Testing and Documentation

All test cases will provide a failed or success message upon operation

Unit test:

Insertion:

Insertion is the process in which a user will insert data into the database

- Test case 1:
    - Check to see if user inserted correct type of data (date created, comments, title, etc.).
        - Can't put a string type into a int field (title of research into a dateCreated).
    - User is prompted to re-enter data if incorrect data is entered, before any processing occurs.
- Test case 2:
    - Check to see if user filled out all required data (no blank input fields when data is required).
- Test case 3:
    - Check to see if user has correct permissions (only researchers and system admins can upload new data into the system).
- Test case 4:
    - Check to see if data was successfully inserted (backend testing).

Deletion

- Test case 1:
    - Check to see if user has correct permissions. Only the researcher of the project can delete items from their own project.
- Test case 2:
    - Check to see if data was deleted correctly and fully in database (backend testing).

Updating

- Test case 1:
    - Check to see if user updated correct type of data(date created, comments, title, etc.).
    - Can't put a string type into a int field (title of research into a dateCreated).
- Test case 2:
    - Check to see if user has correct permission(researcher can update).
- Test case 3:
    - Check if data was deleted(backend testing).

Login/logout

- We will have code the checks for a user credentials(username/password)
    - present a success message on correct validation
    - user has to be in database
        - check to see is user is in database
    - user upon first time to website will need to register themselves
        - If user tries to login without being registered, present error message.

- Can assign themselves as an admin of the project, and can invite students/data scientist

User privilege (There will be several levels of privileges)

- System Admin-level 1 in database
  - has all privileges; reading, writing, updating, deleting, inserting, user account management, manipulating UI elements, download/upload
  - Test privileges whenever this admin tries to use these operations.
- Researcher-level 2 in database
  - has all privileges; reading, deleting, updating, inserting, writing, upload/download
  - Test privileges whenever this researcher tries to use these operations.
- Student-level 3 in database
  - has all privileges; reading, download
  - Test privileges whenever this student tries to use these operations.
- Data scientist-4 in database
  - has all privileges; reading, download
  - Test privileges whenever this data scientist tries to use these operations.

User Acceptance testing

- Give product to Goggin's and TA's, and have them determine if the product meets all the specific requirements (basically, is it acceptable).
- Specific requirements:
  - Upload/download
  - Contribute to existing
  - Save
  - Browse/search
- Have them provide feedback.

Integration Testing

- **Integration testing** is the phase in software **testing** in which individual software modules are combined and tested as a group. It occurs after unit **testing** and before validation **testing**.
- We will using the third approach, sometimes referred to as the umbrella approach, requires testing along functional data and control-flow paths. First, the inputs for functions are integrated in the bottom-up pattern discussed above. The outputs for each function are then integrated in the top-down manner. The primary advantage of this approach is the degree of support for early release of limited functionality. It also helps minimize the need for stubs and drivers. The potential weaknesses of this approach are significant, however, in that it can be less systematic than the other two approaches, leading to the need for more regression testing.
- The group will present their code and explain the changes they made.
- We will have code reviews if necessary.
- The group will then discuss if the project is ready for integration.
- Once the group agrees everything is good to go, the project leader will merge the project.

- If any conflicts happen here, they should be fixed.

Regression Testing

- **Regression testing** is a type of software **testing** that verifies that software previously developed and tested still performs correctly even after it was changed or interfaced with other software. Changes may include software enhancements, patches, configuration changes, etc.
- In computer security, a **sandbox** is a security mechanism for separating running programs. It is often used to execute untested or untrusted programs or code, possibly from unverified or untrusted third parties, suppliers, users or websites, without risking harm to the host machine or operating system.
- Regression testing will include integration testing (explained above), along with unit tests.
- Regression testing will be used so that is a bug in the application, we can easily fix it.
- A possible product we can use is called MircoFocus: www.microfocus.com/products
- To prevent conflicts, we developed a sandbox, so all testing occurs here.
    - Our testing occurs on a site which is a sub domain of the main site.
    - Once completed testing, we will integrate it with the main site.
    - Out main site is what clients see, the sub domain site is what we use and see.

Verification and validation

- Regression Testing – **Verification**
- Integration Testing – **Verification**
- User Acceptance – **Verification**
- Unit Test
    - Updating – **Validation and Verification**
    - Insertion – **Validation and Verification**
    - Deleting – **Validation**
    - Login/logout-**Verification**
    - User privilege (There will be several levels of privileges)-**Validation**

Future testing:

- Upload/download
- Browsing/searching