

---

## INSTALLATION INSTRUCTIONS

---

Application does not allow parsing another custom database, use the database hosted on AMAZON RDS.

You can:

- Search for authors based upon custom criteria options
- Retrieve the probable candidate details.
- Shortlist candidates
- Send email notifications

- **INSTALLING AND RUNNING**

- **METHOD 1**

Requirements

- JDK 1.8 or newer. It is freely available from [Oracle](<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>)

Installing and Running on Windows/Linux/Mac OS

- Download the executable jar file
- Run the following commands on the terminal:

```
$ cd path_to_folder_containing_jar_file
$ java -jar jar_file_name.jar
```

- **METHOD 2**

Downloading through GIT if above does not work

- Go to [github.ccs.neu.edu](https://github.com/ccs.neu.edu) and clone the main repository
- Open the project in eclipse on your local computer
- Go to team7/src/main/java/com/team7.
- Open LoginUI.java which contains the main
- Go to Run on the menu -> Run Configurations
- Inside Run Configurations click on the Arguments tab.
- Enter the following in the VM arguments section:  
(These arguments are used for Eclipse to handle the parsing from the huge dblp file)
- Xms1g -Xmx4g
- XX:-UseConcMarkSweepGC
- XX:+UseG1GC -XX:+UseStringDeduplication
- Click on run.

- **TECH STACK AND DEPENDENCIES**

The application uses several hooks for

- \* [MySQL] - Database hosted on cloud (Amazon RDS)

- \* [Swings] - Java GUI

- **DEVELOPMENT TEAM**

- Biyanta Shah

- Sravani Beeram

- Swayambita Das

- Vivek Malik

.....

## GENERAL ARCHITECTURE OF THE SYSTEM.

.....

The system contains three components:

1. Front End
2. Query Engine
3. User Interface

Below is the explanation for each component

### FRONT END

The front end of the system is responsible for analyzing the publication data. There are three interfaces which make up the front end of the system.

#### 1. ParseDatabase

```
public String parseXml(File xmlDataFile)
```

- *The input xml file parsed via JAXB parser to obtained the desired data*

- *The data is stored in the respective tables (Author, Paper, Article, Author\_Details and Conference)*

- *If the operation is a success it returns "success" else "failure"*

#### 2. SchemaDB

There are three functions which help in inserting data into the database

```
public void dbSetUp()
```

- *The initial skeleton for the database is set here*
- *Tables are created with appropriate column names and required indexes*

```
public Connection getConnection()
```

- *Valid username and password is set for database connection*
- *Obtains a connection to the database (Amazon RDS)*
- *The URL for connection, username and password is obtained from the db.properties in the config folder.*

```
public boolean insertData(Object object_name)
```

- *Input is a User object of the logged in user*
- *Inserts data like username, password, role and conference name into User table*

### **3. Committee**

```
public String ParseFiles(File textFile)
```

- *Parsing the files present in the committee folder and fetches the required data*
- *Stores the data in the Committee table*
- *If the operation is a success it returns "success" else "failure"*

## **QUERY ENGINE**

The query engine is responsible for executing queries against the representation that the front end creates. There are two interfaces that build up the query engine.

### **1. QueryBuilder**

```
public List<String> createQuery(List<SearchParameter> searchParameter)
```

- *The input search parameters are validated first.*
- *If they fail the validation, null is returned*
- *If they pass the validation, a SQL query is generated based on the search parameters*

- The generated SQL query is then run against our database to find the authors who satisfies the searching criteria
- The resultant list of authors is returned back

```
public boolean validateQuery(List<SearchParameter> searchParameter)
```

- The search parameters passed by user is validated here
- Checks like proper format of author name, conference name, published year are checked here
- If any of the passed parameter fails the validation, it returns a false

```
public ResultSet sendQuery(String searchQuery)
```

- The SQL query generated by createQuery() is executed in the function
- It forms a connection with the Database, executes the query and returns the obtained ResultSet

## **2. Search Display**

```
public List<String> search(List<SearchParameter> searchFilter)
```

- This expects the search parameters from the user
- These search parameters are sent to createQuery() of query engine, which validates the search parameters and returns SQL query
- The SQL query is sent to query engine for execution and the obtained result is returned back

```
public String sendEmail(Set<String> authors, String userName)
```

- Input is a list of authors and the username of the logged in user
- It sends an email to the logged in user with the list of authors in body of the mail
- If email sent successfully it returns "success" else "failure"

```
public ResultSet candidateDetails(Set<String> authors)
```

- For the list of Authors sent as an input, it calls the query engine to form a query to fetch details of the authors

- Once the query is formed, it sends the query to `sendQuery()` of query engine to execute the query
- The obtained `ResultSet` is returned back

## USER INTERFACE

The user interface is responsible for providing end users facilities for entering and executing queries and then displaying the results. There are two interfaces for the same.

### Login

```
public boolean login(String username, String password)
```

- Checks for authentication of the provided username and password
- If a valid username and password provided, then returns a true else false

```
public String logout()
```

- Logouts the user

### Register

```
public String createUser(String userName, String password, String role, String confName)
```

- Checks if the username is already used
- If username is used already, returns "exists"
- If username is proper, it checks for the password validity, if password fails validation it returns "failure"
- Else it creates a new user and inserts in the User table

```
public boolean verifyIfUserExists(String userName)
```

- Checks if the username is already registered in the system
- If present returns true, else false

.....

The flow of the entire system is as shown below:

---

## USER MANUAL

---

### VALID INPUTS FOR THE UI

- If the user does not close the pop up's then they disappear in 3 seconds.

Valid Inputs for **LOGIN**:

**Username:** shahbiyanta@gmail.com

**Password:** abc

**Output:** Will direct user to the search page

**Username:** sravs@test.com

**Password:** abc

**Output:** Will direct user to the search page

**Username:** ""

**Password:** any string

**Output:** Please enter a username

**Username:** shah@gmail.com

**Password:** ""

**Output:** Please enter a password

Any other username and password will be valid if registered through the application with valid email ID

Inputs for **REGISTER**:

**Username:** xyz@@gmail.com  
**Password:** any string  
**Output:** Enter a valid email ID

**Username:** xyz@.com  
**Password:** any string  
**Output:** Enter a valid email ID

**Username:** xyz@  
**Password:** any string  
**Output:** Enter a valid email ID

**Username:** xyzgmail.com  
**Password:** any string  
**Output:** Enter a valid email ID

**Username:** xyz@hokus.pokus  
**Password:** any string  
**Output:** Enter a valid email ID

**Username:** ""  
**Password:** any string  
**Output:** Please enter a username

**Username:** shah@gmail.com  
**Password:** ""  
**Output:** Please enter a password

**Username:** shah@gmail.com  
**Password:** abc  
**Output:** Will direct user to the search page

Inputs for **SEARCH UI**

**Input:** Year : -1  
**Output:** Please enter a valid year

**Input:** Year : 2000  
**Output :** List of authors who have published papers in 2000

**Input:** Year : 200  
**Output :** Please enter a valid year

**Input:** Year : 20000  
**Output :** Please enter a valid year

**Input :** "/blank"  
**Output :** Please enter a valid value

**Input:** CountOfPapers : One

**Output:** Please enter a valid paper count

**Input:** CountOfPapers : 1

**Output :** List of authors who have published one paper in total

#### Inputs for **DISPLAY UI**

If user does not click on any “save” then error: Please select some author to view their details

#### Inputs for **SAVED AUTHORS UI**

If user does not click on any “select” then error: Please select some author to send them an email

PAGE NAME	BUTTON	CONDITION	FUNCTION
LOGIN UI			
	Login	If credentials are valid and user is already registered	Logs user into the system and redirects to search page
		Credentials are not valid	Throws an error pop up. “Invalid Credentials: username and password”. Close and log in with valid credentials or register new user
		If user name is empty	Throws an error pop up. “Please enter username”.



		If password is empty	Throws an error pop up. "Please enter password".
	New User? Click to register	If user is not registered	Redirects user to register page
REGISTER UI			
	Register	If user does not exist in the database and email is valid.	Redirects user to search page.
		If email is invalid. Following examples of email are invalid: 1. xyz@@gmail.com 2. xyz@.com 3. xyz@ 4. xyzgmail.com 5. xyz@hokus.pokus	Throws an error pop up. "Enter a valid email ID". Close and enter valid email
		If the user exists in the database	Throws an error pop. saying user exists in the database. "Username already exists". Close pop up and go to login page.
		If user name is empty	Throws an error pop up. "Please enter username".
		If password is empty	Throws an error pop up. "Please enter password".

	Login	If user remembers being a registered user or the application reminds him	Redirects user to login page, to log into the application
SEARCH UI			
	Submit	If none of the rows are left empty	Redirects user to results page.
		If any of the search parameter rows are empty	Throws error pop up. "Please enter a value for search in row".
		If the year is String or a number != 4 numbers	Throws error pop up. "Please enter a valid year".
		If the count of papers is anything other than a number	Throws error pop up. "Please enter a valid paper count".
		If the search parameters entered, do not generate any data (such conditionals that do not exist in the database)	Throws error pop up. "Please widen your search parameters".
	Logout	If the user decides not to have any more searches	Error pop up saying to widen your search parameters since records with such conditions do not exist

DISPLAY UI			
	Search UI	If user wants some other search criteria apart from the ones already selected	Redirects user to search page. The present search results will not be saved.
	Candidate Details	If user clicks on "save" button for few authors	Redirects to Saved Authors page
		If user does not click on any "save" button and still clicks on Candidate Details	Throws an error pop. "Please select some author to view their details".
	Logout	If user decides not to have any more searches	Redirects user to login page. Ends that users 'session'.
SAVED AUTHORS UI			
	Logout	If user decides not to have any more searches	Redirects user to login page. Ends that users 'session'.
	Search UI	If user wants some other search criteria apart from the ones already selected	Redirects user to search page. The present list will not be saved.

	Send Email	If some users are selected	Will send an email with the list of the authors selected, to all users of the same conference as the 'logged-in' user. Throws a pop up "Email sent successfully"
		If no user is selected	Throw an error pop up, saying that users need to be selected to send an email