# SOEN390 - Testing Plan

## Overview

Test plan for Sprint #2. The following will demonstrate how we plan to cover our testing plan from Unit Tests to full System Tests. During Sprint #2 we will start planning automated tests using Continuous Integration.

## Unit Testing

* Discussion Forum:
  + Test that a user can create a new discussion thread.
  + Test that a user can post a reply to an existing discussion thread.
  + Test that discussions are displayed in the correct order (e.g., latest first).
  + Test that users can edit their own posts.
  + Test that users can delete their own posts.
  + Test that unauthorized users cannot delete or edit posts.
* Property Registration:
  + Test that a user can register a property/unit.
  + Test that required fields (e.g., address, owner information) are validated.
  + Test that duplicate property registrations are prevented.
  + Test that property details can be updated.
  + Test that only authorized users (e.g., admins) can modify property details.
  + Test that property registration data is stored correctly in the database.
* Unit, Locker, and Parking Details:
  + Test that unit details can be added, updated, and deleted.
  + Test that unit details are properly validated (e.g., unit number format).
  + Test that locker details can be added, updated, and deleted.
  + Test that parking details can be added, updated, and deleted.
  + Test that associations between units, lockers, and parking spaces are correctly handled.
  + Test that only authorized users can modify unit, locker, and parking details.

## Integration Testing

* Discussion Forum:
  + Test the integration between the front-end and back-end systems to ensure that discussion threads are properly created, displayed, and updated.
  + Test the integration between user authentication and the discussion forum to verify that only authenticated users can post or reply to threads.
  + Test the integration between the discussion forum and notification system to ensure that users receive notifications for new posts or replies.
* Property Registration:
  + Test the integration between the front-end user interface and the back-end system to ensure that property registration forms are properly submitted and processed.
  + Test the integration between property validation logic and the database to ensure that duplicate registrations are prevented.
  + Test the integration between property registration and user authentication to ensure that only authorized users can register properties.
  + Test the integration between property registration and email notifications to notify administrators or relevant parties about new registrations.
* Unit, Locker, and Parking Details:
  + Test the integration between the user interface and the back-end system to ensure that unit, locker, and parking details can be added, updated, and deleted as expected.
  + Test the integration between unit, locker, and parking details and the database to ensure that changes are persisted correctly.
  + Test the integration between access control logic and unit, locker, and parking details to ensure that only authorized users can modify them.
  + Test the integration between unit details and other related modules (e.g., billing system) to ensure consistency in data across the system.

## 

## 

## System Testing

* Discussion Forum:
  + Test the entire discussion forum module end-to-end, including user interaction with the front-end interface, data processing in the back-end, and persistence in the database.
  + Test different user roles (e.g., regular user, admin) to ensure proper access control and permissions.
  + Test error handling and recovery mechanisms, such as what happens if the database connection is lost during a post submission.
* Property Registration:
  + Test the entire property registration workflow, from initial submission to approval and storage in the database.
  + Test edge cases, such as submitting invalid property information or exceeding character limits in input fields.
* Unit, Locker, and Parking Details:
  + Test the end-to-end flow of managing unit, locker, and parking details, including creation, updates, and deletion.
  + Test data consistency across different modules, such as ensuring that changes in unit details reflect correctly in related modules like billing or access control.
  + Test usability and accessibility of the user interface for managing these details.