

# ResearchMate

## User's Manual

CSCI-P565 (Software Engineering I)

### Project Team

Gulshan Madhwani

Jayendra Khandare

Shubham Basu

Xinquan Wu

## 1. Product Overview

Even though with ongoing technology advancements, the researchers and PhD students from various fields don't have any online platform where they can share ideas, and look for sponsorships. Although there are some platforms for computer scientists and students who are working in some kind of computer related field, other fields don't work on these platforms. Also, they might feel outnumbered due to sheer lack of people from diverse fields. It is imperative to have people with same interests to create a collaborative environment.

With **ResearchMate**, we are trying to counter that problem. Anybody can use ResearchMate, no matter what interests them or which field of study they come from. Users can create groups with people having same interests, discuss various problems they are facing, and PhD students can connect with their advisors and other researchers who can help them excel in their fields and possibly find a sponsor.

## 2. System Requirements

### **Software Requirements -**

1. Server: For running and maintaining server, a system with following functionalities is required
  - a. Node.js (v6.11.3): Node.js is a set of libraries for JavaScript which allows it to be used outside of the browser. It is primarily focused on creating simple, easy to build network clients and servers.
  - b. mongod(v2.6.12): mongod is the primary daemon process for the MongoDB system. It handles data requests, manages data format, and performs background management operations.
  - c. mongo(v2.6.12): mongo is an interactive JavaScript shell interface to MongoDB, which provides a powerful interface for systems administrators as well as a way for developers

to test queries and operations directly with the database. mongo also provides a fully functional JavaScript environment for use with a MongoDB.

2. User-Interface: The UI is accessible from most of the updated versions of popular browsers.
  - a. Chrome: 62.0.3202.94 and above
  - b. Firefox: 38.0.1 and above
  - c. Internet Explorer: IE10 and above
3. Support for server application: There are several modules provided for node by some third parties which are essential for running servers. All are included in package.json file along with their version numbers.
4. Platform: There is no specific platform which is required for running the server as long as the requirements specified above are satisfied. Nonetheless for maintenance and debugging, a Linux system with any distribution can be used. (For development and debugging, developers used Ubuntu 16.04 LTS.)

#### Hardware Requirements –

- Operating System – Windows, Linux, MacOS
- Hard drive – Approx 50 MB for web application code. Approx 10 GB for storing files such as publication papers, user images, icons, etc.
- RAM – minimum 4 GB
- Processor – minimum Intel Dual core

### 3. Product Installation

Product installation requires the system to have following packages:

1. npm package manager: This is a tool which helps keep track of various node modules. A detailed explanation and installation guide is available at: <https://github.com/npm/npm>.
2. Once npm is installed, go under main project folder -> backend. Make sure backend folder contains package.json file (package.json file stores all the dependencies required for backend). Run “npm install” command that install all the node modules required by server.

Installation can be done by following basic steps given below:

1. Database setup:

Run a mongod daemon with following command:

Silo: "numactl --interleave=all mongod -- dbpath /u/<username>/<path>/ --port <port #>"

Where,

<username> - username of the installer / admin

<path> - the location where you want store data related to ResearchMate

<port #> - the port number with which you want to run this instance of mongod

local system: sudo mongod --dbpath <path> --port <port #>

<path> - the location where you want store data related to ResearchMate

<port #> - the port number with which you want to run this instance of mongod

2. Open /backend/server.js file and change line 3: "var portNumber = <port #>";.

Where,

<port #> - the port number where you want to run the server

By default, this port number is set to "54545".

3. Open /backend/server.js file and change line 66:

```
"var connection = mongoose.connect('mongodb://<location>:<port#>/<dbname>', {
useMongoClient: true });"
```

Where,

<location> - the location where you are running mongod instance

For silo, <location> = "silo.soic.indiana.edu"

For local system, <location> = "localhost"

<port#> - the port number set in step 1.

4. Change working directory to /backend and run command: "npm install".

Successful execution of the above command gathers all the required node modules.

5. Change working directory to /backend and run command: "node server.js".

Successful executing of this command starts the server application and shows:

"Server running at silo.soic.indiana.edu: <port #>"

which means, the server is running at port number set in step 2.

6. Successful execution of steps 1 to 5 enables the server. ResearchMate can be accessed at:
  - a. If run on local system: "<http://localhost:54545>"
  - b. If run on silo: "<http://silo.soic.indiana.edu:54545>"

## 4. Product Operation

After the successful execution of steps described in section 3 above, you can access ResearchMate at the given link in section 3.6.

[1] Signup

The image shows a 'Sign up' form overlaid on a background of a person working at a desk with a laptop and a cup of coffee. The form has the following fields and elements, each with a red arrow pointing to it and a number:

- 1: First Name input field
- 2: Last Name input field
- 3: Username input field
- 4: Email input field
- 5: Password input field
- 6: Phone Number input field
- 7: Please select carrier dropdown menu
- 8: Create Account button
- 9: Google sign up button (G logo)

- 1) First name – (Required) Allow to enter first name
- 2) Last name – (Required) Allow user to enter last name
- 3) Username – (Required) Enter a unique username
- 4) Email – (Required) Enter a valid email address to which verification email is sent
- 5) Password – (Required) Min 8 characters. Should have at least 1 lower case, 1 upper case, one digit and one special symbol
- 6) Phone – (Required) Used for dual authentication while login
- 7) Phone Carrier – (Required) Dropdown that allows user to select phone carrier such as AT&T, Verizon, etc.
- 8) Create account – Button used to create a new account
- 9) Google signup

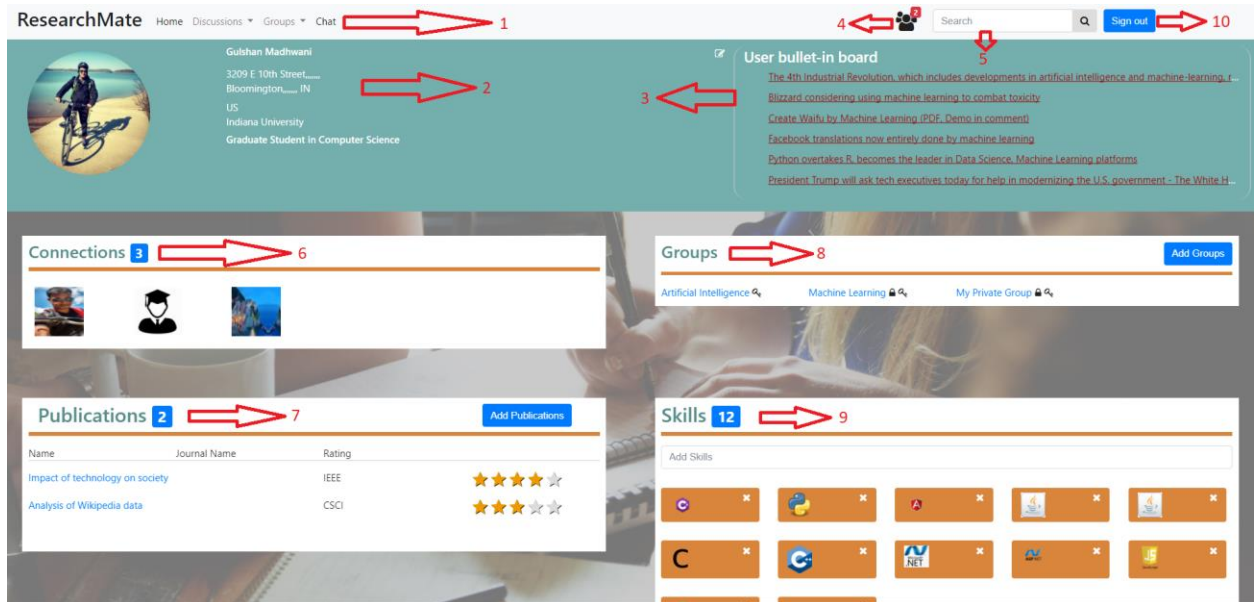
[2] Login:

The image shows a 'Login' form overlaid on the same background as the sign up form. The form has the following fields and elements, each with a red arrow pointing to it and a number:

- 1: Username input field
- 2: Password input field
- 3: Login button
- 4: Google sign in button (G logo)
- 5: Forgot username link
- 6: Forgot password link
- 7: Sign up button

- 1) Username – (required) Enter username for login
- 2) Password – (required) Enter valid password for login
- 3) Login – Button to allow the user to login
- 4) Sign in – Allow user to sign in using google authentication
- 5) Forget username – Can be used when user forgets the username
- 6) Forget password – Can be used when user forgets password
- 7) Sign up – Redirect user to sign up page

### [3] User profile



- 1) Navigation bar – allow user to navigate across website
- 2) Basic User info
- 3) User bulletin board
- 4) View pending requests
- 5) Search textbox
- 6) Connections area to display all the connections that user have.
- 7) Publications area to display user publications
- 8) Groups area to display groups user is part of.
- 9) Skills area to display user skills
- 10) Sign out – to sign out of application

### [4] Post a discussion

The screenshot shows a web form titled "What's on your mind!". It contains the following elements:

- A text input field labeled "Post:" with the placeholder text "Post a question...". A red arrow labeled "1" points to this field.
- A dropdown menu labeled "Select group". A red arrow labeled "2" points to this dropdown.
- A text input field labeled "Add tag". A red arrow labeled "3" points to this field.
- A "Tags:" label below the tag input field.
- A blue "Post" button. A red arrow labeled "4" points to this button.

- 1) Question to be posted
- 2) Group in which discussion is to be posted. By default, it is not posted to any group (posted in common discussion page)
- 3) Any tags that user wants to add. Tags added on enter key press.
- 4) Post button to post the discussion.

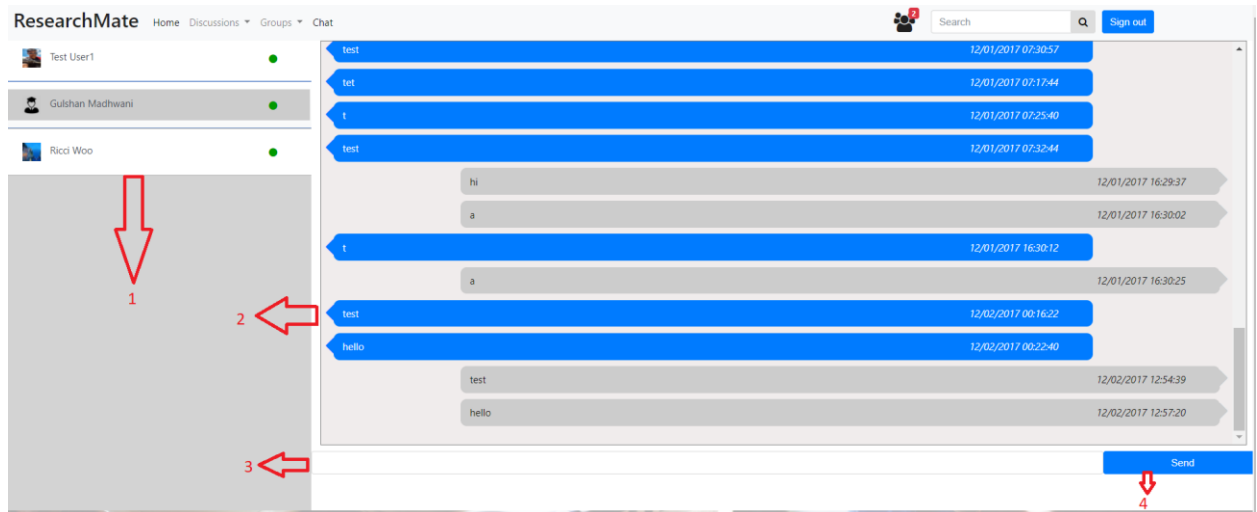
[5] Group creation

The screenshot shows a web form titled "Create New Group". It contains the following elements:

- A text input field labeled "Group Name". A red arrow labeled "1" points to this field.
- A text input field labeled "Description". A red arrow labeled "2" points to this field.
- A dropdown menu labeled "Public". A red arrow labeled "3" points to this dropdown.
- A blue "Create Group" button. A red arrow labeled "4" points to this button.

- 1) Group name (required) – desired group name
- 2) Description (optional) – desired group description
- 3) Group access level – Private or public
- 4) Create group button to create the group as per input

## [6] Chat



- 1) User online list
- 2) Past/current messages
- 3) Text box to enter message to be sent
- 4) Send button to send message to user

Detail in this section how to use your product. The format for this section is at your discretion. Assume that the software has been successfully installed. Guide the user through program start-up, describe the various modes of operation or sequence of steps to perform actions within your tool, and describe how to close down the program. Include screen shots, and detailed descriptions of the necessary steps to use your system. Assume that the audience for this artifact is not a software engineer.

## 5. Known Defects

- While chatting with multiple users, if incoming message is from different user as the active user window, the message is displayed as coming from active user.
- If publication is not uploaded while adding new, it does not show any error or progress.
- No checking for admin user. Values are hard-coded
- Sometimes search crashes the server due to undefined error.
- Update password does not have any pattern validation
- After uploading the profile picture, user needs to refresh the page to reflect changes.
- On few of the pages (eg. Groups page), button label does not show proper status. For example, when group is requested and then user refreshes page, the text changes to "join" when it should be requested.

## Revision History

Revision	Date	Change Description
Initial document	12/03/2017	

**Page Author/Creator:** [Adeel Bhutta](#)  
 Last Modified: 8/23/2016