|  |
| --- |
|  Version 1.1| April 11, 2012 | Team Alpha |

****

|  |  |
| --- | --- |
| Alpha | Schedule 360 Certified Test Cases NF |

Revision History

|  |  |
| --- | --- |
| **Date** | **Changes from Application Specification Version 1.1** |
| April 11, 2012 | Added Test Cases for Non-Functional Requirements |
| April 20, 2012 | Format Changes  |
|  |  |
|  |  |
|  |  |

**TEST CASE 2.3.1**

|  |  |
| --- | --- |
| Description |  Verify application compatibility with Apple iPhone version 3GS and newer. Verify that application can be installed successfully. |
| Expected Result |  Application is installed successfully.  |
| Pass/Fail Criteria |  If the content does not appear obviously installed on the device then the test is a fail. |

**Test Steps**

1. Install the application being tested. The application must install without error.
2. During installation, the version number must be presented to the user. The version number must match the version number appended to the filename.
3. Verify the application has been successfully installed on the iPhone by navigation to the install location. The application should be represented as an icon in the location.

# TEST CASE 2.3.2

|  |  |
| --- | --- |
| Description |  Verify that the software shall be available for download from the iPhone App Store. |
| Expected Result |  Software shall be available for download. |
| Pass/Fail Criteria |  If the software cannot be found on the iPhone App Store, then the test is a fail. |

**Test Steps**

1. Navigate to the iPhone App Store from the iPhone version 3GS or newer.
2. Search “Schedule 360”
3. A download option should appear for the software. If the software is not available, then the test is a fail.

# TEST CASE 2.3.3

|  |  |
| --- | --- |
| Description |  Verify that the filesize of the application does not exceed the maximum limit for iPhone apps transmitted over a 3G connection. |
| Expected Result | The filesize will be less than 20 MB.  |
| Pass/Fail Criteria | If the filesize is more than 20 MB, the test is a fail. |

**Test Steps**

1. Install the application (refer to Test Case 2.3.1).
2. Go to iTunes on a different computer which has iTunes installed.
3. Sign in to iTunes.
4. Select the App Store menu in the tray at the left.
5. Left click on the Schedule 360 application then Click Command+i on Mac or Click Control+I on Windows.
6. A window will pop up showing you the size of the program. If the filesize is greater than 20 MB, the test is a fail.

# TEST CASE 2.3.4

|  |  |
| --- | --- |
| Description |  Verify that the Schedule 360 software is understandable to users who perform scheduling functions on a daily basis. |
| Expected Result | The software will be understandable to a set of Beta testers.  |
| Pass/Fail Criteria | If the selected group of Beta testers rate the software an average of less than 3.0 on a scale of 5.0 on understandability (sample size = 50), then the test is a fail. |

**Test Steps**

1. Select a group of Beta testers that use electronic scheduling software a minimum of 10 minutes per day on average.
2. Have the Beta testers perform a set of 10 functional tasks.
3. Ask the users to fill in the rating sheet that describes the usability of the software.
4. Analyze score results and perform statistics for each category for average, maximum, minimum, and standard deviation.
5. If the understandability score average is less than 3.0 then the test is a fail.

# TEST CASE 2.3.5

|  |  |
| --- | --- |
| Description |  Verify that the software functions are accessible through a GUI application interface. |
| Expected Result | The software functions will be accessible through the GUI. |
| Pass/Fail Criteria | If software functions are not accessible, the test will be failed. If response time to any action takes more than 3 seconds, the test is a fail. |

**Test Steps**

1. Install the application (refer to Test Case 2.3.1).
2. Navigate to the application icon.
3. Open the Schedule 360 application.
4. The software should open to the Graphical User Interface. If the GUI does not appear, the test is a fail.
5. Click through various icons in the GUI.
6. Verify that there is visual feedback. Response time to any action should take fewer than 3 seconds. If response time to any action takes more than 3 seconds, the test is a fail.

# TEST CASE 2.3.6

|  |  |
| --- | --- |
| Description | Verify the GUI application interface has an ease of use such that the user can begin performing operations without prior training or exposure to the application.  |
| Expected Result | Users will immediately be familiar with the application. |
| Pass/Fail Criteria | If the selected group of Beta testers rate the software an average of less than 3.0 on a scale of 5.0 on ease of use (sample size = 50), then the test is a fail. |

**Test Steps**

1. Select a group of Beta testers that use electronic scheduling software a minimum of 10 minutes per day on average.
2. Have the Beta testers perform a set of 10 functional tasks.
3. Ask the users to fill in the rating sheet that describes the usability of the software.
4. Analyze score results and perform statistics for each category for average, maximum, minimum, and standard deviation.
5. If the understandability score average is less than 3.0 then the test is a fail.

# TEST CASE 2.3.7

|  |  |
| --- | --- |
| Description |  Verify that the GUI application interface has readable text such that a user with 20/20 eyesight or corrected vision can read the text in ambient lighting from a distance of 6 inches to 24 inches. |
| Expected Result | A user with normal or corrected eyesight can read the text. |
| Pass/Fail Criteria | If 5% of users with normal or corrected vision cannot read the text, the test is a fail. |

**Test Steps**

1. Select a group of Beta testers that use electronic scheduling software a minimum of 10 minutes per day on average.
2. Have the Beta testers perform a set of 10 functional tasks.
3. On the ratings sheet, ask the user whether or not he or she has 20/20 eyesight or otherwise corrected vision (i.e. contact lenses or prescription eyewear).
4. On the ratings sheet, ask whether the user can read the text at an arm’s length or shorter distance.
5. On the ratings sheet, record whether the group of Beta testers answered a YES or NO.
6. If greater than 5% of the users (sample size = 50) answer NO to the question, the test is a fail.

# TEST CASE 2.3.8

|  |  |
| --- | --- |
| Description |  Verify the power consumption of the application does not drain the iPhone battery more than 10% when run continuously for 30 minutes. |
| Expected Result | The application will not drain the battery excessively. |
| Pass/Fail Criteria | If the application drains the battery by more than 10% with 30 minutes of continuous usage, the test is a fail. |

**Test Steps**

1. Turn on the iPhone Battery Percentage option on the iPhone:
	1. Tap “Settings”
	2. Tap “General”
	3. Tap “Usage” and you’ll see the screen with information on the iPhone usage since last charge, including standby time, call time and [data](http://www.intomobile.com/2011/09/07/howto-turn-battery-percentage-meter-your-iphone/) usage.
	4. Turn on the toggle for “Battery Percentage”.
2. Record the battery percentage remaining.
3. Navigate to the application icon.
4. Open the Schedule 360 application.
5. Begin continuous usage for 30 minutes when connected to the Cloud.
6. Close the application.
7. Record battery percentage remaining.
8. If batter percentage remaining has decreased by greater than 10%, the test is a fail.

# TEST CASE 2.3.9.a

|  |  |
| --- | --- |
| Description | Verify the software has the flexibility to operate on the iPhone operating system without interrupting the system’s normal operation. |
| Expected Result | The application will not interrupt normal voice call operations. |
| Pass/Fail Criteria | If the application interrupts iPhone voice calls, the test is a fail. |

**Test Steps**

1. Install the application (refer to Test Case 2.3.1).
2. Navigate to the application icon.
3. Open the Schedule 360 application.
4. With the application running, a user on a second phone must call the iPhone.
5. The incoming call must be indicated to the user on the iPhone.
6. Answer the call on the iPhone.
7. Conduct a short conversation from the iPhone to the caller on the second phone.
8. If any interference occurs from the running application, the test is a fail.
9. End the call in the normal way on the iPhone. If the call cannot be ended, the test is a fail.
10. With the application running, make a phone call from the iPhone to the second phone.
11. The call must be indicated on both devices. If the call is not indicated, the test is a fail.
12. A user must answer the second phone.
13. Conduct a short conversation from the iPhone to the person on the second phone.
14. If any interference occurs from the running application, the test is a fail.
15. End the voice call from the second phone. If the call is not ended on both devices, the test is a fail.
16. With the application running, place a test call to an emergency number. If the call cannot be made, the test is a fail. NOTE: Please check with your jurisdiction on how to appropriately make test calls to an emergency number.

# TEST CASE 2.3.9.b

|  |  |
| --- | --- |
| Description | Verify the software has the flexibility to operate on the iPhone operating system without interrupting the system’s normal operation. |
| Expected Result | The application will not interrupt normal text message operation. |
| Pass/Fail Criteria | If the application interrupts iPhone text messaging, the test is a fail. |

**Test Steps**

1. Install the application (refer to Test Case 2.3.1).
2. Navigate to the application icon.
3. Open the Schedule 360 application.
4. With the application running, a user on a second phone must text the iPhone.
5. The incoming text message must be indicated to the user on the iPhone. If the text message is not indicated, the test is a fail.
6. Read the text message and choose to reply.
7. Send the reply.
8. The reply must be received by the second phone. If the text is not received, the test is a fail.
9. With the application running, navigate to create a new text message.
10. Send a text message to the second phone.
11. The message must be received by the second phone. If the message is not received, the test is a fail.

# TEST CASE 2.3.9.c

|  |  |
| --- | --- |
| Description | Verify the software has the flexibility to operate on the iPhone operating system without interrupting the system’s normal operation. |
| Expected Result | The application will not interrupt operation of key functions. |
| Pass/Fail Criteria | If the application interrupts iPhone key functions, the test is a fail. |

**Test Steps**

1. Install the application (refer to Test Case 2.3.1).
2. Navigate to the application icon.
3. Open the Schedule 360 application.
4. With the application running, ensure that the contacts menu can be accessed.
5. Enter data into a contacts field. If the data cannot be entered or is altered in any way, the test is a fail.
6. Create a new contact.
7. Edit the contact information and save.
8. Delete the contact.
9. If any of the above actions are not possible, the test is a fail.
10. With the application running, navigate to the messages application and create a new message.
11. Save the message to the drafts folder. Open the message and attempt to edit.
12. Delete the message from the drafts folder.
13. If any of the above actions are not possible, the test is a fail.
14. With the application running, navigate to the web browser.
15. Go to a web page known to be accessible on the network.
16. If a data connection to the web page is not created, the test is fail.

# TEST CASE 2.3.10

|  |  |
| --- | --- |
| Description |  Verify the software does not allow more than 1 user to be connected to a personal profile at any given time. |
| Expected Result | No more than 1 user can access a unique profile at once. |
| Pass/Fail Criteria | If multiple users can login to the profile, the test is a fail. |

**Test Steps**

1. Install the application (refer to Test Case 2.3.1) on 2 separate iPhones.
2. Navigate to the application icon.
3. Open the Schedule 360 application.
4. Login to a unique profile with a defined username and password.
5. Attempt to login to the same profile from the second iPhone.
6. If the second phone is able to login, the test is a fail.

# TEST CASE 2.3.11

|  |  |
| --- | --- |
| Description | Verify the software synchronizes calendar information from internet connected devices without interrupting normal operation of the connected devices.  |
| Expected Result | The synchronization operation will not interrupt normal use of the synched devices. |
| Pass/Fail Criteria | If normal operation is interrupted, the test is a fail. |

**Test Steps**

1. Install the application on a separate computing device which is not an iPhone.
2. The application should install on the device without presenting an error message to explain that it cannot be installed. If an error occurs, the test is a fail.
3. Launch the application. The application should run on the device without presenting an error message that the device cannot be run. If an error message occurs, the test is a fail.
4. While the application is running, briefly navigate the user interface of the device. The user interface should be functional and text should be readable. If normal functionality of the user interface does not occur, the test is a fail.
5. From the application, choose the “Synch devices” operation. A message should occur which informs the user that calendar synchronization is occurring. If a message does not appear, the test is a fail.
6. While the synchronization is occurring, briefly navigate the user interface of the device. The user interface should be functional and text should be readable. If normal functionality of the user interface does not occur, the test is a fail.
7. Close the application via the user interface. If the application cannot be closed, the test is a fail.
8. Uninstall the application from the device. If an error occurs during uninstall, the test is a fail.

# TEST CASE 2.3.12 [ANALYSIS]

|  |  |
| --- | --- |
| Description | Verify the software allows a user to access his or her unique profile while blocking access to the profile from unintended users. |
| Expected Result | The software should block access to unintended users. |
| Pass/Fail Criteria | If the analysis concludes that the application is not secure, the test is a fail. |

**Test Steps**

1. The Schedule 360 Application will be provided to a third-party firm with security analysis software, such as Veracode, IBM, or AppLabs.
2. If the software is deemed to fail commercial quality security standards, the test is a fail.

# TEST CASE 2.3.13

|  |  |
| --- | --- |
| Description |  Verify the software is modifiable such a programmer can make a minor change to the code easily and without changing the software functionality.  |
| Expected Result | A programmer can modify the code with minimal effort and without changing the software architecture. The software should maintain its normal functionality after modification. |
| Pass/Fail Criteria | If a programmer cannot modify the code easily or the functionality changes, the test is a fail. |

**Test Steps**

1. Ask a programmer to make a slight modification to the user interface.
2. If a modification cannot be made or the modification takes longer than 3 hours, the test is a fail.
3. Install the application.
4. If the application cannot be installed, the test is a fail.
5. Briefly navigate the application to ensure normal operation. If an error occurs or normal operation is not possible, the test is a fail.

# TEST CASE 2.3.14

|  |  |
| --- | --- |
| Description |  Verify that software maintenance can be applied directly to the server location on the cloud |
| Expected Result | The software maintenance can be applied to the server location. |
| Pass/Fail Criteria | If maintenance cannot be applied, the test is a fail. |

**Test Steps**

1. Apply normal software maintenance to the Schedule 360 application to its location on the Cloud Server.
2. Access the application from the iPhone.
3. Check application info for “Date and time of latest maintenance”.
4. If maintenance date and time are incorrect, the test is a fail.

# TEST CASE 2.3.15

|  |  |
| --- | --- |
| Description | Verify the software code is maintainable in order for a programmer to fix a bug in the code easily and without changing the software functionality.  |
| Expected Result | A programmer can maintain the code with minimal effort and without changing the software architecture. The software should maintain its normal functionality after the fix. |
| Pass/Fail Criteria | If a programmer cannot maintain the code easily or the functionality changes, the test is a fail. |

**Test Steps**

1. Ask a programmer to perform maintenance to the code in order to fix a software defect.
2. If maintenance cannot be made or the activity takes longer than 3 hours, the test is a fail.
3. Install the application.
4. If the application cannot be installed, the test is a fail.
5. Briefly navigate the application to ensure normal operation. If an error occurs or normal operation is not possible, the test is a fail.

# TEST CASE 2.3.16

|  |  |
| --- | --- |
| Description | Verify that the software reliability can run continuous operation for 1 hour without encountering a software fault. |
| Expected Result | The software will not encounter a fault. |
| Pass/Fail Criteria | If the software encounters a fault, the test is a fail. |

**Test Steps**

1. Install the application (refer to Test Case 2.3.1).
2. Navigate to the application icon.
3. Open the Schedule 360 application.
4. Begin continuous use of the application for 1 hour.
5. If the application delivers an error message, quits unexpectedly, or behaves in an unexpected manner, the test is a fail.

# TEST CASE 2.3.17

|  |  |
| --- | --- |
| Description | Verify that the software error handling notifies the user of an error once it is identified. |
| Expected Result | The software will identify and notify the user of a software error. |
| Pass/Fail Criteria | If the software does not identify or notify the user of a software error, the test is a fail. |

**Test Steps**

1. Ask a programmer to pre-program a small software bug into the code.
2. Install and open the application.
3. Navigate to the functional area where the bug was applied.
4. Attempt to use the function.
5. Verify that the function behaves according to the predefined error.
6. If an error message is not given at the time of the error, the test is a fail.

# TEST CASE 2.3.18

|  |  |
| --- | --- |
| Description | Verify that the software has an ease of use that allows scheduling operations to be completed within a reasonable amount of time. |
| Expected Result | The software will be easy to use for a group of Beta testers |
| Pass/Fail Criteria | If the selected group of Beta testers rate the software an average of less than 3.0 on a scale of 5.0 on ease of use (sample size = 50), then the test will be failed.  |

**Test Steps**

1. Select a group of Beta testers that use electronic scheduling software a minimum of 10 minutes per day on average.
2. Have the Beta testers perform a set of 10 functional tasks.
3. Ask the users to fill in the rating sheet that describes the usability of the software.
4. Analyze score results and perform statistics for each category for average, maximum, minimum, and standard deviation.
5. If the ease of use score average is less than 3.0 then the test will is a fail.

# TEST CASE 2.3.19

|  |  |
| --- | --- |
| Description |  Verify that the software output correct scheduling information from communicable devices such that misinterpretation of the data is minimal. |
| Expected Result | The software will maintain data integrity and output correct information. |
| Pass/Fail Criteria | If the software gives incorrect scheduling information, the test is a fail. |

**Test Steps**

1. Install the application (refer to Test Case 2.3.1).
2. Navigate to the application icon.
3. Open the Schedule 360 application on the iPhone.
4. Synchronize the Schedule 360 application on the iPhone with a calendar on a different device. The calendar on the other device should have at least 20 appointments defined.
5. Cross-check the merged data in the Schedule 360 calendar.
6. Verify the scheduling information from the other device was applied correctly.
7. If an inconsistency is found in the Schedule 360 calendar, the test is a fail.
8. Open the Schedule 360 application on a device other than the iPhone.
9. Synchronize the Schedule 360 application on the device with the iPhone calendar. The iPhone calendar should have at least 20 appointments defined.
10. Cross-check the merged data in the Schedule 360 calendar.
11. Verify the scheduling information from the iPhone was applied correctly.
12. If an inconsistency is found in the Schedule 360 calendar, the test is a fail.

# TEST CASE 2.3.20

|  |  |
| --- | --- |
| Description | Verify the software is compatible with the following calendars - Outlook, Google, Yahoo, and Lotus Notes.  |
| Expected Result | The software will be compatible with the identified calendars. |
| Pass/Fail Criteria | If the software is not compatible, the test is a fail**.** |

**Test Steps**

1. Install the application (refer to Test Case 2.3.1).
2. Navigate to the application icon.
3. Open the Schedule 360 application on the iPhone.
4. Synchronize the Schedule 360 application on the iPhone with an Outlook calendar on a different device. If an error message is given or the data cannot be synchronized, the test is a fail.
5. Synchronize the Schedule 360 application on the iPhone with Google calendar on a different device. If an error message is given or the data cannot be synchronized, the test is a fail.
6. Synchronize the Schedule 360 application on the iPhone with Yahoo calendar on a different device. If an error message is given or the data cannot be synchronized, the test is a fail.
7. Synchronize the Schedule 360 application on the iPhone with a Lotus Notes calendar on a different device. If an error message is given or the data cannot be synchronized, the test is a fail.