**NoSQL Tutorial**

This tutorial aims to create a java based user management application using MongoDB, a cross-platform, document oriented database that provides high performance, high availability and easy scalability. MongoDB works on concept of collection and document.

**Prerequisites**

1. MongoDB
2. MongoVUE
3. Eclipse with WindowBuilder

**Task 1 – Run MongoDB server**

1. Extract MongoDB zip file and move its contents to a location of your choice (eg: c:\mongodb).
2. MongoDB requires a data folder to store its files. You may create this directory in a location of your choice. The default location for the MongoDB data directory is c:\data\db.
3. Open up a command prompt and navigate to the bin directory in the mongodb installation folder.
4. Run mongod.exe by using the following command:

mongod.exe --dbpath "C:\mondogb\data"

This will show waiting for connections message on the console output which indicates that the mongod.exe process is running successfully.

**Task 2 – Create new database and collection for storing user data**

1. Connect to MongoDB server using MongoVUE.
2. Create new database by right clicking on the server node and selecting “Add Database”. Name the new database as “usermanager”.
3. Add new collection to usermanager database by right clicking on the database node and selecting “Add Collection”. Name the new collection as “user”.

**Task 3 – Create client application**

1. Open Eclipse and create a new java project.
   1. Select File > New > Project.
   2. Expand ‘Java’ node and select ‘Java Project’. Click ‘Next’.
   3. Name the project as ‘UserManager’ and click ‘Finish’.
2. Add MongoDB Java driver to the classpath.
   1. Right click project node and select ‘Properties’.
   2. Select ‘Java Build Path’ and go to ‘Libraries’ tab.
   3. Click on ‘Add Externam JARs’ button and browse for the jar file.
   4. Click ‘OK’ to close the window.
3. Add ‘DBManager’ singleton class for managing database connections
   1. Right click src node > New > Class.
   2. Name the new class as DBManager.
   3. Click ‘Finish’.
   4. Add the following code to the new class:

**import** java.net.UnknownHostException;

**import** com.mongodb.DB;

**import** com.mongodb.MongoClient;

**public** **class** DBManager {

**private** **static** DB *database*;

**public** **static** DB getDatabase() {

**if**(*database* == **null**) {

MongoClient mongo;

**try** {

mongo = **new** MongoClient("localhost", 27017);

*database* = mongo.getDB("usermanager");

} **catch** (UnknownHostException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** *database*;

}

}

1. Create User entity class.
   1. Add a new class and name it as ‘User’.
   2. Add the following code to the User class:

**package** lk.sliit.usermanager.model;

**public** **class** User {

**private** **int** id;

**private** String firstName;

**private** String lastName;

**private** String email;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getEmail() {

**return** email;

}

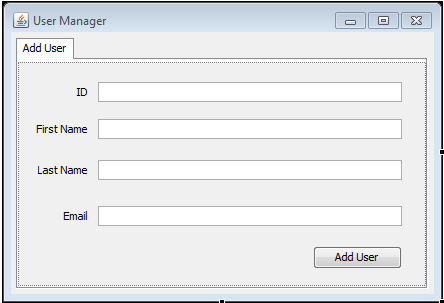
**public** **void** setEmail(String email) {

**this**.email = email;

}

}

1. Add a new jFrame to the project and name it as ‘Main’
   1. Right click on src node and select New > Other.
   2. Search for ‘jFrame’ and click ‘Next’.
   3. Name the new jFrame as ‘Main’ and click ‘Finish’.
2. Design the following user interface using the design tool



1. Code for the ‘Add User’ button is given below:

btnAddUser.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

User newUser = **new** User();

newUser.setId(Integer.*parseInt*(txtId.getText()));

newUser.setFirstName(txtFirstName.getText());

newUser.setLastName(txtLastName.getText());

newUser.setEmail(txtEmail.getText());

DBObject doc = *createDBObject*(newUser);

DB userDB = DBManager.*getDatabase*();

DBCollection col = userDB.getCollection("user");

WriteResult result = col.insert(doc);

}

});

**private** **static** DBObject createDBObject(User user) {

BasicDBObjectBuilder docBuilder = BasicDBObjectBuilder.*start*();

docBuilder.append("\_id", user.getId());

docBuilder.append("firstName", user.getFirstName());

docBuilder.append("lastName", user.getLastName());

docBuilder.append("email", user.getEmail());

**return** docBuilder.get();

}