3 Specific Requirements

3.1 Functional Requirements

3.1.1 Functional Requirement – New User Page (Matthew Redenius)

This is a web page for the user to become a registered user. A user name, password, and ODU email will be required for security. This information will be saved to the database. When a faculty user creates a profile, their email will be queried in the database against pre populated known faculty emails for authentication. Upon completion the user will be navigated to a blank “edit profile” page. The following requirements must be met:

1. Provide each user type with the ability to input information with the following fields:
   a. Username
   b. Password/Confirm Password
      i. Must be at least 8 characters
      ii. Must contain at least one capital letter
      iii. Must have at least one number
   c. Email (Must have ODU domain)

3.1.2 Update User Profile (Kevin Ashley)

This is a web page form that provides an authenticated user with the capability to edit their profile information based on their designated user type. When the user navigates to this form, the form fields will be pre-populated with data that has been previously saved to the database. The following functional requirements shall be provided:

1. Provide each user type with the ability to edit personal information with these form fields:
   a. Photo
   b. First name
   c. Last name
   d. Address
   e. City
   f. State
   g. Zip
   h. Email
   i. Phone
2. Provide the Student user type with the ability to edit education information with these form fields:
   a. Grade Level
   b. GPA
   c. Major
d. Distance Learning  
e. Courses Taken  
f. Interest Areas  
3. Provide the Faculty user type with the ability to edit education information with these form fields:  
a. Department  
b. Courses Taught  
c. Research Affiliations  
d. Interest Areas  
4. Provide the ability to save profile changes to database

3.1.3 Research Opportunity Creation (Dinah Watkins)

This is a form entry that inserts research opportunity records created by Faculty and Administrator accounts. Each record is stored in the database as an active opportunity until the deadline/expiration date has passed. A calendar based feature will assist in the designation of an opportunity being marked as expired and removed from being accessible by active searches. The following functional requirements must be met:

1. Provide the capability to create research opportunities to the following user roles:  
a. Faculty user  
b. Administrator  
2. Provide the capability to enter the following information into the fields:  
a. Title  
b. Deadline/expiration  
c. Expired  
d. Paid or unpaid  
e. Description  
f. Availability for Distance Learning students  
g. Keywords/Categories  
h. Requirements  
3. Provide the capability to store each opportunity entered into the database  
4. Pseudo auto-archival that automatically toggles the Expired field of a record once the deadline/expiration date has passed.

3.1.4 Search Opportunities Feature (Matthew Redenius)

This web page provides registered users with a display of filtered/ordered research opportunities. Upon navigation to the “Search Opportunities” page, a list of all current opportunities is displayed in order of most recently added. The user will be able to filter opportunities using keywords as well as arrange them in a desired order. All research opportunities will be retrieved from the database. The following functional requirements must be met:

1. Provide the capability to filter opportunities using a keyword:  
a. Opportunity fields that will be searched:
i. Title
ii. Description
iii. Company Name
iv. Category

2. Provide the capability to filter out unpaid opportunities
3. Provide the capability to sort opportunities by:
   a. Location
   b. Date Posted
   c. Date Ending

3.1.5 User Profile Search (Kitt Parker)

A web page form that provides a registered faculty user the capability to perform profile searches. The web form will contain data search fields that represent specific information associated with user accounts. User accounts only exist for users who have registered within the ResearchLink product. Searches may return zero to any number of results. The web form will allow the faculty user to sort the results by available numerical fields such as user id. The following functional requirements must be met:

1. Allow only registered faculty users
2. Support multiple account returns on successful searches
   a. Provide user with result page pagination of 10 to 100
   b. Provide faculty user with keyword search capability
   c. Interests
   d. Completed courses
   e. Affiliations
   f. Achievements
3. Provide faculty user with account field search capability
   a. GPA
   b. First name
   c. Last name
   d. Account type
   e. Major
   f. Grade level
   g. City
   h. State
   i. Phone

3.1.6 Web Interface Notification System (John Raha)

This notification system is a web-based message system which sends notifications to registered users. Notifications include messages related to research opportunity creation, account status changes, and information related to a user’s department. Received messages may be marked as read and deleted by the user. The following functional requirements must be met:
1. The system shall trigger a website notification when there is an email notification.
   a. Trigger a notification to students when a new opportunity has been created matching the student's profile list of interests.
   b. Trigger a notification to department members with department news.
   c. The system shall allow administrators to send notifications to all registered users.
2. Notifications will contain a text message summarizing the triggering event.
3. The notifications will be available in the user’s message center.
4. Notifications may be marked as read.
5. Notifications marked as read will be deleted.

3.1.7 Email Notification System (John Raha)

This notification system is the email-based message system which sends emails to registered users. All notifications sent to a user’s email are the same notifications displayed on the website. Users may turn off email notifications at any time. The following functional requirements must be met:

1. Trigger a notification to students when a new opportunity has been created matching the student's profile list of interests.
2. Trigger a notification to department members with department news.
3. The system shall allow administrators to send notifications to all registered users.
4. Notifications shall be sent the registered user’s ODU email address.
5. Notifications will contain a text message summarizing the triggering event.

3.1.8 News/Highlights Feature (Dinah Watkins)

This is a looping feed that displays news/highlight records created by Faculty and Administrator accounts. Each record will be pulled from the database within the allotted time frame they are considered active records. A calendar based feature will assist in archival of news/highlight records that have been active for 60 days. There will be two versions of feeds: public and internal. The following functional requirements must be met for each version:

1. Provide the capability of creating records to the following user roles:
   a. Faculty user
   b. Administrator
2. Provide the capability to submit the record to the database
3. Provide the capability to clear entries made to record being created
4. Provide spell check capability
5. Provide the capability to enter the following information into the fields:
   a. Headline
   b. Keywords/Categories
   c. Description/Content
   d. Public/Internal
   e. Date Posted
f. Expired
6. Query the database to retrieve records based on:
   a. Public/internal designation
   b. Date posted
7. Display the Public active records:
   a. Departmental
   b. Student news/highlights
8. Display the Internal active records:
   a. Departmental news/highlights
   b. Student news/highlights
   c. New research opportunities
   d. ResearchLink update news
9. Pseudo auto-archival that automatically toggles the Expired field of a record once
   60 days has passed since date posted.

3.1.9 Data Administration (Asim Amjad)
The data administration functional area must provide the security, efficiency, and
flexibility required for maintaining the data. This functional area will be capable of
controlling the general access to the system and maintaining the information contained
in the data tables. Research Link must use sixteen tables to create the database.

3.1.9.1 Research Link must use MySQL database to store all information for
research organizations, faculty, and students.
   3.1.9.1.1 academic_subjects Table
   The academic_subjects table will hold the information of the subjects
   which are offered by the ODU. Each subject must have an auto
   incremental id. The following functional requirements must be met:
   1. The attribute id must have a primary key with auto increments.
   2. The attribute name must be non-null.
   3. The attribute department_id must be a foreign key to the
departments table on id attribute.
   4. All attributes in the academic_subjects must be non-null.
   5. The following attributes must appear in the academic_subjects
table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>ID for subjects</td>
</tr>
<tr>
<td>name</td>
<td>Varchar(45)</td>
<td>Yes</td>
<td></td>
<td>Name of the subject</td>
</tr>
<tr>
<td>department_id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Foreign key with departments</td>
<td>ID for each department</td>
</tr>
</tbody>
</table>
### 3.1.9.1.2 categories Table

The categories table will hold the information of the different disciplines for one field. For example, Computer Science includes software engineering, application development, or artificial intelligence. The following functional requirements must be met:

1. The attribute id must have a primary key with auto increments.
2. The attribute name must be non-null.
3. The attribute academic_subject must be a foreign key to the academic_subjects table on id attribute.
4. All attributes in the category table must be non-null.
5. The following attributes must appear in the categories table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>ID for categories</td>
</tr>
<tr>
<td>name</td>
<td>Varchar(45)</td>
<td>Yes</td>
<td></td>
<td>Name of the category</td>
</tr>
<tr>
<td>academic_subject</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Foreign key with departments</td>
<td>ID for each subject</td>
</tr>
</tbody>
</table>

### 3.1.9.1.3 courses Table

The courses table will represent the details of the courses offered by ODU. Each course id must have auto increments. The following functional requirements must be met:

1. The attribute idcourses must have a primary key with auto increments.
2. The attribute course_number must accept a default value null.
3. The attribute academic_subject must be a foreign key to the academic_subjects table on id attribute.
4. The following attributes must appear in the courses table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>idcourses</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>ID for courses</td>
</tr>
<tr>
<td>course_number</td>
<td>Varchar(45)</td>
<td>No</td>
<td></td>
<td>Name of the course</td>
</tr>
</tbody>
</table>
3.1.9.1.4 courses_taken Table
The courses_taken table will hold the information of courses enrolled by students. There must be an auto increment id in this table. This table will use users and courses table data. The following functional requirements must be met:
1. The attribute id must have a primary key with auto increments.
2. The attribute user_id must be a foreign key to the user_id table on id attribute.
3. The attribute course_id must be a foreign key to the courses table on idcourses attribute.
4. The following attributes must appear in the courses_taken table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>ID for courses_taken</td>
</tr>
<tr>
<td>user_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with users</td>
<td>ID for user</td>
</tr>
<tr>
<td>course_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with courses</td>
<td>ID for course</td>
</tr>
</tbody>
</table>

Table 4. courses_taken Table

3.1.9.1.5 courses_taught Table
The courses_taken table will hold the information of courses taught by faculty. There must be an auto increment id in this table. This table will use users and courses table data. The following functional requirements must be met:
1. The attribute id must have a primary key with auto increments.
2. The attribute user_id must be a foreign key to the user_id table on id attribute.
3. The attribute course_id must be a foreign key to the courses table on idcourses attribute.
4. The following attributes must appear in the courses_taught table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>ID for courses_taught</td>
</tr>
<tr>
<td>user_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with users</td>
<td>ID for user</td>
</tr>
<tr>
<td>course_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with courses</td>
<td>ID for course</td>
</tr>
</tbody>
</table>

Table 5. courses_taught Table

3.1.9.1.6 departments Table
The department table will hold the information of the ODU departments. There must be an auto increment id in this table. This table will use users and courses table data. The following functional requirements must be met:

1. The attribute id must have an id with auto increments.
2. The attribute name must have a department name.
3. The following attributes must appear in the departments table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>ID for departments</td>
</tr>
<tr>
<td>name</td>
<td>Varchar(45)</td>
<td>Yes</td>
<td></td>
<td>Department Name</td>
</tr>
</tbody>
</table>

Table 6. departments Table

3.1.9.1.7 interest_areas Table

The interest_areas table will hold the information of the interest for a student, or faculty member. There must be an auto increment id in this table. This table will use users table data. The following functional requirements must be met:

1. The attribute id must have a primary key with auto increments.
2. The attribute user_id must be a foreign key to the user table on id attribute.
3. The attribute category_id must be a foreign key to the category table on category_id attribute.
4. The following attributes must appear in the interest_areas table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>ID for interest</td>
</tr>
<tr>
<td>user_id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Foreign key with users</td>
<td>ID for a user</td>
</tr>
<tr>
<td>category_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with category</td>
<td>ID for a category</td>
</tr>
</tbody>
</table>

Table 7. interest_areas Table

3.1.9.1.8 migrations Table

The migrations table will create a version control of the Research Link database. This table will keep a record of how Research Link database was created and altered over time. The changes to the database will use PHP for migration table instead of SQL. The following functional requirements must be met:

1. The attribute migration must have a non-null value.
2. The attribute batch must have a non-null value.
3. The following attributes must appear in the migrations table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>migration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>batch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.1.9.1.9 notifications Table

The notifications table will send notifications to faculty and students. An algorithm, SendNotification will notify the related users after comparing the deadline date with the current date. SendNotification will be tracked the notifications through a Boolean value which will be a false value by default. Another algorithm, AddNotifications, will add a record in the notification table after notifying the users. The following functional requirements must be met:

1. The attribute id must be a primary key with auto increments.
2. The attribute user_id must have a FK with users table on id attribute.
3. The attribute type_of_notification must accept a default value null.
4. The attribute title_html must accept a default value null.
5. The attribute body_html must use Blob datatype to store the large html file with by accepting default value null.
6. The attribute is_read must use accept a default value null.
7. The attribute created_at must use accept a default value null.
8. The attribute updated_at must use accept a default value null.
9. The following attributes must appear in the notifications table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary Key</td>
<td>Identification number for notifications</td>
</tr>
<tr>
<td>user_id</td>
<td>Int(10)</td>
<td>No</td>
<td>Foreign key with users</td>
<td>Users identification number</td>
</tr>
<tr>
<td>type_of_notifications</td>
<td>Varchar(45)</td>
<td>No</td>
<td></td>
<td>Deadline/New Opportunity</td>
</tr>
<tr>
<td>title_html</td>
<td>Varchar(100)</td>
<td>No</td>
<td></td>
<td>Title of HTML web page</td>
</tr>
<tr>
<td>body_html</td>
<td>Blob</td>
<td>No</td>
<td></td>
<td>HTML file</td>
</tr>
<tr>
<td>is_read</td>
<td>Tinyint(4)</td>
<td>No</td>
<td></td>
<td>Status of the notification</td>
</tr>
<tr>
<td>created_at</td>
<td>Timestamp</td>
<td>No</td>
<td></td>
<td>Record date for a notification creation</td>
</tr>
<tr>
<td>updated_at</td>
<td>Timestamp</td>
<td>No</td>
<td></td>
<td>Record date for an updated notification</td>
</tr>
</tbody>
</table>
Table 9. Notifications Table

3.1.9.1.10 password_resets Table
The password_resets table will update the password for users in case someone forgets a password. The following functional requirements must be met:
1. The attribute email must be non-null.
2. The attribute token must be used for token-based authentication system with a prohibited null values. This attribute must update the current password credentials too.
3. The attribute created_at must do not enter Null values.
4. The following attributes must appear in the password_resets table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>Varchar(255)</td>
<td>Yes</td>
<td>Key</td>
<td>Emails used as user ID</td>
</tr>
<tr>
<td>token</td>
<td>Varchar(255)</td>
<td>Yes</td>
<td></td>
<td>Used for authentication</td>
</tr>
<tr>
<td>created_at</td>
<td>Timestamp</td>
<td>Yes</td>
<td></td>
<td>Record time for updated password</td>
</tr>
</tbody>
</table>

Table 10. password_resets Table

3.1.9.1.11 profile Table
The profile table will create profiles for users - faculty, students, and research organizations. The following functional requirements must be met:
1. The attribute user_id must be a primary key with auto increments. This attribute must have a FK with users table on id attribute.
2. The attribute address must accept default value null.
3. The attribute city must accept default value null.
4. The attribute zipcode must accept default value null.
5. The attribute phone must accept default value null.
6. The attribute user_type must accept default value null.
7. The attribute image must accept default value null.
8. The attribute department must accept default value null.
9. The attribute zipcode must accept default value null.
10. The attribute phone must accept default value null.
11. The attribute user_type must accept default value null.
12. The attribute image must accept default value null.
13. The attribute department must accept default value null.
14. The following attributes must appear in the profile table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id</td>
<td>Int(10)</td>
<td>Yes</td>
<td>Primary Key</td>
<td>ID for users</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Foreign key with</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>users</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Null</td>
<td>Key</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------</td>
<td>------</td>
<td>-------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>address</td>
<td>Varchar(45)</td>
<td>No</td>
<td></td>
<td>Street address of user</td>
</tr>
<tr>
<td>city</td>
<td>Varchar(50)</td>
<td>No</td>
<td></td>
<td>City of user</td>
</tr>
<tr>
<td>state</td>
<td>Varchar(45)</td>
<td>No</td>
<td></td>
<td>State of user</td>
</tr>
<tr>
<td>zipcode</td>
<td>Varchar(45)</td>
<td>No</td>
<td></td>
<td>Zip code of user</td>
</tr>
<tr>
<td>image_name</td>
<td>Varchar(50)</td>
<td>No</td>
<td></td>
<td>User photo</td>
</tr>
<tr>
<td>major</td>
<td>Varchar(50)</td>
<td>No</td>
<td></td>
<td>Only student major</td>
</tr>
<tr>
<td>department</td>
<td>Varchar(45)</td>
<td>No</td>
<td></td>
<td>User department</td>
</tr>
<tr>
<td>gpa</td>
<td>Decimal(2,1)</td>
<td>No</td>
<td></td>
<td>Only student GPA</td>
</tr>
<tr>
<td>distance_learning</td>
<td>Tinyint(4)</td>
<td>No</td>
<td></td>
<td>User way of learning/teaching</td>
</tr>
<tr>
<td>grade_level</td>
<td>Varchar(45)</td>
<td>No</td>
<td></td>
<td>Only student status</td>
</tr>
<tr>
<td>updated_at</td>
<td>Timestamp</td>
<td>No</td>
<td></td>
<td>User’s profile updated date</td>
</tr>
<tr>
<td>created_at</td>
<td>Varchar(45)</td>
<td>No</td>
<td></td>
<td>User’s profile creation date</td>
</tr>
</tbody>
</table>

Table 11. profile Table

3.1.9.1.12 research_agencies Table
The research_agencies table will provide information for the research agency. The following functional requirements must be met:
1. The attribute id must be a primary key with auto increments.
2. The attribute name must accept null values.
3. The attribute image_name must accept null values.
4. The following attributes must appear in the research_agencies table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>Identification for research agency</td>
</tr>
<tr>
<td>name</td>
<td>Varchar(60)</td>
<td>No</td>
<td></td>
<td>Name of the research agency</td>
</tr>
<tr>
<td>image_name</td>
<td>Varchar(45)</td>
<td>No</td>
<td></td>
<td>Image or logo of the research agency</td>
</tr>
</tbody>
</table>

Table 12. research_agencies Table
3.1.9.1.13 research_contact Table
The research_contact table will provide contact information for the research organization in relation to a research opportunity. The following functional requirements must be met:

1. The attribute id must be a primary key with auto increments. This attribute must have FK with research_agency on id attribute.
2. The attribute opportunity_id must have FK with research_opportunities on id attribute.
3. The attribute first_name must accept null values.
4. The attribute last_name must accept null values.
5. The attribute email must accept null values.
6. The attribute phone must accept null values.
7. The following attributes must appear in the research_contact table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>ID for research agency</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Foreign key with</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>research_agency</td>
<td></td>
</tr>
<tr>
<td>opportunity_id</td>
<td>Int(11)</td>
<td>No</td>
<td>ID for an opportunity</td>
<td></td>
</tr>
<tr>
<td>first_name</td>
<td>Varchar(45)</td>
<td>No</td>
<td>Agency’s employee First Name</td>
<td></td>
</tr>
<tr>
<td>last_name</td>
<td>Varchar(45)</td>
<td>No</td>
<td>Agency’s employee second name</td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>Varchar(45)</td>
<td>No</td>
<td>Agency’s email address</td>
<td></td>
</tr>
<tr>
<td>phone</td>
<td>Varchar(45)</td>
<td>No</td>
<td>Agency’s phone number</td>
<td></td>
</tr>
</tbody>
</table>

Table 13. research_contact Table

3.1.9.1.14 research_opportunity Table
The profile table will create profiles for users - faculty, students, and research organizations. The following functional requirements must be met:

1. The attribute id must be primary key with auto increments.
2. The attribute agency_id must be a FK with research_agencies table on id attribute.
3. The attribute user_id must be a FK with users table on id attribute.
4. The attribute description must accept null values.
5. The attribute category_id must be a FK with categories table on id attribute.
6. The attribute department_id must have a FK with department table on id attribute.
7. The attribute distance_learning must accept default value null.
8. The attribute title must accept default value null.
9. The attribute type must accept default value null.
10. The attribute paid must accept default value null.
11. The attribute payment_amount must accept default value null.
12. The attribute app_start must accept default value null.
13. The attribute app_end must accept default value null.
14. The attribute research_start must accept default value null.
15. The attribute research_end must accept default value null.
16. The attribute created_by must accept default value null.
17. The attribute created_at must accept default value null.
18. The attribute updated_at must accept default value null.
19. The following attributes must appear in the profile table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>ID for research opportunity</td>
</tr>
<tr>
<td>agency_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with agencies</td>
<td>ID for research agency</td>
</tr>
<tr>
<td>user_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with users</td>
<td>ID for a user either a research agency or faculty member</td>
</tr>
<tr>
<td>description</td>
<td>Blob</td>
<td>No</td>
<td></td>
<td>Detail about a research opportunity</td>
</tr>
<tr>
<td>category_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with categories</td>
<td>ID for a category</td>
</tr>
<tr>
<td>department_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with departments</td>
<td>ID for a department</td>
</tr>
<tr>
<td>distance_learning</td>
<td>Tinyint(4)</td>
<td>No</td>
<td></td>
<td>Teach/learn from distance</td>
</tr>
<tr>
<td>title</td>
<td>Varchar(100)</td>
<td>No</td>
<td></td>
<td>Title of the person who created opportunity</td>
</tr>
<tr>
<td>type</td>
<td>Varchar(45)</td>
<td>No</td>
<td></td>
<td>Type of opportunity</td>
</tr>
<tr>
<td>paid</td>
<td>Tinyint(4)</td>
<td>No</td>
<td></td>
<td>Opportunity paid/unpaid</td>
</tr>
</tbody>
</table>
### Table 14. research_opportunity Table

3.1.9.1.15 saved_opportunities Table

The saved_opportunities table will store the saved opportunities for each user. The following functional requirements must be met:

1. The attribute id must be a primary key with auto increments.
2. The attribute opportunity_id must have a FK with research_opportunity table on id attribute.
3. The user_id must have a FK with users table on id attribute.
4. The attribute applied must accept null values.
5. The following attributes must appear in the saved_opportunities table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(11)</td>
<td>Yes</td>
<td>Primary key</td>
<td>Unique ID</td>
</tr>
<tr>
<td>opportunity_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with research_opportunity</td>
<td>ID for an opportunity</td>
</tr>
<tr>
<td>user_id</td>
<td>Int(11)</td>
<td>No</td>
<td>Foreign key with users</td>
<td>ID for users</td>
</tr>
<tr>
<td>applied</td>
<td>TinyInt(4)</td>
<td>No</td>
<td></td>
<td>Track if student has applied to opportunity</td>
</tr>
</tbody>
</table>

### Table 15. Saved_opportunities Table

3.1.9.1.16 users Table

The users table will store users in the Research Link database. The following functional requirements must be met:

1. The attribute id must be a primary key with auto increments.
2. The attribute first_name must not accept null values.
3. The attribute last_name must accept null values.
4. The attribute last_name must accept null values.
5. The attribute email must not accept null values.
6. The attribute password must have the following attributes.
   a. Password field must not accept null values.
   b. Password field must consist of at least 8 characters.
   c. Password field must have at least one number.
   d. Password field must have at least one alphabet character.
7. The attribute remember_token must accept null values.
8. The attribute confirmation_code must accept null values.
9. The attribute created_at must accept null values.
10. The attribute updated_at must accept null values.
11. The following attributes must appear in the users table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type</th>
<th>Not-Null</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int(10)</td>
<td>Yes</td>
<td>Primary key</td>
<td>ID for users</td>
</tr>
<tr>
<td>first_name</td>
<td>Varchar(255)</td>
<td>Yes</td>
<td></td>
<td>First name of user</td>
</tr>
<tr>
<td>last_name</td>
<td>Varchar(255)</td>
<td>No</td>
<td></td>
<td>Last name of user</td>
</tr>
<tr>
<td>email</td>
<td>Varchar(255)</td>
<td>Yes</td>
<td></td>
<td>Email address of user</td>
</tr>
<tr>
<td>password</td>
<td>Varchar(255)</td>
<td>Yes</td>
<td></td>
<td>Password for user</td>
</tr>
<tr>
<td>remember_token</td>
<td>Varchar(100)</td>
<td>No</td>
<td></td>
<td>Authentication for password</td>
</tr>
<tr>
<td>confirmation_code</td>
<td>Varchar(100)</td>
<td>No</td>
<td></td>
<td>Send code via email</td>
</tr>
<tr>
<td>confirmed</td>
<td>Tinyint(4)</td>
<td>No</td>
<td></td>
<td>Code status after confirmation</td>
</tr>
<tr>
<td>created_at</td>
<td>Timestamp</td>
<td>No</td>
<td></td>
<td>Account creation date</td>
</tr>
<tr>
<td>updated_at</td>
<td>Timestamp</td>
<td>No</td>
<td></td>
<td>Account updated date</td>
</tr>
</tbody>
</table>

Table 16. users Table