

A3 framework: Aggregation and Achieving of Artifacts

Rosalie Oliva

Computer Science Department, ODU University

CS 411

Professor: Thomas Kennedy

June 15, 2020

Lab 2 v1

Table of Contents

| | | |
|-----|--|----|
| 1 | Introduction | 3 |
| 1.1 | Purpose | 3 |
| 1.2 | Scope | 4 |
| 1.3 | Definitions, Acronyms, and Abbreviations | 4 |
| 1.4 | References | 8 |
| 1.5 | Overview | 11 |
| 2 | General Description | 11 |
| 2.1 | Prototype Architecture Description | 11 |
| 2.2 | Prototype Functional Description | 12 |
| 2.3 | External Interfaces | 13 |

1.Introduction

Faculty members and students lack a framework to aggregate and archive artifacts. Indeed, not having a framework to aggregate provides the inability to track any changes over time. Artifacts may be abandoned; some of them may be lost because of reassignment of responsibilities. Artifacts could be isolated by specialization and their format varies. To help with this societal problem, ODU's Team Crystal have designed a framework that will aggregate and archive artifacts, track changes and supply reports. A³ is a central database with artifacts that can be useful outside of any specific specialization. However, it will help with the normalization of artifacts since none of the current competition provide this solution for students and faculty. Likewise, normalization will allow for comparison and searchability. Our central database will offer that artifacts will remain archived so they will not be lost, and any future changes may be possible.

1.1Purpose

A3 seeks to overcome the challenge of individualization, location, and formatting in academic knowledge management by keeping information available, normalized, and centralized while being enhance by a robust user interface. A3 will allow users to view changes to an artifact, A3 will link artifacts to the original source, A3 allow users to convert documents to markdown for normalization and A3 will automate the update of artifacts. A3 will not allow users to edit archived artifacts, A3 will provide server hosting, A3 will require document templates and A3 will provide continuous integration.

1.2 Scope

The A³ Prototype will be a proof of concept for students, faculties and the application functions required for the real-world product. The prototype will help relieve any risk that comes with collecting large amounts of data for the purpose of academic management.

Feedback will be provided by users and it will be a priority to satisfy customer needs.

Feedback will come from the professors, instructors, and students who will be utilizing A³.

1.3 Definitions, Acronyms, and Abbreviations

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

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.

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.

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 restructured Text 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 virtualizing 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-](https://git-community.cs.odu.edu/tkennedy/csroarsproposal/-/wikis/home)

[community.cs.odu.edu/tkennedy/csroarsproposal/-/wikis/home](https://git-community.cs.odu.edu/tkennedy/csroarsproposal/-/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.

Parenthetical citations - this part and below should not appear in any further work it is for reference only
(J. Brunelle, personal communication, March 2, 2020)

(T.Kennedy, personal communication, February 12,
2020)

Additional requirements for formatting can be found at:

<https://www.cs.odu.edu/~tkennedy/cs411/s20/Public/grammarNotes/index.html> *Additional*

requirements for style APA 7

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

1.5 Overview

This product specification provides the hardware and software configuration, external interfaces, capabilities, and features of the Product A3 prototype. The information provided in the remaining sections of this document includes a detailed description of the hardware, software, and external interface architecture of the A3 prototype; the key features of the prototype; the parameters that will be used to control, manage, or establish that feature; and the performance characteristics of that feature in terms of outputs, displays, and user interaction.

2 General Description

2.1 Prototype Architecture Description

Product A3 is comprised of the following major components:

This paragraph orients the reader to what the product (prototype) “looks like”

- Validate Method: It causes the layout manager to regenerate the window’s layout.
- Web application
- React
- JavaScript
- MySQL

2.2 Prototype functional Description

- User Authentication
- Password
- Guest Account
- Student Account
- Faculty Account
- Administrator Account

Guest View:

- Sorting filters to display artifacts
- Download artifact

Student View:

- Same as Guest plus:
- Create/Export artifact

Faculty View:

- Student/ Guest plus:

- Upload/Update artifacts
- Manipulate owned artifacts
- Grant artifacts

Administrator view:

- Guest/Student/Faculty/Administrator
- Faculty capabilities
- Access/manipulate all artifacts
- Manipulate all users

2.3 External Interfaces

2.3.1 Hardware Interfaces

- Desktop/Laptop
- Mobile/Tablet
- Internet Access

2.3.2 Software Interfaces

- HTML
- CSS
- JavaScript

2.3.3 User Interfaces

- Keyboard
- Mouse
- Monitor
- Touchscreen

2.3.4 Communications Protocols and Interfaces

- HTTPs

