# Week 07: Instructor Notes

## Overview

Units Tests is a skill that very engineer needs to know how to do, from manual tests to automated tests. A unit test covers all the functionality of the code. Units tests run the code with different inputs and checks the output. Good unit tests verify all branching functionality.

## Objectives

By the end of the week, each student will be able to:

* Implement Unit Tests

## Before the Week Begins (Prepare)

Preparation:

* Post Annoucement.
* Coordinate with TA.
  + Discuss struggling students and make plans to support and mentor.
  + Discuss W07 grading expectations.
* Research the EM Spectrum.

Partnership/Group work:

* Check Canvas "groups" to verify that all students are enrolled in a group.

### Labs:

## Electronic Sensor Test Case

## Help students understand the the hardware that is running Dr. Mobile has sensors similar to devices they they are common to devices currently being used by student's electronics.

## Tests steps should follow those found in the Teach 05

## Looking Ahead

Some things to be aware of:

* Teach
  + There is only a Lab for next week, and will be writing manual tests.
* Prove
  + Make sure that the students have install Selenium.
  + Students will be using Code to execute the an automated test.

Next week's announcements:

* Please create and post your Announcement for W08 by Friday of this week, no later than Monday.

# Prepare

## Overview

Unit tests are the first tests programmers implement. Programmers may conduct unit testing before or after designing their code. Units tests run the code with different inputs and checks the output. Good unit tests verify all branching functionality.

One of the first tests that all programmers implement before or after implementing design is a unit test. These test the scope of the functionality of the code. Week 7 is the implementation of Week 5.

## Preparation Material

### Reading

* [See Reading.](../Reading/Reading.html)

As you read, try to answer the following questions:

* Research one of the Software Verification Techniques and provide a 1-2 sentence summary
  + Inspection
  + Analysis
  + Simulation
  + Demonstration
  + Test Case
* For each testing verification technique and identify three Quality Characteristics (Module 05) that would benefit from that technique regarding Unit Testing?

# Teach: How do I write test cases?

## Overview

Testing consists of four different testing types: Inspection, Analysis, Simulation, Demostration Tests. Inspection is looking at the code, application, or resulting data to see the the proper results was retrieved. Analysis is the process of running the software many times and collecting data to see if the data falls with in a acceptable margin, such as performance testing. Simulation using acceptable applications that provide feedback to the system. Demostration is showing the customer/user the actions of a workflow through the system.

Each of these testing types use a similar formate for writing manual tests, called Step & Results/Verify method. The Step & Results/Verify method allows for step by step break down of the test. It is also the initial step toward automating tests. Once the process is layed out, collecting the excepted or unexpected data, automation become easy.

Write different test cases using the Step & Results Verify Method.

1. Write a requirement you want to verify
2. Describe any setup expectations, i.e., Start software with a blank document.
3. Break down the workflow in to steps, where each step is one action. i.e., Click on File: Save
4. For each step, there should be a **result** of what the software did.
5. Finally, after the last step, use **verify** instead of results. This verify must prove prove the requirement is satisfied.

For example here are two different format types of the same test.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | Save Module 7 Test Case Discussion Web Page | | | |
| **Requirement** | Browser shall save Test Case Discussion Description | | | |
| **Test Type** | Demonstration | | | |
| **Pre-Conditions** | Web Brower is open | | | |
|  | | | | |
| **Steps (Actions)** | | **Results/Verify** | **Pass/ Fail** | **Comments** |
| From the file menu, click File: Save Page As... | | Save As dialog window appears |  |  |
| From the Save As dialog, Change the directory to Documents | | Save As dialog now list the contents of Documents |  |  |
| From Save As dialog window, click in the File name: text box to change the name to test.html | | Save As dialog window, File name's text field hold test.html |  |  |
| From Save As dialog window, click the Save button | | Save As dialog window closed |  |  |
| Form the browser file menu, click File: Open File | | Open File dialog window opens |  |  |
| From the Open file dialog window, click on Documents | | Open file dialog window now lists the content of Documents |  |  |
| From the Open file dialog window, Click on the test.html file from the file listing | | test.html file is highlighted and appears in the File name text box |  |  |
| From the Open file dialog window, click the Open button | | Verify that the page that saved the file is identical to the newly opened file. |  |  |

Text Format:

* **Title**: Save Module 7 Test Case Discussion Web Page
* **Requirements**: Browser shall save Test Case Discussion Description
* **Test Type**: Demonstration
* **Pre-conditions**: Web Brower
* **Steps (**[ (A)ction/(Result)/(V)erify ]):
  + **Action**: From the file menu, click File: Save Page As...
    - **Result**: Save As dialog window appears
  + **Action**: From the Save As dialog, Change the directory to Documents
    - **Results**: Save As dialog now list the contents of Documents
  + **Action**: From Save As dialog window, click in the File name: text box to change the name to test.html
    - **Results**: Save As dialog window, File name's text field hold test.html
  + **Action**: From Save As dialog window, click the Save button
    - **Results**: Save As dialog window closed.
  + **Action**: Form the browser file menu, click File: Open File
    - **Results**: Open File dialog window opens
  + **Action**: From the Open file dialog window, click on Documents
    - **Results**: Open file dialog window now lists the content of Documents
  + **Action**: From the Open file dialog window, Click on the test.html file from the file listing
    - **Results**: test.html file is highlighted and appears in the File name text box
  + **Action**: From the Open file dialog window, click the Open button
    - **Verify that the page that saved the file is identical to the newly opened file.**

**Instructions**

You will need to create a test case for 2 of the different topics listed below or choose your own.

Create a test case for any one of the following requirements using the Step & Results/Verify process, using either Inspection, Analysis, Simulation, Demonstration, or Test.

**Topics**

Requirements Topics to write verification tests:

* Brush your teeth: A user shall brush their teeth.
* Flossing your teeth: A user shall brush their teeth
* Making a peanut butter and jelly sandwich: A user shall produce a PBJ sandwich
* Driving kids to school: A user shall deliver a child to school
* Driving self to school: A user shall arrive at a school
* Reading a book: A user shall read and finish a book
* Winning at Tic-tac-toe: A user shall win a game of tic-tac-toe
* Losing at Tic-tac-toe: A user shall lose a game of tic-tac-toe
* Setting up Christmas lights: A user shall set up Christmas lights on one section of the house
* Mathematics: A user shall add two 3-digit numbers together
* Mathematics: A user shall subtract two 3-digit number together
* Mathematics: A user shall multiply two 2-digit numbers together
* Mathematics: A user shall divide a 2-digit number from a 3-digit number.
* Chess: A user shall perform a Rook (castle) move
* Elevator: A user shall go from the 1st floor to the 3rd floor
* Asteroids Game: A user shall win a game
* Scheduler: A user shall schedule a reoccurring event in Outlook
* Mars Rover: A user shall move the Mars Rover in a square
* Binge-watching app: A user shall binge-watch three episodes of a show
* Radio Station: A user shall program the car radio button
* Restaurant Reservation: A user shall make a reservation for dinner
* Rent a Scooter: A user shall rent a vehicle
* Flights Reservation: A user shall create a plan reservation
* Personality Tests: A user shall determine a personality type by running a personality test
* Hotel Reservations: A user shall make a hotel reservation
* Class organizer: A user shall organize five classes in a semester

**Rubric**

Use the following rubric to help understand the expectation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Exceptional 100%** | **Good 90%** | **Acceptable 70%** | **Developing 50%** | **Missing 0%** |
| **Test Case 1 50%** | The test case contains all header information. | The plan contains formatted test cases. | Each test case contains the type of test and at least ten steps | Missing required information | No Test Case given |
| **Test Case 2 50%** | The test case has insights, above and beyond, or contains innovative identifiable aspects | Test case is professionally written up in a plan. | Each test case contains the type of test and at least ten steps | Missing required information | No Test Case given |

# Lab 07 – Electronic Sensor Test Case

## Overview

This lab will help you identify how to test for coverage.

## Assignment

### Research the Mobile Phone sensors.

Pick three sensors. Each provide a brief explanations of it capability in regard to the for [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html). Also a list of “Verify” requirements statements. You will need to pick one of the “Verify” requirement statements and research how that sensor can be tested.

* Camera
  + The camera will essential for capturing information, scanning, and using VR as part of the motion detection workflow.
  + Verify that the camera can pick up close and far range.
  + Verify that the camera can see IR light.
  + Verify that the camera can see bright and low light.
* Barometer
  + Understanding the temperature and change weather patterns helps alert the EMP (Emergency Medical Personal) to potential patient threats.
  + If applicable, verify that the barometer can pick up pressure changes.
* Sound
  + Besides communication recording and making phone calls, it can help with diagnostics of the patient, recording progress, or observation. It can be used to pick distinct and distant noise that the EMP is not able to hear.
  + Verify the range of sound that is detected by the device.
* Wi-Fi/Network
  + Communication can be within line of sight devices or with repeaters. In some cases, it can detect outside elements. Devices can include patient monitoring, device-to-device updates/alerts, or server communication.
  + Verify that the different frequencies that the Wi-Fi can pick up and usage.
* Vibration
  + Most devices these days can go to silent mode, it can also pick up vibrations. The detection allows for searching for trapped people, detecting body injuries, other uses.
  + Verify different types of vibrations.
* Touch
  + The touch capability enhances the user interface, also can include strength tests, writing, etc.
  + Verify the touch capability and sensitivity.
* Finger Print
  + For security purposes, most laptops and tablets come with a figure print id. This can also be used for identification purposes for patients.
  + Verify identification and scanning capabilities.
* Radio
  + Like a headphone cord, any mobile device can pick up FM/AM communication with small antennas. With a more advanced plug-in, the device could pick up other frequencies, from CB to Ham to Emergency broadcasting channels.
  + Verify radio frequencies.
* Acceleromenter / Gyroscope /Compass/GPS
  + Most tables can detect rotation on an X, Y, and Z corrident and gravity and compass directions. Some can detect GPS signals. All of which can provide the location, direction, and velocity.
  + Verify Acceleromenter directions in X, Y, and Z.
  + Verify Gyroscope directions in X, Y, Z
  + Verify Compass, Magentic Field or Lat./Log. In scale and accuracy.

### Write test Cases

Pick three sensors. For each sensor pick one of the "verify" requirement statements to write a manual test.

Use your own computer, mobile device or other equipment software to write the tests. For the sensors, imagine that your current personal electronic device, cell phone or tablet, is the device being used for Dr. Mobile.

Post your tests on the discussion board. Include the test's title, an overview of the tests, including the requirement, test type, and pre-conditions..

#### Format:

* Title:
* Requirement:
* Test Type:
* Pre-conditions:
* Steps: [ (A)ction/(Result)/(V)erify ]
  + A:
  + R:
  + …
  + V:

### Reply

Pick two different of your classmate's tests. Run through each of the tests and make one suggestion or improvement.

## Submission

Post your tests as three separate posts, and two replies

## Rubric

Use the following rubric to help understand the expectation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Exceptional 100%** | **Good 90%** | **Acceptable 70%** | **Developing 50%** | **Missing 0%** |
| **Assignment 70%** | Test case contain all required detail. | Posted 3rd test case | Posted 2nd test case | Posted 1st test case | No original post |
| **Reply 20%** | Improvements are show innovation and insight. | Relies are realivent and positive. | 2nd improvement reply | 1st Reply | No Reply |
| **Professionalism 10%** | The paper is easy to read and communicated. | Properly cited, there are no grammar or spelling errors, and the writing style is "professional." | Found an instance of a spelling error, grammar error, overly verbose wording, poor formatting, or poor writing. | A citation is missing where one is needed (plagiarism alert!). | Gross spelling/grammar errors or other aspects of the writing that make the paper difficult to read. |

# Prove M07: Test Cases

## Overview

Create different types of test cases and understand the uses for each.

## Backstory

Based on the work you did for the Electronic Sensor, you will be using different testing methods. Not all tests are created and executed the same way. Some tests deal with pushing buttons and seeing the results. In contrast, others can be automated because the input and the expected output are known. Some tests need to be done by collecting large amounts of data and analyzing the results. Finally, some tests are as simple as shown during a demo or through code or log inspection.

## Instruction

You are going to extend your tests of the Electronic Sensor Lab. Use the tests that you created from the lab.

* **Blind Test**: Have another person run the tests and ask them for insights. Make any improvements from suggestions from classmates (highlight in red).
* **Checklist**: Review your lab and two of your classmate's labs for common practices and improvements. Create a checklist five things that every test should have and of what should be a good test.

## Make it your Own

The completion of the core of this assignment is 89%. The assignment needs additional personalized work to achieve 100%.

* **Create an additional test** from one of the lab’s sensors with your improvements and checklist. . Mark in blue the improvements that you implemented.

## Submission

Include all activities in one document. Each Test needs to be on a new page. The Inspection checklist and review are on the last page.

Make sure that you upload a copy of your document to iLearn.

## Rubric

Use the following rubric to help understand the expectation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Exceptional 100%** | **Good 90%** | **Acceptable 70%** | **Developing 50%** | **Missing 0%** |
| **Test activities 80%** | Make it your own | Provide checklist | Identify improvements (red) | Blind Test | No answers to questions |
| **Professionalism 10%** | Make it your own | Properly cited, there are no grammar or spelling errors, and the writing style is "professional." | Found an instance of a spelling error, grammar error, overly verbose wording, poor formatting, or poor writing. | A citation is missing where one is needed (plagiarism alert!). | Gross spelling/grammar errors or other aspects of the writing that make the paper difficult to read. |
| **Citations 10%** | One of the citations is a primary source | Contains 3-4 citations other than the reading. | Contains 1-2 more citations other than the reading | Contains citations from the reading | No Citations |