Michael Gibson Acorn Nursery

Overall Status: Behind Schedule,

Summary of Project Tasking: Update personal development folder with reflections and statuses.

* [GitHub/mikegibson66/cse490R](https://github.com/mikegibson66/CSE490R)

Security

Authentication is the process of verifying the identity of a user, typically before granting access to a system or resource. It is a crucial security measure that helps to protect against unauthorized access and data breaches.

Authorization is the process of determining what a user is allowed to do within a system or resource. Once a user has been authenticated, authorization determines what actions they are allowed to perform. For example, an authorized user might be able to read, write, or delete data, while an unauthorized user might only be able to read data.

Access, in the context of information security, refers to the ability to use or approach a resource, such as data, systems, or physical locations. Access can be granted or denied based on various factors, including authentication, authorization, and security controls.

security risk statements for Authentication, Authorization, and Access:

* Authentication

*Condition*: An attacker gains access to valid authentication credentials.

*Event*: The attacker uses the stolen credentials to impersonate a legitimate user.

*Consequences*: The attacker may be able to access sensitive data, perform unauthorized actions, or cause damage to the system.

* Authorization

*Condition*: An attacker exploits a vulnerability in the authorization system.

*Event*: The attacker gains unauthorized access to resources or performs unauthorized actions.

*Consequences*: The attacker may be able to access sensitive data, disrupt operations, or cause financial losses.

* Access

*Condition*: An attacker bypasses security controls to gain unauthorized access to a system or resource.

*Event*: The attacker gains access to sensitive data, performs unauthorized actions, or causes damage to the system.

*Consequences*: The attacker may be able to steal sensitive information, disrupt operations, or cause financial losses.

Software Development Folders

A software development folder, often referred to as a project folder or code repository, is a directory structure or location within a version control system (e.g., Git, SVN) where all the files and documents related to a specific software development project are organized and stored. This folder plays a crucial role in managing and maintaining the software development process and its associated artifacts. Here's a breakdown of what a software development folder typically contains and how it is used in software development:

1. **Codebase**: This is the central part of the software development folder and includes the source code files and scripts that make up the application or software project. These files contain the actual code written by developers.
2. **Task Description Document**: This document is a fundamental component of the software development folder. It contains detailed information about tasks or user stories. It typically includes the following:
   * **Story**: A description of the task or user story, outlining what functionality or feature is being implemented.
   * **Requirements**: An explanation of the specific requirements or acceptance criteria that the story satisfies. This helps ensure that the software meets the intended functionality.
   * **Design Elements**: Details on any design components or architectural elements related to the task. This may include diagrams, mockups, or design documentation.
   * **Change or Addition Rationale**: An explanation of why the change or addition to the code was made. This could include business justifications, bug reports, or feature requests.
3. **Difference Listing**: This document contains information about changes made to the codebase. It provides a summary of what code changes were made, who made them, and when they were made. This can be especially useful for tracking code modifications and reviewing the history of changes.
4. **Unit Test Cases**: This folder contains test cases specifically designed to validate the functionality and correctness of individual units or components within the software. Unit tests are crucial for ensuring that code changes do not introduce new defects.
5. **Review Report with Documents Uploaded**: After code changes are made, developers often conduct code reviews to ensure code quality, adherence to coding standards, and to identify potential issues. The review report documents feedback and findings from the review process. Any relevant documents or feedback provided during the review are stored here.
6. **Code Checked In**: This directory or area within the folder stores the code that has been checked in or committed to the version control system. It represents a snapshot of the codebase at a specific point in time.
7. **Unit Test Results**: This section contains the results of the unit tests performed on the code. It documents whether the code changes passed or failed the unit tests, helping developers identify issues and ensure that the software behaves as expected.

The software development folder serves as a centralized repository for all project-related artifacts, promoting collaboration, version control, and traceability. It enables team members to work on different aspects of the project simultaneously while keeping a record of changes and providing a clear understanding of the project's history. It is an essential component of modern software development practices and is typically organized within a version control system to facilitate collaboration and code management.

Number of Hours:

* Number of Hours worked this week (expectation: 7-9 hours): 9 hours
* Total Number of Hours this semester (anticipated 120/semester): 126 hours

Accomplishments:

* Acrn 43, #49 First draft of the Rest API controller, missing dependencies
* Acrn 39, #48 Draft version of the Repository Layer
* Acrn 39, #47 Draft implementation of user service

*Performance*:

* Last Week’s Performance Average (Actual Total Points/ Total Hours): 9
* This Week’s Performance (Actual Points/Hours): 9

*Challenges encountered and resolutions found*:

* I continue to struggle with my work schedule and attempting to stay in contact with other team members. Working through available videos and using asynchronous communication has been limiting.

Plans/Goals/Tasking for the coming week/sprint:

* I plan to complete the two stories that I am currently assigned to by the end of the week.
* I also plan to start working on the story that I will be assigned for the next sprint.

*Reflection goals and status*:

* I am struggling so far, but I want to make sure that I continue to be consistent with my work.
* I am also going to work on making a better effort to communicate more with my team if I have any questions or concerns.

*Follow-up/Reporting*:

* I will ask my team to help me follow through with my goals.
* I will also work with my team on the weekly status reports.