Lab 2 – A^3 framework

Mike Campbell

CS 411W

Janet Brunelle

Thomas Kennedy

24 June 2020

Version 1

Table of Contents

List of Figures

List of Tables

1. Introduction

Educational organizations employ instructors who have varied, specialized knowledge, and even instructors teaching the same course may have individual knowledge that is not shared (Brunelle). Instructors also have collections of artifacts, or educational materials, that aid them in passing on knowledge to their students. These artifacts exist in a variety of formats and on different platforms based on the instructor's preferences or needs. Sharing artifacts can often be prohibitively complicated because of formatting differences, platform differences, and time constraints (Kennedy). Beneficial artifacts may also be lost to the educational organization due to an instructor changing roles within the organization or moving on to other employment opportunities. The unaddressed need for instructors to archive and aggregate artifacts in a centralized, solutions exist for this problem that include aggregation with change tracking (Carrol, et al., 2003). A^3 seeks to solve this problem because of the considerable benefit it would provide for both the instructors and students.

1.1. Purpose

A^3 will be a framework for aggregating and archiving artifacts, providing a repository of reliable resources for academia. Not only will the artifacts be centralized for convenient access by instructors and students, they will also be normalized in order to facilitate sharing and collaboration among instructors. Additionally, A3 will allow instructors to track changes over time and provide methods to use the most up-to-date artifacts available.

A^3 will include a centralized collection of artifacts where instructors may create repositories and upload artifacts. Changes to artifacts will be tracked within the database for later review. Categorization of repositories and artifacts, for sorting and searching, will be available by user-generated content tags, and users will be able to bookmark artifacts and repositories.

Normalization of artifacts will facilitate comparison features and reduce storage use. Access will be controlled for repositories and artifacts based on user role. Analysis reporting and notifications will also be features available to users.

1.2. Scope

A^3 will be a framework designed for instructors to upload and share artifacts with students and colleagues for reference. Instructors can upload, analyze, update, and generate reports on artifacts. Artifacts and repositories can be categorized for sorting and searching , and notifications will also be configurable by instructors for their convenience.

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, its syntax is closer 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

Team Crystal. (2020, June 15). Lab 1 – A^3 Product Description. Retrieved June 24, 2020 from https://www.cs.odu.edu/~411crystal

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,il 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.

J. Brunelle, personal communication, March 2, 2020

T. Kennedy, personal communication, February 12, 2020

Domes, S. (2017). *Progressive Web Apps with React: Create lightning fast web apps with native power using React and Firebase*. Packt Publishing Ltd.

1.5. Overview

The remaining sections of this product specification detail the architecture, functions and features, interfaces, and specific requirements for A^3.

2. General Description

The A^3 prototype is a designed for faculty to create repositories to which they may upload artifacts for viewing by other faculty and guests. Faculty may also compare artifacts by simple characteristics such as file size and file type and content differences using Diff. Users with appropriate permission level—guest, faculty, or tester—can view repositories and artifacts.

2.1. Prototype Architecture Description

The A^3 prototype requires a single virtual machine running Ubuntu 18.04 on the ODU CS server. Docker and Docker Compose are used for containerization. The prototype is developed using Python 3.8 (or newer) with pydoc and Sphinx for documentation, pycodestyle and Pylint for analysis, and Visual Studio Code as the primary IDE for the development team. Figure 1 illustrates the major components for A^3.

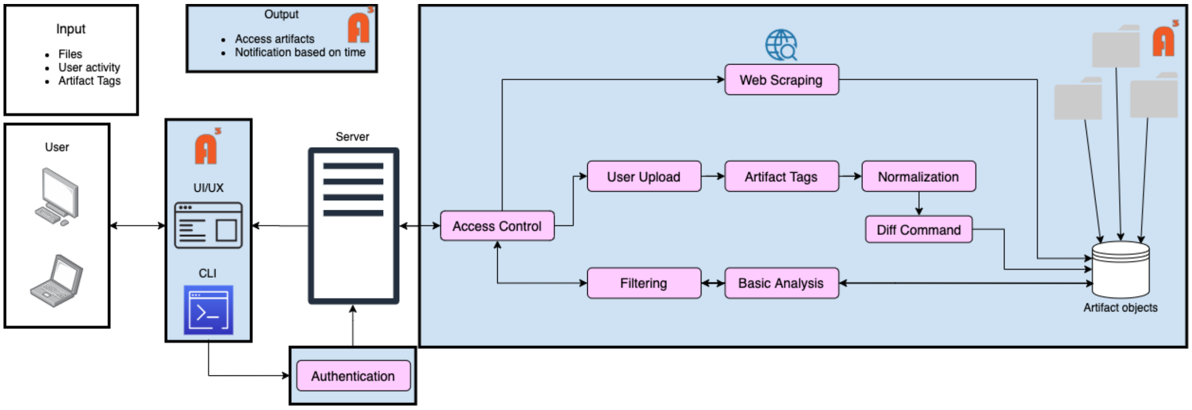


Figure 1: Major Functional Component Diagram for A^3

Version control is managed by maintaining the code repository for A^3 on GitLab, and issue tracking is also handled on the team's GitLab. The GUI is developed with HTML, CSS, and JavaScript with React framework. REST is the design pattern used to develop APIs for the prototype. The VM holds a MySQL database, created based on the schema in Figure 2, which communicates via Flask with the A^3 prototype.

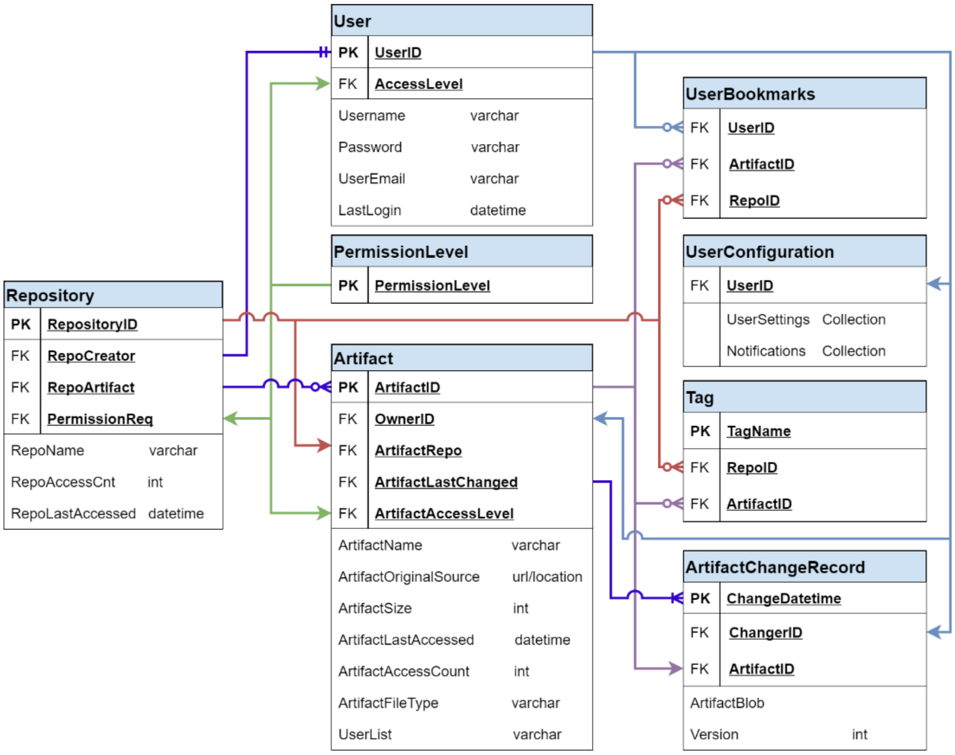
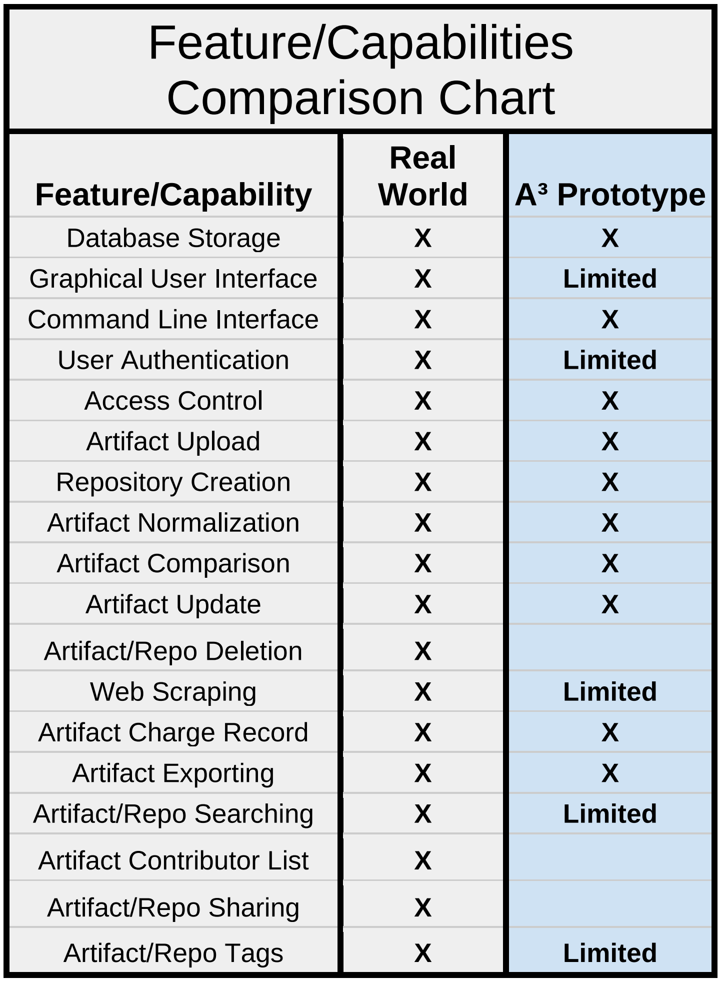


Figure 2: Database Schema for A^3

2.2. Prototype Functional Description

A^3 will contain individual repositories for each instructor, all held in a shared database so that all artifacts are accessible by the intended users. Accessing the artifacts and repositories will be controlled based on user role—e.g., guests, students, and faculty. Users will be able to bookmark artifacts and repositories they wish to access again. To add or update artifacts, instructors may use on-demand and automated uploads via SFTP and web scraping into a centralized repository. Artifacts will be normalized into Markdown format upon upload for storage and comparison. Tagging of artifacts and repositories by users for categorization will allow sorting and searching by content.

A^3 will also feature artifact comparison—comparing simple characteristics (such as artifact size, name, tags) and line-by-line differences using a specialized function, Diff. Artifact changes are tracked by preserving the version history from each change so that on-demand comparisons can be performed. A^3 will have configurable notifications for update reminders for individual artifacts and change notifications for bookmarked repositories and artifacts. Report generation for artifact and repository metrics such as number of users accessing, update frequency, and number of artifacts in repository. Table 1 shows the features included in the prototype in comparison to a real-world production.



Feature/Capabilities Comparison Chart

Feature/Capability	Real World	A³ 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: Real World vs Prototype Features Comparison

2.3. External Interfaces

The prototype will be accessible via CLI using a terminal in the ODU network or remotely through the GUI provided from the A^3 website.

3. Specific Requirements

This section is found in a separate document with the team labs.