

Lab 2: A³ Product Specification

Stephen Ayers

CS 411W

Mr. Thomas Kennedy

July 7, 2020

Version 2

Table of Contents

1. Introduction.....	3
1.1. Purpose.....	3
1.2. Scope.....	3
1.3. Definitions, Acronyms, and Abbreviations.....	5
1.4. References.....	8
1.5. Overview.....	10
2. General Description.....	10
2.1. Prototype Architecture Description.....	10
2.2. Prototype Functional Description.....	11
2.3. Extended Interfaces.....	11
2.3.1. Hardware Interfaces.....	11
2.3.2. Software Interfaces.....	11
2.3.3. User Interfaces.....	11
2.3.4. Communication Protocols and Interfaces.....	12
3. Specific Requirements.....	12
3.1. Functional Requirements.....	12
3.2. Performance Requirements.....	12
3.3. Assumptions and Constraints.....	12
3.4. Non-Functional Requirements.....	12
Appendix.....	12

List of Figures

Figure 1 Prototype functional components diagram.....	10
---	----

List of Tables

Table 1: Features and Capabilities Comparison	
Chart.....	4

1. Introduction

A³ has been in development since early 2020, but the groundwork for the idea that inspired it is much older. A³ was the brainchild of Thomas Kennedy as a way to organize, make more readily available, and easier to update information held by Old Dominion University instructors. Throughout the development of A³ it has become more clear the need for such a product. COVID-19 has forced classrooms across the country to become remote learning institutions at the drop of a dime. Should A³ have been implemented before this pandemic took root it could have eased the stress of instructors by allowing them to more easily share resources without the inconveniences of bureaucracy. A central framework to aggregate and archive artifacts is the solution Old Dominion University has needed, but the degree of that was not clear until now.

1.1. Purpose

A³ is a repository for instructors to upload and share artifacts with students and colleagues for reference material, notify when changes have been made to their artifacts, the normalization of artifacts, the ability to compare, and the ability to tag artifacts and repositories. A³ is not a product for the creation of new artifacts, but simply assisting in the management and distribution of those already created by instructors.

1.2. Scope

The A³ prototype will be implemented over the course of the summer semester of 2020. The prototype will resemble the real world application, but certain aspects of the prototype will be limited or removed entirely. A summary of these features that will be present in the prototype as well of those that will be scaled back or removed entirely can be found below in Table 1.

Feature/Capabilities Comparison Chart		
Feature/Capability	Real World	A ³ Prototype
Database Storage	X	X
Graphical User Interface	X	Limited
Command Line Interface	X	X
User Authentication	X	Limited
Access Control	X	X
Artifact Upload	X	X
Repository Creation	X	X
Artifact Normalization	X	X
Artifact Comparison	X	X
Artifact Update	X	X
Artifact/Repo Deletion	X	
Web Scraping	X	Limited
Artifact Charge Record	X	X
Artifact Exporting	X	X
Artifact/Repo Searching	X	Limited
Artifact Contributor List	X	
Artifact/Repo Sharing	X	
Artifact/Repo Tags	X	Limited

Table 1: Features and Capabilities Comparison Chart

The most notable features being excluded are Artifact/Repo Deletion, Artifact Contributor List, and Artifact/Repo Sharing. There are 5 additional features that will be included, but only in a limited capacity. Those features are Artifact/Repo Tags, Artifact/Repo Searching, Web Scraping, User Authentication, and the GUI. Each of these elements are critical for the real world applications of A³, but the goal for the A³ prototype is to demonstrate its potential and unique characteristics that A³ brings to the market.

1.3. Definitions, Acronyms, and Abbreviations

Aggregate: Data that is composed of smaller pieces that form a larger whole.

Algorithm: Set of instructions designed to perform a specific task.

Angular: A framework for dynamic web apps. Allows for the use of HTML as a template language.

Application Programming Interface (API): Set of functions and procedures allowing the creation of applications that access features of an operating system, applications, etc.

Archive: Contains multiple files and/or folders. May be created by several different utilities and may be saved in different formats.

Artifact: Combination of arte, “by skill”, and factum, “to make”. A file or document.

Backlink: A hyperlink that links from a web page, back to your own web page or website.

Blackboard: A tool that allows faculty to add resources for students to access online.

Centralized: Type of network where all users connect to a central server.

Course Websites from Markdown (CoWeM): A system for building course websites, including notes, slides, and organizational pages, from Markdown documents.

Cascading Style Sheet (CSS): Used to format the layout of web pages. Defines text styles, table sizes, among other things that previously could only be defined in HTML.

Database: Collection of information, that is organized for rapid search and retrieval.

Data Loss: An instance in which information is destroyed by failures or neglect.

Diff: A line by line comparison of normalized artifacts.

Docker: Tool to create, deploy, and run applications by using containers. Allow developers to package up an application, with all parts needed, to be deployed in one package.

Export: Taking data from one program or computer to another.

GitLab: Used to provide internal management of git repositories. Is a self hosted Git-repository management system that keeps the user code private.

Graphical User Interface (GUI): User interface that contains graphical elements. Examples include windows, icons and buttons.

Hypertext Markup Language (HTML): A language used to create web pages. “Hypertext” refers to hyperlinks in a page, and “Markup language” refers to the way tags are used to define page layout.

Hyperlink: An element that links to another file or object.

JavaScript (JS): A language used in web development. While influenced by Java, It’s syntax is more similar to C.

Knowledge Management: The management process of creating, capturing, sharing, retrieving, and storing data, information, knowledge experiences and skills by using appropriate information and network technology.

Markdown: A markup language that can be used to format plain text. Can be converted into another language.

Markup: A language that uses tags to define elements within a document.

MySQL: Open source SQL database management system. Developed and distributed by Oracle Corporation.

Normalization: Converting ingested objects into a small number of pre-selected formats.

Python: An interpreted, object-oriented language.

Personal Learning Environment (PLE): An interface used in flexible online courses. Designed by ODU’s Center for Learning and Teaching.

pydoc: Automatically generates documentation from Python modules. Can be presented as pages of text on the console, served to a web browser, or saved to HTML files.

Pylint: A Python static code analysis tool. Looks for programming errors and warnings from within the code, as well as from an extensive configuration file.

React: A JavaScript library that is used to create User Interfaces for web applications.

reStructuredText: A plaintext markup syntax and parser system. Useful for in-line program documentation.

Secure File Transfer Protocol (SFTP): Secure version of File Transfer Protocol. Facilitates data access and data transfer over a Secure Shell data stream

Sphinx: A Python documentation generator. Converts reStructuredText files into HTML websites and other formats.

Tags: Is a keyword or term assigned to a piece of information.

tox: Aims to automate and standardize testing in Python. Is a generic virtualenv management and test command line tool.

Visual Studio Code: A source code editor that runs on Mac, Linux, and Windows.

1.4. References

- Blackboard Archive Extractor. (2016, December 15) cs.odu.edu. Retrieved March 10, 2020, from <https://www.cs.odu.edu/~cpi/old/411/crystals17/>.
- Carroll, J., Choo, C. W., Dunlap, D., Isenhour, P., Kerr, S., MacLean, A., & Rosson, M. (2003). Knowledge Management Support for Teachers. Educational Technology Research and Development, 51(4), 42-64. www.jstor.org/stable/30221184
- Davenport, T., Long, M. & Beers, M.. (1997). Building Successful Knowledge Management Projects [Working Paper]. Retrieved March 8, 2020, from https://www.researchgate.net/publication/200045855_Building_Successful_Knowledge_Management_Projects.
- Document Management Software | eFileCabinet. (2020). eFileCabinet. Retrieved February 20, 2020, from <https://www.efilecabinet.com>.
- Domes, S. (2017). Progressive Web Apps with React: Create lightning fast web apps with native power using React and Firebase. Packt Publishing Ltd.
- File Sharing and Sync For Education, Schools and Universities - FileCloud. (2020). FileCloud. Retrieved February 20, 2020, from <https://www.getfilecloud.com/file-sharing-and-sync-for-education/>.
- GitHub Features: The right tools for the job. (2020). GitHub. Retrieved March 10, 2020, from <https://github.com/features#team-management>.
- Kennedy, T. (2020, January 21). Home · Wiki · Thomas J. Kennedy / cs-roars-proposal. GitLab. Retrieved 26 April 2020, from <https://git-community.cs.odu.edu/tkennedy/cs-roars-proposal/-/wikis/home>.
- Nvlpubs.nist.gov. (n.d.). Glossary of Key Information Security Terms. From

<https://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf>.

MacFarlane, J. (2006). Pandoc - About pandoc. Pandoc.org. From <https://pandoc.org/index.html>.

Tsapps.nist.gov. (2020). Data Loss Prevention. From

https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=904672.

Xie, I., & Matusiak, K. K. (2016, July 29). Digital preservation. Science Direct (255-279).

Retrieved March 10, 2020, from

<https://www.sciencedirect.com/science/article/pii/B9780124171121000090>

Zeil, S. (2019, December 26). Building the Website. cs.odu.edu. Retrieved 26 April 2020, from

<https://www.cs.odu.edu/~zeil/cowem/Public/buildingTheWebsite/index.html>.

Zeil, S. (2020, January 21). zeil / CoWeM - Course Websites from Markdown. GitLab. From

https://git-community.cs.odu.edu/zeil/Course_Website_Management.

1.5. Overview

The A³ product specification provides an overview of the A³ Framework from a very high level. The production specification will include functional components of the system as a whole, hardware, software, and detailed features contained within the A³ Framework. The interfaces used to interact with A³ will also be included in the product specification. The interfaces will be defined by what exactly they are and how users will interact with each piece in order to use A³ to the fullest of its functionality.

2. General Description

As a whole, the A³ Framework will consist of a database, GUI, and CLI. Other aspects of the A³ Framework do exist, but are not directly visible by each user. Hidden aspects of A³ from users include Docker containers that are responsible for repositories and algorithms that normalize files, Diff files, and Authenticate users.

2.1. Prototype Architecture Description

The major functional components of A³ are all visible in Figure 1. The flow of A³ is created to start with the user on the left hand side of Figure 1. The user flows through to either the CLI or GUI depending on their preference and desired functionality. Upon authentication users will be granted access to the A³ database which will present each user with the appropriate repositories and artifacts as viewable. How users interact with artifacts be it uploading, sorting, or comparing are all done through algorithms. Figure 1 highlights algorithms with purple and green boxes with purple boxes being implemented in the prototype and green boxes being implemented in only a limited capacity.

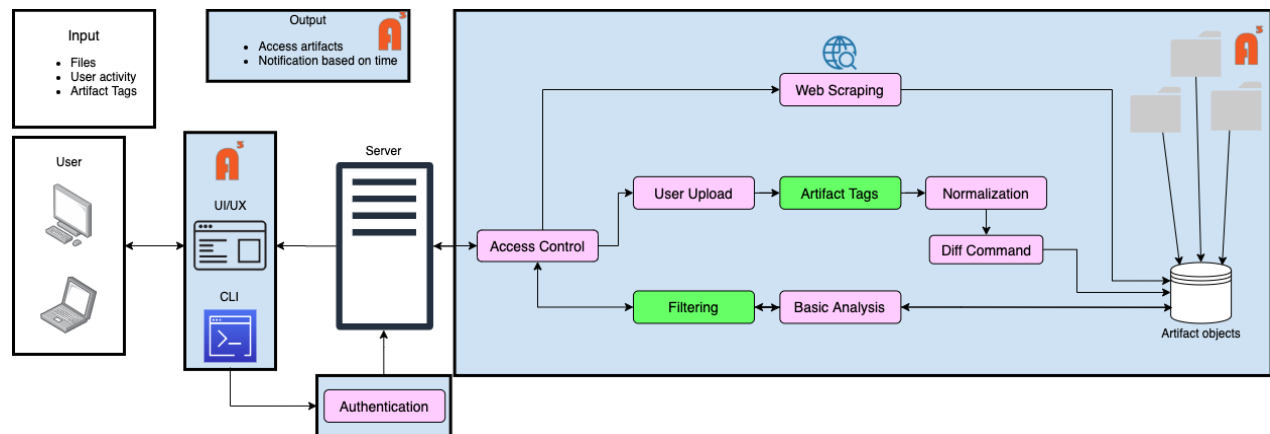


Figure 1 Prototype functional components diagram

2.2. Prototype Functional Description

The A³ prototype will mirror the core functionality and the visual layout of the real world application. The prototype will lack several of the necessary functions of a real world product in addition to having limited functionality of a number of other services. The goal of the prototype is to show the unique aspects that A³ brings to the market. Both Table 1 and Figure 1 each highlight which functionality will and will not be present within the prototype. As a whole users will be able to log into A³, upload artifacts, have them normalized, perform analysis functions on artifacts, export artifacts in various formats, and track changes over time as new versions of artifacts are submitted. All of these functions will be carried out through a CLI or a GUI. Each will be available to users to use depending on their preference and focus.

To summarize the major components that users will interact with are

- ❖ CLI/GUI - This is the actual interaction point between each user and the system as a whole. Users will give commands for A³ through either the GUI or CLI to carry out their wishes be it uploading, downloading, or updating artifacts.
- ❖ Algorithms - Each algorithm is responsible for a different aspect of the user experience. Algorithms normalize, Diff, and search the artifact database.
- ❖ Database - The database holds all repositories and artifacts for each user.

2.3. Extended Interfaces

A³ is designed to be operated on either Windows or Linux machines. The language used to craft A³ will be Python 3.8 or newer with Visual Studio Code as the primary IDE. The database will be managed via MySQL. The GUI will consist of HTML, CSS, and Javascript.

2.3.1. Hardware Interfaces

- None present

2.3.2. Software Interfaces

- Docker & Docker Compose: Docker is used to deploy the database of A³. Docker is necessary to create the containers that will contain the database that can all be deployed within a single package.

2.3.3. User Interfaces

- Keyboard: Data entry and navigation of GUI or CLI.
- Mouse: Assistance in navigation of GUI if desired.
- Monitor: Visual communication with the user will be done through a monitor.

2.3.4. Communication Protocols and Interfaces

- TCP/IP: This is the protocol used for each A³ interface to communicate with the backend resources, be it the GUI or CLI.

****Included in group section****

3. Specific Requirements**3.1. Functional Requirements****3.2. Performance Requirements****3.3. Assumptions and Constraints****3.4. Non-Functional Requirements****Appendix**