 Operations

- User open MigDB product web site.
- User select operating system and download relevant distribution binaries.
- User open installation wizard and install software.
- User open MigDB software application.
- User select connection management tab.
- User provide host, port and password for the MySQL database and save connection information.
- User select more connections and provide MongoDB host, port, directory path to bin folder of Mongo binaries and save connection information.
- User select relational database need to be migrate from object explorer.
- User select tables and column and click next button to move next step of migration.
- User see newly created Mongo database as collections.
- User change mapping models of collection if only user has good experience.
- User select Mongo database and update data and save.
- User copy and paste existing SQL query to the text box and click convert button.
- User drag and drop graphical symbols to generate new NoSQL commands.

2.1.8 Site adaptation requirements

User interfaces, Documentations of MigDB system should be in English language.

2.2 Product functions

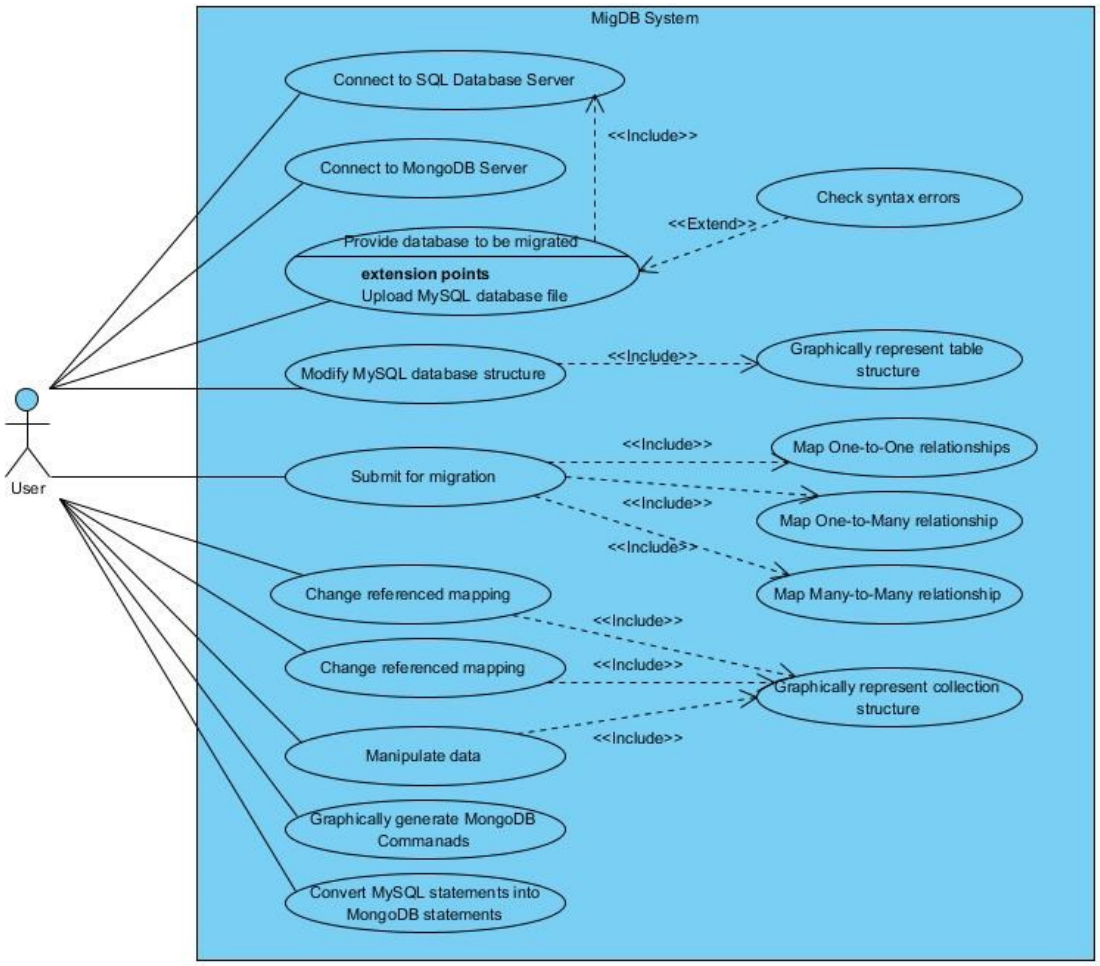

Figure 6: System Use Case Diagram

Use Case Name	Change referenced mapping
Description	When user select embedding from menu bar of referenced collection change current mapping model.
Pre-Condition	Establish connection with MongoDB
Post-Condition	Up and running connection with MongoDB.
Primary Actor	User
Main Success Scenario	• Identify Referenced object and embed to the collection. • Restore all data in collections. • Wait user until process complete • Show completion state to the user.

Table 2: Use Case Scenario

2.3 User characteristics

MigDB system designed especially for the people who need middleware to migrate from relational database to Mongo database and convert existing relational SQL queries into NoSQL commands. Previous experience in database management is not required for this software. User need to have basic knowledge about how to work with general software applications.

2.4 Constraints

User's Desktop Computer:

- Should have JRE 8 or above version
- Should support internet facilities
- Should have MySQL v5.7 or below installed
- Should have MongoDB installed
- Should have processer at least 1GHz or higher
- Should have RAM at least 2GB or higher
- Should have disk space 2GB+Relational backup size

Remote Server:

- Should have RAM at least 1GB or higher
- Should have Apache server installed
- Should have operating system

2.5 Assumptions and dependencies

First module of the MigDB is analyzing Relational database dump. The assumption is all the relational dumps have syntaxes below the version of MySQL v5.7.11.

Users provide database host as localhost has instead of remote host. System need to have more complex configurations to access remote host.

2.6 Apportioning of requirements

MigDB system develop and distribute within four significant iterations. In first iteration team plan to release connection management module, Relational dump analyzing and evaluating module, definition for the N:M mapping and neural network as a service. Second iteration contain definition for 1:1 mapping, definition for 1:N mapping, collection structure displayer and embedded to referenced converter. At the third round team plan to deliver SQL to NoSQL converter, NoSQL command generator, Data Manipulation module and referenced to embedded converter. Final stage of the development is unit testing and system testing with bug fixing.

.

3 Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

Figure 7: Delete Attribute from Collection

Above image is MigDB collection management interface mockup. This interface was designed to manipulate collection structure with user interaction through the GUI. Source of input for this interface is mouse click events. Destination output of this interface is the screen. Appearance of the collection window is totally depend on the attribute count and user need to scroll window to see all the collections. System trigger functionalities on collections based on user's inputs and all the outputs are derived from user inputs. Screen format and window format is available on above image.

3.1.2 Hardware interfaces

- Desktop/ Laptop Computer: MigDB desktop application runs on a computer with any operating system.
- Apache Server: MigDB cloud application interface host on remote server and make available for client applications.
- Internet Connection: Dialup, Modem, 3G or 4G internet connection between desktop computer and internet.

3.1.3 Software interfaces

MigDB development environment is consisted with below tools and technologies.

- Eclips IDE v4.5.2
- Java8
- Apache Tomacat 8
- Ant builder
- Git tool
- Spring MVC
- HTML
- Bootstrap CSS

3.1.4 Communication interfaces

MigDB desktop application needs continues internet connection via any internet technology .e.g. Modem, 3G, 4G or Wi-Fi (IEEE 802.11b,g,n)

3.2 Classes/Objects

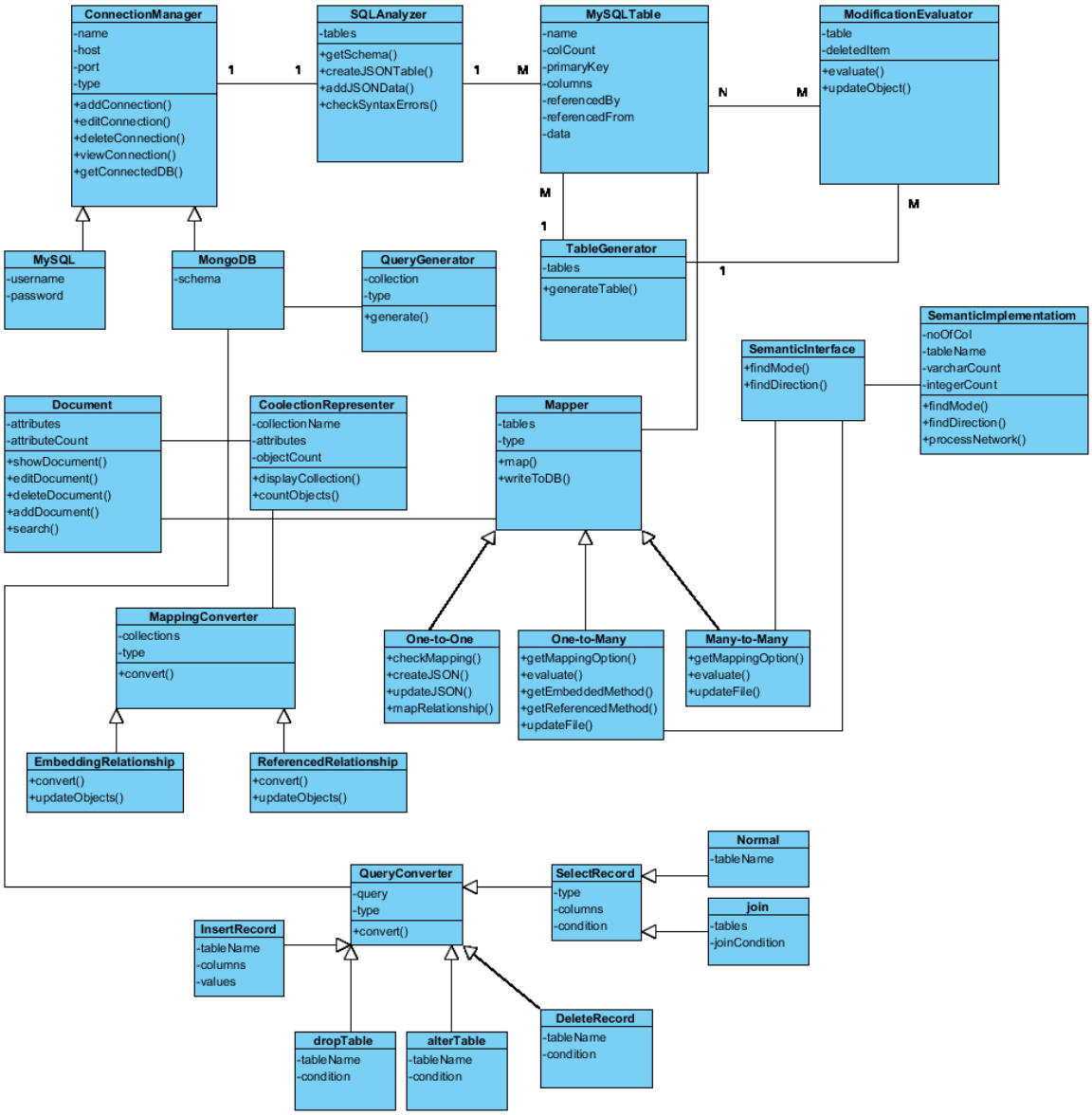

Figure 8: Class Diagram

3.3 Performance requirements

MigDB cloud application was designed to handle multiple simultaneous users using a load balancer. Desktop application has functionality to manage memory of the hosted system and resolve error situations and recover back. Performance of the system is highly depending with the external dependencies of the desktop application. With the increase of number of API calls system performance will be low. Solution is combining all requests for multiple table relationships combine to a one message and send to the server for processing as asynchronous call.

3.4 Design constraints

3.5 Software system attributes

3.5.1 Reliability

Reliability means capability of providing service for a defined time period without any fail. MigDB need to have more powerful exception handling, and crash recovery mechanism to achieve this quality. MigDB is highly focused on system memory consumption and provide more functionalities to manage system memory to avoid having memory leaks. Robustness is more valuable customer satisfaction.

3.5.2 Availability

[6] Availability is define ratio of mean time between failure to the combination of mean time between failure and mean time to recovery. To achieve 99.99% of availability system need to be tested using test automation and manual testing for all the available events and conditions.

3.5.3 Security

MigDB cloud application expose its functionalities to the connected clients via a restful interface. If this interface is identified by a malware and attack on traffic or resources to limit services of the system. To overcome these types of malware MigDB cloud application consist security layer to monitor and identify untrusted requests coming from harmful programs. Cloud service encrypt responses before sending to the client to protect desktop application from threads infected from the internet.

3.5.4 Maintainability

Maintainability means how strongly system was designed to face future requirements. MigDB architecture designed according to [7] Spring framework and it is very flexible to inject new components into execution flow. MigDB project maintain individual configuration files to separate application logic and programming logic for further improvements.

4 Supporting information

4.1 References

[1] MongoDB [Online] Available: https://www.mongodb.org/[Accessed: March 29, 2016].

[2] Semantic Network [Online] Available: https://en.wikipedia.org/wiki/Semantic_network[Accessed: March 30, 2016].

[3] Gansen Zhao, Weichai Huang, Shunlin Liang and Yong Tang, "Schema Conversion Model of SQL Database to NoSQL", in 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Guangdong, China, Nov. 8-10, 2014, p. 355 - 362. Available: IEEE Xplore, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7024609&newsearch=true&queryText=sql%20to%20nosql%20data%20migration . [Accessed: Feb.10, 2016].

[4] Pelica Migrator [Online] Available: https://www.mongodb.com/partners/techgene-pelica-migrator[Accessed: March 29, 2016].

[5] Robo Mongo [Online] Available: https://robomongo.org/[Accessed: March 29, 2016].

[6] Availability Overview [Online] Available: https://msdn.microsoft.com/en-us/library/aa291543(v=vs.71).aspx[Accessed: March 29, 2016].

[7] Spring Framework [Online] Available: https://projects.spring.io/spring-framework/ [Accessed: March 29, 2016].

4.2 Appendices

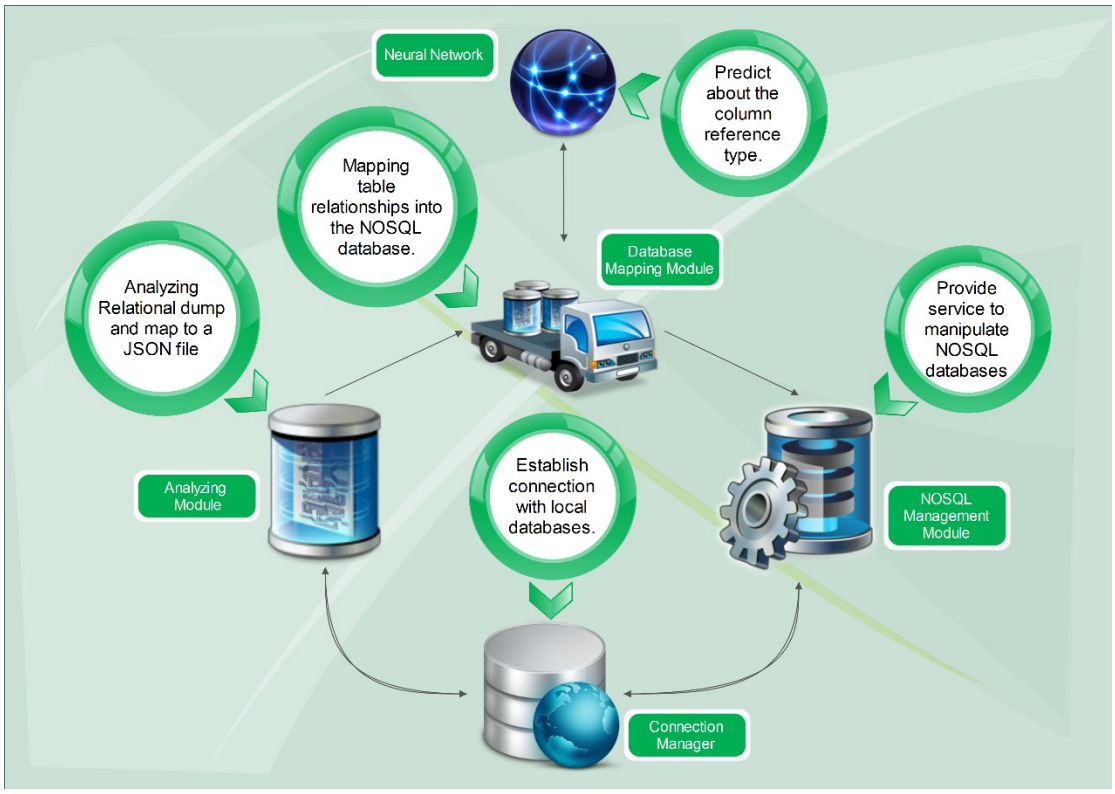

Figure 9: Mind map of MigDB System