

A³ framework

Aggregation and Archiving of Artifacts

1. Introduction

(Guidelines - Introduce the CS 410 product and the approach to demonstrating its characteristics through prototyping (essentially an abstract). This section should

1. *be approximately one page in length.*
2. *summarize the societal problem with some reference/date support.*
3. *outline problem and needed solution characteristics.*
4. *introduce your product by name as the solution.)*

a. Societal problem

- i. Lack of a framework to aggregate and archive artifacts
- ii. Inability to track changes over time
- iii. Cannot be considered centralized(no true aggregation tools)
- iv. Artifacts isolated by specialization
- v. Artifact format varies
- vi. Artifacts may be lost upon reassignment of responsibilities
- vii. Artifacts may be abandoned

b. Solution characteristics

- i. A framework that will aggregate and archive artifacts
- ii. Will track changes
- iii. Will supply reports
- iv. A central database
- v. Artifacts that can be useful outside of specialization

c. Solution

- i. Normalization of artifacts
- ii. Compare Artifacts
- iii. Artifacts will remain archived
- iv. Artifacts will be taggable
- v. Repositories will be taggable

2. Product Description

(Guidelines - Provide a top-level description of CS 410 product for the average reader. Provide a summary of the solution -- and its goals and objectives. This section should be one paragraph minimum.)

A³ is a repository designed for teachers to upload and share artifacts with students and colleagues for reference, notify when changes have been made to said artifacts, the normalization of artifacts, which can then be compared, and the ability to tag artifacts and repositories.

2.1. Key Features and Capabilities

(Guidelines - What does it do? What is significant/unique/innovative about it? What does it accomplish? Describe how this solves the problem.)

- a. Connects users to a single centralized repository
- b. Allows users to create their own repositories to store artifacts
- c. Allows the use of tags on both repositories and artifacts
- d. Allows users to search based on tags
- e. Allows users to search based on last update
- f. Allows users to set profiles to public or private
- g. Allows users to bookmark artifacts or repositories
- h. Allows users to see differences between artifacts with the use of normalization
- i. Notify users when something has been updated

- j. Notify users in a set arbitrary amount of time.

2.2. Major Components

(Guidelines - Provide an overview of the hardware needed to support the solution. Describe how it is structured based on CS 410 MFCD. Define and describe the software to be developed.)

a. Hardware:

- i. Computer with web-connectivity capabilities
- ii. ODU Server
- iii. Single instance server

b. Software:

- i. Language: Python 3.8 or newer
- ii. GUI language: HTML, CSS, and JS
- iii. JS frameworks: Angular and React
- iv. IDE: Visual Studio Code
- v. Documentation: pydoc and Sphinx
- vi. Code Repository: GitLab
- vii. Containerization and deployment: Docker and Docker Compose
- viii. Database: MySQL or MongoDB
- ix. API: REST
- x. Configuration management: tox and VirtualENV
- xi. Analysis: pydocstyle (formerly PEP 8) and Pylint

3. Identification of Case Study

(Guidelines - For whom is this product being developed? Why? Who else might use this in the future?)

A. Developed for:

- i. ODU Computer Science Department

B. What will it be used for:

- i. The aggregation of artifacts by teachers, for the use of students to reference

C. Who else might this benefit:

- i. Academia
- ii. Other colleges
- iii. Students
- iv. Faculty

4. A³ Product Prototype Description

(Guidelines - Provide a top-level description of the CS 411W prototype as it relates to the end product from CS 410 (i.e., the goal). Are capabilities reduced or eliminated? Simulated → modeled? Include a table of comparison between RWP and Prototype either in section 4, 4.1 or 4.2)

4.1. Prototype Architecture (Hardware/Software)

(Guidelines - How will the prototype be structured to demonstrate key features of the CS 410 product. Provide and describe the Prototype MFCD.)

- A. Any operating system (dev / use)
- B. Internet connectivity (dev / use)
- C. IDE (dev)
- D. Python, JavaScript, React Libraries (dev / use)

4.2. Prototype Features and Capabilities

(Guidelines - What does the prototype demonstrate? Why is that significant in showing how the problem is solved? How have you demonstrated success? How does the prototype address the CS 410 project risk mitigation? Describe the functional goals and objectives.)

A. Limited artifact archival and retrieval system

- a. Communal repository of artifacts
- b. User authentication (email / password)
- c. Access controlled environment
- d. CLI and GUI
- e. Artifact upload
- f. Artifact search filtering
- g. Direct resource updating
- h. Web scraped updates

4.3. Prototype Development Challenges

(Guidelines - Describe the expected challenges to be encountered while completing the prototype – e.g., knowledge missing, capability missing, supporting technology issues.)

A. Time constraints

- a. Course timeline

B. Data retrieval and storage

- a. Database
- b. Import/Export

C. Data corruption

- a. Loss

- b. Misinterpretation
- D. Workload distribution
 - a. Group size
 - b. Member contribution
- E. Desires of target audience
 - a. User abilities
 - b. User preferences

5. Glossary

(Guidelines - Alphabetical list of terms 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.

6. References

(Guidelines - Listing of sources.)

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.

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 Ltd.