3. Specific Requirements

This section details the requirements to be met for the production version of A^3 framework. The requirements are grouped by function label, e.g., 3.1.x, and 3.2.x, and further subgrouped by product features and capabilities labeled, e.g., 3.1.x.y, and 3.2.x.y, where x and y represent subheadings. Each requirement is followed by a line containing the information that denotes who the Originating and Modifying authors are. The format for such information is as follows:

(O: Last Name, M1: Last Name, M2: Last Name, …, MN: Last Name)

As shown, O represents the Originating author and MX represents the Modifying Author where X is the sequence of modification from 1 to N. This notation is intended to be used solely for software configuration management purposes only. This notation allows for evaluation of the requirement from an origination and modification standpoint. Allowing evaluators of the 411W course to assign credit where it is due.

3.1. Functional Requirements

3.1.1. The GUI. The GUI will allow users to navigate through the program to perform functions defined below. The GUI provides a layout to support the functional features of A^3, displaying artifacts being stored, and user data.

3.1.1.1. The system shall allow the user to enter login credentials using username and password fields. The following requirement must be met:

1. Username - text field with no more than 20 characters.
2. Password - text field with no more than 20 characters, hidden from view.

(O: Murphy, M1: Berman)

3.1.1.2. The system shall allow the user to enter login credentials for registering a new account. The following requirement must be met:

1. Username - text field with no more than 20 characters.
2. Password - text field with no more than 20 characters, hidden from view.

(O: Berman, M1: Murphy)

3.1.1.3. The system shall allow the user to search for existing artifacts by accepting search filters in the form of tags and evaluating user attributes. The following will be evaluated before returning found artifacts:

1. Entered search filters
2. User permission/access specifications

(O: Jennings)

3.1.1.4. The system shall display artifacts according to the user's access level paired with their permissions. The following will be examined before display:

1. User access level
2. User permissions
3. Existing artifacts
4. User permitted/owned artifacts

(O: Jennings)

3.1.1.5. The system shall allow the user to update owned artifacts depending on their access level. The following will be examined before display:
1. User access level
2. Username
3. Password

(O: Oliva, M1: Murphy, M2: Ayers)

3.1.1.6. The system shall allow the user to export artifacts. The following will be examined before downloading:
1. Username
2. Password

(O: Oliva, M1: Murphy, M2: Ayers)

3.1.1.7. The system shall allow the user to upload a new artifact depending on their access level. The following will be examined before loading:
1. Username
2. Password
3. User access level
4. User permission owned artifacts

(O: Oliva, M1: Murphy, M2: Oliva)

3.1.1.8. The system shall allow the user to delete owned artifacts. The following will be examined before deleting the artifact:
1. Username
2. Password
3. User access level

(O: Oliva, M1: Murphy, M2: Oliva)

3.1.1.9. The system shall allow the user to submit URLs for web scraping. The following will be examined:
1. Username
2. Password
3. User access level

(O: Oliva, M1: Murphy, M2: Oliva)

3.1.1.10. The system shall allow the user to review success and failure reports, deletion confirmation reports and others. The following will be examined before accessing the reports:
1. Username
2. Password
3. User access level
4. User permissions

(O: Oliva, M1: Murphy, M2: Oliva)

3.1.1.11. The system shall allow the user to create a differential report to view when different commands are being used. The following will be examined before accessing the reports:
1. Username
2. Password
3. User access level

4. User permissions

(O:Oliva, M1: Murphy, M2: Oliva)

3.1.1.12. The system shall allow the user to edit/manipulate owned artifacts. The following will be examined before accessing the artifact:
1. Username
2. Password
3. User permission owned artifact

(O:Oliva, M1: Murphy, M2: Oliva)

3.1.1.13. The system shall allow the user to grant artifact access to other users to read owned artifacts. The following will be examined:
1. Username
2. Password
3. User access level
4. User permissions owned artifacts

(O:Oliva, M1: Murphy, M2: Oliva)

3.1.1.14. The system shall allow tester users to have access to all artifacts, edit/manipulate all artifacts and manipulate all users. The following will be examined:
1. List of usernames and passwords
2. List of roles
3. Users access levels
4. Users permissions owned artifacts

(O:Oliva, M1: Murphy, M2: Berman , M3: Oliva)

3.1.1.15. The system shall allow the user to view attributes of files. The following will be included under attributes:
1. Uploader username
2. Upload date and time
3. File name
4. File size
5. Original file type

(O: Ayers, M1: Berman)

3.1.1.16. The system shall allow the user to create notifications in the form of an email reminder under the following categories:
1. File update required
2. Access request

(O: Ayers, M1: Berman)

3.1.1.17. The system shall allow the user to create progress reports to track attributes changes to files through iterations. The progress report will contain attributes for each version of the requested file. Attributes will include:
1. Uploader username
2. Upload date and time
3. File name
4. File size
5. Original file type

(O: Ayers, M1: Berman)

3.1.1.18. The system shall allow tester users to analyze results and report test problems. The following will be examined:
1. Username
2. Password
3. User access level

(O:Oliva, M1: Murphy, M2: Oliva, M3: Ayers)

3.1.2. The CLI. The CLI will allow users to navigate the system to perform the functions specified below. The CLI provides command-line scripts to support the features of A^3, manipulate artifacts, and display user data.

3.1.2.1. The system shall allow a user to enter login credentials using username and password fields. The following requirements must be met:
1. Username - text field with no more than 20 characters.
2. Password - text field with no more than 20 characters, hidden

(O: Jennings)

3.1.2.2. The system shall allow a user to enter login credentials for registering a new account. The following requirement must be met:
1. Username - text field with no more than 20 characters.
2. Password - text field with no more than 20 characters, hidden

(O: Jennings)

3.1.2.3. The system shall allow users to search for existing artifacts by accepting search filters in the form of tags and evaluating user attributes. The following will be evaluated before returning found artifacts:
1. Entered search filters
2. User permission/access specifications

(O: Jennings)

3.1.2.4. The system shall display artifacts according to the user's access level paired with their permissions. The following will be examined before display:
1. User access level
2. User permissions
3. Existing artifacts
4. User permission/owned artifacts

(O: Jennings)

3.1.2.5. The system shall allow users to export public or permissed artifacts. The following will be examined before downloading:
1. User credentials
2. Artifact state
3. Artifact access level

(O: Jennings)

3.1.2.6. The system shall allow authenticated users to upload new artifacts with descriptive tags. The following will be examined before completing the upload:
1. User authentication status

2. Tag format(s)
3. Artifact state

(O: Jennings)

3.1.2.7. The system shall allow users to upload or update owned or permissed artifacts through file upload.
(O: Jennings, M1: Berman)

3.1.2.8. The system shall allow users to update owned or permissed artifacts through web scraping either ad hoc or scheduled.
(O: Jennings, M1: Berman)

3.1.2.9. The system shall allow users to delete owned artifacts.
(O: Jennings, M1: Berman)

3.1.3. The server. The server will allow the system to communicate with the UI and the database. The server will also perform some preprocessing of data for use in algorithms defined in the GUI and CLI sections.

3.1.3.1. The system shall provide a RESTful API endpoint for each function from the GUI/CLI.
(O: Berman)

3.1.3.2. The system shall communicate with the GUI/CLI using .json files.
(O: Berman)

3.1.3.3. The system shall communicate with the database using SQL.
(O: Berman)

3.1.3.4. The system shall preprocess data for use in the GUI/CLI.
(O: Berman)

3.1.3.5. The system shall be written in Python 3.8 or a later version.
(O: Berman)

3.1.3.6. The system shall be able to create a repository with specified attributes. The attributes are as follows:
1. Repository name
2. Repository visibility (public vs. private)
3. Repository tag(s) (one or more)

(O: Campbell, M1: Murphy)

3.1.3.7. The system shall be able to retrieve a file from the web when provided a specified URL.
(O: Murphy)

3.1.3.8. The system shall be able to collect a website complete artifact when provided a specified URL. Including file types as follows:
1. Java
2. CSS
3. Embedded images

(O: Murphy, M1: Murphy)

3.1.3.9. The system shall convert a supported artifact to Markdown (MD). Supported artifact types are as follows:
1. Portable Document Format (PDF)
2. Microsoft Word Document (DOC and DOCX)
3. Hypertext Markup Language (HTM and HTML)

4. Open Document Text (ODT)
5. Microsoft PowerPoint (PPT and PPTX)

(O: Berman, M1: Murphy)

3.1.3.10. The system shall be able to compare two documents, in order to show changes over time, with comparisons completed line by line.
(O: Ayers, M1: Berman, M2: Murphy)

3.1.3.11. The system shall create and store a change record upon artifact update.
(O: Jennings, M1: Berman)

3.1.4. The database.

3.1.4.1. The database shall consist of tables and fields listed in the following format:

X. Table name

X.N Field Name, Data Type (Data Type Options), Constraint (ConstraintProperties/Options)

Where X is the table and N is the column number.

1. Permission_Level
    - 1.1. Level, Integer, Primary Key
2. User
    - 2.1. User_ID, Integer (Auto_Increment), Primary Key
    - 2.2. Access_Level, Integer, Foreign Key (Reference: [Table] Permission_Level [Field] Level)
    - 2.3. Username, Characters(20 max)
    - 2.4. Password, Characters(20 max)
    - 2.5. User_Email, Characters(20 max)
    - 2.6. Last_Login, Datetime
3. Repository
    - 3.1. Repository_ID, Integer (Auto_Increment), Primary Key
    - 3.2. Repo_Creator, Integer, Foreign Key (Reference: [Table] User [Field] User_ID)
    - 3.3. Permission_Req, Integer, Foreign Key (Reference: [Table] Permission_Level [Field] Level)
    - 3.4. Repo_Name, Characters(20 max)
4. Artifact
    - 4.1. Artifact_ID, Integer (Auto_Increment), Primary Key
    - 4.2. Owner_ID, Integer, Foreign Key (Reference: [Table] User [Field] User_ID)
    - 4.3. Artifact_Repo, Integer, Foreign Key (Reference: [Table] Repository [Field] Repository_ID)
    - 4.4. Artifact_Last_Changed, Datetime, Foreign Key (Reference: [Table] Artifact_Change_Record [Field] Change_Datetime)

4.5. Artifact_Access_Level, Integer, Foreign Key (Reference: [Table] Permission_Level [Field] Level)
4.6. Artifact_Name, Characters(40 max)
4.7. Artifact_Orginal_Source, Text
4.8. Artifact_Size, Integer
4.9. Artifact_Creation_Date, Datetime
4.10. Artifact_Last_Accessed, Datetime
4.11. Artifact_Access_Count, Integer

5. Artifact_Change_Record
   5.1. Change_Datetime, Datetime, Primary Key
   5.2. Changer_ID, Integer, Foreign Key (Reference: [Table] User [Field] User_ID)
   5.3. Artifact_ID, Integer, Foreign Key (Reference: [Table] Artifact [Field] Artifact_ID)
   5.4. Artifact_Blob, Binary Large Object
   5.5. Version, Integer
6. Tag
   6.1. Tag_Name, Characters(20 max), Primary Key
   6.2. Repo_ID, Integer, Foreign Key (Reference: [Table] Repository [Field] Repository_ID)
   6.3. Artifact_ID, Integer, Foreign Key (Reference: [Table] Artifact [Field] Artifact_ID)
7. User_Bookmarks
   7.1. User_ID, Integer, Foreign Key (Reference: [Table] User [Field] User_ID)
   7.2. Artifact_ID, Integer, Foreign Key (Reference: [Table] Artifact[Field] Artifact_ID)
   7.3. Repo_ID, Integer, Foreign Key (Reference: [Table] Repository [Field] Repository_ID)

(O: Campbell, M1: Murphy, M2: Berman)

3.1.4.2. The system shall create a working database if no database exists.
(O: Berman)

3.2. Performance requirements

3.2.1. The system shall allow at least 550 concurrent users to the database. The number of 550 would simulate the equivalent of the Old Dominion University Computer Science Department, with fifty faculty and ten classes worth of students.
(O: Kennedy, M1: Berman)

3.2.2. The system shall take no more than five seconds per 1000 words in the artifact to render a line by line comparison of two artifact's contents. This includes both request and return times to/from the user.
(O: Kennedy, M1: Berman)

3.2.3. The system shall take no more than one second to render an attribute comparison between two artifacts. This includes both request and return times to/from the user.
(O: Kennedy, M1: Berman)

3.2.4. The system shall render any React based web page in no more than one second.
(O: Kennedy, M1: Berman)

3.2.5. The system shall take no more than ten seconds to return a search to the user.
(O: Murphy)

3.2.6. The system shall take no more than fifteen seconds, upon request to web scrape a page, to save a page to a file.
(O: Murphy)

[This space intentionally left blank.]

3.3. Assumptions and Constraints

Condition	Type	Effect on Requirements
Web scraper - Only well formed HTML will be accepted	Assumption	Allows for a minimal HTML parser library
In order to deploy the server and database, the user must have docker	Dependency	Allows for deployment of the server and database without any other software requirements.

3.4. Non Functional Requirements

3.4.1. Security

3.4.1.1. A randomly generated salt will be appended to each user password prior to storage and hashing in the A^3 framework database.

3.4.1.2. SHA-256 will be used to hash the combined salt and user password.

3.4.1.3. The username, salt, and hash will be stored in the user database table

3.4.1.4. Data-in-transit encryption will be achieved by enforcing user connections over HTTPS.

3.4.1.5. User permissions will be confirmed prior to all requests from that user for any restricted resource.
(These were adapted from the requirements created by Punctual Patient during the spring 2020 semester.)

3.4.2. Maintainability

No maintainability requirements will be imposed during the prototype development phase.

3.4.3. Reliability

3.4.3.1. The database shall be operational at all times.
(O: Murphy)

3.4.3.2. The database shall have a regularly performed backup process to prevent loss of data to the user.
(O: Campbell)