Newstex Android App Test Plan

NewsRobots
Version: 1.1
Last Modified Date: 2/20/14

Introduction

Description of our application

Features to be Tested

- Android Features
    - Viewing a story
        - Story that does not exist
        - Story with non-ASCII characters
        - story with excessively long content
        - story with small amount of content
        - Story with a large amount of images
    - Viewing a newsfeed
    - Viewing pre-defined content
    - Searching for content
    - Viewing recent searches
    - Navigating with the drawer
        - Menu items
        - Changing fragment
    - Navigating with the action bar
        - Drawer button
        - Search button
    - Use on different versions of Android
        - performance
        - compatibility issues (possible with the fragments)
    - UI on different screen sizes
    - Gestures
        - Pull down refreshing
        - Pull out drawer
        - Change story
    - Re-installing the app
    - Saving state of the app
        - When locking
        - When switching to home screen
    - Stress
        - Scrolling through a large number of stories
- HangerKit Features
    - Connect to server
    - Send requests to server
        - ID
            - This request just asks the server to initialize the device with the server so that it can grab information from it

  - DESCRIBE
    - This request asks the server to return to us the information about the resources that are used to create the menu and other resources that may be used later on
  - SEARCH
    - Ask the server to search its database for stories that contain a specific string and return the matches to us
  - FSEARCH
    - Asks the server to server its database for stories that contain a specific string and return the matches to us. The matches will be returned as Facets
  - GET Story [id]
    - Asks the server to grab a story from the database with a specific id
- Test concurrency of HangerKitJava by issuing multiple requests at the same time
- Interpret XDR responses into HangerKit resources
  - This includes handling properties that are unknown to the XDR parser.
- Return correct HKResources to the android application
  - HKSearch
  - HKMenu
  - HKStory
  - etc.

Approach

Unit testing

All of our unit tests will be done in HangerKit since it is difficult to unit test the Android application. We are currently writing our unit tests as we go. We will want to have unit tests that cover 75% of HangerKits functionality. There is an eclipse plugin tool called EclEmma that we will use to help us measure the code coverage of our test suite.

Integration testing

We want to promote a state of constant integration. Even if there are some insignificant changes done in HangerKit or the application we want to make sure that everything is working properly so when it comes time to releasing the application there are no issues. So every week we are going to integrate HangerKit and Hive. After the integration we will have a table of features we know worked before, if any of those features broke, and any new features we want to test out.

Automated testing

Since we are going to be putting all of our unit tests in one test project our automated testing will work simply. We have an ant task to run all the tests in the test project. So after every change we make to HangerKit we can just run the ant task and see if we broke anything.

Usability testing

We will want to perform a few different types of usability tests using test subjects and maybe have some small surveys for people to fill out.

Regression testing

Item Pass/Fail Criteria

All unit tests for HangerKit as well as the Android app should pass (green). We also want to make sure that we are testing a good amount of functionality of the application. The specific amount of code coverage we want is discussed earlier in the approach section.

Suspension Criteria and Resumption Requirements

Test Deliverables

- Amount of tests ran
- What types of tests were ran
- How many tests passed
- How many tests failed
    - How many bugs were found
- How long it took to run the tests
- Amount of test coverage

Remaining Test Tasks

Environmental Needs

Schedule

As far as unit testing goes there is no specific time schedule for it. As new functionality is added to Hanger Kit new tests are created to test that functionality and make sure it works. The unit tests will also be added to a test suite so that we can just run the test suite and that takes care of all the unit tests for us. This will make sure that we get some regression testing for Hanger Kit as well.

For the android app we will have a sprint dedicated to testing and fixing it. This most likely will be the sixth sprint since we have another sprint after that to do more testing if we need to. Also that will leave any time to fix the bugs that we find in sprint 6.