# Week 09 Instructor Notes

## Overview

Week 09 will be covering Integration Testing. Students will write a Intergation Plan.

## Objectives

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

* Implement Testing Applications (Cyber, UX, Full-Stack, End-To-End)
* Build an Integration Plan

## Before the Week Begins (Prepare)

Partnership/Group work:

* Integration plan should be worked on in assigned groups.
* Check Canvas “groups” to verify that all students are enrolled in a group.

## During the Week (Teach)

Labs:

* Integration of different sensors into [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html).

## Looking Ahead

To be aware of

Next week's announcements:

* Please create and post your Announcement for W10 by Friday of this week, no later than Monday.

# Prepare

## Overview

In developing software, it is often divided into different pieces, top-down design, and worked independently. At some point, the piece needs to come back together. When all the pieces come together, the problems begin to pile up. Much like making bread, you can't put all the ingredients in the mixer at the same time and expect perfect bread. The Assembly of software is a key component to delivering on time. Understanding how to describe the inputs getting processed to output throughout the entire workflow is vital in delivering the desired product. In addition to making sure that the product is coming together, it also makes sure it works with other 3rd party components, interfaces, and standards.

## Objectives

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

* Implement Testing Applications (Cyber, UX, Full-Stack, End-To-End).
* Build an Integration Plan.

## Preparation Material

To be prepared for this module's activities, please read the following and be prepared to start or complete activities for Teach One Another and Prove assignments.

### Reading

* [See Reading.](https://webmailbyui-my.sharepoint.com/personal/wac3_byui_edu/Documents/GitHub/cse270-course/Reading/Reading.html)

As you read, try to answer the following questions:

### Understanding Integration Testing

* What is Integration Testing?
* What is a Test Plan vs. Test Procedure vs. Test Data?
* What are the risks of procrastinating until the end of the schedule to integrate all the pieces?
* Create a checklist of 10 items to look for that would determine when integration testing would be complete or the definition of done?

## Additional Information: Test Plan Templates

* Test Plan Template - wwwn.cdc.gov
  + <https://www2a.cdc.gov/cdcup/library/templates/CDC_UP_Test_Plan_Template.doc>
* Test Plan & Procedures - GW IT - George Washington University
  + <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=26&ved=2ahUKEwi874Dbm53mAhUW7J4KHSKZCtMQFjAZegQIChAC&url=https%3A%2F%2Fit.gwu.edu%2Fsites%2Fg%2Ffiles%2Fzaxdzs1131%2Ff%2Fdownloads%2F%255BProject%2520Name%255D%2520-%2520Test%2520Plan%2520%2526%2520Procedures.docx&usg=AOvVaw3cUgOfQhWcvxwE9rycV2h8>
* Test Plan Template – Strong QA
  + <https://strongqa.com/uploads/document/doc/29/test-plan-template-04.docx>
* Easy QA
  + [https://geteasyqa.com/qa/best-test-plan-template](https://geteasyqa.com/qa/best-test-plan-template/)
* [Test Plan & Procedures](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=26&ved=2ahUKEwi874Dbm53mAhUW7J4KHSKZCtMQFjAZegQIChAC&url=https%3A%2F%2Fit.gwu.edu%2Fsites%2Fg%2Ffiles%2Fzaxdzs1131%2Ff%2Fdownloads%2F%255BProject%2520Name%255D%2520-%2520Test%2520Plan%2520%2526%2520Procedures.docx&usg=AOvVaw3cUgOfQhWcvxwE9rycV2h8) - GW IT - George Washington University

# Lab 09 - Integration of Sensors

## Overview

[Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html) uses a series of sensors to diagnose, monitor, and do search and rescue  The lab for this week is to integrate [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html) software with the sensor software.   
The TiddlyWiki software is simulating [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html). Likewise, we will use another version of the TiddlyWiki to simulate the sensor software and hardware. In this assignment, you will merge the two TiddlyWikis and then test for correctness.   
There are three sets of sensors to be integrated:

* Add Internal Sensors
  + These sensors measure the status of the hardware built into [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html) device. Such as power levels, GPS location, etc.
* Add External Sensors
  + The External Sensors are sensors that are not directly connected to [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html) but communicate back to [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html). Examples are Monitoring bracelet, etc
* Add Communication
  + The Communication Sensors communicate connectivity to External Sensors and the home base.

## Assignment

Complete the following steps for this assignment.  
**Steps:**

* Merge Sensors TiddlyWiki into [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html) TiddlyWiki
* Test Internal Sensors from [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html) TiddlyWiki
  + Write automated tests for required sensors:
    - Power Sensor (Battery and Solar)
    - Sound
    - Camera
    - GPS
  + Stretch Goal
    - InfraRed
    - Echo
    - Surface Scan
    - Finger Print
* Test External Sensors from [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html) TiddlyWiki
  + Required Sensors
    - Heartbeat
    - Temperature
    - Blood pressure
  + Stretch
    - Oxygen Level
* Test Communication Sensors from [Dr. Mobile](https://byui-cse.github.io/cse270-course/Labs/DrMobil.html) TiddlyWiki
  + Required
    - Network
  + Stretch
    - Sattelite
    - HAM Radio

## Submission

Submit the following code items and results:

* Automated Test and results of Internal Sensors
* Automated Test and results of External Sensors
* Automated Test and results of Communication Sensors

## Rubric

Use the following rubric to help understand the expectation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Exceptional 100%** | **Good 90%** | **Acceptable 70%** | **Developing 50%** | **Missing 0%** |
| **Merge 15%** | Provide evidence that manual tests were successful in verifying merge. | Wrote Manual Test to verify merge | Successfully merged TiddlyWikis | Attempted to merge TiddlyWiki | No original post |
| **Internal Sensors 25%** | All optional tests ran successfully | All required automated tests ran. | Wrote automated tests for Internal Sensors | Manual Tests for Internal Sensors | No Attempt to complete Internal Sensor Tests |
| **External Sensors 25%** | All optional tests ran successfully | All required automated tests ran. | Wrote automated tests for Internal Sensors | Manual Tests for Internal Sensors | No Attempt to complete Internal Sensor Tests |
| **Communication Sensors 25%** | All optional tests ran successfully | All required automated tests ran. | Wrote automated tests for Internal Sensors | Manual Tests for Internal Sensors | No Attempt to complete Internal Sensor Tests |
| **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, incomplete citation, 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. |

 The Distribution of points starts at 50 percent for minimal expected participation, results of additional work increase the score.

# Lab 09 - Integration of Sensors

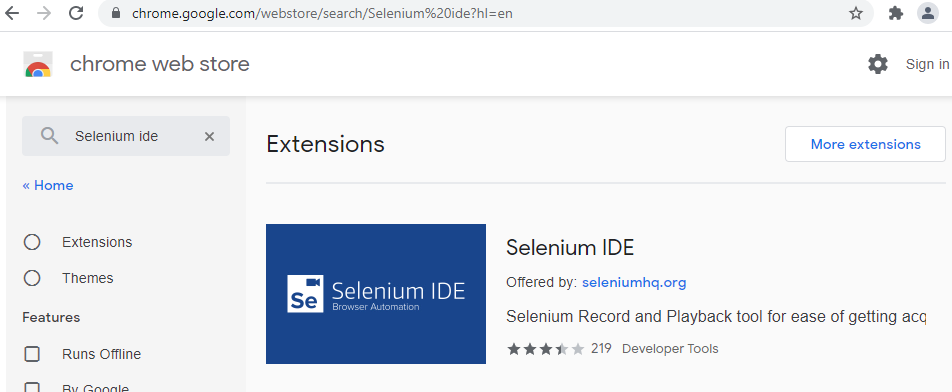
## Overview

In this lab we will use Selenium Software IDE to record our test cases from last weeks assignment.

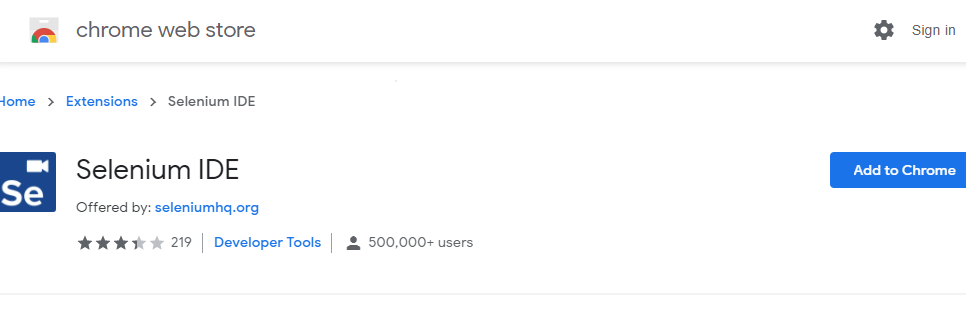
## Assignment

### Add the Selenium IDE plug in.

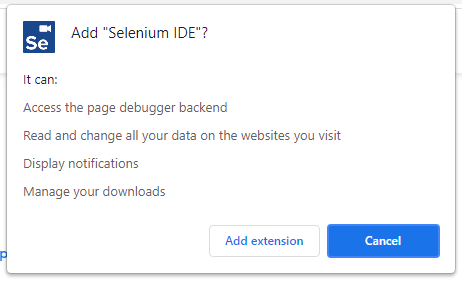
The Selenium IDE work for Firefox and for Chrome. For the sake of the lab, I will be giving directions for downloading Chrome’s extention. Goto Chromes web store and select Selenium IDE.



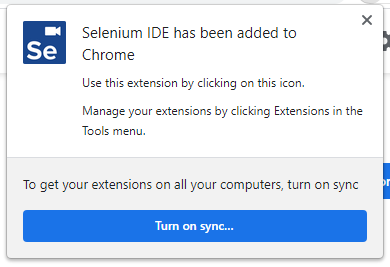
Add the Selenium IDE extension, by selecting Add to Chrome.



Agree to the permissions, by clicking Add Extension:



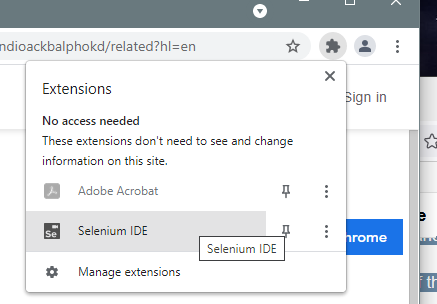
Reading information box



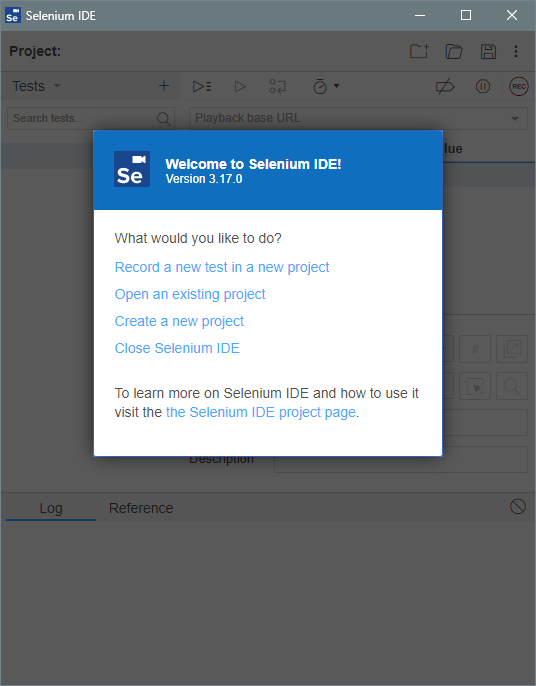
### Open Selenium IDE Project

Save off the [Dr. Mobile Selenium IDE Project File](https://webmailbyui-my.sharepoint.com/personal/wac3_byui_edu/Documents/GitHub/cse270-course/Labs/DrMobileTests.side).

Click on the Selenium icon in the plug-ins



It will then bring up Selenium IDE.



Click “Open an existing project”, select the location of where you saved the file, and select the file, “DrMobileTests.side”. (Note: \*.side, stands for Selenium IDE)

You should see, the following tests in the diagram.

Graphical user interface, application

Description automatically generated

### Run Dr. Mobile Tests

Pre-Conditions: In the top-left corner, under Project: , verify that Tests is selected and not Test suites or Executing.

**As part of this step you need to take screen shots of your IDE. Please open and save a document that you will submit as part of this lab.**

* **Run Dr. Mobile \_H\_ome page test**, by clicking on the test, and clicking on the Run iconGraphical user interface, application

  Description automatically generated

Graphical user interface, text, application

Description automatically generated

The test should bring up Dr. Mobile Webpage, Login and show the \_H\_ome page. In the Log, you should see lines that says:

* + **4.assertText on css=.toc-item:nth-child(1) > .tc-tiddlylink with value Functions OK**
  + **'Dr. Mobile \_H\_ome page' completed successfully**

Save a screen shot of this test run to your document.

* Run the Add Patient C test. The test should Login, Add Patient C to the list of Patients, and verify by going to Diagnosis page. You should see the following in the Log:
  + **15.assertText on css=.evenRow:nth-child(5) > td:nth-child(2) with value Broken Leg OK**
  + **'Dr. Mobile Add Patient C' completed successfully**

Save a screen shot of this test run to your document.

* Run the Diagnosis Patient. The test should Login, Add Patient C, and Enter diagnosis vitals. You should see the following in the Log:
  + **17.assertText on css=.evenRow:nth-child(1) > td:nth-child(2) with value 101 OK**
  + **echo: Check for Heartbeat**
  + **echo: Check for O2 Level**
  + **echo: Check for Blood Pressure**
  + **'Dr. Mobile Diagnosis Patient C' completed successfully**

Save a screen shot of this test run to your document.

### Modify Dr. Mobile – Diagnosis Patient Test

You will notice in the last run of the Dr. Mobile Diagnosis Test, that there were three echo statements after the assertText and the before the ‘successfully’ statement. We are going to change those statement to ‘assert’ statement.

An assert statement is the same to a verify statement in our previous manual tests.

In on Line 18 of the Dr. Mobile Diagnosis Patient Commands, change the Command ‘echo’ to ‘assert text’. You can either pick it from the drop down menu or copy and paste it in. You need to update the target by using the select target in page button next to the Target’s textbox, and select the table cell that contain the Heartbeat or type in the following: “css=.evenRow:nth-child(2) > td:nth-child(2)”. For the value it should be 67.

Repeat the process on lines 19 and 20.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Line | Command | Target | Value |
| Temperature | 17 | assert text | css=.evenRow:nth-child(1) > td:nth-child(2) | 101 |
| Heartbeat | 18 | assert text | css=.evenRow:nth-child(2) > td:nth-child(2) | 67 |
| O2 Level | 19 | assert text | css=.evenRow:nth-child(3) > td:nth-child(2) | 95% |
| Blood Pressure | 20 | assert text | css=.evenRow:nth-child(4) > td:nth-child(2) | 120/80 |

Run test again. Save a screen shot of this test run to your document.

### Record ORDER Supplies Test..

For this test you will be recording steps using Selenium IDE.

You will clicking on Communicate tab, and then the Order Supplies tab. After that you will need to click on each of the checkbox widgets. Try it now. Notice that the status changes from Order to ‘Need to Order’

Now lets record the steps. From Selenium IDE window, click on the Dr. Mobile Order Supplies test. Run the test. You should see it Login and bring up the Home page.

Click on line 2, the line under line 1, blue.

Graphical user interface, text, application, email

Description automatically generated

Now click on the (Rec) button. The browser should come up with it indicating that Selenium is now recording.

Graphical user interface, application

Description automatically generated

Run through the previous steps:

* Click on the Communicate Tab
* Click on the OrderSupplies Tab
* Click on each of the items to order.

Click on the Selenium IDE and click the stop ([]) button.

Run the test.

Take a screen shot of the Selenium IDE app. Save a screen shot of this test run to your document.

Save the Dr. Mobile Project File. Graphical user interface, application

Description automatically generated Close the IDE.

## Submission

Submit the following::

* Document containing the screen shots.
* Saved Selenium IDE File.

https://www.softwaretestingmaterial.com/test-plan-template/

https://www.selenium.dev/selenium-ide/docs/en/api/commands

## Rubric

Use the following rubric to help understand the expectation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Exceptional 100%** | **Good 90%** | **Acceptable 70%** | **Developing 50%** | **Missing 0%** |
| **Run Dr. Mobile tests 45%** | Evidence of running Diagnosis test | Evidence of running Add Patient | Evidence of running \_H\_ page | Evidence of IDE installed | No evidence |
| **Modify Diagnosis tests 25%** | Blood Pressure assert step was successful | O2 Level assert step was successful | Heartbeat assert step was successful | Temperature assert step was successful | No evidence of Modified Tests |
| **Run Automated tests 25%** | All items were successful | Some of the items were successfully executed | Screenshot of recording was made | Attempt was made to record | No recording found |
| **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, incomplete citation, 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. |

 The Distribution of points starts at 50 percent for minimal expected participation, results of additional work increase the score.

# Prove: Integration Plan

## Overview

An Integration test ensures all the sub-units work together correctly, both within the system and outside products and interfaces. An Integration plan lays out the resources and a new set of quality characteristics (Performance, Recovery) not found in Unit or Regression Testing. In some cases, additional tools need to be developed, like Simulators, to complete the job when elements are not available in the Test Bed. The Integration Plan's developments are the high-level tasks, addressing and mitigating any issues, identifying tests, and presenting the schedule for integrating all the components.

## Backstory

This week is an extension of Module 07, where you are now going to wrap the tests into a test plan.

## Instruction

Complete the following tasks to create a test plan:

* Title Page
  + Test Team name
  + Project Name
  + Signature block: with authors names and date
* Section 1: Introduction of Project
  + Provide a description of the project. And the purpose of the Plan.
* Section 2: Requirement
  + Create verification requirements based on the workflow from Week 08 Lab.
  + Create 5 or more requirements from the following summary of the requirements:
    - User can login into Dr. Mobile, go to the Home Page, add a new Patient, Diagnosis the Patient and Monitor the Patient.
    - User can order new supplies.
  + Number each requirement.
* Section 3: Quality Model
  + Identify at least 5 Quality Characteristics.
    - Use your Prove 5 Quality Models to list the qualities that you identified as part of the system.
  + For each Quality Characteristic identify at what level (Unit, Regression, Integration) should the test cases be completed.
* Section 4: Test Cases
  + List the Test Cases from assignments Week 08 Lab.
  + Identify the requirements mapping to the Test Cases.
* Section 5: Test Data
  + Identify data that you need to run tests from Test Cases that you worked on in Week 08.
* Section 6: Test Harness.
  + Identify and list the software that you need to run the automated tests or any additional items you need to run tests for manual, analysis, demo, or inspection tests cases (Prove 08)
* Appendix A: Lesson Learned
  + Provide a summary of lessons that you have learned about testing so far in this class.

## Make it your Own

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

Create a checklist (Professional skill) of what you should find in the Integration Plan. Contact one other group in the class, and ask to review their Plan. Complete a review of the other group's Plan. Turn in any findings from the review.

Or

Other items to include:

* Identify interfaces between project components, components and 3rd party software, and legacy code.
* Identify Stub Elements, Simulators, and Performance elements.
* Identify which Tools are to be used (Week 6).

## Submission

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

The Plan should be a minimum of 8 pages long, one page per section, including separate title and appendix pages. Please provide enough detail that a fictitious co-worker could understand it, not just an outline but not a formal paper.

## Rubric

Use the following rubric to help understand the expectation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Exceptional 100%** | **Good 90%** | **Acceptable 70%** | **Developing 50%** | **Missing 0%** |
| **Test Plan 80%** | Make it your own. | Include a reflection in the appendix of the Plan | The Plan contains all the required elements. | Has two or more significant issues | 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, incomplete citation, 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. |