Testing Information

Smart Door

CSIR
ZEBRA-V

Eduan Bekker (12214834)
Albert Volschenk (12054519)
Zühnja Riekert(12040593)

Publication Date: 20/10/2014

Version 1.0

Change Log

16/10/2014	Version 0.1	Document Created	Zühnja Riekert
16/10/2014	Version 0.2	Document Layout	Zühnja Riekert
17/10/2014	Version 0.3	Unit testing & integration testing	Zühnja Riekert
18/10/2014	Version 0.4	Non-functional testing	Zühnja Riekert
19/10/2014	Version 0.5	Non-functional testing continued	Zühnja Riekert
19/10/2014	Version 0.6	Edited document	Eduan Bekker

Contents

Change Log 2
Testing information 4
Unit testing & Integration testing 4
Non-functional testing 4
Availability tests 5
Security tests 6
Performance tests 6
Accuracy 7
Usability tests 8

Testing information

Unit testing & Integration testing

The Smart Door Android application is developed in the Java programming language using the Android Software Development Kit (SDK).

For all the Java classes that does not make a call to any Android API, the normal JUnit framework was used for testing. All other classes were tested using Android's extensions to JUnit.

In places where unit testing was implemented we followed the test-driven-development methodology. For unit testing the necessary mock data was first created. A test was then written to test unimplemented functionality with the intentions of failing, which it did. After the functionality was added, tests were executed again to make sure that the new functionality meets the contract.

For integration tests, Android's JUnit extensions were used to test components of the user interface. Components like buttons, Android fragments and Android Activities were tested in this manner.

When we started this project we were very uncertain as to how Unit testing should be done. Admittedly we only learned the correct procedures, after many of our operational code has already been written. Also due to time constraints the percentage of code covered by testing is very low. At least 20% of the code is covered by unit tests.

As due time crept closer less code was unit tested. Also some integration tests regarding buttons and other layout tests were performed with older code and never updated as our application code was updated.

We learned from this experience and would definitely do unit testing before our actual program code with future projects.

Non-functional testing

Tests were performed on roughly 4 different devices; 2 Tablets and 2 phones.
Tablets: one was fairly new with Android version 4.4.2, the other older with Android version 4.0.3.
Phones: one had a 4.3 Android version and the other 4.4.2.

Facial recognition tests

Many tests were performed to find the correct settings suitable for facial recognition on different devices. Settings included different image scales, resolutions and threshold values, each explained in the following table:

Setting	Influence
Image scale	The amount to scale the resolution of the image down for face, nose and eyes detection. • **Higher value** will increase speed, but lower accuracy. • **Lower value** will lower speed, but increase accuracy.
Resolution	• **Higher values** will increase accuracy and lower file size. • **Lower values** will lower accuracy and increase file size.
Threshold	This is the Threshold value for the facial recognition algorithm. • **Higher value** will make the threshold less strict and allow more people in. • **Too low value** will lock everyone out. (This value is also dependent on the resolution and quality of the camera)

Availability tests

Tested uptime of different devices:

Uptime was tested by leaving the application running on an Android phone, connected to power, for roughly 5 hours. During the time 100% uptime was achieved. It is safe to assume that uptime of the system will be dependent on the uptime of the device it is being deployed to.

Failure during facial recognition or voice identification processes

Older devices tend to have weaker microphones and/or cameras that may cause a delay and some irritation. Even with an older device settings can be changed until it is most optimal. Unfortunately for better performance it is less secure. If forever reason a valid user can not gain access because he/she cannot pass the facial and voice tests, the option for entering username and password will always be there.

Tests without any network connections:

It was tested that on some Android devices the speech to text does not work when the Internet is down, unless one downloads an offline speech recognition pack for that device.

When there is no Internet connection on application startup, there is no Twitter display. As soon as there is an Internet connection, it takes at maximum a minute to realize it and start updating the feed. When the Internet connection goes down during runtime and after the feed has already been displayed, the feed does not disappear, but is unable to update.

A TCP connection is used between the SmartDoor application and the Raspberry Pi server that will open the door. When there is no connection, there will be no way to open the door unless it is by force.

Conclusions:

- When Internet is down, one can still login and open doors, the only unavailability will be the Twitter feed that won't update or display during application startup.

- If there is no connection to the Raspberry Pi and the device on which the Smart Door application runs, then the application won't be able to open the door.

Security tests

Security measures:

- Passwords stored in database are encrypted.
- Multi factor authentication, where a user's face must first be recognised as a valid user's face and have a valid voice for that user before a user can gain access.
- Only administrators have the ability to add or delete other users, give other users admin access and tamper with the application settings.

Security tests

As soon as the correct combination of settings were found to identify all users in database correctly, the system was too lenient and identified a random user, that was not in the system, as a valid user (managed to pass the facial recognition as well as the voice recognition steps in some cases).

When settings were changed to be more secure:

- Unauthorized users could not log in.
- Users that could pass login were identified correctly.
- The system was so strict that some users that were valid users could not login via face and voice identification. Pin login option will always be available.

The system can be deluded by:

- Placing a valid user's face within view of the camera
- Faking her voice or playing a recording.

Performance tests

Tested with up to 10 users saved on the system.

Facial recognition and identification

Face, eyes and nose detection is completely independent to the amount of users on the system.

The time it takes for Smart Door to recognise a user's face is unaffected by the amount of users within the system.

As the amount of users increased, the face training operation took linearly longer. It takes roughly around 0.2 seconds per image trained on. If the default setting of training on 5 faces is used, then the system takes around 1 second per user to train on. Training only takes place on application startup, when a user is added and when a user is deleted. Due to limitations of the recognition algorithms, when a user is added or removed, the whole set needs to be retrained. Since the application won't be closed regularly (preferable to have it open and available all the time) it does not have much an effect on application use.

Overall the time used to identify a face is equivalent to the amount of images the face recognizer takes to use for identification. The amount of images is a setting that can easily be changed within the app. With the amount of images to recognise upon being 3, the time used to recognize a face is easily less than 5 seconds.

Voice identification

The amount of users saved on the system has no effect on the time it takes to train the voice identifier when a new user is added.

The time it takes to identify a user's voice is linearly equivalent to the amount of users on the system. The voice pattern of user trying to gain access is compared to each user on the system. To limit this, 5 random users within the system are selected to be compared to the user trying to log in.

Overall voice identification takes longer than 5 seconds.

System waking

Rather than waking the system on motion detection, the system wakes as soon as a face is recognised or with touch input. Waking the system takes no longer than 2 seconds.

Accuracy

With previous tests the accuracy of the facial recognition was much more accurate than that of the voice identification. With the voice identification, the ratio of a person's voice being identified as his or her own seemed random. We were able to optimize the voice identification algorithm successfully for better results.

Previous tests also lead to discover the ideal environment for the best accuracy:

- Each device needs to be tweaked differently regarding the settings for best accuracy results.
- Constant lighting needs to be used.
- The light source direction should never deviate.
- A silent environment works best.
- Calibrating the noise level of the environment before using the application tends to improve accuracy.
- The environment for gathering training data should be the same as the environment where the application will be used.

The following is results performed on 10 individuals with a close to ideal environment:

- Facial recognition identified 8 out of 10 users correctly. The 2 who were incorrectly identified were both siblings with someone on the system.
- Voice identification identified 7 out of 8 voices correctly the first time.

Other tests performed with those 10 individuals: Where a face was identified as being a specific user, we tested how many times someone else could fake that person's voice and pass the voice identification process:

- 4 out of the 10 times an unauthorized user could pass the voice identification process.

Usability tests

Tests were performed by 6 willing participants. 1 High school boy, 1 older folk and 4 random students. Each was asked to individually perform certain operations and fill in a form. On the form they could rate (out of 10) how easy each task was to perform and give some feedback.

They were asked to perform the following tasks:

1. Switch to the normal username and password login system.
2. Login as root user with the following details: “root” as username and “root” as password.
3. Go to voice settings.
4. Explain what the microphone’s auto calibration does? (Hint: Use help options)
5. Auto calibrate the microphone and save the new settings.
6. Go to face settings, change the number of photos that should be used to recognize a person on to 3 and save the new settings.
7. Go to the Add User page and add yourself.
    a. Adding details
    b. Adding face
    c. Adding voice
8. Log out as root and then login again by using your new profile you created.
. Face login
a. Voice login
9. Open the door using voice commands.
10. What was your overall impression of the application’s user interface?

Results:

Task number	1	2	3	4	5	6	7.a	7.b	7.c	8.a	8.b	9	10
User 1	10	4	9	8	10	7	5	9	10	10	10	9	8
User 2	10	10	8	10	10	7	10	10	10	10	10	10	10
User 3	10	10	10	8	10	10	10	10	10	-	-	6	9
User 4	10	10	10	7	9	4	9	8	10	10	10	5	8
User 5	9	10	10	10	10	9	10	10	10	7	9	9	8
User 6	8	8	7	6	7	10	8	9	8	7	7	5	8

(Scale: out of 10, where 10 is easy and 1 is extremely difficult)
The feedback from these participants proved to be most valuable for improving Smart Door’s usability.

The following was mentioned with regards to usability improvement:

- In Add user the process of adding oneself as a user should be explained.
- Voice identification is a bit difficult.
- When one is logged in, there should be some mention of when to do voice commands.
- Could not scroll at classic login
- Press back from voice settings should take it to settings menu
- Add user menu should be scrollable.

The following has been improved on as a result:

- Classic login has been made scrollable
- Add user has been made scrollable
- Application now clearly explains what one should do during the add user and voice identification processes.