CS3300: Introduction to Software Engineering

Test Plan for Smartwatch

Version 1.0: June 9, 2015
Team: Smartwatch / Team 4

Contents

1 Introduction

1.1 Overview (Executive Summary)

The product we are testing is the Smartwatch and mobile phone portion of the product. The smartwatch will be tested for connectivity to the phone as well as the ability to properly send and receive messages. The mobile phone portion of the application will be tested for its ability to send and receive messages from the server as well as group the users. The application will finally be tested for its ability to allow the user to create, switch between, and leave groups.

1.2 Assumptions

We are assuming that the Android Wearable SDK will be functional on the watch and that it will allow us to connect between the phone and the watch. We also assume that the server side API and other sections we are not responsible for will be compatible with the setup we create on our devices.

1.3 Definitions and Acronyms

Term	Definition
Test #	Test Case Number / Identifier
Requirement	Requirement that the test cases are validating (number / identifier)
Action	Action to perform or input to produce
Expected Result	Result expected when action is complete
Actual Result	What was actually seen
P / F	Pass / Fail indicator. Checkmark = Pass. “F” = Fail
Notes	Additional notes, error messages, or other information about the test.
UI	User Interface
IDE	Integrated Development Environment
QA	Quality Assurance
API	Application Programing Interface

1.4 References

1. Android Development (Google): http://developer.android.com/training/index.html
2. Android Wear Development (Google): https://developer.android.com/training/building-wearables.html

1.5 Items Not Covered By These Test Cases

As of right now, we are not testing for some user cases of what quick replies are most appropriate. We're not testing for various different hardware compatibility or visual set ups.

2. Quality Control

2.1 Test Plan Quality

The test plan will be assessed by how many issues are found using it. When issues are found outside of the test plan, the test plan should be updated to find these issues in the future.

2.2 Adequacy criterion

Testing will be considered "complete" on any given item when the function can complete its assigned objectives independently without error and when we have verified that all existing features continue to run without error after integrating the new feature. Functionally, each feature should be able to complete each standard and edge case, without error, and produce the expected results.

2.3 Bug Tracking

Bugs will be tracked using Github Issues. Issues will keep track of the bugs assignees, status, updates and any comments from others.

3. Test Strategy

3.1 Testing Process

Unit tests will be executed automatically with each revision. At each milestone the full test plan will be used to verify the application using a combination of scripted and manual tests. These will include integration, system, and regression tests.

3.2 Technology

Most of our tests will done through physical testing, but we will also be using standard android debug tools. We will use additional tools if required, but for the most part, the standard debug tools and physical testing should be sufficient for this application.

4. Test Cases

The following table defines, for each test case, its purpose, the steps necessary to perform the test, the expected result, the actual result, pass/fail information, and optional notes. The table is the most important section of the Test Plan and Report document and is filled in by the test team as the testing process proceeds. Note that failed tests should result in bug reports.

Date test performed: ____________________________
Tester: ____________________________

Test #	Requirement or Purpose	Action / Input	Expected Result	Actual Result	P/F	Notes
1.a	Functionality	Send Message	A new message will be successfully sent to the server, and will be displayed properly on the app			
1.b	Functionality	Receive Message	A message sent from any device will be properly received and displayed			
1.c	Functionality	New Message Notification	After receiving a new message a notification will be displayed on both the phone and wearable devices.			
1.d	Functionality	Notification Dismisses On Read Message	The Notices must vanish after a user reads them			

1.e	Functionality	Notification Dismissal	A user can dismiss a notification			
1.f	Functionality	Create Chat	A user can create a chatroom			
1.g	Functionality	Join Chat	A user can join a chatroom			
1.h	Functionality	Leave Chat	A user can exit from a previously joined chatroom.			
1.i	Functionality	Switch Chat	A user can change which chat is displayed on the phone			
1.j	Functionality	Sign In	A user can login successfully			
1.k	Functionality	Sign Out	A user can sign out successfully			
1.l	Functionality	Update Settings	A user can update their notification settings.			
1.m	Functionality	Change Quick Reply Templates	A user can change their quick reply templates.			
1.n	Functionality	Read Message (Watch)	The user can read new messages on the watch.			
1.o	Functionality	Send Message (Watch)	The user can successfully send a “quick			

			reply” message from the watch.			
1.p	Functionality	Switch Chat (Watch)	A user can change which chat is displayed on the watch.			
1.q	Functionality	Notification (Watch)	A user can receive new message notifications from the watch.			

Guidelines:

- Although the information in the table above indicates that each test comprises a line, actual descriptions may require multiple lines.
- The Test # can represent a category or set of related tests. In these situations, add a suffix to the number using small Roman letter. For example, use Test 2.a, Test 2.b, …
- All tests should have a specific purpose. For example, a test may evaluate the satisfaction of a specific requirement, in which case the requirement number should be included.
- A test may comprise a series of steps, specified in the Action/Input field. Each step should include relevant user input and data files. That is, the information in this document is a prescription for executing the test.
- Some steps may be common to more than one test. There is no need to duplicate this content. In these cases, the Action field can refer to the Test # and step numbers that should be performed.
- Expected Results describe the exact output that should be produced by the test. Actual Output is what happened when the test was run. Note that for non-functional tests, the contents of this field might include elapsed time, memory consumed, etc.
- P/F denotes Pass / Fail. A test passes if its expected output matches its actual output.
- The Notes field can be used to provide additional details (e.g., to the debugging team).
- There may be cases where some tests are too expensive to perform or simply not cost/effective. In these cases, these test should be “grayed out” to indicate that they were not performed.