

LAB 1 - A³ DESCRIPTION

Lab 1 – A³ framework

Aggregation and Archiving of Artifacts

Rosalie Oliva

CS 410

Professor J. Brunelle

4 May 2020

Table of Contents

1. Introduction.....	Error! Bookmark not defined.
2. Product Description.....	Error! Bookmark not defined.
2.1 Key Features and Capabilities.....	4
2.2 Major Components (Hardware/Software)	4
3. Identification of Case Study.....	5
4. iTracker Product Prototype Description	5
4.1 Prototype Architecture (Hardware/Software)	5
4.2 Prototype Features and Capabilities	5
4.3 Prototype Development Challenges.....	5
5. Glossary	6
6. References.....	9

1. Introduction

Based on research 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. Therefore, 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 society problem, ODU's Team Crystal have brought 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.



Fig1: Repositories and knowledge Bases

2. A³ Product Description

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

A³ will connect users to a single centralized repository. It allows users to create their own repository to store artifacts and depending in the user capability level they may be able to edit it and update it. The centralized database will allow the use of tags on both repositories and artifacts, users will be able to search based on tags or last update. Nevertheless, because sometimes users would like their artifacts not seen it will allow users to set their profiles to public or private, those that choose private will have the option to invite another user to their repositories and artifacts. Likewise, apart from guest users A³ will allow users to bookmark artifacts or repositories. It will allow users to see differences between artifacts with the use of normalization. Users will be notified when something has been updating it or when users are set in arbitrary amount of time.

2.2 Major Components

a. Hardware:

A³ is a web application that will run in a computer with web-connectivity capabilities. The application will run with ODU server and a single instance server.

b. Software:

A³ will use language python 3.8 or newer. The GUI language that will be use is HTML, CSS and JavaScript. The JavaScript that will be use are Angular, because offer strong opinions on how applications should be structured, and React, a JavaScript library. The IDE: Visual Studio Code. The documentation: pydoc and Sphinx. We will be using code Repository: GitLab and for fontainerization and deployment: Docker and Docker Compose. The database: MySQL or MongoDB . The configuration management: tox and VirtualENV and analysis: pydocstyle (formerly PEP 8) and Pylint

3 Identification of Case Study

A³ is being built for ODU Computer Science Department to help students and faculty members. The application will be used for the aggregation of artifacts by teachers, students and it will be helpful for guests that try to access to ODU databases. This application can be use in the future for Academia, other colleges, other faculty members and students as well.

4 A³ Product Prototype Description

4.1 Prototype Architecture (Hardware/Software)

4.2 Prototype Features and Capabilities

4.3 Prototype Development Challenges

5. Glossary

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.

LAB 1 - A³ DESCRIPTION

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.

LAB 1 - A³ DESCRIPTION

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

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/>.

LAB 1 - A³ DESCRIPTION

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

LAB 1 - A³ DESCRIPTION

(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.