

Debugging and Testing Tools

The Xcode debugger UI visually provides any potential errors within source code without having to type a single console command. However, Swift provides a tight integration of compiler REPL and debugger in the form of LLDB. LLDB contains an enormous amount of useful debugging commands that should not be overlooked for usage in integrated debugging, recovery from failure, robust expression evaluation, and consistent result formatting. Using this tool we can explore variable values and states, receive overall app states, control app's execution flow, and much more. I expect to be using the LLDB tool throughout the development process to properly debug modules and evaluate application state.

Test cases and test methods are built into Xcode and Swift as classes, namely XCTestCase, a primary class for defining test cases, and XCTestCase, a base abstract class for creating, managing, and executing tests. Utilizing these classes, unit tests for test cases and methods can be added to a test target for purposes of confirming that the code behaves as expected. Another technology being utilized for testing is Appium, which is a cross-platform test automation tool for many different types of applications, including mobile. This application is compatible between Android and iOS applications, which is helpful to developers who are building applications in varying development environments. I plan to utilize this application for not only this project but CSC415's collaborative innovation project as well.

Test Cases

Table 1. Test Case Design Table

Test Case Type	Functionality Tested	Inputs	Expected Output	Actual Output
Functionality	Create account	e-mail, password	Redirect to initial identification survey	
Error-Handling	e-mail does not exist or is invalid	e-mail	Notify user that e-mail does not exist	
Functionality	Take initial identification survey	Survey question answers	Redirect to homepage	
Error-Handling	Incorrect survey question formats	survey question answers	Highlight invalid answers	
Functionality	Recommend daily caloric/macro intake	Survey question answers, total nutrition	Display of daily calories/macros to homepage	
Functionality	Self-report food eaten	Text to search database, foods to add	Update of daily calories/macros and total nutrition	
Error-Handling	Food search does not exist	Test to search database	Display message notifying that the food cannot be found	
Functionality	Self-evaluate weight	Weight, day recorded	Display updated weight, updated weights chart	

Test Case Type	Functionality Tested	Inputs	Expected Output	Actual Output
Error-Handling	Invalid weight measurement	Weight	Notify user that weight update is invalid, disable update weight button	
Functionality	Send daily nutrition notifications to the user	Update status of daily nutrition object	Reminder notification to user	
Error-Handling	User has application open	Update status of daily nutrition object, user's online status	Do not send reminder notification to user	