

Test Case Design

Unit Testing

The XCTest framework developed by Apple is the best tool for unit testing iOS apps. Test classes can be created that can run assert methods on various functions of the app and give a record of what tests passed and failed. I will run unit tests on the classes in TaskWranglr to ensure that each component is functioning as intended.

Integration Testing

Integration testing will be used to test whether the system is functioning as a whole. Since TaskWranglr is such a small system this testing can be done manually by running in XCode's simulator. Tests will be done to ensure that every use case is working as intended and that the app can be closed down and restarted without a loss in data.

System Testing

System testing can be done using Instruments, the debugging application developed by Apple. Instruments contains functionality for testing many aspects of an app including memory usage, time profiling, and system calls. TaskWranglr does not store any sensitive information and is used by a single user so common security issues such as improper authentication are inapplicable. Apple's libraries for accessing calendar data and storing data in persistent memory will be used so these libraries will ensure that any data is handled securely. The only security related problems that will need to be tested are problems related to memory usage. Tests will be done to ensure that the app is using a reasonable amount of memory and CPU time. Performance testing can be done by downloading the app onto an actual device. Stress testing will be used to ensure that there is correct error handling if the user tries to enter tasks without enough time to complete them, ensure correctness of a schedule if the user has a busy schedule and a lot of tasks etc. UI tests can be done using either the built-in simulator or by downloading the app onto an actual device to insure that the UI is displaying the correct behavior.

Functionality Tested	Inputs	Expected Output	Actual Output
Schedule View	Tasks have been entered and a calendar connected	A schedule of when the user should do each task should be displayed alongside their calendar events.	
List View	Tasks have been entered and a calendar connected	A list of non-completed tasks are shown in a list form. Any tasks past their deadline are highlighted in red.	
Adding a task	The add button has been tapped	The screen changes to a form to input name, deadline, duration etc.	
Adding a task	Info has been inputted	The create task button should not be active until all required information has been added. The cancel button should return to the previous screen.	
Adding a subtask	User has selected the add subtask button	A screen shows similar to the add task screen for the user to input data. It behaves similarly to the add task screen.	
Updating a task	A task has been tapped either in the schedule or list view	Task details are displayed with the option to edit them	
Updating a task	The edit button has been tapped	A screen should show similar to the	

		add task screen allowing changes to be made. The cancel button should return the user to the previous screen.	
Dismissing a task	User taps dismiss button on task detail screen	The task as a whole is deleted or its duration has been changed if only part of it has been completed. There is an option to undo the action.	
Dismissing a task	User swipes task in list or schedule view	The task as a whole is deleted or its duration has been changed if only part of it has been completed. There is an option to undo the action.	
Calendar is up to date	The app is showing the schedule view	The events shown are only from calendars enabled in the app and all events are up to date with the most current version of the user's calendar. None are missing.	
Scheduling Algorithm	The app has just been opened or has switched to the schedule screen	The schedule is up to date with calendar events and tasks. All tasks are scheduled to be completed before their due date(assuming that it has not already passed) and there are no time conflicts.	

		The schedule would be considered a reasonable schedule by a human.	
Scheduling Algorithm	Algorithm has just run	<p>If tasks consisting or more time than available have been added an error message is displayed to the user.</p> <p>If the current schedule will result in a loss of sleep the user will be informed of this.</p>	
Adding or removing Calendars	User has navigated to the add/remove calendars screen	All available calendars are displayed and the user can enable or disable them.	
Adding or removing Calendars	User has added or removed a calendar	Events from a new calendar are displayed in the schedule and events from a removed calendar are no longer shown.	