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

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

DECLARATION

I hereby certify that the project work entitled "MigDB" submitted to the Sri Lanka Institute of Information Technology, is a record of the original work done by me under the guidance of our supervisor Dr. Anuradha Karunasena and this project work is submitted in the partial fulfillment of the requirement for the award of the degree of Bsc (Hons) degree of Information Technology. The results embodied in this document was not a copy of any document created by any organization, University any other institute of a previous student project group at SLIIT and was not copied from the internet or any other source.

……..

Madhusanka K.P.L.K

Contents

List of Tables

List of Figures

1. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description about functions, requirements and specifications of MigDB. It will explain the purpose and features of the system, the interfaces of the system, Functional and non-functional requirements and how the system will work. This document is intended for both the stakeholders and the developers of the system. This document will explain the problem that is going to be solved and how the system is used to overcome those problems. And also it provides an understanding about how the research that is being carried out throughout the project.

1.2. Scope

MigDB is divided to following sub systems for divide workload among development team.

- Connection Manager
- SQL DB Analyzer
- SQL DB Modification Evaluator
- One-to-one Mapping
- **One-to-many Mapping**
- Many-to-many Mapping
- Semantic Network for Collection Mapping
- MongoDB Collection Modifier UI
- **MongoDB Modification Evaluator**
- Conversion of Referencing to Embedding
- **Conversion of Embedding to Referencing**
- **MongoDB Data Manipulator**
- Graphical Query Generator
- SQL Query Converter

This Software Requirements Specification covers the requirements for release of Many to Many Relationship Mapper, conversion of embedded mapping to referencing, MongoDB Modification Evaluator and MongoDB Data Manipulator of MigDB.

The Many to Many Relationship Mapper will map sql many to many relationships between tables into MongoDB collections when migrating a database from SQL to MongoDB. The relationships will be mapped by embedding or referencing model depending on the decision made by Semantic Network for Collection Mapping module. With those mapping modules it will be easier to move to MongoDB from MySQL relational databases.

Conversion of embedded mapping to referencing module is a sub module under MongoDB management module. It changes the MongoDB mapping schema to referencing from embedding. It occurs with user's interaction to the existing mapping schema done by system during the migration. When an experienced user using our system and if he discovered an issue or an enchantment to the system he can change the schema and learn to the system.

MongoDB Data Manipulation Module enables users to select a document inside a collection and modify the document. Since the data will show in a table view it will be easier to manage those data. With the user friendly GUI users can add, edit and delete documents, embed new documents easily. User will not face to any difficulties when using this system as novice users.

When a user changes the existing mapping schema MongoDB Modification Evaluator evaluates those changes and verify that the modification is not violating any rule in MongoDB. If the change is violating any rule in MongoDB it will alert the user. Also it check the changes done in Data Manipulation Module and evaluate them.

1.3. Definitions, Acronyms, and Abbreviations

1.3.1. Acronyms

GUI	Graphical User Interface
RDBMS	Relational Database Management System
IDE	Integrated Development Environment
JDK	Java Development Kit
SQL	Structured Query Language
NoSQL	Not Only SQL
JSON	Java Script Object Notation

Table 1 Acronyms

1.4. Overview

The main goal of this research is to come up with a system that migrates RDBMS with its relationships and data to MongoDB and creating a MongoDB management tool which will not requires the knowledge of MongoDB and JSON. The MigDB can be divided into three main sub systems as Connection Manager, Relationship Mapper and MongoDB Manager. But for the development purposes we divided the whole system to many more components. We can list down the these component in the execution order when the system doing a migration.

- There is a connection manger to establish the connection with both MySQL and MongoDB servers
- Users can give SQL dump file which also possible without connecting to the SQL server. Therefore SQL Database Analyzer to analyze dump file and understand table structures, relationships along with data. If there is no any errors the structure and the data will be mapped to an intermediate JSON file.
- SQL database Modification Evaluator is to graphically represent the table structure that allows users to make changes to the existing structure and evaluate for any violations.
- When mapping SQL relationships into tables there are two mapping models called embedding and referencing. Therefore to decide the mapping model for each relationship the system has a semantic network which will be hosted in a cloud server. The client system can access the neural network via a restful web service.
- One to one mapper to map one to one relationships to MongoDB.

- One to many mapper to map one to many relationships considering the decisions taken by neural network.
- Many to many mapper to map many to many relationships considering the decisions taken by neural network.
- MongoDB Collection Modifier UI module is a Graphical representation of the migrated database which enables users to customize the structure.
- MongoDB Modification Evaluator to evaluate the changes done in the previous module restrict the changes that violates MongoDB rules and constraints.
- There's a module to change the existing embedded mapping to referencing and update data.
- Another module to change the existing referenced mapping to embedded mapping and update data.
- MongoDB data manipulator to handle all the data level modifications to the MongoDB with a user friendly GUI.
- Graphical Query generator to access generate MongoDB commands by an interactive GUI.
- SQL Query generator module to convert SQL queries to MongoDB commands.

The following sections, the overall description and specific requirements and supporting information will cover the product perspective, functions and operations with user interfaces of Many to Many Relationship Mapper, conversion of embedded mapping to referencing, MongoDB Modification Evaluator and MongoDB Data Manipulator of MigDB. And also it will contain user characteristics and constraints which can be useful for the end users.

2. Overall Descriptions

With the increasing maturity of NoSQL databases as well as the situation of reading more than writing on large volumes of data, many applications turn to NoSQL and pick it as data storage system. When moving to NoSQL migration of existing data to NoSQL databases was an essential task. And when someone moves to NoSQL for the first time managing NoSQL databases also becomes very difficult because of lack of experience. MigDB will provide solutions to those problems when the user select MongoDB as NoSQL database.

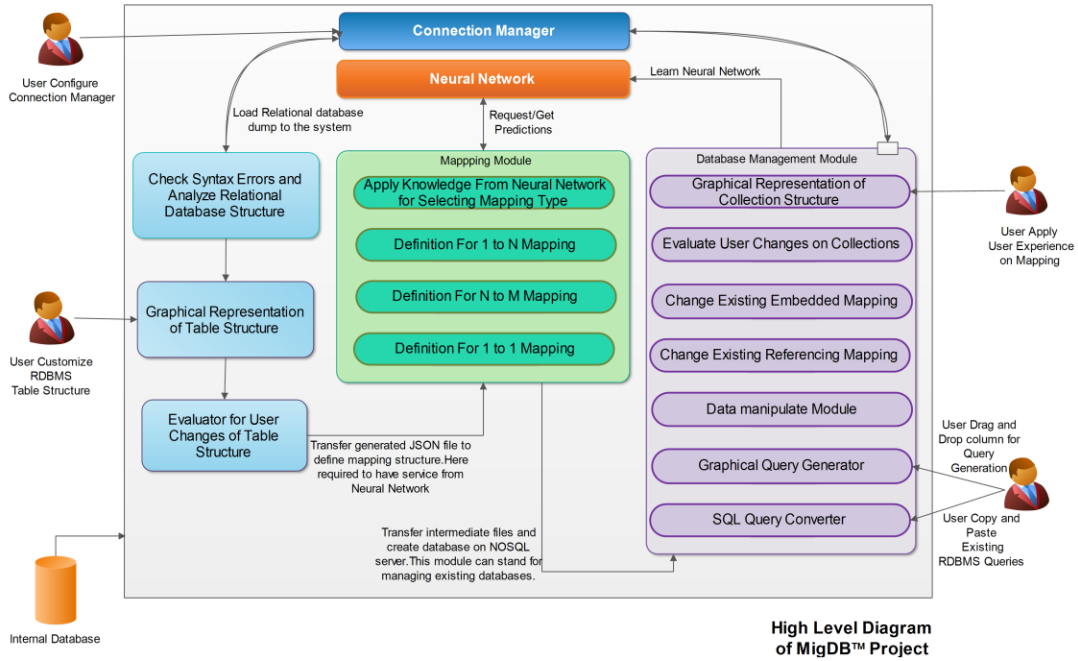

Figure 1: System Diagram

2.1. Product perspective

There are some products in the market that does database migration and database management in MongoDB. The purpose of the MigDB is to overcome problems in those existing products and to introduce new features and functionalities.

Pelica Migrator

Pelica Migrator is a database migration tool which is launched by Techgene in 2014. The tool allows users to migrate data from any RDBMS including MySQL, Oracle, SQL Server 2008, PostgreSQL, DB2 and Sybase to MongoDB. There users need to select the database that need to migrate and user can select or unselect tables as well as columns that they need to migrate. It migrated selected tables along with selected columns into documents inside collections on MongoDB. It also replicates data into the MongoDB. Pelica Migrator also provides a feature called Version Controlling which stores and displays migration selection history for future reference. But it only converts the relational database tables just as they are into collections without considering the relationships among the tables. The relationships were in the Relational database are not mapped as embedded collections or as references.

NoSQL Manager for MongoDB Professional

NoSQL Manager for MongoDB Professional is a management, administration and development tool for MongoDB with shell and GUI. The tool facilitates users to create collections, add/ modify documents within collections, add/ modify fields within documents using the GUI while also providing the ability to create functions, users and user roles. The documents or collections can be exported as CSV or JSON formats and users can back up the databases and restore. The Mongo shell is capable of auto-completing Mongo commands. This tool is also capable of migrating MS SQL or MySQL databases but only transfers tables and data just as they were without considering relationships just as Pelica Migrator.

Robo Mongo

Robo Mongo displays the collections based on three ways such as tree-mode, table mode and text-mode. It also provides the auto-completion of command. But difficult to use by novice users since it requires the knowledge on JSON and MongoDB commands.

In the market place there are some more tools available. A brief comparison of available tools along with the features with MigDB is shown in the table below.

Features	Research paper [1]	Pelica Migrator	No-SQL Manager	Robo Mongo	UMongo	Admin mongo	Query Mongo	MigDB
Evaluate data changes to the schema								✓
Migration of table structures		✓	✓					✓
Migration of data		✓	✓					✓
Selection of most appropriate relationship mapping option from embedding or referencing	(Only embedding)							✓
User interactions to mapping								✓
Graphical management tool for MongoDB				✓	✓	✓		✓
Query conversion with joining							(only select queries)	✓
Graphical query generation								✓
Does not require JSON or MongoDB knowledge		✓						✓

Table 2: Comparison with existing Products

Unlike existing migration tools the MigDB is capable of mapping Relationships within between tables in to MongoDB collections. Since it take mapping decisions from an internal neural network learned by experienced users who used the MigDB earlier the relationships will be mapped in the most efficient way considering the query processing speed and disk space consumption.

In the migration process The Many to Many Mapper will executed after one to one and one to many mappers are executed which means the intermediate JSON file is updated with one to one and one to many mappings. The Many to Many Mapper will identify the many to many relationships by reading the JSON file and send the information to neural network which need to take decisions about mapping. Then following the decision made by neural network's decision the mapper will update the JSON file.

After migrating a database it needs to be managed. For that purpose when users use an existing tool which requires background knowledge about MongoDB and JSON the novice users might be in trouble. With the MigDB management tool users will not require any background knowledge. They can manage the whole database from the GUI which will feel like using a relational database management tool.

2.1.1. System interfaces

The Many to Many Mapper will use a restful web service to access the neural network which is located in cloud for get mapping decisions. The data manipulation module will need to access MongoDB server to read and write data. The MongoDB Modification Evaluator will require to access SQLite database since it will contain the rules and constraints need to evaluate user changes.

2.1.2. User interfaces

Since the Many to Many Mapper is act as an intermediate function when migrating a database it does not have an interface. But the MongoDB Manipulation module is mostly depends on the interface. It needs to be very user friendly focusing on novice users. The data inside collections need to show in an easily understandable manner.

2.1.3. Hardware interfaces

The system can be installed on any PC or Laptop which fulfills the hardware requirements of MigDB. In order to get mapping decisions the system will require an internet connection.

2.1.4. Software interfaces

MongoDB server and MySQL server should be successfully configured in the user's device. For the developing purposes it will require Eclipse IDE, Oracle Scene Builder, and SQLite Query Browser like tools.

2.1.5. Communication interfaces

The devices internet connection will be used as the communication interface to access web services and registration purposes.

2.1.6. Memory constraints

To installation it will only require enough space to store system file which will be less than 50 MB. But when migrating a database the device need a free space around the size of the relational database.

For developing it will need minimum of 4GB ram and 10GB hard disk space

2.1.7. Operations

- Users need to register in MigDB web site in order to download the standalone application and use web services
- They need to install and configure MongoDB database in their devices and have credentials to access database.
- They need to select the database that want to migrate and make modifications before migration if necessary.
- After the migration users can change the mapping schema if they wish to change
- User can do any modification do any modification to an existing Mongo database like add, delete and update data.

2.1.8. Site adaptation requirements

- All the interfaces and documentation should be available in English language.
- Users should be able to browse the web, register and download the setup from web.

2.2. Product functions

Since MigDB includes so many various functions here will discuss functionalities only related to Many to Many Mapper, conversion of embedded mapping to referencing, MongoDB Modification Evaluator and MongoDB Data Manipulation Module

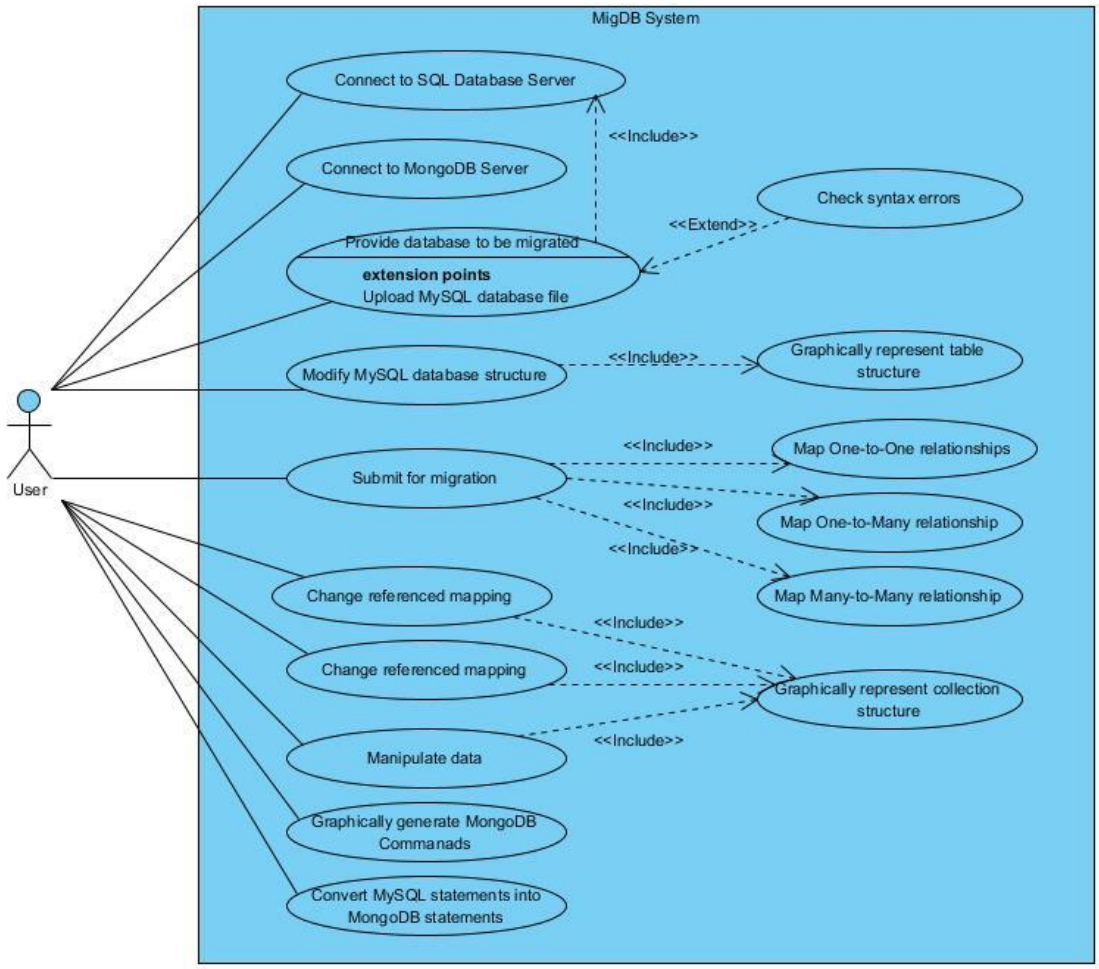

Figure 2: Use Case Diagrm 1

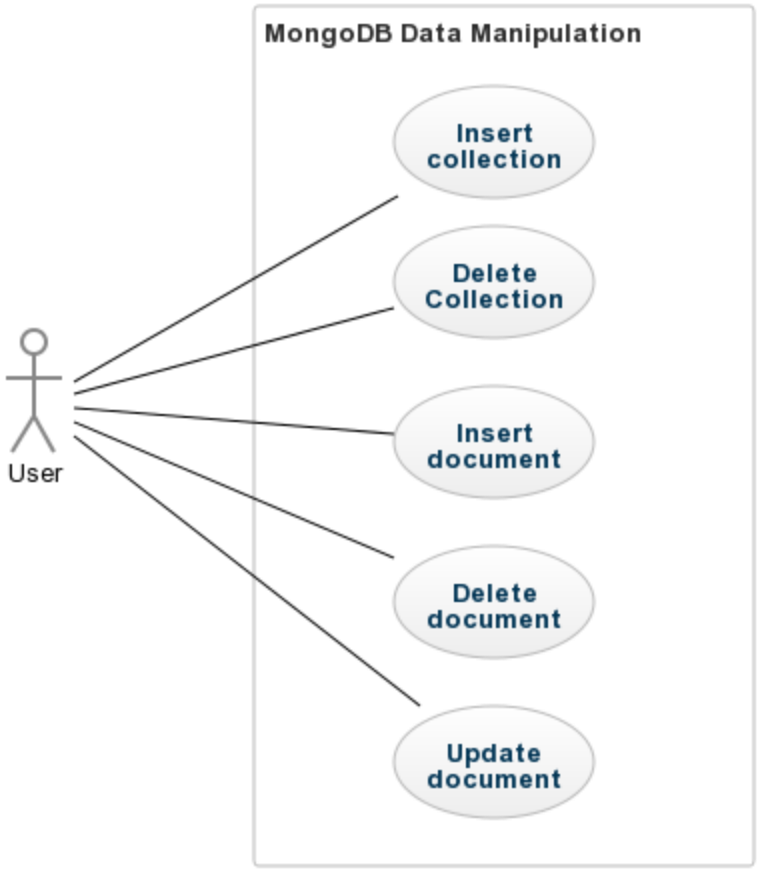

Figure 3: Use Case Diagram 2

Use Case	**Insert collection**
Pre-Conditions	Should have configured and connected Mongo Database
Actor	User
Description	1. Select the database 2. Give a name to collection 3. Submit changes
Extensions	2a. If duplicate show error Message

Table 3: Use Case Scenario 1

Use Case	Delete collection
Pre-Conditions	Should have configured and connected Mongo Database
Actor	User
Description	1. Select the database 2. Select the collection want to delete 3. Confirmation to delete 4. Submit changes
Extensions	2a. If it referenced or referencing other collections notify user 3a. if canceled do nothing

Table 4: Use Case Scenario 2

Use Case	Insert Document
Pre-Conditions	Should have configured and connected Mongo Database
Actor	User
Description	1. Select the database 2. Select the collection 3. Add data to the document 4. Confirmation to insert 5. Submit changes
Extensions	4a. if canceled do nothing

Table 5: Use Case Scenario 3

Use Case	**Delete Document**
Pre-Conditions	Should have configured and connected Mongo Database
Actor	User
Description	1. Select the database 2. Select the collection 3. Select the document that need to be deleted 4. Confirmation to delete 5. Submit changes
Extensions	5a. If it referenced or referencing other collections notify user

Table 6: Use Case Scenario 4

Use Case	**Update Document**
Pre-Conditions	Should have configured and connected Mongo Database
Actor	User
Description	1. Select the database 2. Select the collection 3. Edit the data that need to be modified 4. Confirmation to update 5. Submit changes
Extensions	5a. If the changed data referenced or referencing by other collections notify user

Table 7: Use Case Scenario 5

2.3. User characteristics

MigDB is mainly focused on novice users of Mongo databases. But it is expected to use by experienced users to learn the mapping schema by changing mapping schema. Anyone can use the system as a MongoDB learning tool.

2.4. Constraints

User need to satisfy the hardware requirements.
Installed device should have a proper internet connection.
MongoDB should successfully configured in the device.

2.5. Assumptions and dependencies

It is assumed that the user are capable to read and understand simple English used in Interfaces and having experience about computer basics.

2.6. Apportioning of requirements

Since there is no any specific client product will not be customized by any user requirements in the development stage. The developing will be followed by the same structure as the components are organized.

3. Specific requirements

3.1. External interface requirements

3.1.1. User interfaces

MigDB requires user's interaction in many steps in the migration procedure and management tool. This section will only discuss about interfaces which can be found in Many to Many Relationship Mapper, conversion of embedded mapping to referencing, MongoDB Modification Evaluator and MongoDB Data Manipulator. Among above modules user Interfaces will be found only in Mongo DB Data Manipulator.

Figure 2 is used for browsing documents inside collections. It will list down all the document with their object ids and other most commons key values. User can select a particular object and go inside to show all the attributes with that document.

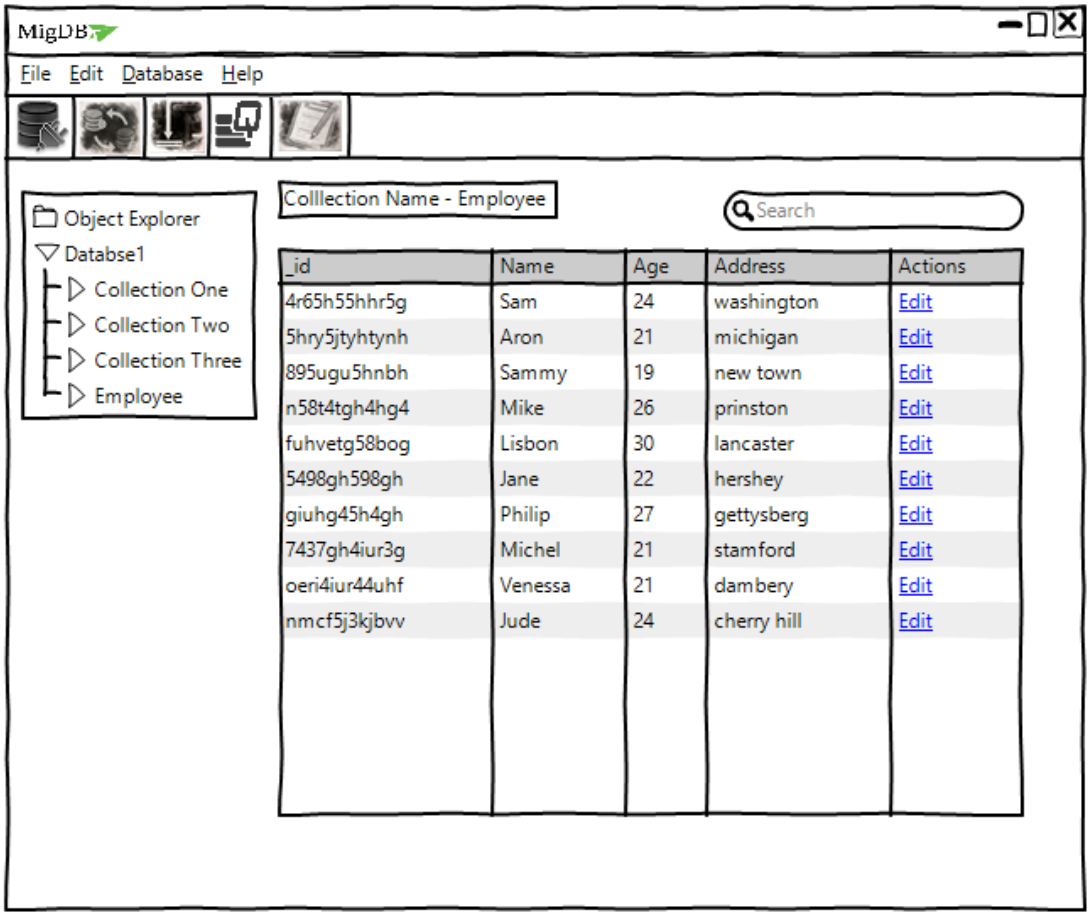

Figure 4: User Interface 1

When a user selects an object it will display the document in a format like figure 3. Here user can do all the modifications to the document easily.

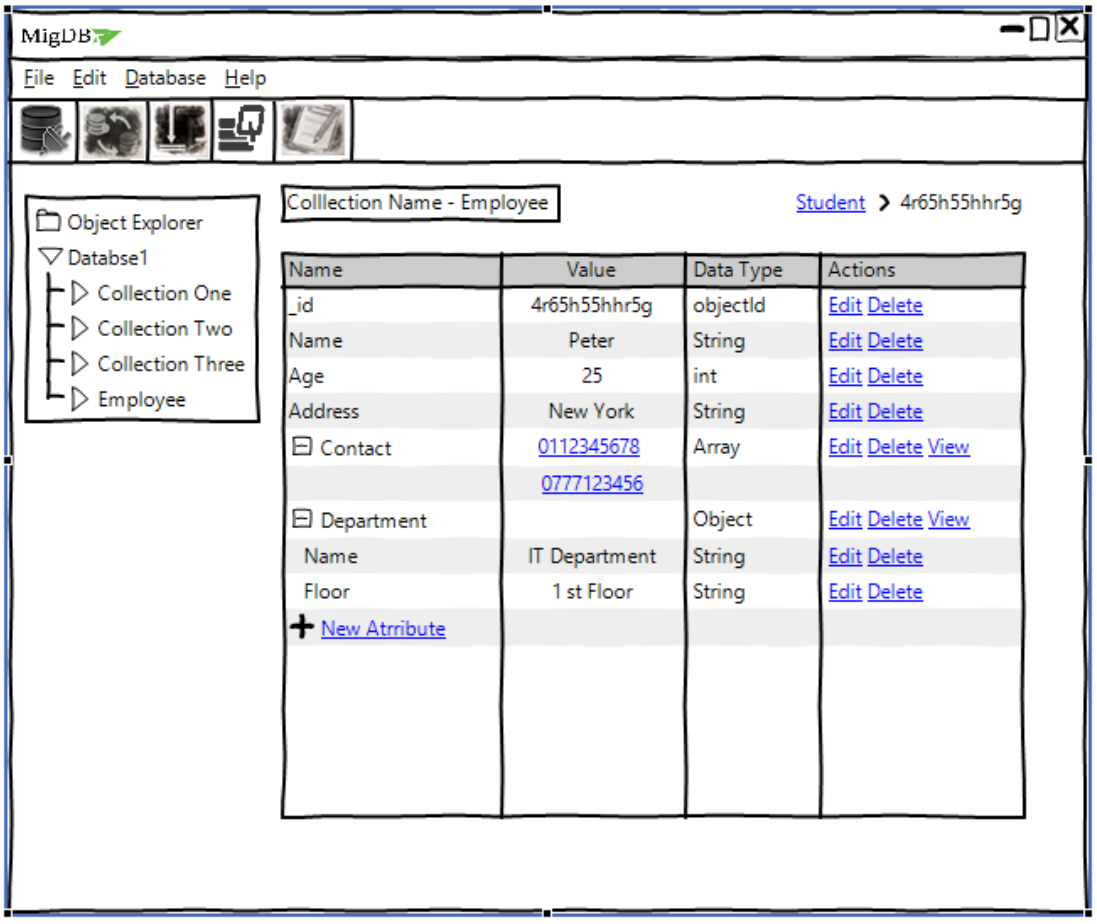

Figure 5: User Interface 2

When a user need to add an attribute to the document he can fill the following form and add the attribute.

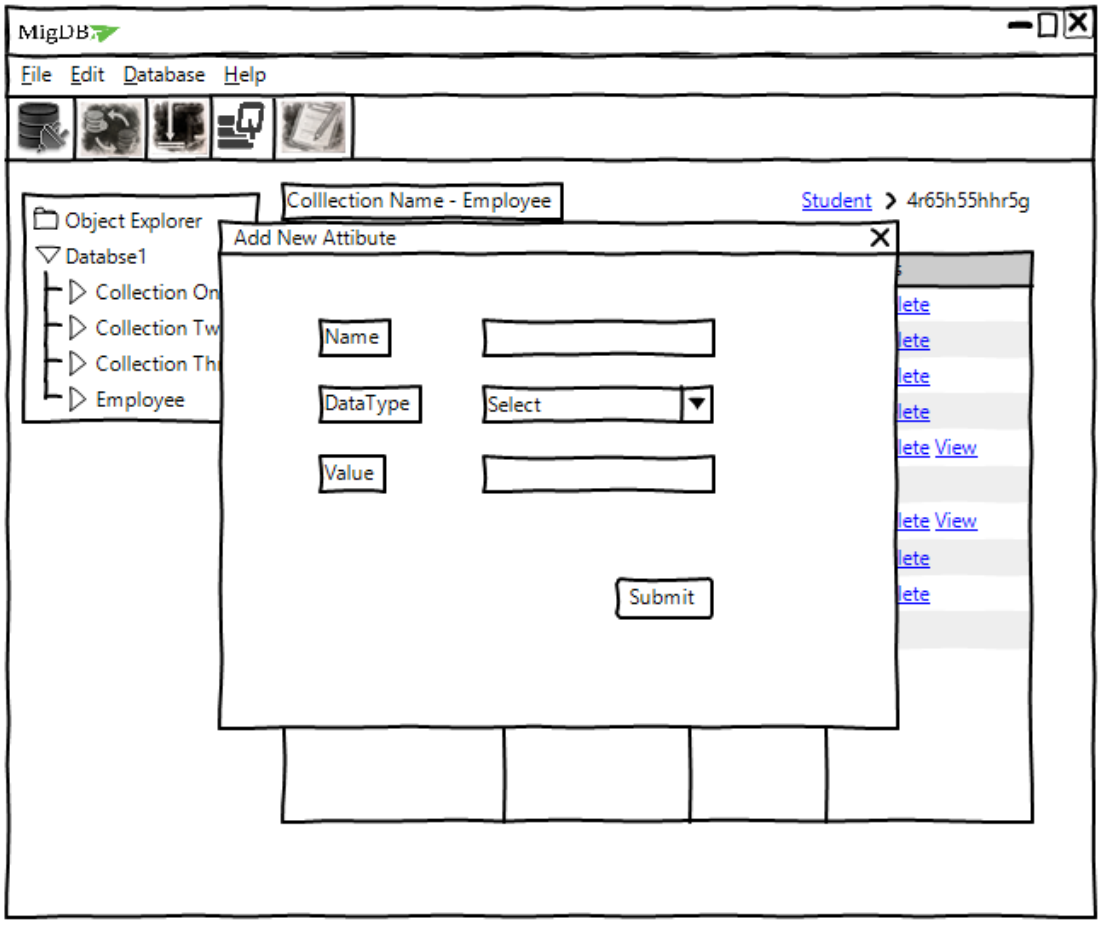

Figure 6: User Interface 3

When user deletes an attribute it will check by the evaluator

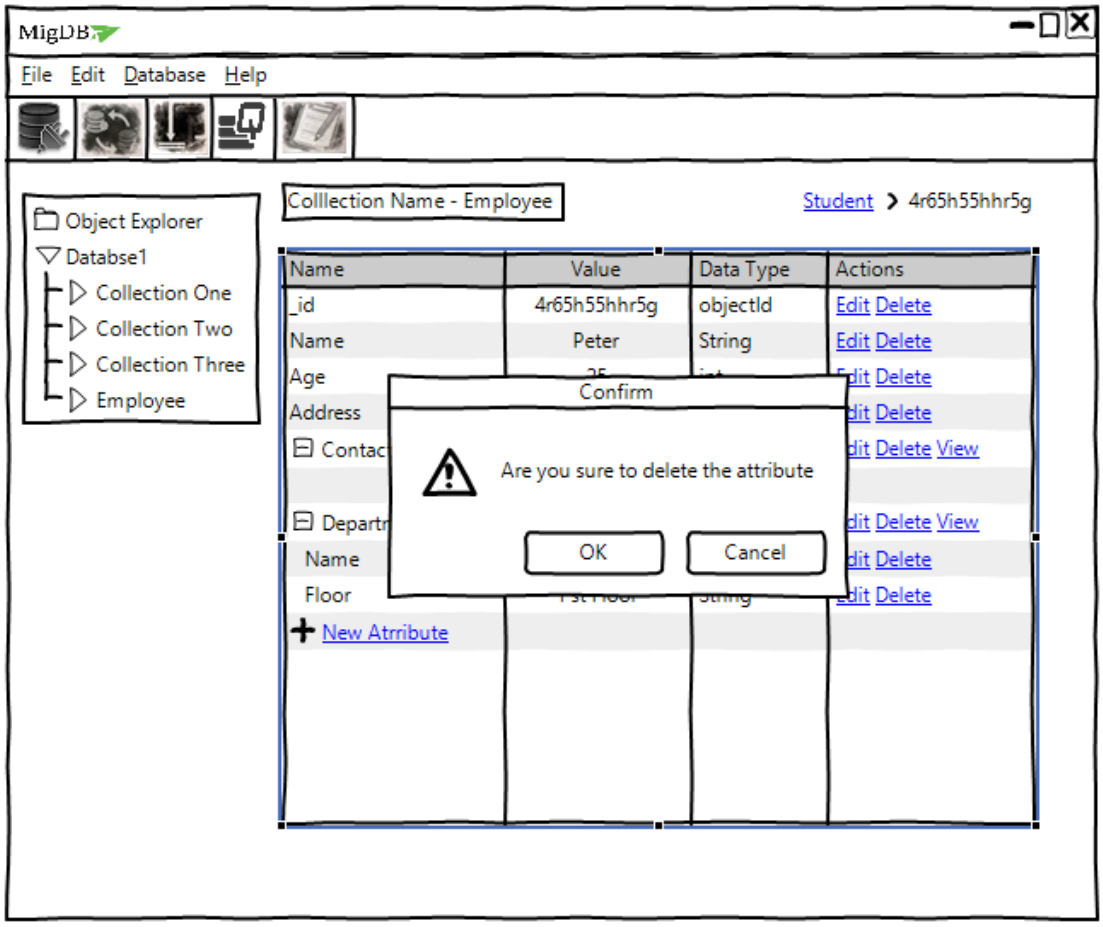

Figure 7: User Interface 4

3.1.2. Hardware interfaces

- Desktop or Laptop to run the software.
- Wi-Fi router or dongle to connect to the internet
- Web server to host the web site and neural network

3.1.3. Software interfaces

Since the MigDB standalone application is developed using java language, java JDK 8, and eclipse Mars.2 (4.5.2) IDE will need to develop software.

MongoDB should be successfully configured in order to establish the connection

SQLite query browser will need to store system configurations, rules and constraints.

3.1.4. Communication interfaces

A router or a modem will need to register, download the system.

To access web service it will need an internet connection while running the system.

3.2. Classes/Objects

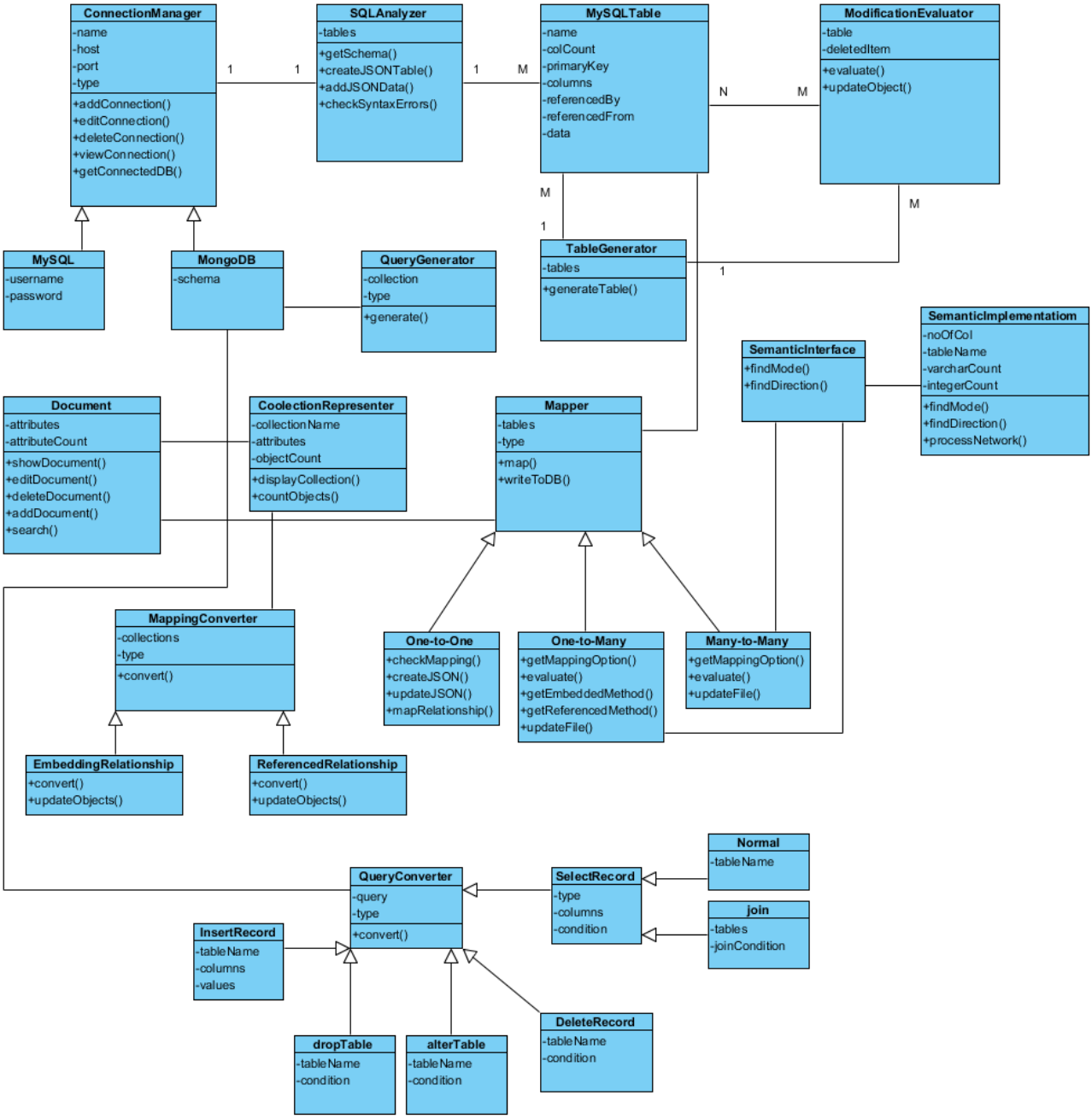

Figure 8: Class Diagram

3.3. Performance requirements

The client application is capable to handle all the operations that can be performed without freezing under required hardware and software requirements.
The cloud server can handle up to 1000 simultaneous users and handle all the operations without freezing.

3.4. Design constraints

Java should be used as the implementation language.

SQLite is used to store configurations, rules and constraints since it should be internal to the system.

The user interfaces should be very simple and easily understandable to the user.

3.5. Software system attributes

3.5.1. Reliability

The reliability of a system is typically measured as its mean time to failure (MTTF), the expected life of the system. The reliability of the MigDB is a measure of the ability to keep operating under huge databases.

3.5.2. Availability

MigDB's target is to achieve high availability around 99% while having communication interfaces with internet. However system should be able to minimize the downtime which should occur during communication.

3.5.3. Security

The MigDB website is store users passwords in an encrypted format. There will be used a secured payment gateway to handle online payments. Since the system is dealing with client's databases that has valuable data and it requires internet connection it should be much secured.

3.5.4. Maintainability

Since the client application is a stand-alone application there should be a proper version controlling in the system.

4. Supporting information

4.1. References

[1] 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].

4.2. Appendices

System Architecture

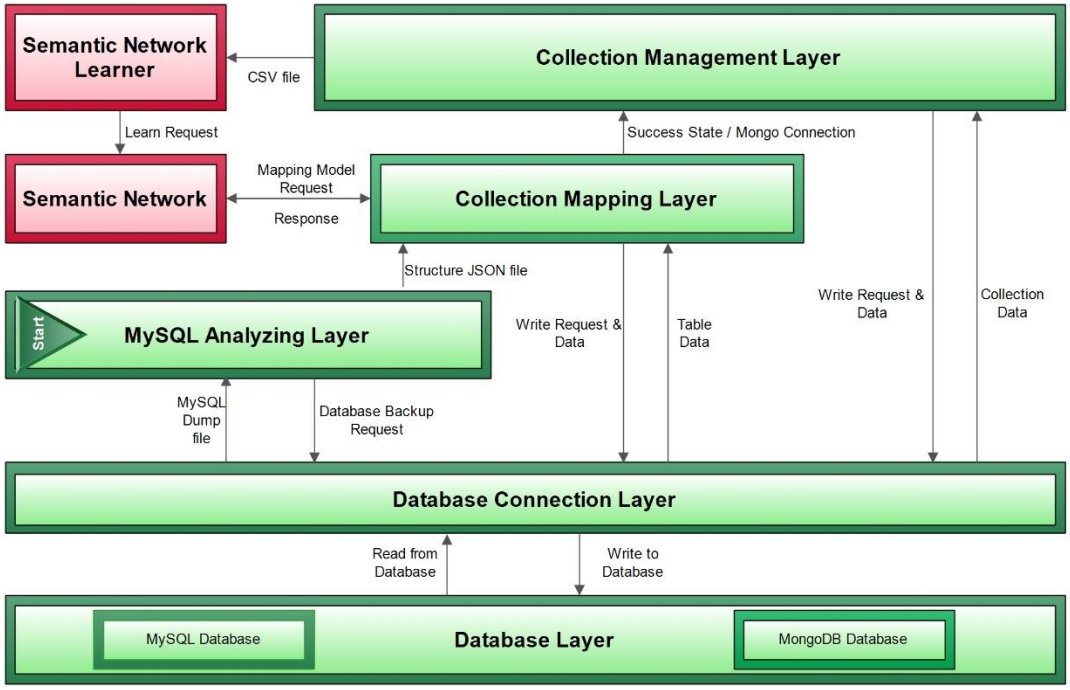

Figure 9: System Architecture