



# IDDQD 4390 Project Portfolio

04.22.2020

---

David Gulde

Dubs Lasley

Haley Smith

Matt Webb

## Overview

### *a. Details of Project*

Our project is a curriculum builder made for users who want to combine elements and make a competency. Their Competency name and description are the choices of the user and they can add whichever elements they wish to make up their competency. The user chooses from a list of dispositions, knowledge elements, and skill levels. It is completely customizable.

### *b. Principle features*

Our website allows the user to create an account, sign in, and log out. They are able to build their own competency made up of dispositions, knowledge elements, and skill levels.

### *c. Stakeholders and End-users*

The end-users of our web application would be students, professors, and anyone who would like to know what constitutes in making a competency such as knowledge elements, skill levels, and dispositions.

## Team and Structure

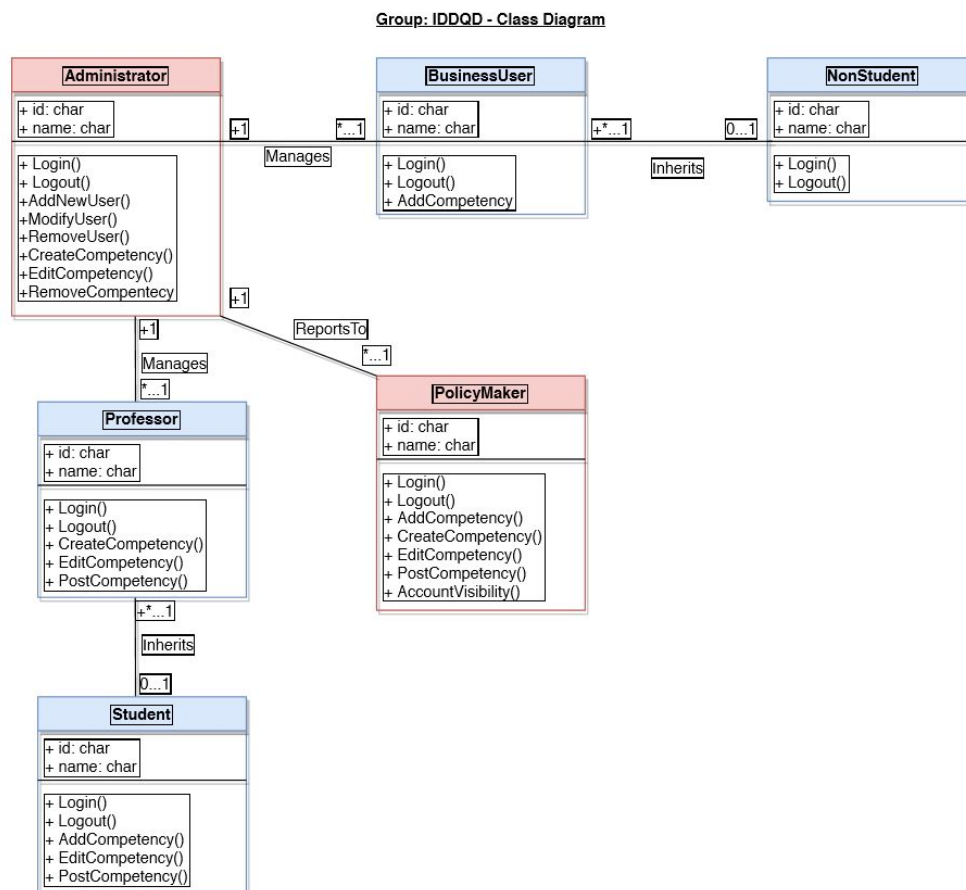
- a. Haley Smith was the SCRUM master and was in charge of spring planning, reviews, and retrospectives. Organizing and allocating work was part of the daily duties of the SCRUM master. The SCRUM master was also appointed as the Team Lead of the group which proved to be a two person task. 50% of the team organization was done by Haley and the other half by Matthew Webb.
- b. Matthew Webb was the development operations lead and in the first few sprints allocated his time to our VPS, web service, and the environments we developed our project out of. He also played a major role in keeping all of the team's work organized with GitKraken and Github. He managed the code repository and handled merge pain across branches by keeping all environments up to date and in sync. He played at least 50% if not more of the Team Lead role along with his development role.
- c. Dubs Lasley was in charge of the User Interface of our project. He went full force into his role and came up with a great layout from the get-go. Unfortunately, we had to go with a different idea that would allow for the layout to be integrated into several pages. He took on the task and found another layout that we used for the rest of the project. Every new component we added Dubs made the appropriate

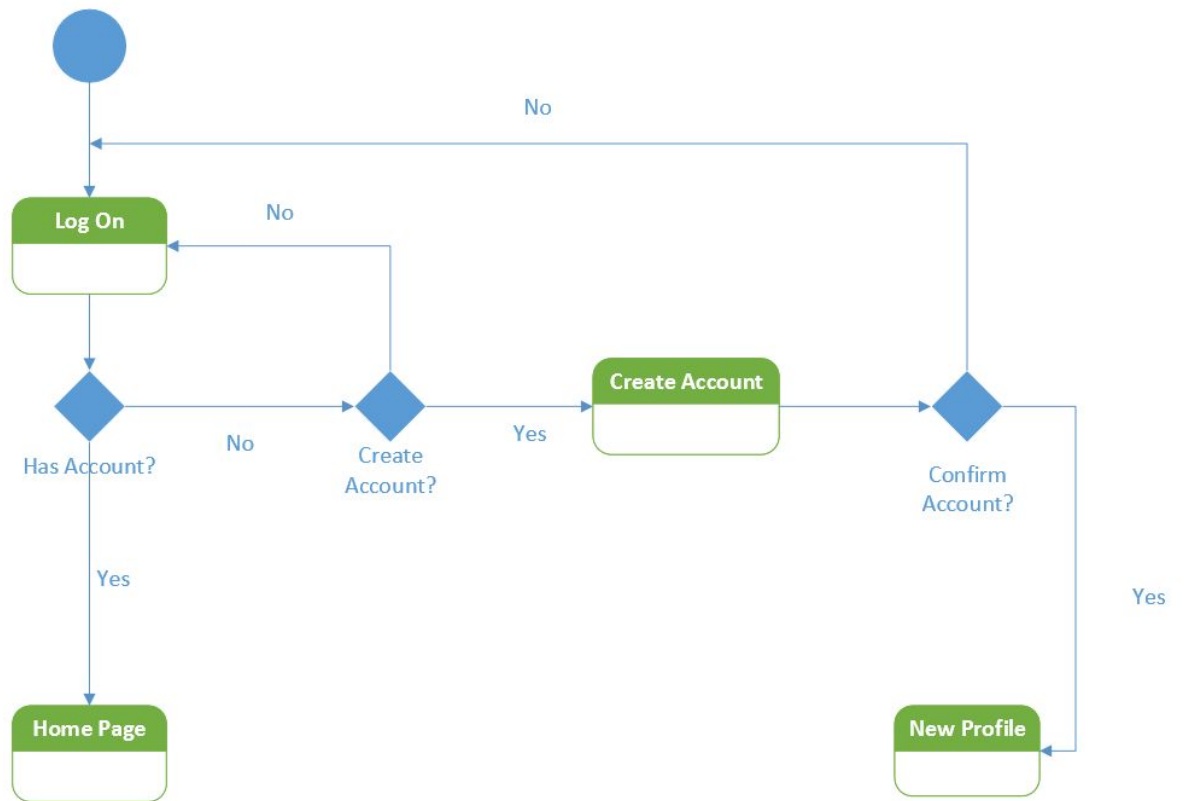
changes right away and even some nice enhancements for more esthetically pleasing user experience.

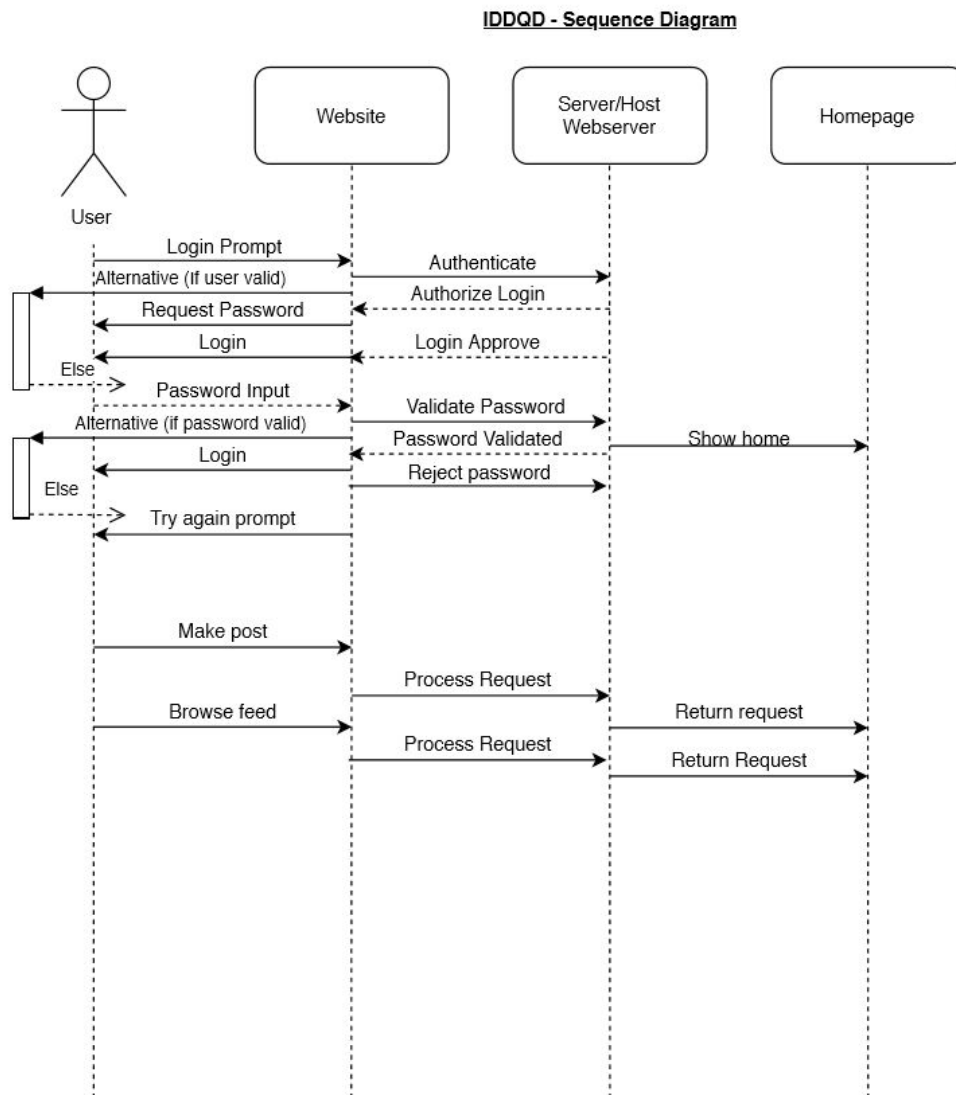
- d. David Gulde was our technician lead for this project. He made sure that all architectural aspects of our code are within the SOLID guidelines. He also tested the functionality of the bulk of our code to make sure that we had everything we needed and where we needed it. He troubleshooted errors when they came up with the best solution for our goals.
- e. Our team grew our communication and cooperation skills with every sprint. We all pulled our weight the best we could and gave each team member the opportunity to contribute to pieces of the project that they were the best at.

## Systems Analysis and Design Diagrams

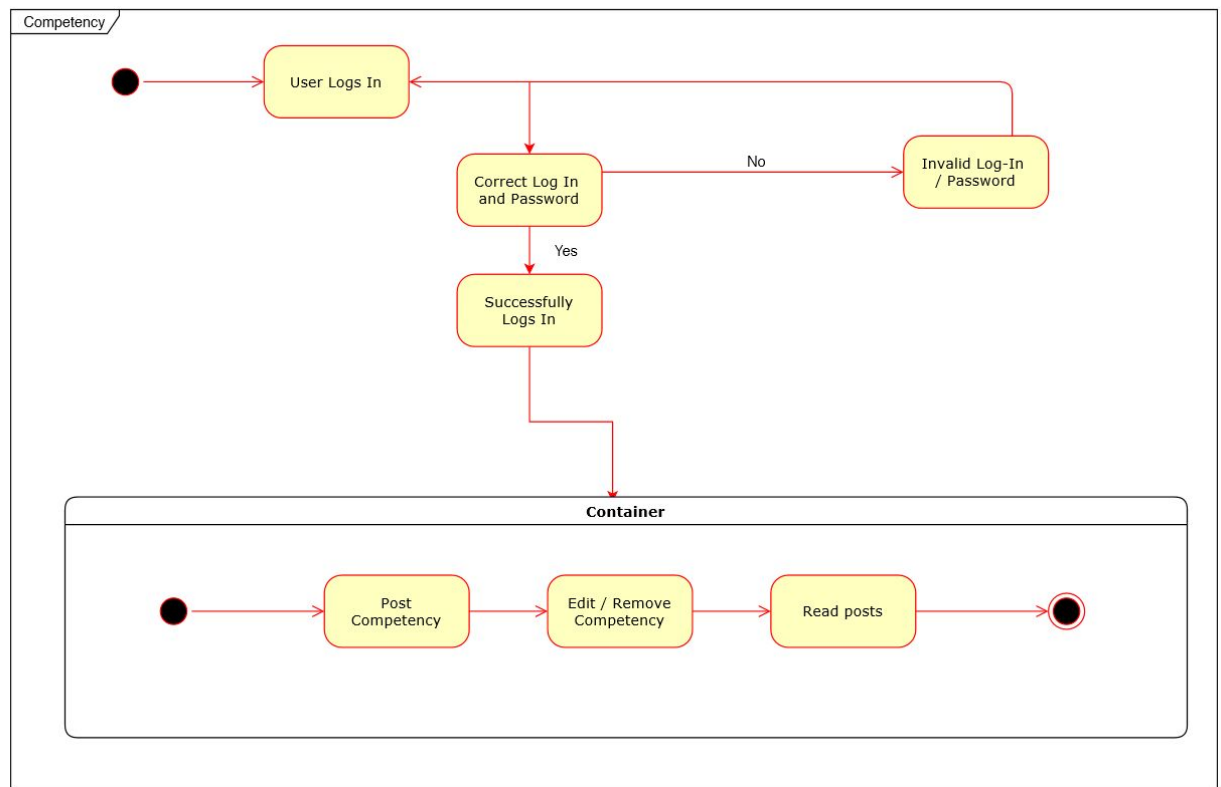
### a. Class Diagram



**b. State****c. Sequence**



d. Activity



## Scrum Process

### a. Overview of the Product Backlog

Our backlog varied every sprint depending on our needs at the time. The backlog consisted of PBI's such as User-Friendly Interface, Building a Competency, Managing Knowledge Elements, Managing Skill Levels, Managing Dispositions, and New Account Registration.

### b. Sprints

#### ❏ Evidence of Review

Our sprint reviews were conducted with Dr. Babb, Angelica Alonzo, and Phelps Murrel. We shared our sprint goal, achievements of the previous sprint, and our vision for the next sprint's goal. Our team recorded each review session as a referral if we needed it.

This link contains every recording that was done by our team. Reviews, retrospectives, and daily scrums included:

<https://drive.google.com/open?id=1sFYNMQNC4bQuFXeqH87qtUWzf-PZcYcE>



#### ❑ *Evidence of Retrospective*

After each sprint, we had a reflection of how the past two weeks had gone. We discussed our progress, team dynamic, and our success in communicating. We reflected on what went well for us and things that we should repeat the next sprint to achieve our goals. We discussed what we should stop doing that was not providing positive or productivity to our project or team goals.

#### ❑ *Evidence of Planning*

At the beginning of every sprint, we had a meeting the same day we had our sprint review to plan the following sprint. We talked collectively about what was achievable in the next two weeks and prioritized our goals. We also allocated our tasks in accordance with our roles. However, we were all willing to help any member of our team with their work regardless if it was not part of our “job”. We found that our most productive times were when we were all working together to accomplish a part of our project.

#### ❑ *Goal*

Our goals for each sprint varied depending on what was completed in the previous sprint. We did have some carryover for some sprints that we had to allocate more of our team’s time to in order to get all of the goals for the sprint done. We always had a focused goal for each team member each sprint that way there was no question as to what each of us was responsible for. At the end of each daily scrum, the SCRUM master would ask the team if they were all aware of what they were responsible for before the meeting had ended to ensure that there was little communication error.

#### ❑ *Definition of Done*

Our team’s definition of done relied on the individual who worked on their part to be able to show the rest of the team what they accomplished and that it worked. It also had to be successfully run from another team member machine and the VPS to assure the quality of the piece of the project they had completed.

#### ❑ *Backlog*

The backlog’s consisted of the current goal of the team in priority order. We built the backlog at each sprint planning meeting and we all in agreeance of the items that were needed the sprint following. The effort and remaining work came into play later on in our project. In the beginning, the effort level for each Project Backlog Item was determined by our team in a meeting typically the sprint planning meeting. Later in our planning, we utilized the

remaining work feature of Azure DevOps to determine how much work we still needed to be done on each of our tasks. During our meetings, we subtracted the hours that we had worked on a certain task after we had shown the team our progress.

#### ❏ *Evidence of Daily Scrum*

Our team conducted our daily scrum meeting three times a week at roughly the same time. We also went beyond the minimum three times and had many more meetings when they were required to get some finishing touches of our project done before the sprint review. Our team has every daily scrum meeting recorded as evidence of our meetings.

#### ❏ *Burndowns*

##### **Burndown for Sprint B:**



Note: Our burndown chart shows work remaining but we continued our hours of work that remained over the weekend before the next sprint review.

##### **Burndown for Sprint C:**





Note: The bulk of our work was spent over the weekend finishing the project. On 5/1/2020 there were four hours of remaining work for our team to complete.

#### ❏ *Task allocations*

Each of our tasks was allocated according to our role that was assigned at the beginning of the project. Development related tasks were allocated to Matt until the development portion was completed. User interface elements of the project were allocated to Dubs and some tasks were allocated to Haley the SCRUM master. Technical and code-driven tasks were allocated to David the technical lead and toward the middle and end of the project both David and Matt, the development lead were working on code-based elements of the project. To wrap everything up the team came together to go over every component of the project no matter what the focus was and put the finishing touches on the web application. We each participated in every category.

## Technical Requirements

In the code of our project, our Development lead put in a “resources” folder that documents every step of development including tools necessary for deployment, deployment requirements, and development tools.

- a. *Development Tools, Deployment Tools, and Deployment Requirements*

[Certbot](#) - Certificate Authority

[Digital Ocean](#) - Cloud Computing Host

[Ubuntu Server 18.04](#) - OS

[NGINX](#) - load balancer, reverse proxy

[Git Bash](#) - for ssh and development

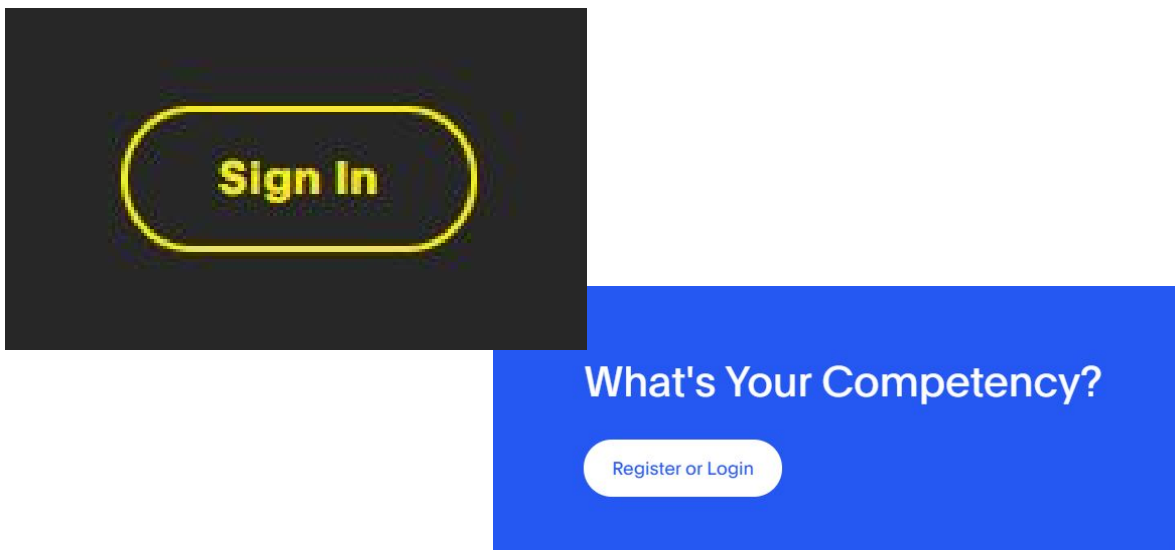
[VS Code](#) - an integrated development environment

[Node.js](#) - Javascript Engine

## User Guide

How to sign in

1. Click Sign In or Register and log in



2. Fill in the information accordingly and click login

## USE A LOCAL ACCOUNT TO LOG IN.

Email

Password

☐ Remember me?

Log in

[Forgot your password?](#)

[Register as a new user](#)


[Resend email confirmation](#)

How to create a competency :

1. Click on Competencies



2. Click the green button to create a competency.



Create your Competency!

ID	Name	Description
1	Disposition A	Do Stuff about A
2	Test	Test
3	Test	Test
4	Test of Enumerated Loop	Loren ipsum
5	Test Competency	Test Description
6	Final Test	Cleaning House
7	Computer	Tech
8	C#	I know how to code

3. Give you competency a name and a description, then click next

## Create Competency - Step 1

Provide competency name and description.

Competency Name	<input type="text"/>
Competency Description	<div></div>

Provide an appropriately-descriptive name.

Provide a competency description that summarizes the set of tasks of constituent competencies.

Next

4. Check the corresponding box according to your disposition

## Select Dispositions - Step 2

Select Dispositions for the competency.

Competency Name: Test

Competency Description: Test

ID	Name	Description	Category	Discipline	Select
1	Proactive	With Initiative (Nwokeji, Stachel, & Holmes, 2019) / Self-Starter (Clear, 2017) Shows independence. Ability to assess and start activities independently without needing to be told what to do. Willing to take the lead, not waiting for others to start activities or wait for instructions.	Habits	Information Systems	<input type="checkbox"/>
2	Purpose-Driven	Purposefully engaged / Purposefulness (Nwokeji et al., 2019), (Clear, 2017) Goal-directed, intentionally acting and committed to achieve organizational and project goals. Reflects an attitude towards the organizational goals served by decisions, work or work products. e.g., Business acumen.	Qualities	Information Systems	<input type="checkbox"/>
3	Self-Directed	Self-motivated (Clear, 2017) / Self-Directed (Nwokeji et al., 2019) Demonstrates determination to sustain efforts to continue tasks. Direction from others is not required to continue a task toward its desired ends.	Qualities	Information Systems	<input type="checkbox"/>

Next

5. Choose atomic or composite, and click next.

## Step 3

Is this an Atomic or Composite competency?

- ☐ Atomic
- ☐ Composite

Next

6. Choose your disposition etymology using the checkboxes, and the level of your skill with the corresponding dropdown menu on the same row. When you are finished, click next

### Step 4 - Skill Level

ID	Name	Description	Cartesian Index	Semiotic Index	Etymology	Selected	Skill Level
1	Control Structures		1	1	Programming	<input type="checkbox"/>	0 Understand ▾
2	Design Patterns		1	1	Software Engineering	<input type="checkbox"/>	0 Choose a Skill ▾
3	Data Normalization		1	1	Database	<input type="checkbox"/>	0 Choose a Skill ▾

Next

7. Click submit competency

Submit My Competency

## Source Code

a. *Repository*

<https://github.com/MJW3BB/CIDM4390-IDDQD>

b. *Requisite commits and Commit Messages*

All commits and commit messages are located in each of the branches. Each team member possessed their own branch and pushed the changes to that branch then their changes were either merged or pushed to dev then production then master branch to be pushed to the VPS. Once pushed to the VPS the code was tested and made sure it was functioning.