**Test plan**

**There is not enough requirements - no info about devices to be covered, no target audience, no info about a team which developed the application, so detailed test plan cannot be created (or creation such test plan will be something like an essay). Please see a plan how the app is going to be tested below.**

Phase 0. Requirements testing

Before the implementation the provided requirements should be validated to make sure there is no inconsistencies, wrong interpretations etc. For example, there is no info how user can change parameters of a request after sending, no info how UI should look like (labels, descriptions, colors), does the app support default values and validations etc. All these (and many more) questions should be asked, and clarified requirements should be covered by test cases. But for now let’s imagine we are good with the requirements provided (no need to do like that in real life).

Phase 1. Smoke testing

The goal of this phase is to make sure the app can be used and it works without any blockers or criticals. All test cases should be run on the most widely used OS versions (according to statistic on first quarter of 2021 iOS14 and Android 10 are the most widely used versions)

Phase 3. Functional and non-functional testing

This phase requires functional and non-functional testing on the same OS versions as the test cases from the previous phase. Test cases should cover positive and negative checks for all three parameters in use, check UI, validations, results. Additional test cases can validate how the app works in different device’s positions (landscape and portrait view), check what happen when incoming calls/sms received, run the app when the device is out of memory of low battery. Also connection type (wifi/mobile data usage) can be checked.

Phase 4. Cross-platform and cross-device testing

This phase requires to make sure the app works correctly on different OS versions and different resolutions. There is need to pay attention to test on different iOS versions (I think up to iOS10) and Android versions (up to Android 6.0). Resolutions depends on platform (both Android and iOS uses different ones), but at least several cases should be executed for large-resolution devices (like iPads, Notes etc.).

Phase 5. Additional activities

This phase requires to put all data up to date. All test cases should be maintained, regression test cases should be written, autotests (if needed) should be developed and set up. Also info about the app should be added to confluence (if exists), all Jira tickets should be up to date as well. If some issues will not be fixed comments should be left in corresponding tickets (not a bug, as designed, skipped etc.).

**Test cases (checklist)**

**The app is quite small and functionality is too weak, so there is no need to create detailed test cases and checklist will be enough.**

1. **Smoke checks**

* Make sure the application can be installed to and launched on iOS and Android-based devices PASSED
* Make sure correct default values are listed FAILED #01
* Make sure the app retrieves correct posts/number of posts when default parameters requested:
  + Feed URL = <https://run.mocky.io/v3/01a05594-3f31-405e-bf81-edefc1624aad>
  + Number of posts to display = 10
  + Update interval = 01:00. *ER: 10 latest posts are received* FAILED #02, #08
* Make sure the app sends new request in 60 seconds automatically PASSED
* Make sure posts can be viewed and scrolled (if number of posts do not fit to device resolution) PASSED
* Make sure Save and Reset buttons work correctly PASSED
* Make sure UI is correct and there is no unexpected/corrupted elements are displayed FAILED #03, #04
* Make sure the app can be uninstalled successfully PASSED

1. **Functional checks**
   1. Feed URL (as I understood there is only one valid URL, so checks relating to Feed URL are mostly negative):
      1. Make sure leading spaces are trimmed (best practice) FAILED #05
      2. Make sure Feed URL cannot be empty, consists of ‘spaces’ only, contains trailing ‘spaces’. ER: Warning occurs FAILED #06
      3. Make sure Feed URL cannot be invalid (without protocol, sub domain, root domain etc.) ER: Warning occurs
      4. Make sure Feed URL can be copied/pasted/manually changed PASSED
      5. Make sure the app handles too long Feed URL value correctly and does not crash PASSED
      6. Make sure encoded Feed URL value is unsupported (best practice) FAILED #06
      7. Make sure entered/pasted value clears by tapping Reset. ER: Default value is shown PASSED
      8. Make sure entered/pasted value saves by tapping Save PASSED
      9. Make sure warnings are correct and red-colored (best practice) FAILED #06
   2. Number of posts to display (there is no info about the max number of posts to be returned)
      1. Make sure the app accepts the following numbers: 1; 100; 1000 FAILED #02
      2. Make sure Number of posts to display (NPD) contains valid default value when first launch – 10 FAILED #02
      3. Try to use some huge numbers (like i16, int, int64) to make sure the app does not crash FAILED #02
      4. Make sure zeros, blank, negative, decimal, special characters or letters cannot be saved. ER: Warning occurs FAILED #07
      5. Make sure NPD values can be copied/pasted/manually changed PASSED
      6. Make sure entered/pasted value clears by tapping Reset. ER: Default value is shown PASSED
      7. Make sure entered/pasted value saves by tapping Save PASSED
      8. Make sure warnings are correct and red-colored (best practice) PASSED
   3. Update interval
      1. Make sure Update interval accepts the following values: 00:30; 60:00; 15:15 FAILED #10
      2. Make sure Update interval cannot save the following values: 00:00; 00:29; 60:01; 00:60 FAILED #10
      3. Make sure single zeros, negative values, special characters and letters cannot be saved. ER: Warning occurs PASSED
      4. Make sure Update interval format can be MM:SS only (other formats are not supported) FAILED #08
      5. Make sure Update interval values can be copied/pasted/manually changed PASSED
      6. Make sure entered/pasted value clears by tapping Reset. ER: Default value is shown PASSED
      7. Make sure valid entered/pasted value saves by tapping Save FAILED by #10
      8. Make sure warnings are correct and red-colored (best practice) PASSED
   4. Reset
      1. Make sure default values cannot be reset PASSED
      2. Make sure valid values entered to all fields can be reset to default ones PASSED
      3. Make sure invalid values entered to all fields can be reset to default ones. ER: Corresponding warning occur (pay attention that individual checks for saving negative values were performed for every field separately) PASSED
   5. Save
      1. Make sure default values can be saved PASSED
      2. Make sure valid values entered to all fields can be saved PASSED
      3. Make sure invalid values entered to all fields cannot be saved. ER: Corresponding warning occur (pay attention that individual checks for saving negative values were performed for every field separately) PASSED
   6. Refresh data
      1. Measure time to refresh posts for the following values: 00:30, 00:59; 01:00; 9:59; 60:00. ER: New posts receive after each time value left PASSED
      2. Measure time to refresh posts for default value PASSED
      3. Make sure refresh data functionality is not interrupted by incoming calls, locking/unlocking device, switching data type (wifi, mobile, Bluetooth etc.), switching between different apps etc. FAILED #11, #12
   7. Number of posts and view posts
      1. Make sure correct number of posts retrieved when the following values are used: 1; 100; 1000 FAILED #02
      2. Make sure 10 posts retrieved when default value is used FAILED #02
      3. Make sure each post has the following data:
         1. Author
         2. Time (format should be DD/MM/YYYY HH:MM), user’s timezone is required FAILED #13, #14, #15
         3. Message body PASSED
2. **Non-functional checks**
   1. Make sure all labels are named correctly PASSED
   2. Make sure buttons and fields are correctly aligned PASSED
   3. Make sure font style, color and size is correct for all elements (including posts retrieved) FAILED #03
   4. Make sure background color is correct PASSED
   5. Make sure the app works correctly in both portrait and landscape views FAILED #15
   6. Make sure app’s cache can be deleted PASSED
   7. Verify how the app works in background
3. **Cross-platform and cross-device testing** *(spent some time trying to resolve an environment issue, decided to skip these tests then)*
   1. Run the most critical test cases on the following platforms:
      1. Android 11 SKIPPED
      2. Android 9 SKIPPED
      3. Android 8 SKIPPED
      4. Android 8.1 SKIPPED
      5. Android 7 SKIPPED
      6. Android 6 SKIPPED
      7. iOS 14 SKIPPED
      8. iOS 13 SKIPPED
      9. iOS 12 SKIPPED
      10. iOS 11 SKIPPED
      11. iOS 10 SKIPPED
   2. Run the most critical test cases on the following resolutions:
      1. 640 x 1136 SKIPPED
      2. 750 x 1334 SKIPPED
      3. 1080 x 1920 SKIPPED
      4. 720 x 1280 SKIPPED
      5. 1080 x 2400 SKIPPED
      6. 1080 x 2340 SKIPPED

*The tests to executed in p. 4 can be run on some Android and iOS emulators. List of versions and resolutions may be modified according to usage statistic.*

**Issues**

#01. High. Default Feed URL is incorrect (if taking into account that the default value is listed on GitHub)

#02. Blocker. Number of posts to display value is ignored (default value is 10, but 50 latest posts retrieved. Manually entered values are also ignored)

#03. Blocker. Values entered to Feed URL, Number of posts to display and Time interval are not shown (seems to be font color equals to background color)

#04. Medium. Settings and Posts list labels are truncated (background overlaps them)

#05. Low. Leading ‘spaces’ for Feed URL are not trimmed (e.g. Postman and Fiddler handles such cases by trimming leading ‘spaces’)

#06. High. Feed URL has no validation (no warning message occurs when Feed URL is invalid for any reason including encoded URL)

#07. Medium. Number of posts to display has no validation for decimal, letters, special characters. Pay attention that warning message occurs when zero or blank value

#08. High. Update interval values are displayed in wrong format (the app should display values in format MM:SS, but now it is MMSS)

#09. High. Update interval cannot be entered in correct format (there is no auto-separation minutes from seconds by “:” sign and user is also unable to enter “:” sign from the keyboard)

#10. Critical. Manually entered values to Update interval values in ranges 00:30-00:59 and 02:00-60:00 cannot be saved due to warning message occurs. Pay attention that if such values are pasted the app works as expected

#11. Medium. No warning message when new posts are unable to be retrieved (when data is turned off)

#12. Medium. Lock/unlock a device interrupts a refresh process (e.g. if Update interval = 01:00 and a device is locked and immediately unlocked when 00:10 remains, the app will refresh posts not in 00:10, but in 01:00)

#13. Medium. Time of post has incorrect format (now it displays as MMM DD HH:MM:SS YYYY instead of DD/MM/YYYY HH:MM)

#14. Medium. Day of week is retrieved and displayed (there is no info that day of week should be displayed for every post retrieved)

#15. Medium. No user’s timezone is displayed (now user’s timezone is not listed)

#16. Blocker. Settings screen cannot be scrolled (there is no way to save/reset parameters using low resolutions and landscape mode)

**Enhancements**

#17. Locate label (for all fields) above corresponding field. In this case each warning message will appear below corresponding field

#18. Auto-correct values entered to Update interval field by applying minutes and seconds separation “:” sign

#19. Scroll up posts retrieved after each refresh to see the latest ones. Now if the list is scrolled down a refresh does not focus on the latest posts

#20. Add an ability to change parameters. Now to change Feed Url, Number of posts to display and Update interval there needs to re-install the application due to clearing cache does not remove saved values

**Bug report (example)**

ID: #16

Summary: Low resolution. Unable to manage settings due to Save and Reset buttons are unavailable

Priority: Blocker

Severity: Critical

Description: The defect is reproduced when the app is running on devices with low resolutions like 480x800 or 720x1280 (landscape mode). In these cases Settings screen cannot be scrolled, so settings cannot be saved and reset

Preconditions:

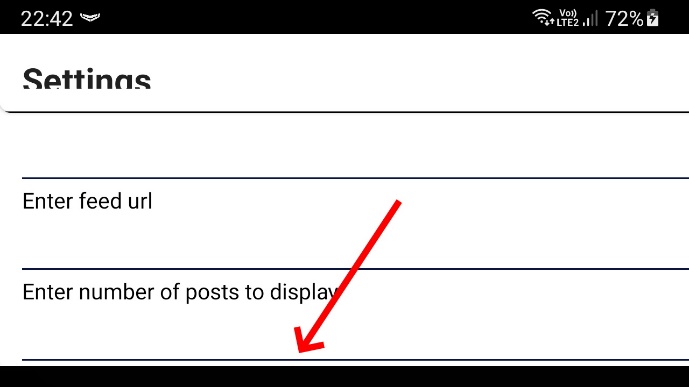
* Use a device with 720x1280 resolution
* Use landscape orientation

Steps to reproduce:

1. Launch the app
2. Scroll down the screen to Reset setting

Expected result: The screen is scrolled down and settings are reset

Actual result: The screen cannot be scrolled down, Reset button is unavailable



**Results**

The application is not ready for the release. Due to an environment issue and three blockers/one critical priority issues found testing was performed on Android device only. All issues above have priorities. To make sure the app is ready for release all High, Critical and Blocker issue should be fixed and verified. Also need to revise list of Medium issue due to some of them need to be fixed as well. After that actions relating to Phase 4 should be done and, if no high-priority issues is found the app can be released.

The following blockers were found:

* #02. Number of posts to display value is ignored. There is no chance to get any number of posts, the app retrieves 50 latest posts only
* #03. Values entered to Feed URL, Number of posts to display and Time interval are not shown. There is only one way to view values in the fields – select them. Even though it is not functional issue it will block working with the app for mostly all users
* #10. Manually entered values to Update interval values in ranges 00:30-00:59 and 02:00-60:00 cannot be saved due to warning message occurs. Several issues were found for Update interval field and they are linked to each other, but this one blocks the app usage
* #16. Settings screen cannot be scrolled. The issue can be a blocker for users with low-resolution devices, they are unable to use the app in any way

**Additional**

I do not think that QA can give lots of ways how to fix issues. For some bugs (e.g. UI-related) QA can propose some options to be implemented. As example provide a developer with a text of warning message to be added etc.

Have not faced with a situation in my experience where manual QA covers the code by unit tests. Interesting strategy, I would look at it if I have such a chance

Have not enough experience in implementing monitoring services to mobile apps, but it looks like a task for developers not QA