**Faculty Research**

**User Documentation**

**Database Connectivity & Access**

Lowell Pence, [lxp3901@rit.edu](mailto:lxp3901@rit.edu)

Campbell Schweickhardt, [rcs5166@rit.edu](mailto:rcs5166@rit.edu)

Vincent Cheng, [vkc9448@rit.edu](mailto:vkc9448@rit.edu)

|  |  |  |
| --- | --- | --- |
| **Revision History** | | |
| **Version** | **Date** | **Description** |
| 1.0 | 3/28/19 | * Add a Entity Relationship Diagram * Add data dictionary |
| 2.0 | 4/4/19 | * Add definition of the API * Add UML diagrams * Add the database model * Update the data dictionary |
| 3.0 | 4/11/19 | * Update the Entity Relationship Diagram * Update the data dictionary |
| 4.0 | 4/25/19 | * Update the Entity Relationship Diagram (normalized) |
| 5.0 | 4/28/19 | * Update the API * Update the UML diagrams * Update the data dictionary |
| 6.0 | 4/29/19 | * Update the API * Update the UML diagrams * Update the Entity Relationship Diagram |

Index

[**Faculty Research Project Summary**](#_t2an8qqwhaa0) **4**

[**Entity Relationship Diagram**](#_nuiaryz7wuza) **4**

[**Data Dictionary**](#_tz1b8eq2dsnv) **5**

[**Faculty Research Library API**](#_ubyxf5zag1ur) **5**

[**UML Diagrams**](#_6dt5l4wjjtzn) **8**

## 

## Faculty Research Project Summary

Developed a database and infrastructure that would be required by a front-end developer to build a database application. The Faculty Research application will be a useful tool to keep track of what each person is working on and interests to find collaborators.

## 

## Entity Relationship Diagram

## 

## 

## 

## Data Dictionary

**Project** = A project is a table that provide abstracts information using text field that contains the information for research assignments, projects, or interest.

**ProjectKeys** = A project keys is a table that locate the abstracts information using the project ID from the project table.

**Contributors** = A contributors is a table that includes the author of the research assignment that the facility or student was involved with.

**User** = A user is a table that is use for authentication, requires the contributors (faculty/student) id and password to update, insert, or delete any abstracts information to the project table.

**UserKeys** = A user keys is a table that locate the abstracts information using the user ID from the user table.

**Keywords** = A keyword is a table that contains the keywords to locate the abstracts information for a specific research assignments or projects.

## Faculty Research Library API

The model must be able to Create, Update, and Delete only by the faculty. Reading the database will be public. These operations do the following:

|  |  |
| --- | --- |
| **Class** MYSQLDB | |
| **Responsibilities** | A data layer class to connect to the research database  Methods:  + connect() - Load the driver and connect to database  + close() - Close the database  + executeStmt() - Executes a prepared statement  + setData() - Update the data in the database  + setData(stmt) - Change data in the database using a prepared statement  + setData(sqlString) - Update data in database, does not use prepared statements  + startTrans() - Starts the transaction  + endTrans() - Ends the transaction  + rollbackTrans() - Rolls back the transaction if a failure occurs  + PrepStmt() - Execute sqlQuery faster and more secure |
| **Collaborators**  **(uses)** | java.sql.\*  java.util.ArrayList  java.util.Collections  com.mysql.cj.jdbc.Driver  research.sql  DLException.java |

|  |  |
| --- | --- |
| **Class** Users | |
| **Responsibilities** | Search for information from the database  Methods:  - resultSetFormatter() - Store projects from result set to call the various output formatters  + findUserByName() - Find users in the database by a given name  + findUsersByKeyword() - Find users in the database that are associated with a given keyword  + findUserByUsername() -Finds any user in the database by a given username  + findUserByUserType() - Finds any user in the database by a given userType  + addKeywordToUser() - List of Keywords from the database associated with projects  + findUserByProjID() - Find users given a projID from the database  + login() - Check if user is a user, privileges check from the data  + userEditPassword() - Update password information to the database |
| **Collaborators**  **(uses)** | java.util.ArrayList  java.sql.\*  ArrayList<User>  MySQLDB.java  DLException.java |

|  |  |
| --- | --- |
| **Class** User | |
| **Responsibilities** | Retrieve and add data at the prompt of the Users class.  Methods:  + getUsername() - Return the username from the database  + getName() - Return the name from the database  + getUserType() - Return the user type from the database  + setUsername() - Add the username to the database  + setName() - Add the name to the database  + setUserType() - Add the user type to the database |
| **Collaborators**  **(uses)** |  |

|  |  |
| --- | --- |
| **Class** Projects | |
| **Responsibilities** | Search for information from the database  Methods:  - resultSetFormatter() - Store projects from result set to call the various output formatters  + findProjectByName() - Find all projects in database by a given name  + findProjectByOwner() - Finds any projects in the database by the owner ID  + findProjectByID() - Finds any projects in the database by a given contributor  + findProjectByKeyword() - Finds any projects in the database by a given keyword  + findProjectByContributor() -  + addProject() |
| **Collaborators**  **(uses)** | java.sql.ResultSet  java.sql.ResultSetMetaData  java.sql.SQLException  java.util.ArrayList  ArrayList<Project>  MySQLDB.java  DLException.java |

|  |  |
| --- | --- |
| **Class** Project | |
| **Responsibilities** | Retrieve and add data at the prompt of the Projects class.  Methods:  + getProjID() - Return project ID from the database  + getProjectName() - Return the project name from the database  + getAbstractText() - Return the abstract information from the database  + getOwnerID() - Return the owner ID from the database  + setProjID - Add project ID to the database  + setProjectName() - Add project name to the database  + setContributors() - Add the contributors to the database  + setAbstractText() - Add the abstract information to the database  + setOwnerID() - Add owner ID to the database |
| **Collaborators**  **(uses)** | import java.util.ArrayList; |

## 

|  |  |
| --- | --- |
| **Class** DLException | |
| **Responsibilities** | Uses supplied error messages to log errors  Methods:  + writeLog() - Writes out all error information to a text log file (mySQL\_server\_log.txt). |
| **Collaborators**  **(uses)** | java.io.\* |

|  |  |
| --- | --- |
| **Class** Admin | |
| **Responsibilities** | The privilege to add and edit account information for the users  Methods:  + addUser() - Add user information to the database  + editPassword() - Modify the password in user table from the database  + editUserType() - Modify the user type in user table from the database  + editName() - Modify name in user table from the database  + deleteUser() - Remove user information from the database |
| **Collaborators**  **(uses)** | import java.util.ArrayList;  import java.sql.\*;  MySQLDB.java |

## 

## 

## UML Diagrams



