

QA TEST PLAN

AccuWeather iOS App

VERSION HISTORY

Version No.	Created By	Reviewed By	Date	Comments
1.0	Amrita Basu	--	24.01.2018	--

Table of Contents

1. Introduction.....	4
2. Scope.....	4
3. Assumptions/Risks.....	5
3.1. Assumptions.....	5
3.2. Risks.....	5
4. Test Strategy/Types.....	6
5. Test Criteria.....	6
5.1. Entry Criteria.....	6
5.2. Exit Criteria.....	6
6. Test Environment.....	7
7. Test Estimates.....	7
8. Test Deliverables.....	7

1. INTRODUCTION

This Test Plan has been created to clearly communicate the testing scope, strategies, assumptions, estimates and deliverables specific to testing the AccuWeather iOS app.

2. SCOPE

In Scope:

- Ability to add/delete locations
- Retention of last saved location data between app pause and resume/kill and relaunch
- Location accuracy(some standard source to be approved as the point of reference)
- Weather data accuracy(some standard source to be approved as the point of reference)
- Interruptions(Calls/Texts)handling. The app should stay on the most recent screen after the user finishes taking the call/reading the text
- Celsius-Fahrenheit and vice versa conversion should be accurate
- Icons representing weather conditions should be in accordance with the data displayed
- Data accuracy of 'Minute By Minute' feature
- Layout of 'Minute By Minute' feature should be as per the approved design
- Data accuracy of the 'Radar' feature
- Animation on the 'Radar' screen should be smooth
- Tapping on any random AccuWeather video should load the appropriate content
- Playback controls of the video player should be smooth and as expected.
- Tapping on the cross icon should immediately cancel the video
- Data and location accuracy on iPad
- Layout adherence on iPad
- Data and location accuracy on Apple Watch
- Layout adherence on Apple Watch
- Users should be able to view/send any AccuWeather stickers on iMessage
- Offline behaviour: Weather data displayed should be the last synced data with appropriate message informing the user that he/she is offline. Message should not urge the user to turn data back on but politely inform about the connectivity status

- App behaviour should be consistent across device orientation changes
- Data should not refresh upon configuration change
- Data should refresh upon app resumption/relaunch
- If notification access is provided, notifications should arrive at the chosen frequency
- In the absence/revoking of an app permission, the app should behave appropriately in accordance with the criticality of the permission (as per Apple's specifications)

Out of Scope:

- Load Testing
- In-depth performance testing
- In-depth security testing

3. ASSUMPTIONS/RISKS

3.1. Assumptions:

- The devices used for testing are running iOS 10.0 or later
- The devices are capable of connecting to/disconnecting from network as required and receiving calls/texts
- The tester will test only app functionalities and certain non-functional aspects like performance under no/slow/full network connectivity and app behaviour on absence/revoke of permissions
- All the current functionalities have been business-approved and any additional new features coming up after the testing begins will be in scope for the next test cycle
- The tester has access to the app code for the purpose of automating test cases

3.2 Risks:

- Network speed may fluctuate from time to time, thereby affecting the testing. In case of significant fluctuations, timelines may need to be pushed.
- Device hardware issues may occur unexpectedly.

4. TEST STRATEGY/TYPES:

- 4.1. Manual/Automation Testing: All the test cases will first be executed manually. Once the testing team is confident that the app is stable and the main functionalities are present, automation of the test cases will be done, as feasible.
- 4.2. System Testing: This will include testing of all the core functionalities in scope of the latest update.
- 4.3. Regression Testing: This will ensure all the functionalities existing prior to the latest update are not broken in any way. Regression testing will commence once system testing is complete.
- 4.4. Device Testing: Testing of the core functionalities should be done across compatible devices such as iPhone, iPad and iWatch. The app should adhere to Apple's responsive design guidelines.
- 4.5. OS versions testing: Testing should be done on all possible iOS versions that the app supports i.e. iOS 10 and all the later versions(as of the time of creating this document).
- 4.6. Carriers and Network Connectivity Testing: Testing should be done across a wide range of network carriers and with no/slow/full connectivity.
- 4.7. Interruptions handling: The app should handle interruptions like calls/text/notifications gracefully. There should be no app crashes after the user is done with the interruption nor should he/she be navigated back to the main screen. The app should resume from the most recent screen.

5. TEST CRITERIA:

- 5.1. Entry Criteria:
 - The app specs have been business-approved
 - The app has been confirmed as ready to be tested by the Dev
 - Test plan document has been reviewed and signed off
 - Test cases and test data have been designed, reviewed and signed off
 - Appropriate test tools and/or framework are ready
 - The devices to be used for testing are in working condition
- 5.2. Exit Criteria:
 - 100% run of the test pack(System plus Regression)
 - At least 80% of the test cases should pass
 - All the priority 1/Severity 1 defects have been identified, fixed, retested and closed

6. TEST ENVIRONMENT:

Compatible fully functional devices (iPhone, iPad, Apple Watch) running iOS 10.0 or later.

7. TEST ESTIMATES:

Sl. No.	QA Activity	Estimated Efforts
1.	Requirements Analysis and Clarifications	3 days(including buffer time of 1 day)
2.	Test planning document creation, review and sign-off	2 days
3.	Test cases and test data creation, review and sign-off	3 days(including buffer time of 1 day)
4.	Test environment setup	2 days
5.	System Testing – execution of functional test cases and simultaneous defects tracking and retesting(manual plus automation as feasible)	4 days(including buffer time of 1 day)
6.	Regression Testing of the existing functional test cases(already automated plus manual)	2 days
7.	Basic non-functional testing: 1. Basic performance testing across multiple devices, OS versions and network carriers and speeds 2. Interruptions handling	3 days
8.	Basic non-functional testing: App behaviour upon revoking permissions of varying criticalities	1 day
9.	Retesting and closing of any remaining high-priority/high-severity defects	2 days(including buffer time of 1 day)
10.	Delivering test results, defects report and Requirements Traceability Matrix	2 days
	Total	24 days(including buffer time)

8. TEST DELIVERABLES:

- 8.1. Test Plan document to be delivered by QA team
- 8.2. Test Cases and Test Data to be delivered by QA team
- 8.3. Test Execution Results to be delivered by QA team
- 8.4. Defects Report to be delivered by QA team
- 8.5. Requirements Traceability Matrix to be delivered by QA lead