# Acceptance Test Plan

Test A: The UI is simple and effective.

Test by asking users if they have any problems using the application.

Process

Give the application or paper prototype to a random person (prefer non-engineer) to see if they have any problems using it.

Results

Able to make rule (yes/no): \_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test B: Implement certain rules that were given to us by Samsung.

Test by making sure that the rules work in our application.

Process

Run the application and make rules that use those functionalities, make them activate.

Results

Wifi (yes/no): \_\_\_\_\_\_

Location (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test C: Check that the rules engine is working properly.

Test by making any rule become active.

Process

Set up one of the rules to easily activate, then activate and make sure nothing crashes.

Results

Able to add rule (yes/no): \_\_\_\_\_\_

Rule is activated when appropriate (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test D: Evaluate battery life while the app is running.

Test by leaving a phone out that has the application running on it.

Process

Set up one of the phones with the application on it and compare battery life to when the phone does not have the application on it.

Results

Battery life without app: \_\_\_\_\_\_

Battery life with app: \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test E: How often the app checks to evaluate rules.

Test by waiting for any rule become active.

Process

Set up one of the rules to easily activate and then wait for activation.

Results

How long it takes after proper time before it activates: \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test F: Test the sharing functions.

Test by trying to share a rule.

Process

Set up one of the rules on a phone and use some service to try to share that rule.

Results

Able to share rule via email (yes/no): \_\_\_\_\_\_

Able to share rule via NFC (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test G: Check extensibility to 3rd party programs.

Test by making a rule that uses a 3rd party program.

Process

Set up one of the rules with a 3rd party program that you can easily activate, then activate it.

Results

Able to add rule (yes/no): \_\_\_\_\_\_

Rule is activated when appropriate (yes/no): \_\_\_\_\_\_

Rule functions correctly (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test H: Check extensibility to tablets.

Test by trying to put on tablet.

Process

Install the app on a tablet then try to run the app.

Results

Able to install app (yes/no): \_\_\_\_\_\_

Able to run app (yes/no): \_\_\_\_\_\_

All functionality still present (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test I: Set up Bitbucket.

Test by making the Bitbucket account and creating the project.

Process

Set up the project on Bitbucket and make sure it updates correctly.

Results

Able to make project (yes/no): \_\_\_\_\_\_

Updates (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test J: Set up Mercurial.

Test by making Mercurial update the Bitbucket project.

Process

Install Mercurial and try to pull all of the information, then update the Bitbucket through it.

Results

Able to pull information (yes/no): \_\_\_\_\_\_

Able to update Bitbucket (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test K: Make sure the Boolean algebra works correctly.

Test by making a rule with Boolean algebra present and trying to activate.

Process

Set up one of the rules to easily activate that has multiple causes, then activate and make sure nothing crashes.

Results

Able to add rule (yes/no): \_\_\_\_\_\_

Rule is activated when appropriate following Boolean algebra (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test L: Check the drag and drop UI.

Test by asking users if they have any problems using the drag and drop UI.

Process

Give the application to a random person (prefer non-engineer) to see if they have any problems using it.

Results

Able to make rule using drag and drop (yes/no): \_\_\_\_\_\_

Functionality of rule still present (yes/no): \_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test M: Check that the accounts connect correctly.

Test by making any rule that uses accounts.

Process

Set up one of the rules to easily activate that uses accounts, then activate.

Results

Able to add account (yes/no): \_\_\_\_\_\_

Account is able to connect to service (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test N: Make it so that a single receiver receives all of the rules instead of multiple receivers.

Test by checking output of the receiver to what it should be getting.

Process

Have the receiver print out all of the stuff it receives and compare to what it should be getting.

Results

Receiver receives all information (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test O: Check that the event handler correctly receives callbacks from the underlying system.

Test by having the event handler print out what it receives.

Process

Set up the application and having the event handler print out all the callbacks it receives.

Results

Event handler prints out desired callbacks (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_

Test P: Understand what an action is in respect to the rules database.

Test by queuing up multiple intents.

Process

Set up multiple rules that you can then quickly queue up to see if they are being stored in database correctly.

Results

Database is storing intent queue correctly (yes/no): \_\_\_\_\_\_

[Passed/Failed]

\_\_\_\_\_\_\_\_\_\_\_\_