

Software Development Capstone Topic Approval Form

The purpose of this document is to help you clearly explain your capstone topic, project scope, and timeline. Identify each of these areas so that you will have a complete and realistic overview of your project. Your course instructor cannot sign off on your project topic without this information.

Note: You must fill out and submit this form. Space beneath each number will expand as needed.

Any cost associated with developing the application will be the responsibility of the student.

INFORM MENTOR:

Potential use of proprietary company information: (Y/N)

No proprietary company information will be used.

ANALYSIS:

1. Project topic AND description:

This is an application for cataloging and reporting file information from disparate sources.

This will crawl target file systems to gather and catalog file information (i.e. file system location, size, modification date) as well as meta data contained within files (i.e. Author, GPS location, resolution) as is supported by the file type and its relevance to customer requirements.

2. Project purpose/goals:

The marketing department has two issues: They are running out of shared disk space; and they recent staff turnover has made it difficult to locate critical marketing material. They are required to remove files that are no longer needed and to remove unnecessary redundancies before they can require additional storage space. They need to re-organize all marketing files and define governance so marketing materials can be more easily located in the future. Previously, there was no official governance, and marketing materials have been scattered across loosely structured file shares, personal workstations and even on DVDs stored in physical file cabinets. It was determined that all files need to be centrally cataloged. Each cataloged file needs to include information about the file system location and contents of the file, including file type, size, and creation and modification dates. For file types that support/include it, meta data information should be extracted, including title, description, producer, composer, author, resolution, resolution, duration, dimensions, bit rate and geographic location.

There are solutions on the market that could be used, such as file system linters and multimedia cataloging software.

However, affordable file system linters do not deal well with large amounts of files, or they don't have the ability to catalog files from several removable media sources in a single instance. Affordable multimedia cataloging software is mainly geared toward single-user instances. Other more expensive solutions also include functionality which will never be used by the marketing department, making their cost unjustifiable.

The solution needs to be able catalog files from various source media, such as local hard drives, shared drives and DVDs. In the case of removable media, the solution needs to be able identify physical media from which it was obtained. Once all file data is collected it needs to be viewed, filtered, sorted and grouped as a whole, so management staff can develop file storage governance (i.e. folder structure, storage medium, file sizes and types) as well as what resources are needed in order to bring everything into compliance. This will also allow them to periodically re-catalog files so they can monitor for non-compliance or to identify areas where the governance may need to be amended.

DESIGN and DEVELOPMENT:

1. Application Type (select one):

- Mobile (indicate Apple or Android)
- Web
- Stand-Alone

Web application hosted on IIS.

2. **Programming/development language(s) that you will use:**

Server-side code will be written in c# and will utilize XML data. Database communication will be accomplished using T-SQL queries. This will also serve HTML and CSS content as well as JSON data through RESTful web services. Client-side code will be developed using TypeScript, which will transpiled into JavaScript. The Angular framework will be used along with Bootstrap for client code to facilitate a dynamic, rich user interface that is compatible with both mobile and desktop displays.

3. **Operating System(s)/Platform(s) that you will use:**

Application will be developed using Visual Studio on a Windows 10 laptop. Further validation will be completed by deploying the web app to Windows Server 2016 and testing its operation there.

4. **Database Management System you will use:**

The target DBMS will be Microsoft SQL Server 2016.

5. **Estimated number of hours for:**

- i. Planning and Design: 10
- ii. Development: 100
- iii. Documentation: 10
- iv. Total: 120

6. **Projected completion date:** Wednesday, January 21, 2021

IMPLEMENTATION and EVALUATION:

1. **Describe how you will approach the execution of your project:**

Development will use an adapted Iterative Agile approach, whereby the software developers will strive to adhere to Agile values and principles. From the customer's perspective, the product "release" is defined as the completion of the project, rather than the completion of an epic, milestone or sprint. Furthermore, time frames of epics will be constrained so as to give the customer a higher degree of predictability of product cost and delivery.

- a. Work with customer to create use cases and user stories to define what the customer needs to do. Ensure that a concrete listing of file types to be supported is determined. This is to include the meta data to be extracted from each file type and how the customer needs to use the meta data.
- b. Create wireframe and UML diagrams. Create a list of requirements (functional and non-functional) for completion of the initial release. Include time and cost estimates for each requirement. Associate requirement with use cases and/or user stories.
- c. Review diagrams and requirements with customer. Customer may request modifications, defer requirements for a future release, or omit them if they are too expensive. If customer makes any modifications, revert to step "b".
- d. Organize requirements into epics. Epics will be developed within separate software modules, where possible.
- e. At the start of each epic, define milestones (if needed) and tasks. Epics will be fulfilled in 1 or more sprints.
- f. Perform final unit tests and user acceptance tests.
- g. Deploy to customer site.

☒ **This project does not involve human subjects research and is exempt from WGU IRB review.**

STUDENT SIGNATURE

Leonard E. Egan

By signing and submitting this form, you acknowledge any cost associated with development and execution of the application will be the responsibility of the student.

COURSE INSTRUCTOR'S NAME:

Charles E. Faller

COURSE INSTRUCTOR APPROVAL DATE:

12/18/2020
