Sprint 01 Individual Project

Purpose:

* Communicate effectively in a written and oral professional context.
* Apply various tools and agile principles for project management
* Apply computer science theory such as utilizing design patterns for Django framework to improve the maintainability, modularity and reusability when building a SaaS application

**Effort:** Individual [CS3300 Academic Integrity](https://docs.google.com/document/d/1cORsFi1YrqW5ChfJu0G67Fjm8HwEMse47DVqXfEn2n4/edit#heading=h.w1yj4lpdz8sh)

Reminder if you share work/code/solutions or use someone else's code you will receive a 0 and be reported to the CS chair and UCCS Dean of Students

Points: 60 pts see rubric (Can hand in one day late for 10% deduction)

Deliverables: Word document or PDF of reflection and learning questions only (not this entire document) and video

# Description

Implement the requirements below for your first sprint to build your app for your customers.

[Description](#_ygw0xg8r8r1w)

[App Sprint 01 Project Requirements](#_jxnd9u77mq08)

[Model](#_pec6dbkpn6jx)

[Behavior](#_1t88tplm85hm)

[User Interface](#_z2pb803zukhj)

[Project Management](#_uu945d3w6fow)

[Sprint Backlog, Tasks and Project Tracking](#_16os8gl3ezxl)

[Development and Version Control](#_4jiwuy4oab7h)

[Develop App](#_axnx83ao9zcc)

[Professional Communication: Customer Sprint 01 Demonstration](#_89f85flbcnhx)

[Learnings and Reflection](#_ped7f38piv)

[Learnings](#_xy15n9kne0ii)

[Reflection](#_z2o6jo8ww8xr)

# App Sprint 01 Project Requirements

Implement the following in your app for sprint 01. If you decide to do more, do that after you implement the following first.

## Model

Model (class) that represents the item attributes and methods. Must have at least 4 attributes with at least one required and not all the same type. For example, not all text fields.

## Behavior

* Display home page - overview of app and any other information you would like on home page.
* Display list of items that contains a way to navigate to view, create, edit, delete
* Display detailed list of item information
* Forms to create/update/delete item information

## User Interface

* Navigation bar with at least two menu items
  + Home
  + Menu to list of items
* Not be just black and white
* Have a static image on website
* Some type of organized layout to display listed items
* Buttons to create, edit and delete
* Incorporate at least 4 items to make your web app accessible for people with varied abilities.
* Be responsive to different sized screens

# Project Management

You are going to organize and prioritize your tasks for your first iteration.

Set up a public repository in github for your app.

## Sprint Backlog, Tasks and Project Tracking

You need user stories for the above requirements. For each user story you will break it down into tasks. You should also have tasks for setting up your environment. These user stories and tasks will be stored in a github project with your repository code and documentation for your project.

You will manage your project backlog and tasks in github as shared in class when creating your Kanban Board. It should contain

Board Columns for Task Board

* Sprint Backlog (User stories)
* To Do (Tasks)
* In Process (Tasks)
* Done (Tasks)

Create Issues for User Stories and add to Sprint Backlog

* User stories should represent all requirements for this sprint
* Should include S1, S2, etc in front of the user story.

Create tasks for all the work needed to

* Add tasks for the user stories and setting up your environment (you may have other tasks such as the styling the user interface)
* Include (time estimate) (story number) for each task with brief description (Actual Time)
  + (3 h) (S1) Item detail view - urls, views, html
* As you work on each task move them to in progress and then to done. Update title a place to put actual time (AT 4h) after the estimate and story number. For example
  + (3 h) (S1) Add project model class and migrate to database (AT 5h)

When you create your sprint 01 project you must change the permission from private to public. Have someone test that your repository and project can be viewed by them so that the TAs and Deb can see.

## Development and Version Control

Create a UML diagram of the class that represents your model attributes and methods. Must have at least 4 attributes with at least one required and aren’t all of the same type. For example, not all text fields. Include this in your github repository.

Just a simple representation that includes data types.



|  |
| --- |
| Work Requests |
| Task Number: int |
| Name of Submitter: string |
| Contact Info: string |
| Date: string – MM/DD/YYYY |
| Location: string |
| Sub-location: string |
| Description: long string |
| Type of Issue: string |
| Photo of issue: file (need to R&D) |
| Worker Assigned: string |
| Is\_closed: Boolean |
|  |
| View Functions |
| createRequest() |
| updateRequest: requestID |
| closeRequest: requestID |
| deleteRequest: requestID |
| changeSubLocationDropDown: requestID?? (chained dropboxes) |
|  |
| Model Functions |
| \_\_str\_\_: return task number |
| get\_absolute\_url: reverse the url for other functions |

## Develop App

Develop an app to meet sprint 01 requirements for the app and your customer report.

Use git and github to track and store your app development using best practices for version control.

# Professional Communication: Customer Sprint 01 Demonstration

Create a video screencast to demonstrate the requirements for sprint 01 as if you are presenting to your customer.

* Give brief overview of app and problem statement
* Demonstrate behavior of app
* Explain how app is accessible for varying abilities
* Demonstrate the responsiveness of the app

The video should be no more than 5 minutes. Prepare a script and use the pause feature to be efficient with your time and not ramble to make sure you demonstrate all the requirements.

# Learnings and Reflection

Create a separate document for your Learnings and Reflection

Link to github repository version control:

Link to github project task board:

Link to UML class diagram for model.

## Learnings

Descriptions and explanations should contain technical vocabulary, your own words, code snippets, diagrams (digitally or drawn on paper).

1. Explain how you utilized version control for your spint01 development.
2. Explain how you implemented the functionality for the user to create an item.
3. Explain how you implemented the functionality to display a list of items on a web page.
4. Explain how you used Bootstrap's framework of html and css (elements and classes) to implement the navigation menu in the user interface.

## Reflection

Critically analyze your learning experience by providing a meaningful reflection that includes evidence related to the experience.

Write two to three paragraphs reflecting on the sprint and full stack development utilizing agile methodologies. Think about where you started the class understanding of developing software and where you are now. Here are some questions to help guide your reflection.

* How do you feel about breaking the user stories into tasks? How did the estimates compare to your actual time to complete tasks? How do you feel about utilizing version control?
* What were your successes? What do you think is your biggest accomplishment?
* What lessons did you learn throughout this process? What would you do differently?
* What was the most challenging? What problems did you encounter and how did you solve them? What resources helped you the most?
* What do you think is your biggest strength as a software developer?