

CGI SOAR PROJECT

Intranet Site

[Project Overview](#)

Olivia Whitman, Tim Stadtlander, Zaren Wienclaw

Contents

OVERVIEW	2
SCHEDULE	2
DESIGN GROUP	2
BACKEND GROUP	2
FRONTEND GROUP	3
PROJECT BREAKDOWN	4
ARCHITECTURE	4
TECHNICAL OVERVIEW	4
REQUIREMENTS	4
ANNOUNCEMENTS	4
TIMESHEETS	4
SECURITY	5
GUIDELINES	5
ANGULAR & TYPESCRIPT	6
C#	6
CSS	6
HTML	6
GIT	6
VISUAL STUDIO PROJECT OVERVIEW	6

OVERVIEW

Welcome to the SOAR Project. This project is designed to use the skills you have learned throughout SOAR to build a fully functional application that could be delivered to a client. This project will have you touch all areas from database to UI.

The project that you've been given to complete is to create an intranet website. This site will have two main functions: displaying announcements and time tracking. While there will be issue tracking that contains your tasks throughout the day, you should use this document as a reference when you're unsure about a task. If things are still unclear, please ask for assistance.

SCHEDULE

DESIGN GROUP

Many projects have a workflow of *mockups -> sample pages -> interactive sample pages -> final product*. We have tried to lay out the project so that it follows this flow as best as it can, given the constraints on materials covered throughout the day

DAY 1

1. Presentation: Introduction to Grad Academy Project
2. UI templates

BACKEND GROUP

DAY 1

1. Models
2. Services and Repositories

Day 2

1. Web API
2. Static Repositories
3. Presentation: Introduction to Unit Testing
4. Unit Testing
5. Entity Framework Repositories

CGI SOAR Project – Intranet Site

FRONTEND GROUP

DAY 3

1. Angular UI
2. Prep for Grad Academy Project Presentation

DAY 4

1. Wrap up project
2. Prep for Grad Academy Project Presentation
3. Present the project

PROJECT BREAKDOWN

ARCHITECTURE

Our application will be following the hexagonal architecture. This is also sometimes referred to as the onion architecture or ports and adapters architecture.

TECHNICAL OVERVIEW

The following will lay out some of the technical details of the project:

- Source Control: git, hosted in Azure DevOps
 - https://dev.azure.com/HMBSolutions/_git/GAP-Students
- Work Tracking: Azure DevOps
 - https://dev.azure.com/HMBSolutions/GAP-Students/_sprints/taskboard/GAP-Students%20Team/GAP-Students
- Languages and/or frameworks: C#, TypeScript, Angular
- Dev Database Server: SQL LocalDB
- Unit testing framework: XUnit
- Build System: Azure DevOps
- Test Environment: Azure

REQUIREMENTS

ANNOUNCEMENTS

- REQ-ANN-001 – The main page should display a list of current active announcements
- REQ-ANN-002 – Announcements can be scheduled in advance
- REQ-ANN-003 – Announcements can expire
- REQ-ANN-004 – Announcements can have high priority
- REQ-ANN-005 – Announcements have a title that cannot exceed more than 500 characters
- REQ-ANN-006 – Announcements have a body
- REQ-ANN-007 – Announcement's body can contain Markdown syntax
- REQ-ANN-008 – Announcements are displayed in 5 items per page intervals
- REQ-ANN-009 – Announcements are ordered by highest priority first
- REQ-ANN-010 – Announcements are ordered by Date second (descending order)

TIMESHEETS

- REQ-TIM-001 – Employees must be able to enter their timesheet for a given week

CGI SOAR Project – Intranet Site

- REQ-TIM-002 – Timesheets are broken up by week
- REQ-TIM-003 – Timesheets are associated with a single employee
- REQ-TIM-004 – Timesheets show 0-N time entries
- REQ-TIM-005 – A timesheet's single day total cannot exceed 12 hours
- REQ-TIM-006 – A timesheet's weekly total cannot exceed 50 hours
- REQ-TIM-007 – Employees can only view their own timesheets
- REQ-TIM-008 – Employees have an email address that is required and may not exceed 300 characters
- REQ-TIM-009 – Employees have a first name that is required and may not exceed 100 characters
- REQ-TIM-010 – Employees have a last name that is required and may not exceed 100 characters
- REQ-TIM-011 – Time entries accept a single numeric value per day of week
- REQ-TIM-012 – Time entries are associated with a single task
- REQ-TIM-013 – Time entries sometimes have notes associated with it. Notes may not exceed 1000 characters
- REQ-TIM-014 – Tasks have a name that may not exceed 300 characters
- REQ-TIM-015 – Certain tasks may require time entries associated with it to have a note

SECURITY

- REQ-SEC-001 – OAuth will be used for authentication
- REQ-SEC-002 – Authentication is required to modify a timesheet
- REQ-SEC-003 – Authentication is required to create or modify announcements
- REQ-SEC-004 – Thorough logging should be implemented

GUIDELINES

In general, each project you are on is going to conform to some sort of coding style guide. Robert C, Martin (Uncle Bob) says the following in his book, *Clean Code*, "the ratio of time spent reading (code) versus writing is well over 10 to 1." While sometimes there can be a performance or security benefit outlined in style guides, the general benefit style guides give is reducing the cognitive load on future developers. Style guides are typically broken up by language, as different languages have different syntaxes and implicit formatting guidelines.

Below are the guidelines we recommend for the project. It would take longer to read all the guidelines than we can spare for the project, but if you are wondering where merge request comments came from perhaps browse the recommended style guide for the language in question.

CGI SOAR Project – Intranet Site

ANGULAR & TYPESCRIPT

[Angular Style Guide](#)

C#

[Microsoft's C# Coding Conventions](#)

CSS

- Short: [AirBnB's CSS Style Guide](#)
- Long: [CSSGuidelin.es](#)

HTML

[W3School's Coding Conventions](#)

GIT

- The branch called “SOAR-2021” will be locked down from direct commits.
- All code changes to SOAR-2021 should be done through merge requests.
- All development code will be in branches following the convention {area}/team-{group number}
 - certain days will address different areas (see [schedule](#)) such as: design, backend, and frontend. This should be used as {area}
 - Throughout the project you will be working in rotating teams of 3; these teams will be assigned a number and should be used as {group number}
- Commits should be associated to the task that you are working on
- Commit messages should be informative

VISUAL STUDIO PROJECT OVERVIEW

HMB.GAP2019.INTRANET.WEBSITE

- Contains our final UI
- The focus should be on how to display information.
- For now, consider this an adapter
 - In our enhancement document we dig into why we could consider this an adapter that implements an entire hexagonal architecture by itself.

CGI SOAR Project – Intranet Site

HMB.GAP2019.INTRANET.API

- Contains the endpoints that will respond to our HTTP requests
- The focus of this project is mapping HTTP requests to services
- There should be virtually no logic or coordination within this project
- The project would be considered an adapter

HMB.GAP2019.INTRANET.CORE

- Contains the models, ports, and business logic of our application.
- This is the core functionality of the product.
- The focus is making sure the actual requirements of the product are met.
- This project should not know how the adapters are implemented, rather it depends on the port(s) existing.

HMB.GAP2019.INTRANET.INFRASTRUCTURE

- This will be an implementation of our backing storage
- This will provide static, “demo”, data for repeatable results
- This would be considered an adapter

HMB.GAP2019.INTRANET.DATA

- This will be an implementation of our backing storage
- This will persist data to a database using Entity Framework Core
- This would be considered an adapter